N° d'ordre NNT : 2020LYSE1005

PH.D. THESIS
co-supervised by:

Université Claude Bernard Lyon 1

École Doctorale ED205 - Interdisciplinaire Sciences Santé

Spécialité de doctorat : Recherche clinique, innovation technologique, santé publique

Università della Calabria

Dipartimento di Matematica e Informatica (DeMaCS)

Dottorato di Ricerca in Matematica e Informatica - XXXII Ciclo
Settore scientifico disciplinare: INF/01

Publicly defended on 16 January 2020, by :
Aldo Marzullo

Deep Learning and Graph Theory for Brain Connectivity Analysis in Multiple Sclerosis

---

in front of the jury:

DE MOMI Elena *Associate Professor, Polytechnic University of Milan*	Examinator
MATTEUCCI Matteo *Associate Professor, Polytechnic University of Milan*	Examinator
MAUCORT-BOULCH Delphine *Full Professor, University Claude Bernard Lyon 1*	Examinator
SCARCELLO Francesco *Full Professor, University of Calabria*	Examinator
WØHLK Sanne *Full Professor, Aarhus University*	Examinator
CALIMERI Francesco *Associate Professor, University of Calabria*	Supervisor
SAPPEY-MARINIER Dominique *Maître de Conférences - Praticien Hospitalier, University Claude Bernard Lyon 1*	Supervisor
TERRACINA Giorgio *Associate Professor, University of Calabria*	Supervisor

UNIVERSITE CLAUDE BERNARD - LYON 1

Président de l'Université	**M. Frédéric FLEURY**
Président du Conseil Académique	M. Hamda BEN HADID
Vice-président du Conseil d'Administration	M. Didier REVEL
Vice-président du Conseil Formation et Vie Universitaire	M. Philippe CHEVALIER
Vice-président de la Commission Recherche	M. Jean-François MORNEX
Directrice Générale des Services	M. Damien VERHAEGHE

COMPOSANTES SANTE

Faculté de Médecine Lyon Est - Claude Bernard	Doyen: M. Gilles RODE
Faculté de Médecine et de Maïeutique Lyon Sud - Charles Mérieux	Doyenne: Mme Carole BURILLON
Faculté d'Odontologie	Doyenne: Mme Dominique SEUX
Institut des Sciences Pharmaceutiques et Biologiques	Directrice: Mme Christine VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur: M. Xavier PERROT
Département de formation et Centre de Recherche en Biologie Humaine	Directrice: Mme Anne-Marie SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

UFR Biosciences	Directrice: Mme Kathrin GIESELER
Département Génie Electrique et des Procédeés (GEP)	Directrice: Mme Rosaria FERRIGNO
Département Informatique	Directeur M. Behzad SHARIAT
Département Mécanique	Directeur: M. Marc BUFFAT
UFR - Faculté des Sciences	Administrateur provisoire: M. Bruno ANDRIOLETTI
UFR (STAPS)	Directeur: M. Yannick VANPOULLE
Observatoire de Lyon	Directrice: Mme Isabelle DANIEL
Ecole Polytechnique Universitaire Lyon 1	Directeur: M. Emmanuel PERRIN
Ecole Supérieure de Chimie Physique Electronique	Directeur: Gérard PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur: M. Christophe VITON
Institut de Science Financière et d'Assurances	Directeur: M. Nicolas LEBOISNE
ESPE	Administrateur Provisoire: M. Pierre CHAREYRON

Ad Alessio,

Acknowledgements

I would like to express my thanks to everyone who have contributed to this work. Prof. Luca Oneto and Prof. Danail Stoyanov for accepting to review this thesis. The examination commitee Prof. Elena De Momi, Prof. Delphine Maucort-Boulch, Dr. Dominique Sappey-Marinier, Prof. Francesco Scarcello, Prof. Sanne Wøhlk.

I would like to thanks my supervisors. Prof. Giorgio Terracina, for his guidance, his patience and all his valuable advices. Dr. Dominique Sappey-Marinier for all the constructive exchanges, and the opportunity to work in a wonderful context. Prof. Francesco Calimeri: supervisor, advisor, friend.

I would like to thanks the Department of Mathematics and Computer Science. Home more than a workplace. Thanks to Hélène Ratiney, chef d'equipe, and Olivier Beuf, director of CREATIS. Merci à Ilaria, Gabriel et Jamila pour ce que nous avons partagé dans mes journées françaises, et après.

Personally, I would address a big thank you to Claudio. I really learnt a lot from you. Your close supervision, full of either strong criticisms and encouraging phrases, really gave me the possibility to express myself.

To Francesco. I do not remember the actual words you told me to write. However, I think we shared much more than an office. You have been one of my best friends during these years and, for this, I just have to say thank you.

Ai miei amici, per la loro presenza.
Alla mia famiglia, per non aver mai smesso di supportarmi e per aver creduto in me più di quanto non abbia mai fatto io.

"...e quella vita resta senza tempo. Perché il presente e il futuro sono il risultato di ciò che abbiamo costruito fino ad oggi e il segno di ciò che è stato influenzerà per sempre ciò che sarà. E soprattutto ciò che saremo."

Abstract

Multiple sclerosis (MS) is a chronic disease of the central nervous system, leading cause of nontraumatic disability in young adults. MS is characterized by inflammation, demyelination and neurodegenrative pathological processes which cause a wide range of symptoms, including cognitive deficits and irreversible disability. Concerning the diagnosis of the disease, the introduction of Magnetic Resonance Imaging (MRI) has constituted an important revolution in the last 30 years. Furthermore, advanced MRI techniques, such as brain volumetry, magnetization transfer imaging (MTI) and diffusion-tensor imaging (DTI) are nowadays the main tools for detecting alterations outside visible brain lesions and contributed to our understanding of the pathological mechanisms occurring in normal appearing white matter. In particular, new approaches based on the representation of MR images of the brain as graph have been used to study and quantify damages in the brain white matter network, achieving promising results.

In the last decade, novel deep learning based approaches have been used for studying social networks, and recently opened new perspectives in neuroscience for the study of functional and structural brain connectivity. Due to their effectiveness in analyzing large amount of data, detecting latent patterns and establishing functional relationships between input and output, these artificial intelligence techniques have gained particular attention in the scientific community and is nowadays widely applied in many context, including computer vision, speech recognition, medical diagnosis, among others.

In this work, deep learning methods were developed to support biomedical image analysis, in particular for the classification and the characterization of MS patients based on structural connectivity information. Graph theory, indeed, constitutes a sensitive tool to analyze the brain networks and can be combined with novel deep learning techniques to detect latent structural properties useful to investigate the progression of the disease.

In the first part of this manuscript, an overview of the state of the art will be given. We will focus our analysis on studies showing the interest of DTI for WM characterization in MS. An overview of the main deep learning techniques will be also provided, along with examples of application in the biomedical domain.

In a second part, two deep learning approaches will be proposed, for the generation of new, unseen, MRI slices of the human brain and for the automatic detection of the optic disc in retinal fundus images.

In the third part, graph-based deep learning techniques will be applied to the study of brain structural connectivity of MS patients. Graph Neural Network methods to classify MS patients in their respective clinical profiles were proposed with particular attention to the model interpretation, the identification of potentially relevant brain substructures, and to the investigation of the importance of local graph-derived metrics for the classification task. Semisupervised and unsupervised approaches were also investigated with the aim of reducing the human intervention in the pipeline.

Résumé

La sclérose en plaques (SEP) est une maladie chronique du système nerveux central, principale cause de handicap d'origine non traumatique chez l'adulte jeune. Il se caractérise par de nombreux processus de démyélinisation inflammatoire qui provoquent une vaste gamme de symptômes, notamment des déficits cognitifs et invalidité irréversible. L'imagerie par résonance magnétique (IRM) est aujourd'hui l'outil de référence pour le diagnostic de la maladie. L'emploi de techniques d'imagerie avancées comme la spectroscopie par résonance magnétique et l'IRM de diffusion (DTI) sont les principaux outils de détection des altérations autres que les lésions cérébrales visibles. Ces techniques ont également permis de mieux comprendre mécanismes pathologiques dans la substance blanche. En particulier, de nouvelles approches basées sur la représentation d'images IRM utilisant la théorie des graphes ont été appliquées avec succès pour l'étude et la quantification des dommages à la substance blanche.

La dernière décennie a vu l'émergence de prometteuses méthodes d'apprentissage profond pour l'étude des réseaux sociaux. Ces méthodes ont ouvert des perspectives fascinantes en neurosciences pour l'étude de la connectivité structurelle et fonctionnelle du cerveau. Grâce à leur capacité à analyser d'énormes quantités de données et à identifier les relations latentes, ce domaine de l'intelligence artificielle a connu un assez grand succès dans la communauté scientifique et s'applique désormais dans de nombreux contextes, notamment le diagnostic médical.

Dans ce manuscrit, nous présenterons les différentes techniques d'apprentissage profond développées dans ce travail concernant l'analyse des images biomédicales et, en particulier, pour la classification et la caractérisation des patients atteints de SEP. Dans ce contexte, la connectivité structurelle est utilisée pour représenter les patients. En fait, la théorie des graphes est devenue un outil sensible pour la détection des altérations causées par les pathologies cérébrales, et peut être combinée à des techniques d'apprentissage automatique afin d'identifier les propriétés structurelles latentes utiles pour étudier la progression de la maladie.

La première partie de ce manuscrit est consacré à la description de l'état de l'art. Cet état de l'art se focalisera sur les études montrant les effets de la SEP sur les faisceaux de SB grâce à l'emploi de l'imagerie de tenseur de diffusion. Une description des principales techniques d'apprentissage profond sera également fournie, ainsi que des exemples d'applicabilité dans le contexte biomédical.

Dans la seconde partie, deux techniques d'apprentissage profond seront proposées, concernant la génération de nouvelles images IRM du cerveau humain et l'identification automatique du disque optique dans les images du fond oculaire.

Dans la troisième partie, nous présenterons les techniques d'apprentissage profond combinées à la théorie des graphiques que développée dans ce travail pour étudier la connectivité structurelle des patients atteints d'une SEP. Nous présenterons en particulier des modèles de réseaux de neurones basés sur la théorie des graphes pour la classification des patients dans leurs formes cliniques. Une attention particulière sera accordée à l'interprétation de ces modèles afin d'identifier les sous-structures cérébrales potentiellement importantes. Enfin, nous explorerons des approches semi-supervisées et non supervisées pour réduire l'intervention humaine dans les processus de décision.

Sommario

La sclerosi multipla (SM) è una malattia cronica del sistema nervoso centrale, principale causa di disabilità non traumatica nei giovani adulti. Essa è caratterizzata da numerosi processi di demielinizzazione infiammatoria che causano un'ampia gamma di sintomi, inclusi deficit cognitivi e disabilità irreversibile. Per quanto riguarda la diagnosi della patologia, l'introduzione della risonanza magnetica (RM) ha rappresentato la più importante rivoluzione degli ultimi 30 anni. Tecniche avanzate di risonanza magnetica come la volumetria cerebrale, l'imaging a trasferimento di magnetizzazione e l'imaging a tensore di diffusione (DTI), sono oggi i principali strumenti per rilevare alterazioni diverse dalle lesioni cerebrali visibili e hanno contribuito ad una migliore comprensione dei meccanismi patologici che si verificano nella materia bianca. In particolare, nuovi approcci basati sulla rappresentazione di immagini di RM tramite la teoria dei grafi sono stati applicati con successo per lo studio e la quantificazione di danni della materia bianca.

L'ultimo decennio ha visto nascere nuovi promettenti metodi basati sul deep learning per lo studio di reti sociali, che hanno aperto intriganti prospettive in neuroscienza per lo studio della connettività cerebrale strutturale e funzionale. Grazie alla loro capacità di analizzare imponenti quantità di dati e individuare pattern latenti, infatti, questo campo dell'intelligenza artificiale ha ottenuto un discreto successo nella comunità scientifica ed è oggi applicato in numerosi contesti, inclusa la diagnosi medica.

In questo elaborato saranno presentate le diverse tecniche basate sul deep learning da noi sviluppate per il supporto all'analisi di immagini biomediche e, in particolare, per la classificazione e la caratterizzazione di pazienti affetti da SM. In questo contesto, i pazienti sono rappresentati utilizzando informazioni circa la loro connettività strutturale. La teoria dei grafi, infatti, costituisce un potente strumento per analizzare la connettività cerebrale e può essere combinata con tecniche di apprendimento automatico al fine di individuare proprietà strutturali latenti utili ad investigare la progressione della malattia.

Nella prima parte di questo manoscritto forniremo una panoramica dello stato dell'arte. Ci concentreremo sugli studi riguardanti l'importante ruolo delle DTI per la caratterizzazione della SM. Una panoramica sulle principali tecniche di deep learning sarà inoltre fornita, insieme ad esempi di applicabilità nel contesto biomedico.

Nella seconda parte, due tecniche di deep learning saranno proposte, riguardanti la generazione di nuove immagini di RM del cervello umano e l'individuazione automatica del disco ottico in immagini del fondo oculare.

Nella terza parte saranno presentate le tecniche di deep learning combinate con la teoria dei grafi da noi elaborate per lo studio della connettività strutturale di pazienti affetti da SM. Presenteremo in particolare modelli di reti neurali basate su grafi per la classificazione dei pazienti nelle rispettive forme cliniche. Particolare attenzione verrà posta sull'interpretazione dei modelli al fine di identificare sotto-strutture cerebrali potenzialmente rilevanti. Infine, esploreremo approcci semi-supervisionati e non-supervisionati al fine di ridurre l'intervento umano nei processi di decisione.

Contents

List of Figures

List of symbols and abbreviations

Latin letters

$\vec{B_0}$	Magnetic field

Greek letters

$\lambda_{1,2,3}$	EigenValues of diffusion tensor
$\vec{\varepsilon}_{1,2,3}$	EigenVectors of diffusion tensor
λ_1	Axial Diffusivity
λ_r	Radial Diffusivity

Abbreviations

ANN	Artificial Neural Network
ASP	Answer Set Programming
CIS	Clinically Isolated Syndrome
CNN	Convolutional Neural Network
CNS	Central Nervous System
CSF	Cerebrospinal Fluid
CST	Cortico-Spinal Tract
CPD	Canonical Polyadic Decomposition
DI	Diffusion Imaging
dMRI	Diffusion Magnetic Resonance Imaging
DNN	Deep Neural Network
DSC	Sørensen-Dice Score Coefficient
DTI	Diffusion Tensor Imaging
EDSS	Expanded Disability Status Scale
GANN	Generative Adversarial Neural Network
GCNN	Graph Convolutional Neural Network
GM	Grey Matter
GNN	Graph Neural Networks
GPU	Graphic Processing Units
HC	Healthy Control
MRI	Magnetic Resonance Imaging
MS	Multiple Sclerosis
OD	Optic Disc
PP	Primary Progressive

RR	Relapsing Remitting
SP	Secondary Progressive
SVM	Support Vector Machine
WM	White Matter

Introduction

Among the cause of non-traumatic disability in young adults, Multiple Sclerosis (MS) is one of the most important and frequent. MS is an inflammatory and neurodegenerative autoimmune disease that gradually disrupt communication randomly in the central nervous system. It may cause a wide range of symptoms, including cognitive deficits and irreversible disability. Despite the fact that the etiology still remains unknown, there is a common agreement to describe individual patient's disease form based on four clinical profiles that illustrate temporal information about the ongoing disease process. Patients usually experience a first neurological episode, called Clinically Isolated Syndrome (CIS). This form may evolve into the Relapsing-Remitting (RR) course and, successively, into a Secondary-Progressive (SP) course. The remaining 15% of patients start directly with the Primary-Progressive (PP) course. The exact cause of MS is still unknown, as well as the exact evolution mechanisms of the disease. The large unpredictability course in an individual MS patient, in fact, represents one of the most disturbing aspects and makes clinical management difficult.

Magnetic resonance imaging (MRI) is nowadays the reference tool for the diagnosis and the investigation of neurological diseases, including MS. In particular, the T1-weighted sequences (with and without contrast agent injection) and T2 allow us to identify and characterize white matter lesions and to measure the lesion load on the one hand, and cerebral atrophy on the other hand. However, these measures are only moderately correlated with the patient's clinical condition, and in particular with his disability. Besides, these techniques offer little information about the underlying pathophysiological mechanisms. The use of advanced imaging techniques such as magnetic resonance spectroscopy (MRS) and diffusion MRI offers the prospect of a better understanding of these mechanisms and could eventually lead to a better prediction of the clinical course of the patient.

Artificial Neural Networks (ANN) constitute a family of machine learning methods loosely inspired by the human brain. Due to their effectiveness in analyzing large amount of data, detecting latent patterns and establishing functional relationships between inputs and outputs, ANN have gained attention in the scientific community, and are nowadays widely applied in many contexts such as computer vision, speech recognition, but also medical diagnosis, prognosis and survival analysis. In recent years, Neural Network-based approaches for classifying patients affected by Multiple Sclerosis in their respective clinical profiles have achieved remarkable results. In particular, graph-based techniques exploit the structural connectivity representation of the white matter networks to well detect and characterize the presence of pathological mechanisms in the brain

networks that are invisible to conventional MRI. Quantifying damages in the brain white matter networks by means of functional and structural brain connectivity using graph theory, indeed, achieved promising results, becoming one of the most successful approach of the last decade. Such networks consist of nodes, corresponding to segmented cortical regions, and links, reconstructed by tractography of DTI from WM fibers-tracts.

In the first part of this work, a general introduction about the main concept object of this thesis is provided. It is divided in three chapters. In the first, we illustrate the concept of MRI and diffusion MRI, and their role in human brain investigation, including the MS context. We describe how structural connectivity representation of the white matter networks can be extracted from diffusion MRI images and the role of complex network analysis in detecting MS pathological alterations. In the second chapter, MS disease is illustrated by describing the main aspects of the disease. The presentation begins with the description of the world epidemiology. Then, elements of physiopathology are recalled. Finally, the clinical aspects are presented, starting with the natural history of the disease and the description of the different clinical phenotypes. The third chapter illustrates the main concept behind deep learning. Starting from the definition of ANN, a detailed description of the various techniques will be provided, along with different examples of their application in the biomedical domain. In the last chapter, we will discuss in more detail the scope of this thesis.

In the second part, we propose two deep learning approaches adapted to biomedical analysis. In the first chapter, we present the application of a specific type of Generative Models, namely Generative Adversarial Neural Networks, to the generation of new, unseen, MRI slices of the human brain. In the second chapter, a deep learning method to automatically detect the optic disc in retinal fundus images is develped.

In the third, and the last part, graph-based deep learning techniques are applied to the analysis of brain structural connectivity of MS patients. The first chapter propose a Graph Neural Network method to classify MS patients in their respective clinical profiles. Particular attention was given to the model interpretation and to the investigation of the importance of local graph-derived metrics for the classification task. In the following two chapters, semisupervised and unsupervised approaches are explored with the purpose of reducing the human intervention in the pipeline. In the last chapter, a general logic based framework will be presented, intended as a proof-of-concept showing how the integration of rule-based systems and neural networks can provide a remarkable impact in simplifying the study of complex mechanisms. Several experiments will be illustrated to investigate whether specific structures of the brain are more likely to be affected by the pathology.

Finally, we draw the conclusions of this work and highlight the most interesting perspectives for clinical applications.

I State of the art

Chapter 1

Magnetic Resonance Imaging

Contents

1 Magnetic Resonance Imaging

Understanding the structural basis of functional connectivity patterns requires a comprehensive map of structural connections of the human brain. Magnetic Resonance Imaging (MRI), is a method of imaging the interior of structures in a non-invasive way. It is an important diagnostic method because it is non-invasive, safe, and yields information that cannot be obtained with any other techniques. Thanks to the ever-increasing accuracy and resolution, imaging techniques have recently expanded in new directions. Technological developments in non-invasive neuroimaging combined with powerful network modelling tools, have opened up new frontiers.

1.1 Principle of Magnetic Resonance Imaging

MRI makes use of the magnetic properties of certain atomic nuclei. Atomic nuclei consist of protons with positive charge and neutron charge with zero charge. These nuclear particles are characterized by specific intrinsic properties, such as mass and electric charge. At the base of the phenomenon of nuclear magnetic resonance there is a property called spin, which can be represented as a motion of rotation of the particles around its own axis. This behaviour can be compared to a perfectly balanced not rotating spinning: it lacks moment angular and, if touched, falls immediately. When the top is rotating around on its own axis, if touched or disturbed it does not fall immediately, but enters a movement of rotation of its rotational axis around the direction of the earth's gravitational field, called precession movement. Atomic nuclei possess intrinsically this property, as if they were in constant rotation, but their intrinsic angular moment or spin is influenced by magnetic fields and not gravity, unlike angular "classic" momentum like that possessed by a spinning top. In more detail, when the atomic nuclei is placed in a magnetic field (denoted with $\vec{B}_0$) the direction of the spins follows the direction of $\vec{B}_0$. The alignment of the spin with the magnetic field $\vec{B}_0$ generates a magnetization $\vec{M}$ defined as:

$$\vec{M} = \frac{\Sigma \vec{\mu}}{dV}$$

where $\vec{\mu}$ represents the magnetic moment in the magnetic field $\vec{B}_0$. The frequency of precession is governed by the Larmor equation, defined as:

$$\vec{\omega} = -\gamma \vec{B}_0.$$

where γ is the magnetogyric ratio and every nucleus has its own specific value. Under the influence of a radio frequency (RF) wave, it is possible to perturbate the magnetization created by the field $\vec{B}_0$. This perturbation leads to the transition of the nuclei from their state of energy (resonance), simultaneously producing a radio signal. This is detected using antennas (coils) and can be used for making detailed images of body tissues. Indeed, due to differences in tissue characteristics the rate of relaxation differs in each part of the brain and thus sends a slightly different signal back to the receiver coil. By decoding where that signal came from a 3D images of brain structure can be created. In general, MRI is possible only for nuclei with non-zero spin, or nuclei for which

both protons and neutrons are not even numbered. For the formation of biomedical images the hydrogen nucleus H+ (consisting of a single proton) is used, which due to its abundance and high gyagnagnetic constant is ideal for producing and returning the greatest quantity of signal.

Recovery of Longitudinal Magnetization and Relaxation times

The RF pulse brings the proton system into a situation of imbalance due to the amount of energy absorbed and the consequent increase in potential energy that generates instability and a tendency to restore the conditions of initial equilibrium. The actual longitudinal magnetization is a function of the tissue-specific relaxation rate, the time needed for the realignment of the magnetization with the main magnetic field.

The longitudinal recovery describes the regrowth of the magnetization component in the z direction. It is a relaxation time constant which is an intrinsic property of each tissue. After a 90° pulse, when all the z component is tipped into the transverse plane (M), T1 is the number of milliseconds it takes to grow to the 63% of the original orientation (M_z). The relationship is described by the following equation:

$$M_z(t) = M_z(0)\left(1 - e^{-\frac{t}{T1}}\right) \tag{1.1}$$

T2, or transverse, relaxation describes the decay of the signal in the xy plane. It occurs due to the interactions between spins as energy is released followed an RF pulse. T2 decay is the number of milliseconds for 37% of the magnetization in the xy (M_{xy}) plane. It is described by the equation:

$$M_{xy}(t) = M_{xy}(0)e^{-\frac{t}{T2}} \tag{1.2}$$

1.2 Conventional MRI Sequences

The basic MRI techniques to obtain brain images are called conventional MRI (cMRI) sequences. With this name, we usually refer to a well-defined set of standard MRI acquisition techniques that allow to obtain simple, yet informative anatomical *in-vivo* images of the brain, or, in general, human body. By varying the sequence of RF pulses applied and collected, different types of images are created. In particular, consider the two paramenters *(i)* Repetition Time (TR), the amount of time between successive pulse sequences applied to the same slice *(ii)* Time to Echo (TE), the time between the delivery of the RF pulse and the receipt of the echo signal. By varying TR and TE the most common MRI sequences are produced, including T1-weighted and T2-weighted scans and Fluid Attenuated Inversion Recovery (Flair). These techniques were particularly successful for many reasons. They do not use x-ray radiation, unlike traditional x-ray or Computed Tomography (CT) imaging. They are non-invasive and provide images with excellent contrast detail of soft tissue and anatomic structures like grey and white matter in the brain or small metastatic lesions (cancers).

T1-Weighted Imaging

T1-weighted sequence is obtained by tuning the TR parameter to be less than T1 time (500 ms) and the TE value to be less than T2 time (14 ms). T1-weighted sequence is particularly sensible, providing the best contrast for paramagnetic contrast agents (e.g. a gadolinium-containing compounds). This property is extremely important especially in clinical settings where contrast agents are essential to perform a correct diagnosis, particularly in brain-related pathologies.

T2-Weighted Imaging

T2-weighted sequence is obtained by tuning the values of the TR and TE acquisition parameters to be greater than T1 (4000 ms) and less than T2 (90 ms), respectively. T2-weighted images provide better contrast between pathological tissue and normal tissue. In particular, dominant signals come from: fluids (like Cerebrospinal fluid) that result to be white, grey matter (grey) and white matter.

T2 FLAIR Imaging

The FLAIR sequence is similar to a T2-weighted image except that the TE and TR times are very long (greater than 9000 ms and 114 ms, respectively). By doing so, abnormalities remain bright but normal cerebrospinal fluid (CSF) is attenuated and made dark. This sequence makes the differentiation between CSF and an abnormality much easier and for this reason, T2-FLAIR images are useful to help the diagnosis of several neurodegenerative pathologies.

2 Diffusion MRI

The goal of diffusion MRI (dMRI) and specifically Diffusion Tensor Imaging (DTI) is to image the diffusion of water in the brain. The key idea behind diffusion rely on the concept that water is always moving (Brownian motion, [Brown (1828)]). In a perfectly homogeneous environment, water moves with equal probability in all directions. However, the human body environment has complex diffusion properties and the relative proportion of the water distribution between intra and extracellular compartments is affected by the pathologic processes. Diffusion weighted imaging (DWI) provides qualitative and quantitative information about the diffusion properties. It allows the mapping of the diffusion process of molecules in biological tissues, in vivo and non-invasively. A special kind of DWI, diffusion tensor imaging (DTI), has been used extensively to map white matter tractography in the brain.

2.1 The Physical Phenomenon

The term *Diffusion* refers to a random transport phenomenon, which describes the transfer of material from one spatial location to another over time. The rate of diffusion in three dimension at a fixed temperature was described by Einstein in 1905 [Einstein (1956)]:

$$r^2 = 6Dt$$

where r^2 is the mean square displacement of the molecules, t is the diffusion time and D is a constant value defined as follow:

$$D = \frac{k_B T}{6\pi\eta R}$$

k_B is the Boltzmann constant, T is the temperature of the medium, η is the dynamic viscosity of the medium and R is the radius of the spherical particle. In the case of studying *in vivo* the brain structure in humans, the type of diffusion being investigated is water self-diffusion, meaning the thermal motion of water molecules in a medium that itself consists mostly of water [Thomsen *et al.* (1987), Mukherjee *et al.* (2008)].

2.2 Acquisition Sequence

Modern diffusion-weighted sequences all trace their origin to the Pulsed Gradient Spin-Echo (PGSE) technique developed by Edward Stejskal and John Tanner in the mid-1960's [Stejskal and Tanner (1965)]. This pioneering work allowed for the first time to obtain *in vivo* images of the brain showing the diffusion in tissue.

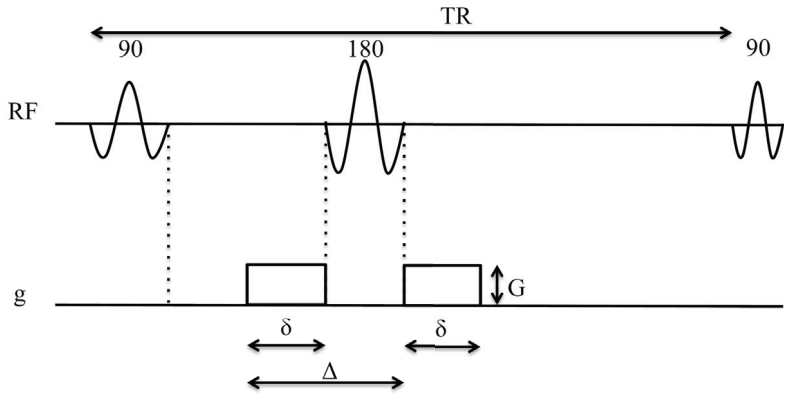


Figure 1.1: Representation of Pulsed Gradient Echo-Spin sequence. δ represents the duration of each gradient, Δ is the interval between the onset of the diffusion gradient before the refocusing pulse and that after the refocusing pulse, G is the amplitude of the diffusion gradient and *RF* indicates radiofrequency pulses.

As shown in Figure 1.1, the diffusion-weighted pulse sequence is composed by the addition of two symmetric, strong diffusion-sensitizing gradients (DG's), applied on either side of the 180°-pulse. Immediately following the second DG, an image acquisition module is placed. This is typically an echo-planar sequence using rapidly oscillating phase and frequency gradients that generate multiple gradient echoes.

Molecular motion results in loss of signal intensity due to incomplete rephasing of water proton spins, which change position between and during the applications of the 2 diffusion-sensitizing

gradients [Mukherjee *et al.* (2008)]. This diffusion-weighted contrast can be fit to an exponential model:

$$S = S_0 e^{-b \cdot ADC}$$

$$\log(S) = \log(S_0) - b \cdot ADC \tag{1.3}$$

where S represents the diffusion weighted intensity in a specific voxel, S_0 is the signal intensity in the same voxel obtained without the application of diffusion gradients, and ADC is the apparent diffusion coefficient.

The value of b, who represents a measure of the diffusion weighting, is defined by the following equation:

$$b = \gamma^2 G^2 \delta^2 \left(\Delta - \frac{\delta}{3} \right) \tag{1.4}$$

where γ is the gyromagnetic ratio, G is the amplitude of the diffusion gradient, δ represents the duration of each gradient and Δ is the interval between the onset of the diffusion gradient before the refocusing pulse and that after the refocusing pulse. Its unity is seconds per square millimetres. Typical values of b used in clinical applications range from 600 to 1500.

According to the equation of S is then possible to obtain the value for the ADC in each voxel. The equation can be rewritten as follow:

$$ADC = \frac{\log \frac{S}{S_0}}{b}$$

ADC value is a quantitative parameter used to study and measure diffusion changes due to the presence of various pathologies related to the brain [Albers (1998), Maier *et al.* (2010), Balashov and Lindzen (2012)].

2.3 Diffusion Tensor Imaging

In fibrous tissues including white matter, water diffusion is relatively unimpeded in the direction parallel to the fiber orientation. Conversely, water diffusion is highly restricted and hindered in the directions perpendicular to the fibers. Thus, the diffusion in fibrous tissues is anisotropic.

As we showed in the last two sections, dMRI is a powerful tool to obtain a large range of interesting information by simply studying the diffusion of the water in the brain. Unfortunately, except for the ADC value, representing and exploiting dMRI information is not an easy task and a big effort in development of new mathematical models is needed.

The application of the diffusion tensor (DT) to describe anisotropic diffusion behaviour was introduced by Basser *et al.* [Basser *et al.* (1992), Basser *et al.* (1994)]. In this elegant yet powerful model, diffusion is described by a multivariate normal distribution. Technically, it is proportional to the covariance matrix of a three-dimensional Gaussian distribution that models the displacements of the molecules. The symmetric matrix $D \in \mathbb{R}^{3\times3}$ is the diffusion tensor matrix defined as

follow:

$$D_{ij} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix}$$

Diagonalization of this matrix allows to obtain the eigenvalues and the eigenvectors. The matrix D can be then written as:

$$D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \overrightarrow{\varepsilon_1} \\ \overrightarrow{\varepsilon_2} \\ \overrightarrow{\varepsilon_3} \end{pmatrix}$$

$\lambda_1, \lambda_2, \lambda_3$ represent the eigenvalues and $\overrightarrow{\varepsilon_1}, \overrightarrow{\varepsilon_2}, \overrightarrow{\varepsilon_3}$ represent the eigenvectors of the diffusion ellipsoid. Those values and vectors allows to obtain simple and clear information about the shape of the diffusion tensor model as showed in Figure 1.2. The major eigenvector of the diffusion tensor points in the principal diffusion direction (the direction of the fastest diffusion). In anisotropic fibrous tissues the major eigenvector also defines the fiber tract axis of the tissue.

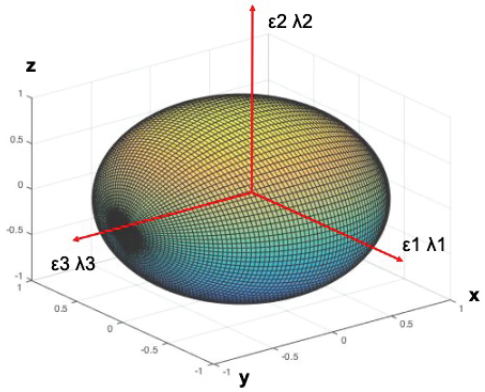


Figure 1.2: Representation of the diffusion tensor with its eigenvectors $\overrightarrow{\varepsilon_1}, \overrightarrow{\varepsilon_2}, \overrightarrow{\varepsilon_3}$ and eigenvalues $\lambda_1, \lambda_2, \lambda_3$.

2.4 Estimate White Matter Tracts Course: Tractography

DTI is often viewed by estimating the course of white matter tracts through the brain via a process called tractography [Mori *et al.* (1999)]. The word tractography refers to any method for estimating the trajectories of the fiber tracts in the white matter. Amongst the many methods proposed for tractography, *streamline tractography* is one of the most widely used. This method outputs discrete curves or trajectories (tracts) by successively stepping in the direction of the principal eigenvector (the direction of the fastest diffusion) [O'Donnell and Westin (2011)]. This process is simply described in figure 1.3. The first set of voxels, also called "seed", used to start this iteration chain are usually selected in two different ways according to the type of tractography. For global brain tractography, usually the seed voxels are randomly selected from the whole WM.

For structure analysis, like investigation of a specific WM tract, the seed are selected by the user according to a specific anatomic knowledge i. e. atlas.

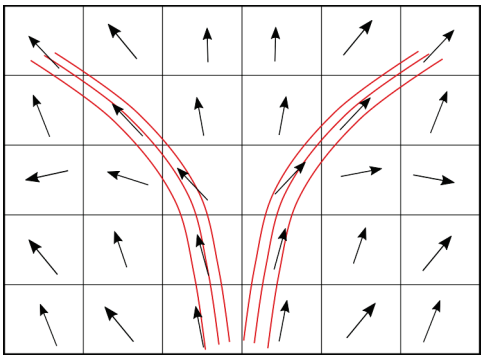


Figure 1.3: Schematic demonstrating the tractography algorithm using DTI information. Arrows represent primary eigenvectors in each voxel. Red lines are reconstructed trajectories.

A large number of techniques have been proposed in the literature [Fillard *et al.* (2011), Jbabdi and Johansen-Berg (2011), Mangin *et al.* (2013)] and an exhaustive evaluation would be prohibitive. The algorithm we previously described belong to a particular family of algorithm called *deterministic*. One of this algorithm was proposed in [Hagmann *et al.* (2007)]. Deterministic algorithms for tractography are quite fast and allow to obtain quite good results in terms of accuracy in WM fiber reconstruction. Limitation of this type of tractography is related to the accuracy of the path followed by the fibers. Indeed, these algorithms just follow one of the principal directions without taking into account other options that could give better results. In order to overcome this limitation, a new family of *probabilistic* algorithm was developed in [Behrens *et al.* (2003)]. Probabilistic algorithms repeat the deterministic version many times by randomly perturbing the main fiber directions each time, and produce maps of connectivity. Such maps indicate the probability that a given voxel is connected to a reference position [Fillard *et al.* (2011)].

3 Brain Connectivity Analysis using Graph Theory

We have known since the nineteenth century that the neuronal elements of the brain constitute a formidably complicated structural network. Modern non-invasive imaging techniques applied to the human brain, allow to map its intricate networks of anatomical regions and neural pathways at near-millimeter resolution. The resulting large-scale networks provide a comprehensive description of the brains structural connectivity, also called the human *connectome* [Sporns *et al.* (2005)].

3.1 Graph Theory

Graph theory is a mathematical theory for the study of organized data in the form of networks. The first introduction of graph theory was proposed by Euler in 1735 to address the problem of the seven königsberg bridges. Graphs are mathematical structures used to model pairwise relations between objects. A graph is composed by vertices (also called nodes) which are connected by

edges (also called links). Formally, a graph is an ordered pair $G = (V,E)$, where V is the set of vertices and E is the set of edges defined as:

$$E \subseteq \{\{x,y\}|(x,y) \in V \times V\}$$

which are unordered pairs of vertices. In more detail, this type of object may be called precisely an undirected graphs. A directed graph, instead, is a graph in which edges have orientations.

In a broader sense a graph is an ordered triple $G = (V,E,\omega)$. In this context, the function ω associates to every edge a value (e.g $\omega : E \to \mathbb{R}$). These values are usually known as *weights*. Thus, the graph can be weighted or unweighted.

Graphs are powerful models since they describes relationships. Isolated objects, indeed, are difficult to be considered interesting. When start considering their interactions with the surrounding environment, instead, the object become interesting. A graph can describe any type of relation by using edges between nodes. Each node is the graph, is connected with a subset of nodes defining its *neighbourhood*.

Neighbourhood For an undirected graph $G = (V,E)$, the neighbourhood $N_G(v)$ of a vertex v is the set of all neighbours of v, i.e., $N_G(v) = \{u|u,v \in E$. A similar definition can be applied to directed graphs, considering ordered pairs of nodes as edges. In this case, two set of neighbours can be distinguished: in-neighbours and out-neighbours. In general, the $N_G(v)$ notation refers to out-neighbours.

Degree The *degree* of a node represents the number of edges connected to that node. For an undirected graph, it counts the number of neighbour nodes; for a directed graph, it is the number of outcoming/incoming edges; for a weighed graph, it represents the sum of weights corresponding to connected edges.

$$d_i = \sum_{j \in V} a_{i,j}$$

Individual values of degree reflect the importance of nodes in the network. Degrees of all nodes in the network comprise the *degree distribution*, which is an important marker of network development and resilience. The mean network degree, or *global degree*, is commonly used as a measure of density. Neurobiologically, nodes with high value of degree, interacts both structurally and functionally with many other nodes of the network, thus representing important nodes [Sporns *et al.* (2005)]. Moreover, most neurobiological systems have been shown to have a broad degree distribution, with a small but important admixture of highly interconnected nodes and others with much less degree.

Path A path in a graph is a sequence of adjacent vertices. More formally, for a graph $G = (V,E)$, $Paths = \{P \in V^+|1 \leq I < |P|, (P_i, P_{i+1}) \in E\}$ is the set of all paths in G, where V^+ indicates all positive length sequences of vertices.

3.2 DTI Based Structural Connectivity

The importance of the connectome comes from the realization that *structure* and *functionality* of the brain are intrinsically connected by means of multiple levels of brain connectivity [Sporns (2013)]. These types of brain connectivity provide different informations, reflecting complementary aspect of the brain network organization. Structural connectivity, in particular, describes anatomical connections, linking a set of neural elements. At the scale of the human brain, these connections generally refer to trajectories of white matter pathways extracted from DTI. Functional connectivity (derived from functional MRI) is generally created from time series observations, and describes patterns of statistical dependence among neural elements. Within the formal framework of graph theory, a brain connectivity is a network, comprising a set of nodes, corresponding to segmented cortical regions, and edges, their mutual connections. Structural brain connectivity can be processed into network form, by means of a complex task, involving several steps and computational resources. Nodes are generally derived by parcellating cortical and subcortical gray matter regions, usually by defining a random parcellation into evenly spaced and sized voxel clusters. Once nodes are defined, their structural couplings can be estimated, and the full set of all pairwise couplings can then be aggregated into a connection matrix. The resulting network can be examined with tools and methods coming from graph theory. This offers a large set of experimental methodology for detecting, analysing, and visualizing network architectures. A major promise of human connectomics is that it will lead to a deeper understanding of the biological substrates underlying brain and mental disorders [Bassett and Bullmore (2009)]. The primary aim, however, is to map patterns of structural brain connectivity and uncover their relationship to emerging patterns of brain dynamics. In the vision that, structure and functionality of the human brain are intrinsically related, we may think to use brain connectivity to uncover potential dynamics of brain and mental disorders, as well as brain injury and recovery [Sporns (2013)]. A graphical overview of the mentioned techniques is reported in Figure 1.4.

3.3 Complex Network Analysis

Complex Network Analysis is a new multidisciplinary approach to the study of complex systems aiming at characterizing brain networks with a small number of neurobiologically meaningful and easily computable measures [Rubinov and Sporns (2010)]. The foundation for this study has been laid by the graph theory, thus most if not all of that measures come from it.

In the brain connectivity domain, several measures variously capture functional integration and segregation, quantify centrality of individual brain regions or pathways, characterize patterns of local anatomical circuitry, and test resilience of networks to insult [Rubinov and Sporns (2010)]. Different graph metrics can be estimated to measure brain network properties which can be analysed to better characterize brain networks.

These metrics are often represented in terms of individual network elements (such as nodes or links). Measurement values of all individual elements comprise a distribution, which provides a more global description of the network. This distribution is usually characterized by its mean and standard deviation, although other features, such as distribution shape, may be also important.

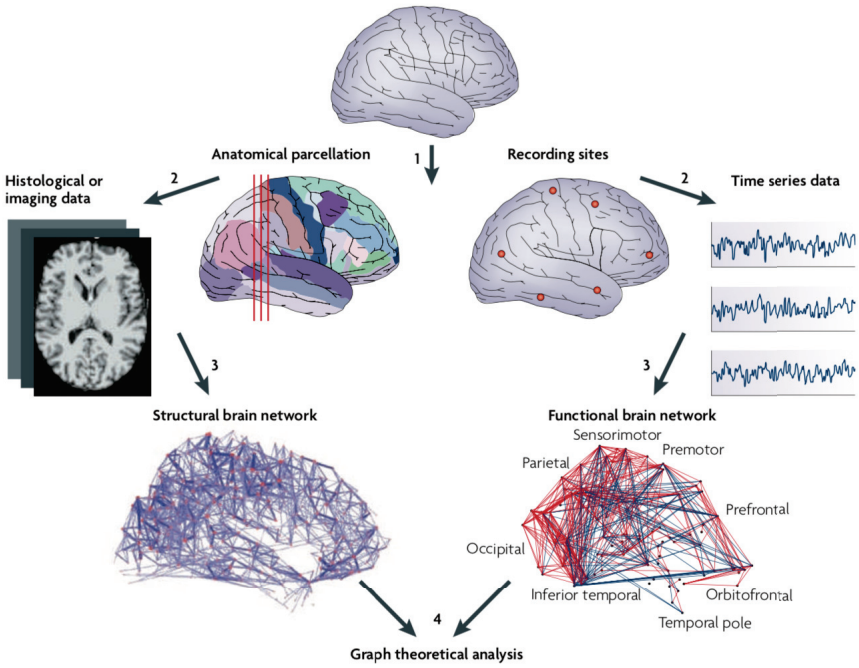


Figure 1.4: Structural and functional brain networks created through three main steps: *(i)* definition of network nodes *(ii)* estimation of a continuous measure of association between nodes *(iii)* generation of an association matrix by compiling pairwise associations between nodes. Image from Bullmore *et al.* [Bullmore and Sporns (2009)].

Graph metrics reflect different characteristic of the network and can be divided in four groups:

- measures of integration, that estimate the ease with which brain regions communicate, starting from the basic concepts of *path* and *shortest path*;
- measures of segregation, that primarily quantify the presence of such groups, known as *clusters* or *modules*, within the network;
- measures of centrality, that variously assess importance of individual nodes as *hubs* for promoting functional integration;
- measures of resilience, that measure the *vulnerability* of the network in terms of functional integration.

Measures of Integration

Shortest Path Length The *Shortest Path Length (SPL)* between nodes i and j is the shortest path that reach j starting from i. For an undirected (or directed) graph is the shortest number of edges between i and j; for a weighed graph is the sum of all the edge weights in the shortest path between i and j. This is computed as:

$$s_{i,j} = \sum_{a_{u,v} \in G_{i \Leftrightarrow j}} a_{u,v}$$

where $G_{i \Leftrightarrow j}$ is the shortest path between i and j. Note that $s_{i,j} = \infty$ for all disconnected pairs i, j.

Short path lengths promote functional integration, since they allow communication with few intermediate steps, thus minimizing the effects of noise or signal degradation. SPL is used as basic measure for integration measurements, the most important one, the Characteristic Path Length.

Characteristic Path Length The *Characteristic Path Length (CPL)* of the network is a global metric and is the average of all the shortest path length between all pair of nodes. It is computed as:

$$C = \frac{1}{q} \sum C_i = \frac{1}{q} \sum_{i \in V} \frac{\sum_{j \in V} s_{i,j}}{q-1}$$

where C_i is the average path length between i and all the other nodes, and $q = |V|$.

CPL represents the most commonly used measure of functional integration. Short CPL promotes functional integration of the network. Unfortunately, is not always possible to compute the measure since disconnected graphs are defined to have infinite length. For this reason, Global Efficiency is preferred as measure of functional integration.

Global Efficiency The *Global Efficiency* of a graph is the average of the inverse shortest path length for all pair of nodes. It is computed as:

$$G = \frac{1}{q} \sum_{i \in V} \frac{\sum_{j \in V} s_{i,j}^{-1}}{q-1}$$

where $q = |V|$ and $s_{i,j}$ is the shortest path from i to j.

Global efficiency is maximum for a fully-connected network, while minimum for a totally disconnected one. This measure is influenced by short paths; thus, the shortest the paths, the lower the measure.

Measures of Segregation

Modularity The *Modularity* is the degree to which the network may be subdivided into clearly delineated and non-overlapping groups. Such groups are referred to as *community* or *modules*, and represent aggregated sets of highly interconnected nodes. Unlike most other network measures, the optimal modular structure for a given network is typically estimated by means of optimization algorithms; one of the most commonly used is based on Newman's Q-metric [Newman and Girvan (2004)] coupled with an efficient optimization approach:

$$Q = \sum_{u \in M} \left[p_{u,u} - \left(\sum_{v \in M} p_{u,v} \right)^2 \right]$$

where the networks is subdivided into M non-overlapping modules and $p_{u,v}$ is the proportion of edges that connect nodes belonging to module u with nodes belonging to module v.

Measures of Centrality

Betweenness Centrality The *Betweenness Centrality (BC)* of a node i is the sum of the fraction of all-pairs shortest paths that pass through i [Sporns *et al.* (2005)]. It is computed as:

$$b_i = \frac{1}{(n-1)(n-2)} \sum_{h,j \in V} \frac{\rho_{(h,j)\to i}}{\rho_{h,j}}$$

where $\rho_{h,j}$ is the number of shortest paths between h and j, and $\rho_{(h,j)\to i}$ is the number of shortest paths between h and j that pass through i.

This measure is useful for identify bridging nodes that connect disparate parts of the network (*hubs*), since such this kind of nodes have high BC. The mean of the BC measurements or *Global Betweenness Centrality (GBC)*, represents the degree to which the graph is able to maintain hubs.

Measures of Resilience

Assortativity Coefficient The *Assortativity Coefficient* (AC) of a network is the Pearson correlation coefficient between the degrees of all nodes on two opposite ends of a link [Newman (2002)]. Networks featuring a positive AC are likely to have a comparatively resilient core of mutually interconnected high-degree hubs. On the other hand, networks featuring a negative coefficient are likely to have widely distributed high-degree hubs. AC is computed as:

$$A = \frac{l^{-1}\sum_{(i,j)\in E} d_i d_j - [l^{-1}\sum_{(i,j)\in E} \frac{1}{2}(d_i + d_j)]^2}{l^{-1}\sum_{(i,j)\in E} \frac{1}{2}(d_i^2 + d_j^2) - [l^{-1}\sum_{(i,j)\in E} \frac{1}{2}(d_i + d_j)]^2}$$

where $l = \sum_{i,j\in V} a_{i,j}$ is the total number of links and $d_i = \sum_{j\in V} a_{i,j}$ is the degree of node i.

The aforementioned network measures are not intended to be an exhaustive list. Nevertheless, they represent the most informative metrics, useful to describe importance of brain regions in the human connectome. The node degree is one of the most easily accessible graph measures and it is also highly informative, as well as the distribution of node degrees across the whole network. In many cases, the degree of a node is highly correlated with other more complex influence measures. Many of these measures capture the "centrality" of network elements, for example the betweenness centrality. This measure is, in turn, related to communication processes, but is also often found to be highly correlated with the measure of "closeness centrality", quantifying the proximity of each node to the rest of the network.

Another class of measures concerns vulnerability aspects of networks. For example, the decrease (or increase) in global efficiency due to the deletion of a single node or edge, or the variation of assortativity, in conjunction with a decrease (or increase) of network density, should be used when attempting to identify crucial areas of the network [Sporns (2013)].

Closer analysis of brain networks has shown that modules are network communities of densely interconnected neural elements that share common input and output projections, exhibiting similar physiological responses [Hilgetag and Kaiser (2004)]. To this end, hubs perform important integrative roles in structural networks. It should be noted that there is no unique way of detect-

ing these hubs with graph theory tools. Bridging nodes indeed, are generally identified by their high degree, high centrality, and diverse connection profiles that straddle the boundaries between modules [Sporns *et al.* (2007)]. A detailed analysis of the topology of human brain structural connectivity, revealed a rich club of highly interconnected hub regions including portions of the superior frontal cortex, superior parietal cortex, and the precuneus, in addition to several subcortical regions including the thalamus, hippocampus, and part of the basal ganglia [van den Heuvel and Sporns (2011)].

The study of brain networks is a promising frontier. The development of new analytic techniques and modelling approaches, in parallel with continued methodological refinements in the area of human neuroimaging, continue to allow ever more detailed analyses of human structural and functional networks. Graph methods have proven useful for capturing how networks vary across individuals, how they evolve across the time, and how they behave in presence of a wide variety of brain and mental disorders.

Chapter 2

Multiple Sclerosis

Contents

1 Introduction

Multiple sclerosis (MS) is a chronic autoimmune, inflammatory neurological disease of the central nervous system (CNS). MS attacks the myelinated axons in the CNS, destroying the myelin and the axons to varying degrees. It constitutes the leading cause of non-traumatic disability in young adults (from 20 to 40 years old) and remains without well-known etiology [Compston and Coles (2008)]. The course of MS is highly varied and unpredictable. In most patients, the disease is characterized initially by episodes of reversible neurological deficits, which is often followed by progressive neurological deterioration over time.

2 Epidemiology

In recent years, knowledge of the geographical distribution of the disease and its survival data, and a better understanding of the natural history of the disease, have improved our understanding of the respective roles of endogenous and exogenous causes of MS. A pioneering study on that field is the one described in [Kurtzke (1980)] and later updated in [Kurtzke (2000)]. As reported in [Kurtzke (1980)] and how it is shown in 2.1, the geography repartition of MS worldwide is not equally distributed. It has heterogeneous prevalence worldwide: it is highest in North America (140/100,000 population) and Europe (108/100,000), and lowest in East Asia (2.2/100,000 population) and sub-Saharan Africa (2.1/100,000). The global median prevalence of MS has increased from 30/100,000 in 2008 to 33/100,000 in 2013, as also confirmed by the MS International Federation [Leray *et al.* (2016)]. However, given the mortality, prevalence, incidence, evolution of the gender ratio and geographical distribution of MS in France, this country can now be more precisely defined as a high-risk country for MS.

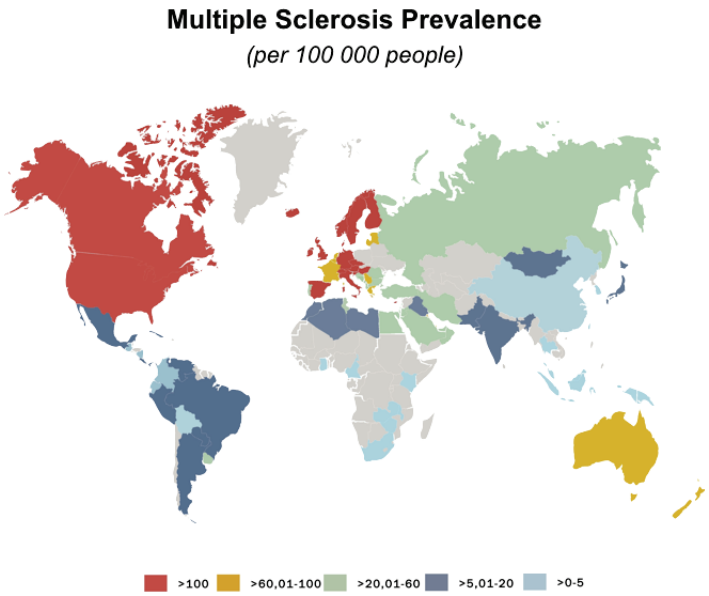


Figure 2.1: Worldwide Multiple Sclerosis prevalence. Image and data from https://www.msif.org/about-us/advocacy/atlas/.

Regarding the risk factors for MS, the most significant environmental factors are Epstein-Barr virus (EBV) infection, particularly if it occurs after infancy and is symptomatic. The role of smoking in the risk of MS has been confirmed, but modest. On the contrary, vaccines, stress, trauma and allergies have not been identified as risk factors, while vitamin D involvement has not yet been confirmed [Leray *et al.* (2016)]. From a genetic point of view, In [Kurtzke (2000)] the authors studied the changes in the risk exposure to MS between population migrating to different regions of risk. As major result, the authors found that adolescents migrating before the age of 15, present the same risk factor of their original region. Contrary, adults who migrate acquire the risk factor typical of the new risk zone. These results show how the geographic partition of MS is related to environmental factors linked to a genetic susceptibility. More recently, studies of genome-wide associations have identified immunogenetic markers (IL2RA, IL7RA) and more than 100 genetic variants have been reported. Most of these are involved in the immune response and often associated with other autoimmune diseases.

3 Etiology

The CNS is the part of the nervous system consisting of the brain and spinal cord. It is referred to as “central" because it combines information from the entire body and coordinates activity across the whole organism. The brain is the most complex organ in the human body; the cerebral cortex (the outermost part of the brain and the largest part by volume) contains an estimated 15-33 billion neurons, each of which is connected to thousands of other neurons. Neurons, are the basic units of the nervous system which communicate with other cells via specialized connections called synapses. They have a diameter ranging from 5 to 150 μm and they are composed by three parts: cell body, dendrites and axons (Figure 2.2). Axons are responsible for the nerve impulse transmission and they are surrounded by myelin sheath cells in the CNS. This substance is a membrane who facilitates the nerve impulse transmission along the axons.

MS is characterized by an abnormal immune-mediated response who attacks the myelinated axons of neurons, inducing a progressive destruction of myelin. As myelin helps to speed up the nerve impulse transmission along the axon, a destruction of myelin decreases the capability of the axon to transmit the nerve impulse. The inflammatory processes seem to start after a cell-mediated response. In this part, macrophages recognize the myelin basic protein (MBP) and present this to the T lymphocytes. These, after their activation, cross the blood brain barrier (BBB) and trigger the immune response and relative inflammation. Recent studies also suggest that the beginning of an antibody-mediated response, with an abnormal production of antibody for myelin destruction, plays an important role in the progression of the inflammation with relative tissue damage [Disanto *et al.* (2012)].

Demyelination in specific tissue area usually start without axonal damage [Noseworthy *et al.* (2000), Lucchinetti *et al.* (2000), Compston and Coles (2008)], oligodendrocytes destruction and axonal damages are induced when repeated attacks appear in time. Indeed, at the beginning of the pathology, remyelination process and pathological demyelination alternate. During this process, the oligodendrocyte progenitor cells differentiate in oligodendrocytes in order to repair the

damaged tissue [Goldschmidt *et al.* (2009), Brück *et al.* (2003)]. Unfortunately, the capability of the oligodendrocyte progenitor cells to differentiate oligodendrocytes is reduced in MS. This limitation influences the capability to recover the damaged tissue.

In the intermediate phase, the myelin contained in the tissue affected by the disease, is substituted with scarred tissue.

In the late part, demyelination effects are not present and the tissue area does not contain inflammatory cells. The increased permeability of the BBB and the inflammatory attacks increase the clinical effect related to the neurodegeneration and atrophy. In this part, when all the reparation mechanisms of the tissue are exhausted, the disability progression progressively increases.

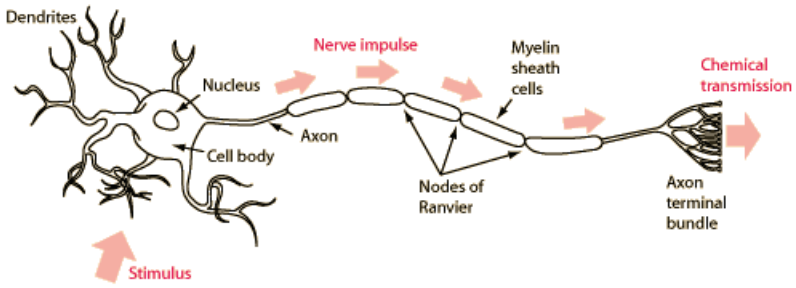


Figure 2.2: Structure of a nerve cell. Image from http://hyperphysics.phy-astr.gsu.edu/hbase/Biology/nervecell.html.

4 Clinical Forms

In Multiple Sclerosis, the course of the disease and the risk for developing permanent disability are very different from one patient to another and the prediction of long-term disability is still an open challenge. According to the current clinical standards four forms of MS are actually recognized (Figure 2.3) [Lublin *et al.* (2014), McDonald *et al.* (2001)].

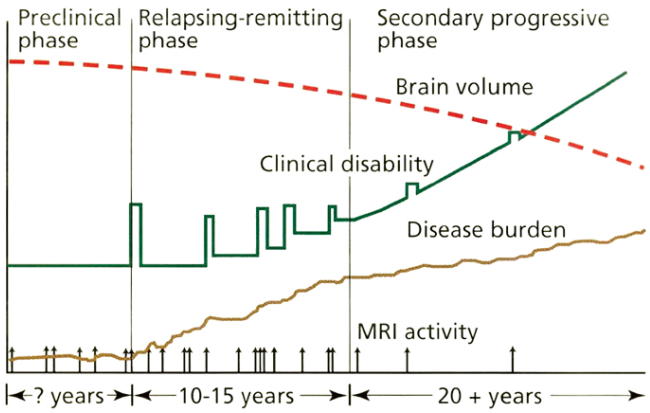


Figure 2.3: Classification of multiple sclerosis clinical forms according to the patients disability progression. Image from http://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/neurology/multiple_sclerosis/

4.1 Clinically Isolated Syndrome (CIS)

Clinically Isolated Syndrome (CIS), is recognized as the first clinical presentation of the disease, that shows characteristics of inflammatory demyelination that could be MS, but has yet to fulfil criteria of dissemination in time. This form of MS is the consequence of the series of one or two consecutive attacks from which the patients recovered completely, generally quite quickly, and without any progression or persistence of disability. Following this first attack, in the 85% of the cases, the progression to a relapsing remitting form (RR) occurs.

4.2 Relapsing Remitting (RR)

In this form of MS, It is characterized by acute episodes of exacerbations, followed by events of complete or partial remission. We often observe unpredictable attacks. During the presence of those new attacks new clinical symptoms appear or old clinical symptoms evolve. This phase has a variable duration and could be followed by a partial or total remission. At this stage, the pathology can be inactive for months or years.

4.3 Secondary Progressive (SP)

The Secondary-Progressive (SP) form is the evolution of the RR form, and is characterized by a permanent form of disability that gradually progresses in the time, independently of the presence of inflammatory attacks. This form involves 30% of MS patients.

4.4 Primary Progressive (PP)

Primary progressive (PP) form is characterized by the absence of inflammatory attacks. The patients suffer from an accumulation of deficits and disabilities. All these effects can be stable for certain periods or can progressively degenerate in months or years.

5 Diagnosis

The diagnostic criteria for MS have been continuously evolved since 1950's, and gained speed parallel to the development of detailed laboratory methods. The common aim for all the defined criteria up to now, is to establish the dissemination in space and time of the clinical picture caused by the lesions in the CNS, and to rule out other diseases which might mimic MS. At this time, there are no symptoms, physical findings or laboratory tests that can, by themselves, diagnose MS. The Revised McDonald Criteria, published in 2017 by the International Panel on the Diagnosis of Multiple Sclerosis, include specific guidelines for using MRI and cerebrospinal fluid analysis to speed the diagnostic process [Thompson *et al.* (2018), McDonald *et al.* (2001), Polman *et al.* (2011)].

First, differential diagnosis can be use diagnose MS. In medicine, a differential diagnosis is the distinguishing of a particular disease or condition from others that present similar clinical features. More generally, a differential diagnostic procedure is a systematic diagnostic method

used to identify the presence of a disease entity where multiple alternatives are possible. In MS, a systematic process for exclusion of alternative diagnoses is really hard to define, due to the large variety of MS symptoms. Moreover, since the symptoms of MS evolve and change with time, the use of the differential diagnosis is not sufficient to have a general overview of the pathology. An example of a differential diagnosis table for MS, from [Noseworthy *et al.* (2000)], is showed in Figure 2.4.

More complex and more effective methods can be used, compared to the differential diagnosis. MRI is the diagnostic tool with the best sensitivity in the search for MS lesions. It allows to identify lesions, their location, and to evaluate their dissemination in space. In addition, a longitudinal follow-up of the patient makes it possible to identify the appearance of new lesions and thus to control whether there is dissemination over time.

In order to have a full picture of the damages generated by MS in the brain three MRI sequences are needed to detect different type of lesions: T2 sequence, which allows to detect MS lesions as hyper-signal spots, T1 sequence where lesions are characterized by hypo-intensity signal spots and T1 sequence acquired after injection of a contrast agent (like Gadolinium) to detect regions where disruption of hematoencephalic barrier is present.

In [Barkhof *et al.* (1997), McDonald *et al.* (2001), Thompson *et al.* (2018), Polman *et al.* (2011)], MRI derived information were used to derive MRI based criteria for the diagnosis of MS. In more detail, to satisfy the presence of spatial dissemination of the lesions, at least two of this four criteria must hold:

- Lesion showing hyper-intensity in T2 sequence in periventricular region
- Lesion showing hyper-intensity in T2 sequence in near to the cortex
- Lesion showing hyper-intensity in T2 sequence in the supra tentorial region
- Lesion showing hyper-intensity in T2 sequence in the spinal cord

To satisfy the presence of temporal dissemination of the lesions the following criteria must hold:

- Lesion showing hyper-intensity in T1 acquired using contrast agent three months after one attack
- A new lesions showing hyper-intensity in T2 sequence three months after the first MRI exam
- If the first MRI exam was performed 3 months after the first attack, all the new lesions showing hyper-intensity in T2 sequence are considered as expression of a temporal dissemination

An overview of the spatial and temporal dissemination criteria are summarized in Figure 2.5.

6 Medical Treatment

There are different levels of treatment for the disease. The first consists of a re-education of the patient in order to allow him to recover a maximum of compromised functions or, if necessary,

Metabolic disorders

Disorders of B_{12} metabolism*
Leukodystrophies

Autoimmune diseases

Sjögren's syndrome, systemic lupus erythematosus, Behçet's disease, sarcoidosis, chronic inflammatory demyelinating polyradiculopathy associated with central nervous system demyelination, antiphospholipid-antibody syndrome

Infections†

HIV-associated myelopathy* and HTLV-1–associated myelopathy,* Lyme disease, meningovascular syphilis, Eales' disease

Vascular disorders

Spinal dural arteriovenous fistula*
Cavernous hemangiomata
Central nervous system vasculitis, including retinocochlear cerebral vasculitis
Cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy

Genetic syndromes

Hereditary ataxias and hereditary paraplegias*
Leber's optic atrophy and other mitochondrial cytopathies

Lesions of the posterior fossa and spinal cord

Arnold–Chiari malformation, nonhereditary ataxias
Spondylotic and other myelopathies*

Psychiatric disorders

Conversion reaction, malingering

Neoplastic diseases

Spinal cord tumors,* central nervous system lymphoma
Paraneoplastic disorders

Variants of multiple sclerosis‡

Optic neuritis; isolated brain-stem syndromes; transverse myelitis; acute disseminated encephalomyelitis, Marburg disease; neuromyelitis optica

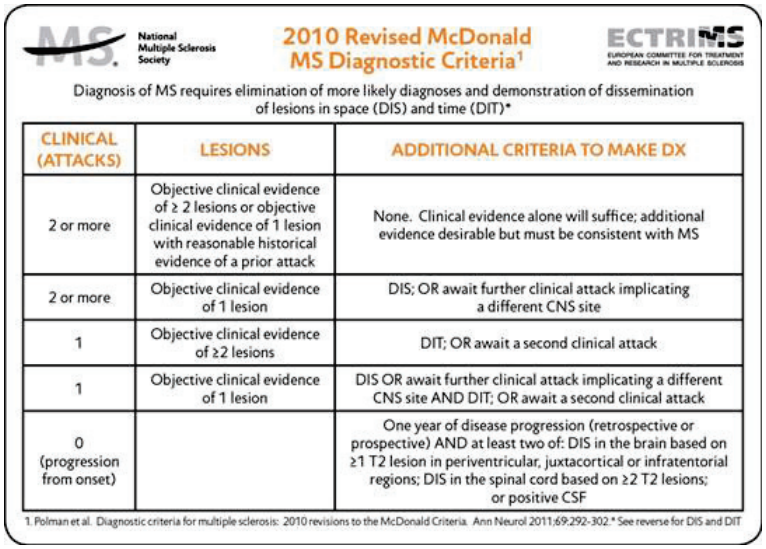
*This disorder or group of disorders is of particular relevance in the differential diagnosis of progressive myelopathy and primary progressive multiple sclerosis.

†HIV denotes human immunodeficiency virus, and HTLV-1 human T-cell lymphotropic virus type 1.

‡In many patients with these variants, clinically definite multiple sclerosis develops or the course is indistinguishable from that of multiple sclerosis.

Figure 2.4: Differential diagnosis table of Multiple Sclerosis. Table from [Noseworthy *et al.* (2000)].

to learn to live with the disability caused by the disease. To this aim, symptoms can be treated to improve the quality of life of the patients without directly affect the pathological process. For instance, for the spasticity injection of botulinum toxin or assumption of antispasmodics are used. Paint is usually treated with the assumption of analgesics or antiepileptic. The second is a treat-



National Multiple Sclerosis Society

2010 Revised McDonald MS Diagnostic Criteria[1]

ECTRIMS EUROPEAN COMMITTEE FOR TREATMENT AND RESEARCH IN MULTIPLE SCLEROSIS

Diagnosis of MS requires elimination of more likely diagnoses and demonstration of dissemination of lesions in space (DIS) and time (DIT)*

CLINICAL (ATTACKS)	LESIONS	ADDITIONAL CRITERIA TO MAKE DX
2 or more	Objective clinical evidence of ≥ 2 lesions or objective clinical evidence of 1 lesion with reasonable historical evidence of a prior attack	None. Clinical evidence alone will suffice; additional evidence desirable but must be consistent with MS
2 or more	Objective clinical evidence of 1 lesion	DIS; OR await further clinical attack implicating a different CNS site
1	Objective clinical evidence of ≥2 lesions	DIT; OR await a second clinical attack
1	Objective clinical evidence of 1 lesion	DIS OR await further clinical attack implicating a different CNS site AND DIT; OR await a second clinical attack
0 (progression from onset)		One year of disease progression (retrospective or prospective) AND at least two of: DIS in the brain based on ≥1 T2 lesion in periventricular, juxtacortical or infratentorial regions; DIS in the spinal cord based on ≥2 T2 lesions; or positive CSF

1. Polman et al. Diagnostic criteria for multiple sclerosis: 2010 revisions to the McDonald Criteria. Ann Neurol 2011;69:292-302.* See reverse for DIS and DIT

Figure 2.5: MS positive diagnosis criteria. Image from http://www.nationalmssociety.org/For-Professionals/Clinical-Care/Diagnosing-MS/Diagnosing-Criteria.

ment of the attacks, which goal is to reduce the number and the frequency of the attacks and then reduce the progression of the disability by increasing the remission. This type of treatment is mainly based on the assumption of corticoids usually *methylprednisolone*. Finally, the treatment of the pathology tries to control its evolution. The main idea is to reduce the demyelination by stimulation of the remyelination process. This treatment relies on the auto-immune nature of MS. Immunomodulators and Immunosuppressors drugs are often use to reduce the effect of MS. In the category of immunomodulators, we recall the interferon β and the Tysabri®. For the Immunosuppressors, mitoxantrone and cyclophosphamide are the most used.

7 Conclusion

MS is a complex disease whose evolution remains difficult to predict. As discussed in this chapter, its origin has not yet been clearly identified and pathological mechanisms are still subject of several researches worldwide. From the general introduction presented in this chapter, we can already see how cMRI acquisition are powerful tools for clinicians to diagnose and study the effect of MS on the CNS. Those results encouraged researcher on MS to use more complex MRI techniques like diffusion imaging and spectroscopy to obtain more sensitive and specific information on normal appearing WM useful for a deeper investigation of the pathology.

Chapter 3

Deep Learning

Contents

1 Introduction

Despite the significant results obtained by the current state of the art, given the complexity and the huge volume of biological data, many traditional computer science techniques and algorithms fail to solve complex biological problems in the real world. Nevertheless, modern computational approaches based on machine learning can address these limitations. *Machine Learning* is an adaptive process that enables computers to learn from experience, learn by example, and learn by analogy. According to T. Mitchell [Mitchell *et al.* (1997)], "a computer program is said to *learn* from experience E with respect to some class of tasks T and a performance measure P, if its performance at task T, as measured by P, improves with experience E". The experience comes in form of data flow usually referred to as *training set*. Machine Learning methods extract a mathematical model from the data to be used in making decisions or predictions in a specific task. Learning capabilities are essential for automatically improving the performance of a computational system over time on the basis of previous results. Such concept is particularly suitable in many context, including the biomedical domain. For example, it is not always easy for expert to say what rules they use for disease analysis and control, data are often noisy or missing, there is the need for redesigning systems whenever the environment changes, etc.

In general, every machine learning algorithm falls into one of three main categories: **supervised learning**, **unsupervised learning** and **semi-supervised learning**.

1.1 Learning Paradigms

Consider a training set, denoted as $\mathscr{D}$. The learning problem can be defined following three different paradigms:

- **Supervised learning**: this kind of learning is used when the answer to the problem is known. In supervised learning, we are able to provide a set of *samples* which have a known label or result. A *model* is prepared through a training process where it is required to make prediction and is corrected when those predictions are wrong. In other words, the algorithm tries to figure out what kind of math needs to be done in order to find a relationship. More formally, The training set is presented to the algorithm in the form $\mathscr{D} = < x,t >$, where each element $< x,t >$ is a pair of input and corresponding desired output coming from some unknown function f. The input variables x are also called *features* while the output variables t are also known as *targets* or *labels*. In a supervised setting the algorithm attempt to find a good approximation of f and use it to predict the output for new input. Depending on the type of the output variables, supervised-learning tasks can be further categorized:

  - *Classification*: The labels t are *discrete*. The algorithm's objective is to find a function f that maps an input x to a category whose label is y. The classification problem could be *binary* or *multi-class* depending on the number of classes that we want to predict that are two for the binary classification problem and more than two in the case of the multi class one.

- *Regression*: Targets t are *continuous* variables. Examples of regression problems are the prediction of the temperature in a building, or the prediction of the selling price of a specific product.
- **Unsupervised learning**: differently from supervised learning, input data are not labelled and we do not have any know result. The training set does not include the target variables but only the input ones $\mathscr{D} = < x >$. A model is prepared by deducing structures present in the input data. Problems of this kind are the one to discover groups of similar examples (*clustering*), or to give a new representation of the data in an high-dimensional space.
- **Semi-supervised learning**: the algorithm is trained upon a combination of labelled and unlabelled data. Typically, a small amount of labeled data is used to guide the research of structures present in the (mostly unlabelled) input data.

s

In the field of computer vision, such techniques represented a revolution. Several methods based on computer vision and image processing techniques to accomplish automatic identification of pathological mechanisms were developed. Such methods mostly exploit recent advances in machine learning. In more detail, Deep Learning methods have recently achieved a breakthrough in a variety of computer vision benchmarks, and are attracting a very strong interest within the computer vision community. Recently, many studies on extending deep learning approaches for graph data have emerged, also opening new perspective in the study of brain networks.

In this chapter, an overview of the main techniques used in this research work will be provided. We illustrate the concept of Deep Learning, with particular attention to the differences between the various existing algorithms, including graph-based techniques and their applications.

2 Artificial Neural Networks

In the last decade, a particular machine learning paradigm, the Artificial Neural Network (ANN), gained a lot of popularity, given their impressive results for the analysis and classification of images in a wide range of applications [Goodfellow *et al.* (2016)]. The study of ANNs has been inspired in part by the observation that biological learning systems are built of very complex webs of interconnected neurons. If we observe the human brain we discover that it is composed by million of little components called *neurons*. In rough analogy, ANNs are built out of a densely interconnected set of simple units, where each unit takes a number of real-valued inputs (possibly the outputs of other units) and produces a single real-valued output (which may become the input to many other units) [Mitchell *et al.* (1997)].

More formally, ANNs aiming to approximate some function f^*. For example, for a classifier a function $y = f^*(x, \theta)$ maps an input x to a category whose label is y. θ is the vector of parameters that the model learn in order to make the best approximation of f^* [Goodfellow *et al.* (2016)]. They are typically represented by composing together many functions. For example, we might have three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ connected in a chain, to form $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$.

In this case, $f^{(1)}$ is called the first layer of the network, $f^{(2)}$ is called the second layer, and so on. The final layer of an ANN is called the *output layer*. The overall length of the chain gives the *depth* of the model. It is from this terminology that the name "deep learning" arises. The learning algorithm must decide how to use the intermediate layers to best implement an approximation of f^*, by iteratively refine parameters θ. Because the training data does not show the desired output for each of these layers, these layers are called *hidden layers*.

In order to understand better what an ANN actually is, we can focus on its main component: the *Artificial Neuron*. Basically a neural network can be seen as an oriented graph, which nodes are called *neurons* and neurons are connected each other by weighted arcs. Each neuron can *fires* or *inhibits* other neurons, depending whether the value computed by its *activation function* exceeds a certain threshold. To get started, we focus on a type of artificial neuron called a perceptron. Perceptrons were developed in the 1950s and 1960s by the scientist Frank Rosenblatt, inspired by earlier work by Warren McCulloch and Walter Pitts [McCulloch and Pitts (1943)]. This simple architecture is a linear binary classifier that maps its input to a single binary value. Given a real-valued vector $\mathbf{x}$ as input and the corresponding real-valued weights $\mathbf{w}$, the output is

$$y = \begin{cases} 1, & \text{if } \mathbf{w} \cdot \mathbf{x} + \mathbf{b} \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

where the $\mathbf{w} \cdot \mathbf{x}$ is the weighted sum of the input vector

$$z = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} (w_i x_i), \tag{3.2}$$

and $\mathbf{b}$ is the *bias* term.

Although we are interested in learning networks of many interconnected units, let us begin by understanding how to learn the weights for a single perceptron. One of the easiest way to learn a weight vector is starting with random weight and iteratively, at each step t, for each example j, apply the perceptron on each example, modifying the weights whenever the example is misclassified. This process is repeated until the perceptron classifies all training examples correctly. Weights are modified at each step according to the perceptron training rule:

$$w_i(t+1) = w_i(t) + \Delta w_i, \text{ for all } 0 \leq i \leq n \tag{3.3}$$

where w_i is the i-th element of the n-dimensional weight vector and Δw_i is given by

$$\Delta w_i = \eta (t_j - y_j) x_{j,i} \tag{3.4}$$

In this formulation t_j is the desired output for the current example x_j, y_j is the prediction produced by the perceptron, and η is the *learning rate* which regulates the severity of the update. The $t_j - y_j$ term expresses the quality of the predictions with respect to the *ground truth* and it is referred to as the *error* E. Then, the perceptron updates its weights proportionally to the error E: the more the predictions are far from the targets the greater the correction to be performed in the

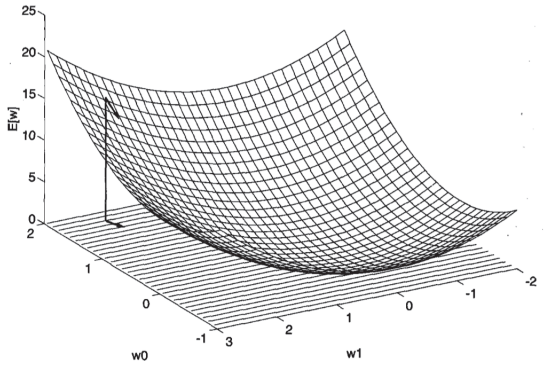


Figure 3.1: Hypothesis space of possible weight vectors and their associated error values.

weights. This procedure is repeated until the algorithm *converges* to a possible solution so that the found model classifies correctly all the examples.

In order to learn the weights of the model several *optimization* techniques could be applied. The following section will give an overview of the most used ones.

2.1 Optimization

One of the widely used learning algorithm in supervised learning is called *gradient descend*. o understand the gradient descent algorithm, it is helpful to visualize the entire hypothesis space of possible weight vectors and their associated E values, as illustrated in Figure 3.1

We can think the search space search space as a valley and we imagine a ball rolling down the slope of the valley. Our everyday experience tell us that the ball will roll to the bottom of the valley. Gradient descendant uses this idea as a way to find a minimum for the function randomly choosing a starting point for an (imaginary) ball, and then simulate the motion of the ball. We could do this simulation simply by computing derivatives of the cost function. Those derivatives would tell us everything we need to know about the local "shape" of the valley, and therefore the direction the ball will roll. More formally, this procedure takes the form:

$$\mathbf{w} = \mathbf{w} + \Delta\mathbf{w} \tag{3.5}$$

where

$$\Delta\mathbf{w} = -\eta\nabla L(\mathbf{w}) \tag{3.6}$$

The learning rate η is interpreted as the step size in the negative gradient direction and it governs the entity of the update. The choice of learning rate is critical since for too large η the gradient descent could "jump" the minimum of the function rather then approach to it. This problem could be alleviated by using a *scheduled* learning rate η_t that decreases over time steps.

Gradient descent has some drawbacks which make the training difficult. For example, the loss function is defined with respected to the whole training set, so we have to process all the data in batch before evaluating ∇L. Furthermore, the loss surface could have several local minima, so the

found solution could not be the global one. For this reason, many variations of the original gradient descent optimization were proposed. *Stochastic Gradient Descent* updates the weights after seeing only a single or a few training examples. In practice stochastic gradient descent has been proven to accelerate the learning, especially in the case of large data sets. It gives also the possibility to escape local minima since the stationary points could be different from one data point to another. Variations of the stochastic gradient descent optimization include:

- *Stochastic Gradient Descent with Momentum* ([Nesterov (1983)]): It consists in adding a *momentum* term in the update rule that has the role of taking memory of the update step at time $t-1$. The update rule becomes:

$$\mathbf{w}^t = \mathbf{w}^{t-1} + \Delta\mathbf{w}^t \tag{3.7}$$

  where

$$\Delta\mathbf{w}^t = \alpha\Delta\mathbf{w}^{t-1} - \eta\nabla L(\mathbf{w}^{t-1}) \tag{3.8}$$

  The effect of α is to force the gradient descent search to maintain the trajectory of the previous step. It also has the effect of promoting the "exploration" when the gradient is unchanging.

- *RMSProp* ([Tieleman and Hinton (2012)]): Also in this case the learning rate is adapted for each parameter. In RMSProp the learning rate for a parameter is normalized by the average of the recent gradients for that parameter.

- *Adam* ([Kingma and Ba (2014)a]): Adam is similar to RMSProp but in this case the learning rate is adapted using the average of both the gradients and their second moments.

2.2 Backpropagation

Multi-layer networks can not be trained using the perceptron's learning rule that needs to compute the error term between targets and predictions for the single unit. In the case of multi-layer networks this prerequisite is not verified because of the presence of multiple hidden-layers. The ground truth is available only for the last layer that outputs the predictions, so it is possible to evaluate the error only after the computation of the whole network. For this reason we need a new procedure able to carry the error information back to all the layers so that the network weights can be updated accordingly.

The solution is the backpropagation algorithm [Rumelhart *et al.* (1986)] which allows to compute the gradients for a general network, having arbitrary feed-forward topology and differentiable activation functions. The partial derivatives of the error with respect to a weight tells us the sensitivity of the error on its value. Then, computing the partial derivatives with respect to the weights of the last layer we can determine the amount by which each of its neuron contributing to causing the error.

Through the application of the *chain rule* it is possible to further compute the partial derivatives of the error with respect to the weights of the previous layer until the contribution of each

weight of the network to the error is determined. The backpropagation algorithm is then composed by three main steps:

1. *Forward* pass:

   (a) The information is propagated forward through the network to compute values from input to output for each neuron.

   (b) The error term is computed.

2. *Backward* pass: The error information is propagated backward through the network starting at last layer and recursively applying the chain rule until the input to compute the partial derivatives on all the weights.

3. Weights update: The partial derivatives are used by the gradient descent optimizer to update the weights of the network.

In order to fully understand the backpropagation procedure, let's consider a multi-layer network with sigmoid activations and the following error function for a single training example:

$$E_d(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{K}(t_k - y_k)^2 \tag{3.9}$$

where E_d is the error on training example d, k is the k-th output unit, K is the number of output units, t_k is the target value of unit k, and y_k is the output of unit k given a training example. Consider now a generic unit j, that could be either an output or hidden unit, and the following notation:

- x_{ji} = the i-th input to unit j
- w_{ji} = the weight associated with the i-th input to unit j
- $sum_j = \sum_i w_{ji}x_{ji}$ (the weighted sum of inputs for unit j)
- y_j = the output computed by unit j
- t_j = the target output for unit j
- σ = the sigmoid activation function
- $following(j)$ = the set of units whose inputs include the output of j

We want to obtain the partial derivatives $\frac{\partial E_d}{\partial w_{ji}}$ to be used in stochastic gradient descent for updating the weights. This will be done by showing the backward pass and weights update step of backpropagation. Start considering only the output units.

Backward pass for output unit j

The output unit j influences the previous layer through w_{ji}, w_{ji} influences the previous layer through sum_j. Thus, the partial derivative of the error on w_{ji} is given by the chain rule:

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial sum_j}\frac{\partial sum_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial sum_j}x_{ji} \tag{3.10}$$

Following the previous reasoning, sum_j influences the previous layer through y_j. Thus we can write:

$$\frac{\partial E_d}{\partial sum_j} = \frac{\partial E_d}{\partial y_j}\frac{\partial y_j}{\partial sum_j}, \tag{3.11}$$

where the first term is

$$\frac{\partial E_d}{\partial y_j} = \frac{\partial}{\partial y_j}\frac{1}{2}\sum_{k=1}^{N}(t_k - y_k)^2. \tag{3.12}$$

Since we are considering only the unit j the partial derivative becomes:

$$\frac{\partial E_d}{\partial y_j} = -\frac{1}{2}2(t_j - y_j) = -(t_j - y_j). \tag{3.13}$$

Consider now the second term $\frac{\partial y_j}{\partial sum_j}$. Since $y_j = \sigma(sum_j)$ it corresponds to the derivative of the sigmoid function that is equal to $\sigma(sum_j)(1-\sigma(sum_j))$. Therefore

$$\frac{\partial y_j}{\partial sum_j} = y_j(1-y_j). \tag{3.14}$$

The overall partial derivatives $\frac{\partial E_d}{\partial sum_j}$ become

$$\frac{\partial E_d}{\partial sum_j} = -(t_j - y_j)y_j(1-y_j), \tag{3.15}$$

where the negative of $\frac{\partial E_d}{\partial sum_j}$ is usually denoted as δ_j. Substituting in Equation 3.10 we obtain

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial sum_j}x_{ji} = -(t_j - y_j)y_j(1-y_j)x_{ji} = -\delta_j x_{ji}. \tag{3.16}$$

Weights update

Now we can compute the stochastic gradient descent update amount for output units as

$$\Delta w_{ji} = -\eta\frac{\partial E_d}{\partial w_{ji}} = \eta\delta_j x_{ji}. \tag{3.17}$$

Backward pass for hidden unit j

Consider now that j is a generic hidden unit. As above we want to compute the amount by which the unit contributes to the error that is expressed by the partial derivatives $\frac{\partial E_d}{\partial w_{ji}}$. Since the unit is located in the middle of the network it influences the error indirectly through the units

whose inputs include the output of j that we have defined as $following(j)$. Thus the gradients become

$$\frac{\partial E_d}{\partial w_{ji}} = \sum_{k \in following(j)} \frac{\partial E_d}{\partial sum_k} \frac{\partial sum_k}{\partial sum_j} x_{ji} \tag{3.18}$$

The $\frac{\partial E_d}{\partial sum_k}$ is the same computed in Equation 3.15 so we can write

$$\frac{\partial E_d}{\partial w_{ji}} = \sum_{k \in following(j)} -\delta_k \frac{\partial sum_k}{\partial sum_j} x_{ji} \tag{3.19}$$

Since the sum_k depends on the output y_j the above partial derivatives can be decomposed in the following way

$$\frac{\partial E_d}{\partial w_{ji}} = \sum_{k \in following(j)} -\delta_k \frac{\partial sum_k}{\partial y_j} \frac{\partial y_j}{\partial sum_j} x_{ji} \tag{3.20}$$

Now notice that sum_k is equal to $\sum_j w_{kj} x_{kj}$, x_{kj} is the output of the hidden unit y_j, and $y_j = \sigma(sum_j)$. The overall partial derivatives become

$$\frac{\partial E_d}{\partial w_{ji}} = \sum_{k \in following(j)} -\delta_k w_{kj} y_j (1 - y_j) x_{ji} \tag{3.21}$$

Weights update

From the following, the stochastic gradient descent update amount for hidden units is

$$\Delta w_{ji} = \eta y_j (1 - y_j) \sum_{k \in following(j)} \delta_k w_{kj} x_{ji}. \tag{3.22}$$

2.3 Activation Functions

The previous example considered a network composed by sigmoid units as in MLP. The sigmoid non-linearity is not the only possible choice. Indeed, each layer of a neural network can employ arbitrary activation functions and the choice of the non-linearity strictly depends on the kind of problem we want to afford. Typical choices are:

- *Sigmoid* or *Logistic*:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.23}$$

  The sigmoid non-linearity takes a real-valued number and "squashes" it into range [0, 1]. It produces outputs similar to the Perceptron step-function with the main difference of being a continuous function. Historically it has been frequently used but it has the main drawback to *saturate*: at either 0 or 1, the gradient is small (near zero). This is not a desirable behavior since during backpropagation it could cause the gradient to vanish, making learning difficult. In this context the initialization of the weights plays a fundamental role. For example too large weights could lead most of the neurons to saturate.

- *Hyperbolic tangent* or *Tanh*:

$$\tanh(z) = \frac{1-e^{-2z}}{1+e^{-2z}} = 2\sigma(2z) - 1 \tag{3.24}$$

The hyperbolic tangent is a continuous function that "squashes" a real-valued number into range [-1, 1]. Like the sigmoid, the tanh non-linearity suffers from the saturation problem, but it is *zero-center* and this is a desirable property for the training since it is proven to help learning. For this reason, tanh is usually preferred to the sigmoid non-linearity.

- *Rectified Linear Unit (ReLU)*:

$$\text{relu}(z) = \max(0, z) \tag{3.25}$$

ReLU non-linearity has become very used in the recent years. Many works prove that ReLU units accelerate the learning convergence with respect to sigmoid and tanh non-linearities. [Krizhevsky *et al.* (2012)] reached 6x improvement in convergence compared with the tanh. The main drawback of the ReLU units is that they can "die" during training. Suppose that most inputs to a ReLU unit are negative. This cause the gradients flowing through the unit to be zero and then each neuron going into the unit will stop to respond to the variation in the error. This problem could cause the "death" of several parts of the network.

3 Deep Learning Architectures

The way neurons and layers are connected determines the type of the architecture of the network. The choice about what type of neural network to use is strictly related to the problem we are facing. Indeed, different types of artificial neural network topographies are suited for solving different types of problems. In this section, a general overview of the main deep learning algorithms is presented. However, in this section, we will provide a more detailed description of the models defined to address the specific problems presented in this manuscript.

3.1 Convolutional Neural Networks

Concerning the image processing field, Convolutional Neural Networks (CNN) play a key role. Since their first definition by Yann LeCun *et al.* in 1999 [LeCun *et al.* (1999)], the impressive results obtained in various tasks allowed an exponential growth of their popularity. The idea of convolution is inspired partly by computer science and partly by biology. The particular structure of these networks achieves the skill of understand *translation invariance*, this means it has to relearn the identify of each object in every possible position. Furthermore, CNNs are fast to train and particularly well-adapted to classify images. Convolutional neural networks use three basic ideas: *local receptive fields*, *shared weights*, and *pooling*. Let's look at each of these ideas in turn. From now on is useful to consider the input layer as a square of neurons, where values correspond to the pixel intensities of the input image.

- **Local receptive fields**: Differently from a fully connected network, where each neuron is connected with every neuron of the previous layer, in a classic CNN architecture each neuron is connected only with small, localized regions of the input image. Such *local receptive field* is slided over by certain quantity of pixels (*stride*) to connect to a second hidden neuron.

- **Share weights and biases**: Convolutional neural networks uses the *same* weights and biases for each hidden neuron, so that the output of i,jth hidden neuron is given by:

$$\sigma\left(b+\sum_{l=0}^{n-1}\sum_{m=0}^{n-1}w_{l,m}a_{j+l,k+m}\right) \tag{3.26}$$

  Where n is the square root the local receptive field's side, σ is the activation function (usually a *sigmoid function*), b is the shared value for the bias, $w_{l,m}$ is a $n \times n$ vector of shared weights and finally $a_{x,j}$ denotes the input values. The equation 3.26 is also know as *convolution*. Shared weights and biases give to convolutional neural networks an important characteristic: all the neurons in the first hidden layer detect exactly the same *feature* (e.g. edges in the images or other types of shape) just at different locations in the input image (translation invariance principle). The map from the input layer to the hidden layer is known as *feature map*. The network structure we described so far has just one feature map and so it can detect just a single kind of localized feature. To do image recognition we will need more than one feature map. And so a complete convolutional layer consists of several different feature maps.

- **Pooling**: In addition to the convolutional layers just described, convolutional neural networks also contain *pooling layers*. They are used to simplify information coming from convolutional layers. In details they work combining the output of a feature map in a single smaller feature map.

In practice, when solving problems in the real world, these steps can be combined and stacked as many times to address different problems. For example, is possible to have two, three or even more convolution layers and data size can be reduced at each layer using pooling. The more convolution steps you have, the more complicated features your network will be able to learn to recognize. However, there exist many examples of data that lack the underlying low-dimensional grid structure. Is this the case for many domains, where data usually describe relationships, and can be naturally modelled by means of graphs [Scarselli *et al.* (2009)]. To recall, a graph is an ordered pair $\mathcal{G} = (V,E)$ where V is the set of nodes (representing objects of a particular domains) and E is the set of edges connecting nodes (representing relations between objects).

3.2 Graph Neural Networks

Since the first definition of Graph Neural Network (GNN) [Scarselli *et al.* (2009)], a huge effort was made to extend neural networks with the purpose of processing graph structures [Perozzi *et al.* (2014),Hamilton *et al.* (2017)]. Differently from previous existing architectures, this models

are defined to capture neighbourhood properties of the nodes. Such mechanisms allow to learn latent representation for the node through a mutual sharing of data among the nodes' neighbours in an iterative manner until convergence. More formally, the main idea behind GNN can be expressed as following: consider a graph $\mathcal{G} = (V, E)$. Each node $v \in V$ can be characterized by a set of features x_v. The goal of a generic GNN layer is to learn a d-dimensional latent representation of each node which encode information of its neighbourhood. Specifically,

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]})$$

where $x_{co[v]}$ denotes the features of the edges connecting with v, $h_{ne[v]}$ denotes the embedding of the neighbouring nodes of v, and $x_{ne[v]}$ denotes the features of the neighbouring nodes of v. The function f is the transition function that projects these inputs onto a d-dimentional space.

GNNs have found applications in many different domains, achieving interesting results, quickly adopting ideas from other successful areas of deep learning to evolve to the architecture we know today as Graph Convolution Networks [Kipf and Welling (2016)]. Promising results were achieved by properly managing structured data and capturing hidden information from graphs.

Concerning the study of neurological alterations, graph-based deep learning models have been defined to handle brain networks. Brain graphs, indeed, are different from social networks in the fact that their nodes cannot be directly described using features. In more detail, novel specific cases of more general convolutional filters that have a meaningful interpretation in terms of network topology have been defined, to directly process the adjacency matrix representation of brain network data. Interesting results were achieved in various tasks, including predicting neurodevelopmental outcomes [Kawahara *et al.* (2017)] and automatic discrimination of neuropsychiatric patients [Phang *et al.* (2019)], and paved the way to the characterization of either cognitive impairments or pathological alterations caused by dfferent brain diseases, including MS.

3.3 Recurrent Neural Networks

The Recurrent Neural Network (RNN) is a class of ANN where connections between neurons form a directed graph along a temporal sequence. This allows to model temporal dynamic behaviour. Unlike feedforward neural networks, RNNs maintain an internal state (memory) to process sequences of inputs. Roughly speaking, the decision of a recurrent network reached at time step $t-1$ affects the decision it will reach one moment later at time step t. So RNNs have two sources of input, the present and the recent past. The particular topology makes them applicable to sequence processing tasks such as speech recognition, text and video processing. More formally, the hidden state at time step t is h_t defined as:

$$h_t = \sigma(W x_t + U h_{t-1})$$

It is a function of the input at the same time step x_t, modified by a weight matrix W added to the hidden state matrix U. The weight matrices determine the importance to give to the present input and the past hidden state The error generated is used to adjust such weights until convergence. The

sum of the weight input and the hidden state is processed by the - usually non-linear - function σ.

Training RNNs is similar to training ordinary Neural Networks using the backpropagation algorithm but with some additional efforts. Since the parameters are shared by all the time steps of the RNN, the gradient at each output depends on the computation of the whole sequence. In order to backpropagate the error we need then to unroll the RNN and use brackpropagation from the last time step to the starting one, proceeding backward in time. This procedure is usually referred to as *Backpropagation Through Time (BPTT)*.

Chapter 4

Topic of the thesis

The main objective of this thesis is to take advantage from findings in recent network analysis [Kocevar *et al.* (2016), Rubinov and Sporns (2010), Sporns *et al.* (2005), Bullmore and Sporns (2009)] and propose a further investigation of the MS disease in terms of variations among clinical forms. To address this task, we exploit recent advances in computer vision, including graph representation of the brain structure in conjunction with deep learning techniques. Such combination represents a powerful tool to model and analyze the complex mechanisms underlying brain pathologies.

To this end, we first explored the capability of various deep learning algorithm at solving image processing tasks. These techniques have been proven capable of extracting highly meaningful statistical patterns in large-scale and high-dimensional datasets, showing great power and flexibility by learning to represent data as a nested hierarchy of concepts.

Next, a second group of experiments is proposed, concerning the exploitation of potential latent features behind brain networks by means of Deep Neural Networks. Graph-based techniques were explored, which combine local network descriptors with the graph representation of the brain structure. Such approach allowed to investigate the role of specific graph-based local metrics for the characterization of the MS disease. In more detail, supervised, semi-supervised and unsupervised techniques were defined to automatically classify MS patients in their respective clinical forms. Furthermore, a general logic-based framework was defined with the main goal of understanding which portions of connectome are significant during relapses of the disease, thus allowing to delineate a small group of vulnerable brain regions.

II Supporting Biomedical Analysis Using Deep Learning

Chapter 1

Biomedical Data Augmentation Using Generative Adversarial Networks

Contents

1 Introduction

The availability of a large amount of data is a crucial issue for applications in many domains. Indeed, proper data are essential in order to understand specific scenarios (for instance, useful information are extracted for predicting the evolution of systems or environments) and develop effective applications. This is especially the case of the biomedical domain; however, in such context collecting a significant amount of "good" data is not always an easy task, due, for instance, to the high costs in terms of money and time required to perform screenings and analyses, or, as in the case of certain pathologies, to the number of case study which is too limited for the creation of data banks large enough to train physicians, experts, or artificial models. A possible way to overcome limited availability, in some domains, is to artificially *create new data*. For many tasks, indeed, this can be achieved by modifying initially available data [Goodfellow *et al.* (2016)]. As an example, new instance images can be obtained by applying linear transformations (i.e., rotation, reflection, scaling, etc.) to already available ones. For instance, this approach is effective while training systems for automatic classification: a classifier just needs to take a complicated, high-dimensional input x and summarize it with a single category identity y, meaning that the main task a classifier has to perform is to be invariant to a wide variety of transformations. In this scenario, generating new $< x, y >$ pairs is just a matter of applying proper transformations to the x inputs in the training set [Goodfellow *et al.* (2016)]: operations like translating the training images a few pixels in each direction can often greatly improve generalization, and many other operations such as rotating or scaling the image have also been proven to be particularly effective for a the specific task of object recognition. Unfortunately, the same approach is not straightforwardly applicable to any task: for example, it is difficult to generate new "artificial" data for a density estimation, unless one has already solved the density estimation problem.

One of the most interesting alternatives, when dealing with image data, consists of learning the latent manifold on which the input images lie, and then sample realistic pictures (and their labels) from this manifold. We refer to *Generative* models as a class of machine learning algorithms which start from a training set consisting of samples drawn from a distribution, and learn how to represent an estimate of that distribution, or samples of it, to some extent. Furthermore Generative models can be trained with missing data and can provide predictions on inputs that are missing data. For example they can be integrated in semi-supervised learning approaches, in which the labels for many or even most training examples are missing.

One of the goals of our proposal is to apply such techniques to new *generative* areas, not explored so far. In particular, we present the application of a specific type of Generative Models, namely *Generative Adversarial Neural Networks* (GANNs), to the generation of new, unseen, MRI (Magnetic Resonance Imaging) slices of the human brain. Interestingly, the model produces samples very similar to real MRI slices, which present realistic features. To the best of our knowledge, this is one of the first attempts to apply GANNs to biomedical imaging, and brain images in particular. We validated the generated images both analytically and via an ad-hoc web platform, where human physicians and experts have been invited to distinguish wether an image is real or artificially generated.

The remainder of the chapter is structured as follows. In Section 2 we illustrate the related literature, and then, in Section 3 we provide a detailed description of our approach. Section 4 presents our experimental evaluation and eventually, in Section 5 we draw our conclusion.

2 Related Works

Artificial generation of natural images is a widely studied task in machine learning, and constituted an ambitious goal for many years [Goodfellow *et al.* (2016)]. Several efforts have been spent to solve the problem of generating realistic high-resolution images, and several novel approaches [Dosovitskiy *et al.* (2015), Dosovitskiy and Brox (2016)] have already been proven to be well-suited for generating realistic images which look very similar to the ImageNet dataset [Russakovsky *et al.* (2015)], also achieving impressive results at high resolutions.

As for the biomedical domain, the problem of automatically generated unseen instances has been addressed by means of many different techniques. Many studies focussed on the reconstruction or the synthesis of an image starting from some initial data [Nie *et al.* (2017)], on the synthesis of a source MRI modality to another target MRI modality [Sevetlidis *et al.* (2016)], on the generation of multi-modal medical images of pathological cases based on a single label map, also outperforming the state-of-the-art methods it has been compared against. Closer to our proposal, in [Costa *et al.* (2017)], the Authors tested the capability of GANNs in generating high quality retinal fundus images from pairs of retinal vessel trees and corresponding retinal images. We use the method proposed in [Denton *et al.* (2015)], as an attempt to generate high-quality MRI slices in order to augment biomedical datasets with a fast and inexpensive method.

3 Proposed Approach

In the following, we describe the background techniques and methods, and provide further details on the proposed approach.

3.1 Generative Adversarial Neural Networks

Generative Adversarial Neural Network is a generative model approach based on differentiable generator networks [Goodfellow *et al.* (2016)]. GANNs are conceived for scenarios in which the generator network must compete against an adversary, in a sort of forger-police relation. Two actors are involved: the *Generator* network (the "forger"), which directly produces samples $x = g(z, \theta^{(g)})$, where g is a given probability distribution that describes the training set; the *Discriminator* (the "police"), that attempts to distinguish between samples taken from the original data and samples drawn from the Generator; in other words, it estimates a probability value given by $d(x, \theta^{(d)})$, indicating the probability that x is a real training example rather than an artificial sample drawn from the model. The best way to describe the GANN training process is as a zero-sum game as defined in game theory, in which the Discriminator's Generator's payoffs are $v(\theta^{(g)}, \theta^{(d)})$ and $-v(\theta^{(g)}, \theta^{(d)})$, respectively. During the learning process, each player attempts to maximize its

own payoff; in such a scenario, the Discriminator is called examine an image and estimate whether it is "real" (i.e., taken from the training set) or "artificial" (i.e. generated by the algorithm). This means that it must learn some general rules that govern the distribution until, at convergence, the Discriminator is no more able to distinguish Generator's samples from real data, so its output is $\frac{1}{2}$ everywhere. On the other side, the Generator should learn how to generate images that look more and more similar to the samples from the training set, in order to fool the Discriminator and make it believe that they are real.

3.2 Laplacian Pyramid of Adversarial Networks

Several GANN models exist. As a first attempt, we take advantage from a recent optimization method which uses a cascade of convolutional networks within a Laplacian pyramid framework (LAPGAN), in order to generate images in a coarse-to-fine fashion [Denton *et al.* (2015)]. The goal is achieved by building a series of generative models, each one able to capture image structure at a particular scale of a Laplacian pyramid.

This approach allows to first generate a very low-resolution version of an image, and then incrementally add details to it. The Generator (G) and the Discriminator (D), indeed, are not trained directly on full-sized images: the training starts with a downsampling at a minimum size which is increased (e.g., doubled) during multiple steps, until the final size is reached. During these steps another pair of G and D are trained to learn good refinements of the upscaled images. This means that G learns how to improve the quality of its input, adding good refinements, and, at the same time, D learns how refined images look like. It is worth noting that this methodology is very closely related to the one a human being typically employ to draw images: start with a rough sketch, and then progressively add more and more details.

3.3 Generating MRI slices of the brain

Our approach uses a GANN to automatically generate MRI slices of the brain; in our work the architecture described in [Denton *et al.* (2015)], public available on github[1], was maintained "as is", except for the size of the output, which has been increased by adding one more convolutional layer. The framework makes use of another convolutional neural network, the *Validator*, in charge of assigning validation scores to generated images and trained once before the Generator network. Artificial images used to train the Validator are created by applying some transformations to real images. These techniques are sometimes combined with each other.

Both the Generator and the Discriminator are convolutional networks trained with stochastic gradient descent, where Adaptive Moment Estimation is used as optimizer. The architecture of the Generator is basically a full laplacian pyramid in one network; it starts with a linear layer, which generates 16×8 images, followed by upsampling layers, which increase the image size to 32×16, 64×32 and then 128×64 pixels. The Discriminator is a convolutional network with multiple branches. Rotations are removed by means of spacial transformer at the beginning; three out of the four branches have also spatial transformers (for rotation, translation and scaling), so they can

[1] https://github.com/aleju/sky-generator

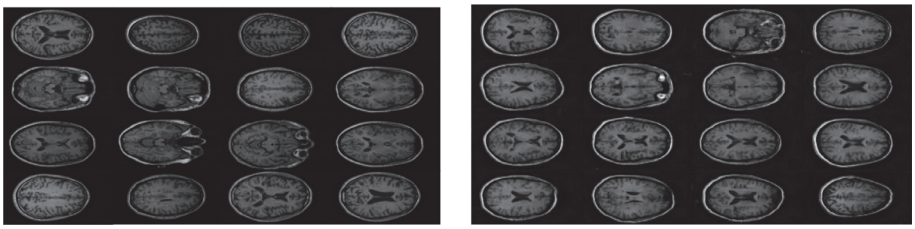


Figure 1.1: Real (left) images compared with artificial (right) images

learn to focus on specific areas of the image. The fourth branch is intended to analyze the whole image.

4 Experimental Analysis

4.1 Dataset description

The dataset consists of 46,737 images representing MRI slices extracted from 77 subjects. Each subject underwent 8 MR scans. The MR protocol consisted in the acquisition of a sagittal 3D-T1 sequence ($1 \times 1 \times 1 mm^3, TE/TR = 4/2000 ms$). In order to have "pure" MRI images, no post-processing was applied to the images.

4.2 Training phase

The training process consists of multiple steps: at first, the Validator and the Generator are trained for a predefined number of steps, 50 and 10, respectively; then, the real training process, i.e., the zero-sum game, starts. Results shown in the following refer to 800 epochs of training. Since images are defined on a grayscale, the framework was used with the *grayscale* parameter enabled, so that only one input channel was used during the operations. In order to perform all the tests, the following workstation was used: x86_64 CPU(s), Intel(R) Xeon(R) CPU E5440 @ 2.83GHz, Linux Debian 4.8.4-1, CUDA compilation tools, release 7.5, V7.5.17, NVIDIA Corporation GK110GL on Tesla K20c.

4.3 Evaluation

Rigorous performance evaluation of GANNs is an important research area, since is not clear how to quantitatively evaluate generative models [Goodfellow *et al.* (2016)]. Indeed, finding an images evaluation method in such a context is not straightforward. When using statistical methods, for instance, it might not be sufficient to look at probability distributions among pixels or part of the images, as "geometric" relations are crucial. Also, the task is quite different from a clustering or classification problem, as the point is to find what kind of "features" allow one to tell if an image is eligible to stay within a given group or not, and not only judge "similarities". This is why, besides quantitative tests, we also conduct human evaluation to evaluate the quality of the generated images [Zhang *et al.* (2017)a].

Table 1.1: Likelihood comparison

Samples	Real	Generated
100	37.31	33.90
1000	34.30	33.92
10000	34.02	33.93

Table 1.2: Inception Score comparison ($\pm$ standard deviation in parenthesis)

Test	Real	Generated
100	1.92 ($\pm$0.26)	1.80 ($\pm$0.29)
1000	1.79 ($\pm$0.06)	1.89 ($\pm$0.08)
10000	1.80 ($\pm$0.03)	1.93 ($\pm$0.03)

4.4 Quantitative image quality assessment

We evaluate our approach by means of two different quantitative methods: *(i)* Estimating the distributions of the real and the generated datasets by means of the Kernel Density function and comparing their likelihood; *(ii)* comparing the Inception Score of the two datasets. For the sake of the present work, for each metric we considered two distributions similar if their distance, in terms of score, is below the empirical threshold of 10%.

Kernel Density function

The approach based on Kernel Density function to evaluate generative models was originally introduced in [Breuleux *et al.* (2011)] and applied on GANNs in [Goodfellow *et al.* (2014)]. The method estimates the probability of the generated dataset, by fitting a Gaussian Parzen window to the generated samples and reporting the likelihood under this distribution. The bandwidth of the Gaussians is obtained by cross-validating the validation set. In our approach, we compute the similarity between the two datasets estimating their distribution by means of the Kernel Density function, so that similar datasets should be represented by similar distributions. Figure 1.2 shows the comparison of the density distribution and the estimated Cumulative Density function under real and generated datasets. As observable, generated images present very similar features with respect to the real dataset.

Inception Score

Inception Score [Salimans *et al.* (2016)] is an automatic method to evaluate samples which is found to correlate well with human judgement. The probability $p(y|x)$ is estimated by applying the Inception model [Szegedy *et al.* (2016)a] to every image in the dataset, so that images belonging to the same distribution should have low entropy. Consequently, if the generated images are distant from the estimated distribution, the marginal should present high entropy. These assumption are used to compute the Inception Score according to the equation $IS = exp(\mathbb{E}_x KL(p(y|x)||p(y))))$, where KL is the Kullback-Leibler divergence and results are exponentiated so the values are easier to compare. In this work we compare the Inception Score computed on both generated and real dataset. Tables 1.1 and 1.2 report a comparison of the likelihood and the Inception Score between both the estimated distributions over 100, 1000 and 10000 samples. Similar scores were obtained when comparing the generated and the real dataset.

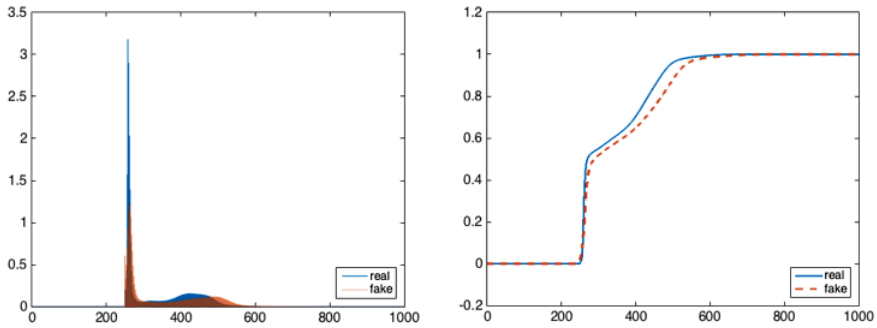


Figure 1.2: Density function (left) and Cumulative Density function (right) comparison of generated (orange) and real (blue) datasets.

4.5 Human evaluation of generated images

The quality assessment of the generated images are evaluated by means of a web platform[2], where physicians and experts are called to distinguish between real and artificial images. More in detail, two sets of 100 images (both 100 real and 100 artificial) was prepared. Each user is proposed, one at a time, images from a set of 20 randomly extracted from the two sets with, probability $\frac{1}{2}$. During each trial, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) are collected, where *positive* is used to indicate real images and *negative* is referred to artificial ones. In order to assess the quality of the delineation, we compute Accuracy ($Acc = \frac{TP}{P+N}$), Precision ($Prec = \frac{TP}{TP+FP}$), Recall ($Rec = \frac{TP}{TP+FN}$) and F1-score ($F1 = \frac{2*Prec*Rec}{Prec+Rec}$) where P is the number of positive samples and N the number of negative ones. We collected 15 tests performed by different experts in neuroimaging field; they achieved (on average) an accuracy, precision, recall and F1-score, to discriminate between real and "artificial" images, of 0.52 ± 0.16, 0.55 ± 0.23, 0.58 ± 0.21, and 0.53 ± 0.17, respectively.

Results obtained after human evaluation show the capability of our method to generate "artificial" MR images similar to real one. The difficulty to differentiate between the two is well underlined by the low values of F1-score obtained by humans in the tests. Furthermore, in order to have a more detailed feedback, we also asked our experts to write down comments describing how did they tell the difference between real and "artificial" images. Among others, we received two interesting observations: the first is about grey and white matter tissues contrast, while and the second about image symmetry. Those two limitations are indeed noticeable in the generated images: they present a low level of contrast between the two tissues, and an high symmetry between the two hemispheres. Based on the quantitative results and the comments obtained from the experts, we can say that our method is definitely appropriate, and still features significant room for improvement.

5 Conclusion

In this chapter we have shown the feasibility of learning to perform the synthesization of unseen high-quality MRI slices of the human brain by means of GANNs. The aim of the study

[2] www.tinyurl.com/mrichallenge

was to ease inexpensive and fast augmentation of biomedical datasets, in order to overcome the lack of real images and allow physicians and machine learning algorithms to take advantage from new instances for their training.

Applications of GANNs have been just started to be studied in literature, and a large variety of applications are still open. As future work, we aim to improve the quality of the generated images, to be more and more similar to real MRI scans; to this aim, a comparison with alternatives models, such as Deep Convolutional Generative Adversarial Networks (DCGAN) [Radford *et al.* (2015)], will be of clear interest, as they are currently emerging in literature. Furthermore, the generation may be improved by allowing the network to add pathological symptoms and provide unseen data of synthesized patients; this might also improve the study of rare diseases. Another perspective is the combination of the generated slices in order to compose a three-dimensional MRI. Eventually, we are planning to better investigate quality and use of quantitative measures for the assessment of the methods.

Chapter 2

Optic Disc Detection Using Fine Tuned Convolutional Neural Networks

Contents

1 Introduction

It is well known in the medical literature that retinal changes reflect systemic microvascular damage associated with different pathological conditions, such as hypertension or diabetes [Cavallari *et al.* (2015)]; interestingly, analysis of such changes can be performed by using non-invasive techniques, like fundoscopy.

Fundoscopy allows to obtain high-definition images of the retina in a few seconds, and constitutes a perfect test in emergency rooms to assess the presence of serious diseases. However, full exploitation of fundoscopy in clinical settings is limited because quantitative information can hardly be obtained through the observer-driven evaluations currently employed in routine clinical practices [Cavallari *et al.* (2015)]. Hence, the development of algorithms capable to extract quantitative information from those images is crucial for the automatic extraction of useful information from fundus images.

Several approaches have been proposed in literature in the last decade [Wong *et al.* (2004)] [Cavallari *et al.* (2015)]. It is worth noting that optic disc (OD) identification is a fundamental step in the analysis of human retina image analysis, as it allows to identify important regions in fundus images. Those regions could be exploited as seed points for semi-automatic algorithms to perform a quantitative analysis of the retinal damage.

Due to the large variations in fundus imaged object appearance, OD detection is a challenging problem of computer vision. Common variations are given by the changes in viewpoint, illumination and intra-class variance [Aytar (2014)]. Many of the methods present in literature are based on two assumptions on the images: *(i)* the optic disc tends to be much brighter than its surroundings, *(ii)* the optic disc is the ultimate source of retinal blood vessels [Lim *et al.* (2015)]. In contrast, proper machine learning techniques, like Deep Learning, in OD detection, make the algorithm able to automatically infer rules for solving this task. In addition, this "data-driven" approach allows to overcome problems related to the quality of the image and the exposure, considering that the disc could be hidden in the image because of several diseases.

Current trends in research proved the effectiveness of convolutional neural networks (CNNs) for solving object recognition tasks, automatically identifying features from large collections of images, with minimum error [Krizhevsky *et al.* (2012)] [Szegedy *et al.* (2016)b]. Moreover, CNNs are fast to train and particularly well-adapted to classify images.

In practice, in many real-world applications, it is not always possible to have a dataset of sufficient size to create a high-performance learner for a target domain from a related source domain [Weiss *et al.* (2016)]. Especially in some scenarios (such as in the case of the biomedical domain), obtaining training data that matches the feature space and predicted data distribution characteristics of the test data is often difficult and expensive. Thus, in order to overcome this limitation, it is common to pre-train a neural network on a very large dataset, and then use the network either as an initialization or a fixed feature extractor for the task of interest. This methodology is known as *Transfer Learning* [Pan and Yang (2009)].

We present here a supervised method for the automatic detection of the Optic Disc in retinal fundus digital images, which properly reuses previous knowledge from a Convolutional Neural

Networks (CNN) pre-trained to detect faces in an image. We illustrate the outcome of an experimental activity that showed high level of accuracy on the *DRIVE*[1], *STARE*[2] and *DRIONS*[3] databases.

The remainder of the chapter is structured as follows. In Section 2, we illustrate related literature and then, in Section 3, we provide a detailed description of our approach. In Section 4 we present our experimental activities, discussing the results in Section 5. Eventually, in Section 6 we draw our conclusion.

2 Related Works

The OD is a yellowish or white circular, sometimes elliptic, region present in color fundus images. It is the entrance of blood vessels and the optic nerve into the retina which size varies from patient to patient. Despite its well-known visual properties, OD detection remains a challenging task because of several issues related to the quality of the image acquired, such as differences in (high or low) levels of brightness, contrast, or saturation. Moreover, several retinal pathologies can hide or modify the appearance of the optic disc and the presence of exudates and other shapes on funds surface can generate confusion in a way such that methods are likely to fail.

Several attempts were provided in order to automatically locate and segment the optic disc and cup. This was usually achieved by taking advantage from many well-known properties of the optic disc, such as its grey level variation along with the high contrast between it and its surrounding [Sinthanayothin *et al.* (1999), Goh *et al.* (2001), Pallawala *et al.* (2004)]. However, this approach has been shown to work well, unless the presence of occlusion due to pathologies like exudates which also appears bright and well contrasted. In [Aquino *et al.* (2010), Yin *et al.* (2012), Tamura *et al.* (1988)], the OD is localized using by means of circular Huge transform, i.e., computing the gradient of the image, determining the best fitting circle. Unfortunately, this approach is quite time consuming and it does not consider that the OD is sometimes partially hidden in the image plane, and so the shape is far from being circular or even elliptic. Finally, in [Xu *et al.* (2013)] one of the best optic cup segmentation results has been achieved by means of optimal linear reconstruction with codebooks.

Fundus images reveal a complex mixture of visual hidden patterns. These patterns could be only observed by skilled human graders, and can potentially be time-consuming. However, the best way of translating this knowledge into an automated procedure still remains an open problem. In contrast to "analytical" methods, which attempt to exploit well-known features of the OD, another relative recent branch of methods is taking place, namely *machine learning based approaches*; these attempt to automatically detect useful properties in order to overcome the above limitations. In [Lim *et al.* (2015), Chen *et al.* (2015)] novel approaches to automatically detect these features by means of Convolutional Neural Networks have been proposed, which have recently been successfully employed for many image segmentation and classification tasks.

[1] http://www.isi.uu.nl/Research/Databases/DRIVE/
[2] http://cecas.clemson.edu/~ahoover/stare/
[3] http://www.ia.uned.es/~ejcarmona/DRIONS-DB.html

3 Proposed Approach

In the following, we describe the background techniques and methods, and provide further details on the proposed approach.

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are particular types of artificial neural network which are biologically inspired by the visual cortex. They are supervised feature learner able to learn complex invariances and patterns with extreme variability, with robustness scaling, shifting, translation, rotation and distortion of the input. Their particular architecture was designed to recognize visual patterns directly from pixel images with minimal preprocessing.

As already described previously, CNN consists of a sequence of layers, each of which make use of a differentiable function to transform one volume of activations to another. Typically three main types of layers are used to build CNN architectures: Convolutional Layers, Pooling Layers, and Fully-Connected Layers. *Convolutional layers* can be seen as sets of filters which are able to extract some specific patterns and particular features from an image. The output of a convolutional layer is a feature map which describes particular characteristics of the image. All neurons in a feature map share the same set of weights and the same bias, so neurons in a feature map are able to detect the same feature. The other feature maps in this layer use different sets of weights and biases, also reducing the number of free parameters, so different types of local features are extracted. *Pooling layers* work as a feature selection layers. They progressively reduce the spatial resolution of each feature map in order to reduce the amount of parameters and the computation in the net by means of local averaging and a subsampling, hence controlling overfitting. *Fully-connected layers* are usually placed at the end of the structure to perform high-level reasoning. They take a vector of neurons from the previous layer as input, and connects it to all its neurons, actually producing a one-dimensional vector. This structure makes CNNs particularly well suited for image recognition and object detection tasks.

3.2 Transfer Learning and Fine-Tuning

Transfer Learning overcomes the assumption, typical of traditional machine learning methodologies, on the fact that training and testing data are supposed to be taken from the same domain, so that the input features space and data distribution characteristics are the same.

A common real-world example that helps at understanding why transfer learning is possible and actually effective is the following. Let us consider two people who want to learn how to play a musical instrument: the piano, for instance; interestingly, one of the two persons has no previous knowledge about music, while the other has been studying guitar since he was child; thus, the latter already has an extensive musical background. Intuitively, the guy with such a background will be able to learn the piano in a more efficient way by *transfer* his previous knowledge to the new task [Weiss *et al.* (2016)].

In order to formalize the definition of Transfer Learning, in this paper we use the notation

defined in [Pan and Yang (2009)]. More in detail, let D be a domain composed by a feature space $\mathscr{X}$ and a marginal probability $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathscr{X}$. For each domain D we can define a task $\mathscr{T}$ composed by a label space $\mathscr{Y}$ and a predictive function $f(\cdot)$ which is inferred from each pair $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in \mathscr{Y}$. Now, let D_s be the *source* domain data and $\mathscr{T}_s$ be the source task. Let D_t be the *target* domain data and $\mathscr{T}_t$ be the target task, Transfer Learning can be formally defined. Given a source domain D_s along with the corresponding source task $\mathscr{T}_s$ and a target domain D_t with a corresponding target task $\mathscr{T}_t$, Transfer Learning is the process of improving the target predictive function $f_t(\cdot)$ by using the related information from D_s and $\mathscr{T}_s$, where $D_s \neq D_t$ or $\mathscr{T}_s \neq \mathscr{T}_t$ [Weiss *et al.* (2016)].

Fine-tuning is one of the major Transfer Learning scenario, which consists in transferring knowledge by training a CNN to learn features for a source domain yet trying to minimize the error in that domain. Hence, the network is trained again on another, more specific, domain. In this setting, we are transferring features and parameters of the network from the source domain to the specific one [Reyes *et al.* (2015)].

3.3 Reinspection Algorithm Model Overview

The model proposed in [Stewart *et al.* (2016)] provides an end-to-end approach for directly generating as output a set of boxes that, once superimposed over the input image, bound specific objects appearing in it. Briefly, the architecture converts an intermediate representation of an image, obtained by means of expressive image features from GoogleLeNet [Szegedy *et al.* (2016)b], into a set of boxes using a recurrent neural network with LSTM units (Figure 2.1). As described by the authors, the LSTM units can be seen as a "controller" that propagates information between decoding steps, and controls the location of the next output. It has been trained to face detection in crowded scenes, and achieved impressive results also in situations where distinguishing nearby individuals is particularly challenging.

More in details, each image is transformed into a grid of 1024 dimensional features descriptors. This vector is somehow representative of the contents of a region, and also holds proper information about the position of the object in that region. At each step, the LSTM predicts a new bounding box $b = \{b_{pos}, b_c\}$, where $b_{pos} = (b_x, b_y, b_w, b_h) \in \mathbb{R}^4$ and $b_c \in [0,1]$ is the confidence that a person is found at that location. Boxes are produced in order of descending confidence values until the LSTM is unable to find another box with a confidence greater than a given threshold. The memory states allow to generate the next hypothesis depending on the previous one. A specific loss function suitable for guiding the learning process towards the desired output has been defined.

3.4 The GoogLeNet-OverFeat algorithm

The proposed model takes advantage from a novel feature extractor called "OverFeat", which is based on the approach proposed in [Sermanet *et al.* (2013)], where the winner of the localization task of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013) is introduced, that obtained very competitive results in the detection and classifications tasks.

Two models are proposed: *fast* and *accurate*, where the second achieves better results with respect to the first, yet requiring, however, much more connections.

GoogLeNet-Overfeat uses the same fixed input size approach proposed by Krizhevsky *et al.* in [Krizhevsky *et al.* (2012)]. Each image is reduced to a minimum size of 256 pixels; then, 5 crops of size 221×221 in a random position are extracted and presented to the network in mini-batches of size 128. Table 2.1 reports some details about the architecture.

3.5 Fine Tuning for Optic Disc Detection

It is known that the OD presents some specific visual properties, such as its circular shape and its variation of brightness with respect to the background. Furthermore, its position varies among precise regions of the eye, and it is the origin of the blood vessels. Unfortunately, all these well-defined features are extremely difficult to combine in the definition of precise algorithmic rules, due to the presence of a large amount of variations in colors, brightness and saturation; in addition, several disorders might cause the presence of defects in the retina: well-demarcated lesions, spots and scar tissue often appear, such that distinguishing between them and the OD become an hard task (see Figure 2.4 for some examples).

In order to successfully face these issues, CNNs can be used, as they constitute a "natural" tool for extracting features from images, and can infer such rules for automatically detecting the OD. Although the real meaning of features learned by the neural network is still unknown, it is reasonable to believe that they represent concepts such as shape, color, proportions, and relationships between objects of an instance.

Our approach uses the ability of the network defined above for abstracting the features learned from the source task. Fine-tuning is then applied in order to transfer such background knowledge to the task of detecting the OD in retinal fundus images. Since training a CNN from scratch can be very time consuming, the model was fine-tuned in order to speed-up the training process. In our experiments the main architecture implemented by [Stewart *et al.* (2016)] was maintained "as is", except for the size of the input images. Our model takes as input a retinal fundus image of size 704×576 and process via the three stages described above, hence generating as output a set of boxes along with their confidence values. The box with the highest value of confidence is the one which bounds the OD with the highest level of accuracy. Since the OD in a retinal fundus image is unique, bounding boxes with a confidence value lower than a predefined threshold (0.8) were removed in order to avoid misclassifications due to the presence of exudates or other brightness shapes; only the one with the higher level of confidence was considered as representative of the position of the OD.

4 Experimental Analysis

4.1 Dataset description

Fine-tuning was performed using the publicly-available structural analysis of retina (*STARE*) database, which provides a set of retinal fundus images along with the location of their optic disc.

Table 2.1: GoogLeNet model architecture

Layer	1	2	3	4	5	6	7	8	Output
Stage	conv + max	conv + max	conv	conv	conv	conv + max	full	full	full
#channels	96	256	512	512	1024	1024	4096	4096	1000
Filter size	7x7	7x7	3x3	3x3	3x3	3x3	-	-	-
Conv. Stride	2x2	1x1	1x1	1x1	1x1	1x1	-	-	-
Pooling size	3x3	2x2	-	-	-	3x3	-	-	-
Pooling stride	3x3	2x2	-	-	-	3x3	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	221x221	36x36	15x15	15x15	15x15	15x15	5x5	1x1	1x1

Table 2.2: Performances in terms of Recall (Rec), True Positives (TP) and mean Dice Score ($\overline{DSC}$) ($\pm$ its standard deviation in parenthesis) obtained with $Learningrate = 0.0003$ and different Epochs.

	DRIONS			**DRIVE**		
Epochs	**Rec**	**TP**	$\overline{DSC}$	**Rec**	**TP**	$\overline{DSC}$
1000	100	110	0.97 ($\pm$0.52)	97.5	39	0.43 ($\pm$0.19)
900	100	110	0.99 ($\pm$0.15)	65.0	26	0.43 ($\pm$0.19)
800	100	110	0.99 ($\pm$0.09)	90.0	36	0.64 ($\pm$0.20)
700	100	110	0.98 ($\pm$0.28)	72.5	29	0.60 ($\pm$0.23)
500	100	110	0.99 ($\pm$0.05)	52.5	21	0.52 ($\pm$0.15)
300	96.4	106	0.94 ($\pm$0.32)	47.5	19	0.49 ($\pm$0.21)
100	0	0	0	0	0	0

STARE dataset consists of 400 PPM images, digitized slides captured by a TopCon TRV-50 fundus camera with 35 degree field of view. Each slide was digitized to produce 605×700 pixel image with 24 bits per pixel.

Two different publicly-available databases are used as validation sets: the "test" *DRIVE* dataset and the *DRIONS-DB* dataset. The "test" DRIVE dataset consists of 40 images acquired by means of a *CanonCR*5 non-mydriatic 3*CCD* camera with a 45 degree field of view (FOV). Each image was captured using 8 bits per color plane at 768×584 pixels. The FOV of each image is circular with a diameter of approximately 540 pixels. For this database, the images have been cropped around the FOV. For each image, a mask image is provided that delineates the FOV.

The DRIONS-DB consists of 110 color digital retinal images affected by potential problems that may distort the detection process of the papillary contour. The mean age of the patients was

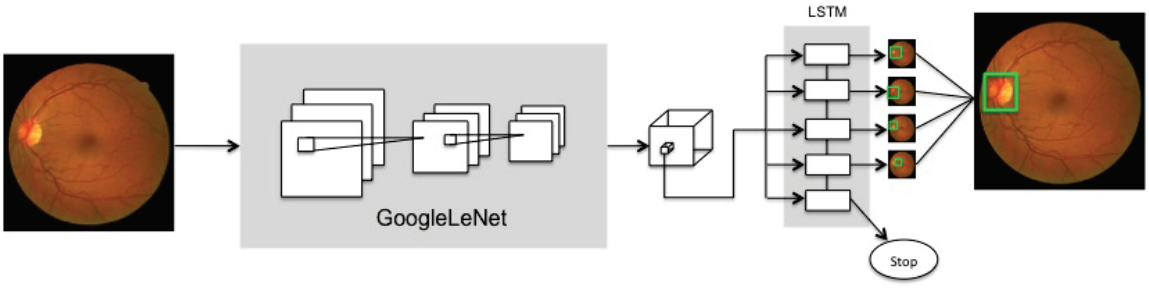


Figure 2.1: Application of GoogLeNet for optic disc detection

53.0 years (*SD*13.05), with 46.2% male and 53.8% female, all of Caucasian ethnicity. 23.1% patients had chronic simple glaucoma and 76.9% eye hypertension. The images were acquired with a colour analogical fundus camera, approximately centered on the optic nerve and they were stored in slide format. In order to have the images in digital format, they were digitized using a HP-PhotoSmart-S20 high-resolution scanner, RGB format, resolution 600×400 and 8 bits/pixel.

4.2 Tuning of Neural Network

Accurate selection of the best parameters to train the Neural Network (NN) is a crucial point to: *i)* obtain the best performances and *ii)* understand the behavior of the neural network according to each parameter. The NN used in this work mainly depends on two parameters: the number of *Epochs* and the *Learning Rate*.

An *Epoch* consists of one full training cycle on the training set, i.e., the times all of the training vectors are used once to update the weights.

The *Learning Rate* is a positive constant, that plays the role of moderating the degree to which weights are changed at each step [Mitchell *et al.* (1997)]. A too small learning rate leads to slow convergence, whilst a too high learning rate to divergence [Duffner and Garcia (2007)].

In order to find the best combination of these two parameters a grid search has been performed. More in detail, the range $[100;1000]$ with a step size of 100 was selected for the *Epochs* and values' set $\{0.000003, 0.0003, 0.003, 0.03\}$ was selected for the *Learning Rate*. These ranges were selected since a finer step size did not significantly improve the results any further.

Since training of NN is computational demanding, the classical CPU-based architecture is not sufficient to obtain results in short time. This is why recently, Graphic Processing Units (GPUs) have become an attractive way of implementing machine learning algorithms that requires large amount of data processing [Upadhyaya (2013)]. GPUs allow to drastically decrease the computational time needed by the NN by using a massive parallelism. The introduction of programmability in GPUs has allowed to perform non-graphical related applications, known as general purpose computing on GPU (GPGPU). In order to perform all the tests, the following workstation was used: x86_64 CPU(s), Intel(R) Xeon(R) CPU E5440 @ 2.83GHz, Linux Debian 4.8.4-1, CUDA compilation tools, release 7.5, V7.5.17, NVIDIA Corporation GK110GL on Tesla K20c.

In this work only the STARE dataset is used to perform the training of the NN. The grid search was then performed using part of the STARE and the performances were computed on DRIONS and DRIVE datasets already described in Section 4.1. It is important stating that the NN never used information derived from DRIONS and DRIVE datasets during the training phase.

Since the main goal of this test is to understand if our method can detect the optic nerve, we measured the performances of the neural network using the Recall measured defined as $Rec = \frac{TP}{TN+FN}$.

Results of the tuning phase are reported in Figure 2.2 and Figure 2.3.

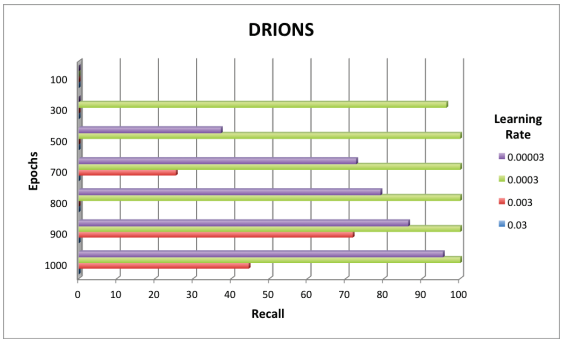


Figure 2.2: Recall performances in optic disc identification according to Learning Rate and Epochs on DRIONS database

4.3 Evaluation

In order to assess the quality of the delineation, the Sørensen-Dice Score Coefficient (DSC) [Zou *et al.* (2004)] was computed according to the following equation:

$$DSC = \frac{2 * |A \cap B|}{|A| + |B|}$$

where A is the voxel set containing the regions with the manually segmented optic disc, B is the voxel set with the region detected by our method. According to this index, we can have three different cases:

1. $DSC = 0$: No overlap
2. $0 < DSC < 1$: Partial overlap
3. $DSC = 1$: Complete overlap

Mean ($\overline{DSC}$) and standard deviation ($\sigma(DSC)$) of DSC were computed using the DSC value obtained for each image in the testing set. Moreover, number of true positives (TP) and recall (Rec) were computed.

Results obtained with different epochs with an optimal $LearningRate = 0.0003$ obtained from the previous section are visible in Table 2.3.

5 Discussion

Figure 2.4 illustrates the accuracy of our method: it shows the optic disc detection in 4 images of the retinal fundus that feature high level of damages. Indeed, the results of our experiments show that the herein proposed model allows to detect the optic disc with an average recall of above 90% on the DRIVE-DB and 100% on the DRIONS-DB, which are composed by images which often feature very high variation in illumination and colors. Furthermore, these results were achieved after about 500 epochs, showing that the network is able to reuse its previous knowledge in order to reduce the training time.

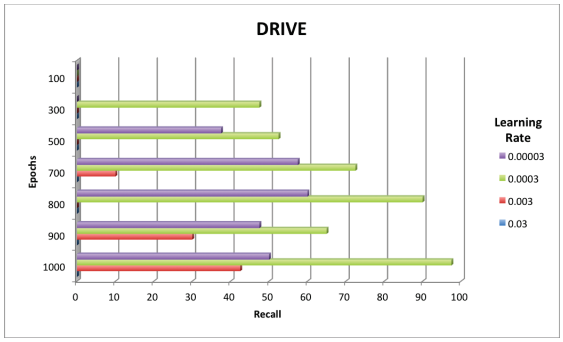


Figure 2.3: Recall performances in optic disc identification according to Learning Rate and Epochs on DRIVE database

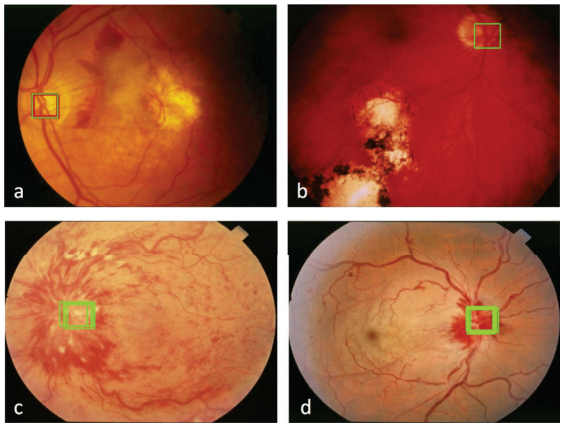


Figure 2.4: Example of optic disc identification in images with high level of damages

Table 2.3: Performances in terms of Recall (Rec), True Positives (TP) and mean Dice Score ($\overline{DSC}$) ($\pm$ its standard deviation in parenthesis) obtained with $Learningrate = 0.0003$ and different Epochs.

	DRIONS			**DRIVE**		
Epochs	**Rec**	**TP**	$\overline{DSC}$	**Rec**	**TP**	$\overline{DSC}$
1000	100	110	0.97 ($\pm$0.52)	97.5	39	0.43 ($\pm$0.19)
900	100	110	0.99 ($\pm$0.15)	65.0	26	0.43 ($\pm$0.19)
800	100	110	0.99 ($\pm$0.09)	90.0	36	0.64 ($\pm$0.20)
700	100	110	0.98 ($\pm$0.28)	72.5	29	0.60 ($\pm$0.23)
500	100	110	0.99 ($\pm$0.05)	52.5	21	0.52 ($\pm$0.15)
300	96.4	106	0.94 ($\pm$0.32)	47.5	19	0.49 ($\pm$0.21)
100	0	0	0	0	0	0

However, we observed that the level of accuracy significantly decreases in the case of the DRIVE dataset, where the network fails at correctly locating the disc for 1 out of 10 images; the main reason is that the dataset used for the training might not be sufficiently representative of the DRIVE-DB. We have to observe, moreover, that reducing the threshold of overlapping used in the evaluation the recall achieved is 100% also in the DRIVE-DB. Hence, we can conclude that the network still locates the Optic Disc in the correct position, but with less accuracy, so that the misclassification might be solved by training the network on a larger dataset for more epochs.

6 Conclusion

It has been known for a long time that retinal changes reflect systemic microvascular damages associated with a number of pathological conditions, such as hypertension or diabetes. In this chapter we presented a method for the automatic location of the optic disc in retinal fundus images. The method is based on a fine-tuning of the weight of a Convolutional Neural Network designed and trained for face detection tasks, and, notably, we also observed how previous knowledge learned from a source task can be easily transferred to solve a related target task, thus overcoming the limitation of insufficient data and time/cost constraints. We carried out an experimental activity aimed at assessing effectiveness and performances of the proposed method, that proved to be effective also in case of low-quality input images.

III Neural Networks for Neurological Complex Network Analysis in Multiple Sclerosis

Chapter 1

Classification of Multiple Sclerosis Clinical Profiles via Graph Convolutional Networks

Contents

1 Introduction

In Multiple Sclerosis, the course of the disease and the risk for developing permanent disability are very different from one patient to another and the prediction of long-term disability is still not possible in a new MS patient. Today's neurologist challenge is to predict the individual patient evolution and response to therapy based on the clinical, biological and imaging markers available from disease onset. Long-term clinical studies have been conducted to determine the clinical predictors of disability accumulation in MS [Degenhardt *et al.* (2009), Soldán *et al.* (2015)]. In RRMS and SPMS, several negative prognostic factors were identified such as the onset of progression, higher early relapse rate, greater disability in the first 5 years, and shorter interval to the second relapse. However, none of these predictors are available at the beginning of the disease [Confavreux *et al.* (2003), Confavreux and Vukusic (2006), Scalfari *et al.* (2010)].

Among the available information with a potential prognostic value at the CIS stage, MRI remains the most promising. A lot of efforts have been concentrated on the identification and characterization of MS lesions [Valverde *et al.* (2017), Brosch *et al.* (2015)]. While conventional T2 lesion load is moderately correlated with the patient clinical status, it can predict the increase of disability scores, such as the Expanded Disability Status Scale (EDSS) and the Multiple Sclerosis Functional Composite (MSFC) [Barkhof (2002)]. Global brain atrophy constitutes a potential marker, as it even exists at the early stages of MS. However, its predictive value is still controversial, probably due to its methodological limitations [Durand-Dubief *et al.* (2012)]. Measurement of subcortical grey matter atrophy could be of special interest if appropriate tools were available in clinic [Hannoun *et al.* (2012)]. Indeed, atrophy in the thalamus was recently reported to be an early marker of the neurodegeneration processes occuring throughout the disease progression [Azevedo *et al.* (2018)]. Regional atrophy in the whole brain was also studied, showing a specific pattern of the atrophy progression within the central nervous system, starting in the posterior cingulate cortex before spreading in the whole cortex [Eshaghi *et al.* (2018)]. More advanced MRI techniques, such as brain volumetry, magnetization transfer imaging (MTI) and diffusion-tensor imaging (DTI) are promising tools in that perspective [Rovira *et al.* (2013)]. Reflecting more specifically the demyelination and remyelination processes, have been shown to predict deterioration of cognitive functions in patients with early MS stages followed during 7 years [Deloire *et al.* (2011)]. However, these advanced techniques are not always available in clinical routine. In contrast, DTI becomes more available in clinical environment and provides an effective mean for the quantification of demyelination and axonal loss in CIS patients [Sbardella *et al.* (2013)]. Furthermore, it has recently been shown that diffusivity measurements in CIS patient's cerebellar white matter (decreased fractional anisotropy) can be predictive of a shorter conversion into a clinically definite MS [Kugler and Deppe (2018)]. Therefore, we propose in this work a new approach for the automatic classification of MS clinical profiles based on brain DTI acquisition.

MRI data are usually represented as images. However, new data representation approaches were developed based on graph theory. Recently applied in neurosciences, graph-based models opened new perspectives for the exploration of brain structural and functional connectivity

by means of graph-derived metrics [Rubinov and Sporns (2010)]. In this context, few machine leaning approaches have been developed for the classification of MS clinical forms. [Stamile *et al.* (2015)] applied Support Vector Machines (SVM) to graph-based representation of the brain for the classification of MS patients clinical courses. In particular, Brain structural connectivity graphs were extracted from DTI data and several experiments were performed to classify RR vs PP, RR vs SP, PP vs SP and RR vs PP vs SP clinical profiles. Both weighted and binary graphs have been considered and the best performances were obtained with unweighted graphs for most of the classification tasks. In [Kocevar *et al.* (2016)], a similar strategy has been used. Six global features (graph density, assortativity, transitivity, global efficiency, modularity and characteristic path length) were extracted from the structural connectivity graphs to enhance the performance of the SVM classification of MS clinical profiles. High level of accuracy were obtained in the HC vs CIS, CIS vs RR, RR vs PP, RR vs SP, SP vs PP and CIS vs RR vs SP tasks. This work demonstrated the better sensitivity of the modularity and assortativity metrics to achieve the best performances. These approaches provided remarkable results on binary classification tasks but were unable to classify the four possible MS profiles at once.

More recently, Neural Networks (NN) based approaches showed promising results for the analysis and classification of images in a wide range of applications [Goodfellow *et al.* (2016)].

More specifically in the context of MRI analysis in MS, Whang *et al.* exploited complex CNN to differentiate MS patients from healthy controls with an accuracy greater than 98% based on T1-weighted MRI [Wang *et al.* (2018)]. The same task was addressed by Maleki *et al.* and Zhang *et al.* where CNN achieved similar results [Maleki *et al.* (2012), Zhang *et al.* (2018)c]. In [Ion-Mărgineanu *et al.* (2017)] the authors demonstrated the potential of using simple CNN to classify MS clinical courses. In particular, they exploited features extracted from magnetic resonance spectroscopic images combined with brain tissue segmentations of grey matter, white matter, and lesions.

Since the first definition of Graph Neural Networks (GNN) [Scarselli *et al.* (2009)], a huge effort was made to extend neural networks with the purpose of processing graph structures data. By implementing a function that maps a graph and its nodes into an m-dimensional Euclidean space, the GNN model can directly process many types of graphs (e.g., acyclic, cyclic, directed, and undirected). An extension of this approach was proposed by [Kipf and Welling (2016)], which introduced a Graph Convolutional Neural Network (GCNN) model that is able to achieve promising results by properly managing structured data and capturing hidden information from graphs.

Graph kernels are historically widely used techniques to solve graph classification problems [Kriege *et al.* (2019)]. These methods use a kernel function to measure the similarity between pairs of graphs. In this way, kernel-based algorithms such as support vector machines for supervised graph learning can be used for processing graphs. Similar to GNNs, graph kernels can embed graphs or nodes into vector spaces by a mapping function. However, differently from GNNs, this mapping function is deterministic rather than learnable. Furthermore, due to a pair-wise similarity calculation, graph kernel methods suffer significantly from computational bottlenecks. GNNs attempt to overcome this limitation by directly perform graph classification based on the extracted graph representations and therefore are much more efficient than graph kernel methods.

Table 1.1: Information on the data set for the different clinical profiles (HC,CIS, RR, SP, PP).

	HC	CIS	RR	SP	PP
Number of patients (%Male/Female)	24(42/57)	12 (50/50)	30 (20/80)	28 (61/39)	20 (45/55)
Age at first scan (years)	35.7 (10.1)	30.88 (6.4)	27.57 (7.8)	27.64 (7.6)	34.99 (6.1)
Disease duration (years)	-	1.50 (1.54)	6.75 (4.81)	13.12 (5.84)	5.90 (2.60)
EDSS median (range)	-	0.5 (0-4)	2.0 (0-4.5)	5.0 (3-7)	4.0 (2.5-6.5)
Total number of scans	24	63	190	199	126

In this work, we used the GCNN model to classify MS patients into four clinical profiles [CIS, RR, SP, PP] using the graph structural connectivity information. Beside the use of brain connectivity graphs directly as an input to the NN, we also investigate the potential role of graph local features in further improving classification performances. Finally, we perform our experiments using both weighted and unweighted connectivity matrices of the brain structure, in order to understand the role played by edge weights in the classification process.

2 Materials and Methods

Our method is divided in three steps: *(i)* structural connectivity information is extracted from the images in order to produce a graph representation of the MRI; *(ii)* a feature matrix is extracted from each graph and local graph metrics are computed; *(iii)* the adjacency matrix, together with the local graph features matrix, is used as input for the GCNN to perform the classification task.

In the following, we illustrate how structural connectivity information are extracted from the images in order to produce a graph representation of the MRI. Then, we describe the NN architecture used for the classification task. Finally, we provide a description of the graph features considered in this work.

2.1 MRI Acquisition and data set description

The MS population consisted of 12 CIS, 30 RR, 28 SP, 20 PP examined longitudinally every 6 months during 3 years and then every year during 4 more years. A total of 580 exams were processed for classification. In addition, twenty-four healthy controls (HC) subjects, age and sex matched with the MS patients, were considered in the experiments. This prospective study was approved by the local ethics committee (CPP Sud-Est IV) and the French national agency for medicine and health products safety (ANSM). Written informed consents were obtained from all patients prior to study initiation. A description of clinical data is reported in Table 1.1. Diagnosis and MS profile were established according to the McDonald criteria [Lublin *et al.* (2014), McDonald *et al.* (2001)], while disability was assessed with Extended Disability Status Scale (EDSS).

MR examinations were performed on a 1.5T Siemens Sonata system (Siemens Medical Solution, Erlangen, Germany) using an 8-channel head-coil. The MR protocol consisted in the acquisition of a sagittal 3D-T1 sequence ($1 \times 1 \times 1$ mm^3, TE/TR = 4/2000 ms) and an axial 2D-spin-echo DTI sequence (TE/TR = 86/6900 ms; 2×24 directions of gradient diffusion; $b = 1000$ $s.mm^{-2}$, spatial resolution of $2.5 \times 2.5 \times 2.5$ mm^3) oriented in the AC-PC plane.

2.2 Brain Structural Connectivity Graph

As previously described by [Kocevar *et al.* (2016)], the data processing for the extraction of brain structural connectivity is composed of three steps:

1. First, each voxel of the T1-weighted MR images is labeled in four classes, depending on the corresponding tissue type [white matter (WM), cortical GM, sub-cortical GM, cerebro-spinal fluid (CSF)]. In order to perform the classification a segmentation of the Cortical and sub-cortical parcellation using FreeSurfer [Fischl *et al.* (2004)] is performed on the T1 images. The segmentation is also used to define the graph nodes (q=84).

2. Second, the diffusion images are pre-processed by applying correction of Eddy-current distortions [Jenkinson *et al.* (2012)] and skull stripping.

3. Third, MRtrix spherical deconvolution algorithm [Tournier *et al.* (2012)] is used to estimate main diffusion directions in each voxel of diffusion images. Starting from the previous tissue-class labeling, a probabilistic streamline tractography algorithm is applied to generate fiber-tracks in voxels labeled as WM voxels. Symmetrical connectivity matrix $\mathbf{A} \in \mathbb{N}_{+}^{q \times q}$ is then generated for each subject through the combination of GM segmentation and WM tractography.

   In detail, let $\Psi : \mathbb{N}_1^2 \to \mathbb{N}$ be the number of fibers connecting two nodes i and j. Then, each element of the connectivity matrix A is $a_{i,j} = \Psi(i,j)$. In particular, A represents the adjacency matrix of the weighted undirected graph $G = (V, E, \omega)$ where V ($|V| = q$) is the set containing the segmented GM brain regions, E is the graph edges set defined as:

$$E = \{\{i,j\} \mid \Psi(i,j) > 0 \; \forall \; 1 \leq i,j \leq q\}$$

   and $\omega : E \to \Psi(E)$ is the weighted function that assigns at each edge $e \in E$ its weight. Roughly speaking, this function is the same as Ψ but is defined only from the elements of the edges set E.

Finally, a weighted undirected graph $\mathscr{G}^1 = (V^1, E^1, \omega)$ is created, starting from the undirected graph $\mathscr{G} = (V, E, \omega)$ by applying the graph function $\Upsilon : \mathscr{G} \to \mathscr{G}^1$. The resulting graph contains only the strongly connected regions with respect to a given threshold $\tau \in \mathbb{R}_{[0,1]}$. In particular, Υ performs the following mapping:

$$V^1 = V \quad E^1 = L(1, \ldots, T), \; T = \frac{(q^2 - q)\tau}{2}$$

where L is the list of graph edges (E) sorted in ascending order of weight. This results in a weighted undirected brain connectivity graph that is used for this work.

2.3 Notation

For the description of our method, we introduce the following notations. We denote scalar values with small letters (e.g., a), 1-dimensional vectors with bold small letters (e.g., $\boldsymbol{a}$), matrices

with boldface capital letters (e.g., $\boldsymbol{A}$) where $\mathbf{A}'$ is the transpose of $\mathbf{A}$. $\mathscr{G} = (V,E)$ is an undirected graph, where V is the set of vertex and E is the set of edges. For each vertex $v \in V$ let $\mathbf{x} \in \mathbb{R}^d$ be the associated feature vector. If not differently specified, given a graph $\mathscr{G} = (V,E)$ we denote by $n = |V|$ the number of nodes of the graph.

2.4 Graph-based Neural Networks

The architecture proposed in this work is composed of one GC layer (k=100) with ReLU activation function followed by a Fully Connected Network with *softmax* activation to handle the multi-class classification problem. Dropout (α=0.3) is used to reduce overfitting. Layers organization and hyperparameters settings were defined following a trial and error process, starting from a minimal topology. We observed that adding more layers did not increase performances.

2.5 Graph Local Features

Feature extraction is an important task for graph classification. Indeed, while adjacency matrices represent exactly the structure of the graph, features encode latent patterns or measure simple characteristics of graphs which could be useful for a better characterization and classification. Particularly, in the brain connectivity domain, several measures are able to detect functional integration and segregation to quantify centrality of individual brain regions or pathways, characterize patterns of local anatomical circuitry, and to test resilience of networks to insult [Rubinov and Sporns (2010)]. These network measures have binary and weighted variants, where weighted variants of measures are typically generalizations of binary variants obtained by considering edge weights in the computation. In this work four local measures were identified[1], according to the method described in [Rubinov and Sporns (2010)]. Below, we recall their definition already introduced in Section 3.3, providing a detailed description of their weighted and unweighted version. Nevertheless, we refer the interested reader to [Rubinov and Sporns (2010)] for a complete description of the most commonly used measures of local and global connectivity, as well as their neurobiological interpretations.

Node Degree

The degree of a node is the number of connections of that node. The weighted version of the metric (strength) also considers the weights of the edges into account. Let N be the set of all nodes in the network and a_{ij} the connection status between i and j, i.e. equals 1 if there is a link between these two nodes 0 otherwise. The degree of an unweighted graph can be calculated as follows:

$$D_i = \sum_{j \in N} a_{ij}$$

The weighted version of the metric (strength) also considers the weights of the edges into account. Let w_{ij} the connection weight between i and j, the weighted degree of a weighted undirected graph

[1] Brain Connectivity Toolbox: https://sites.google.com/site/bctnet/

can be calculated as follows:

$$D_i^w = \sum_{j \in N} w_{ij}$$

Clustering Coefficient

The clustering coefficient is the fraction of triangles around a node and is equivalent to the fraction of node's neighbors that are neighbors of each other. Let t_i be the number of triangles around a node i computed as follows:

$$t_i = \frac{1}{2} \sum_{j,h \in N} a_{ij} a_{ih} a_{jh}$$

The clustering coefficient per each node i is computed as:

$$CC_i = \frac{2t_i}{k_i(k_i - 1)}$$

The weighed version of the clustering coefficient is obtained by replacing the number of triangles t_i with the sum of triangle intensities:

$$CC_i^w = \frac{2}{k_i(k_i - 1)} \sum_{j,k} (\tilde{w}_{ij} \tilde{w}_{jk} \tilde{w}_{ki})^{1/3}$$

where weights are normalized by the largest weight in the network, $\tilde{w}_{ij} = w_{ij}/max(w_{ij})$.

Local Efficiency

Let first define the global efficiency as the average of inverse shortest path length. The local efficiency is the global efficiency computed on the neighborhood of the node, and is related to the clustering coefficient. It can be defined as follows:

$$E_i = \frac{1}{n} \sum_{i \in N} \frac{\sum_{j,h \in N, j \neq i} a_{ij} a_{ih} \left[d_{jh}(N_i) \right]^{-1}}{k_i (k_i - 1)}$$

where E_i is the local efficiency of node i, and $d_{jh}(N_i)$ is the length of the shortest path between j and h, that contains only neighbors of i. By considering weights in the calculation, the formula can be extended to the weighted version as follows:

$$E_i^w = \frac{1}{2} \sum_{i \in N} \frac{\sum_{j,h \in N, j \neq i} (w_{ij} w_{ih} \left[d_{jh}^w(N_i) \right]^{-1})^{1/3}}{k_i (k_i - 1)}$$

Betweenness centrality

Betweenness centrality is the fraction of all shortest paths in the network that pass through a given node. Nodes with high betweenness centrality are considered hub nodes and determine

important regions in a network. In terms of brain networks this measure helps to detect important anatomical or functional connections. It is defined as follows:

$$BC_i = \frac{1}{(n-1)(n-2)} \sum_{\substack{h,j \in N \\ h \neq j, h \neq i, j \neq i}} \frac{\rho_{hj}(i)}{\rho_{hj}}$$

where ρ_{hj} is the number of shortest paths between h and j, and $\rho_{hj}(i)$ is the number of shortest paths between h and j that pass through i. Betweenness centrality is equivalent on weighted networks, provided that path lengths are computed on respective weighted paths.

2.6 Statistical Analysis

To address the variability introduced by each patient, a linear mixed-effects model was applied separately to each local graph metric in each graph node:

$$Response_{ij} = \beta_0 + \beta_1(Clinical\ phenotype_i) + \beta_2(Scan\ Session_{ij}) + b_{0i} + \varepsilon_{ij} \tag{1.1}$$

In this model, the predicted response of interest for subject i at time j is determined by fixed effects, represented by β_1 and β_2. Subject-specific effects are represented by b_{0i}, allowing a random interception per subject i.

The linear mixed-effects models were fitted using the “lme4" package in R [Bates *et al.* (2015)] and the significance of the fixed effects and the interaction term is tested applying the Kenward-Roger approximation to estimate the degrees of freedom using the “car" package [Fox and Monette (2002)].
When the clinical phenotype fixed effect was significant, a post-hoc test was conducted to extract the estimate and the significance of each between class difference. This step was processed using ‘lsmeans’ package in R [Lenth *et al.* (2016)].

2.7 Experimental Settings

T1 and DTI images have been used to obtain a structural $N \times N$ connectivity matrix for each MRI. For each feature vector, normalization was applied so that each value was in the real range $[0,1]$.

The parameter τ was set to 0.35 according to the method described in [Kocevar *et al.* (2016)]. The model was trained using Adam [Kingma and Ba (2014)a] with learning rate 0.001 and early stopping to prevent overfitting. Cross validation with 3 folds was used to provide a more robust evaluation of the model. The quality of the classification was compared by means of the average F-Measure, Precision and Recall [Powers (2011)] achieved during the cross validation. Wilcoxon-Mann-Whitney test [Wilcoxon (1945)] was conducted to test the differences between the global metrics measured between the patient’s groups.

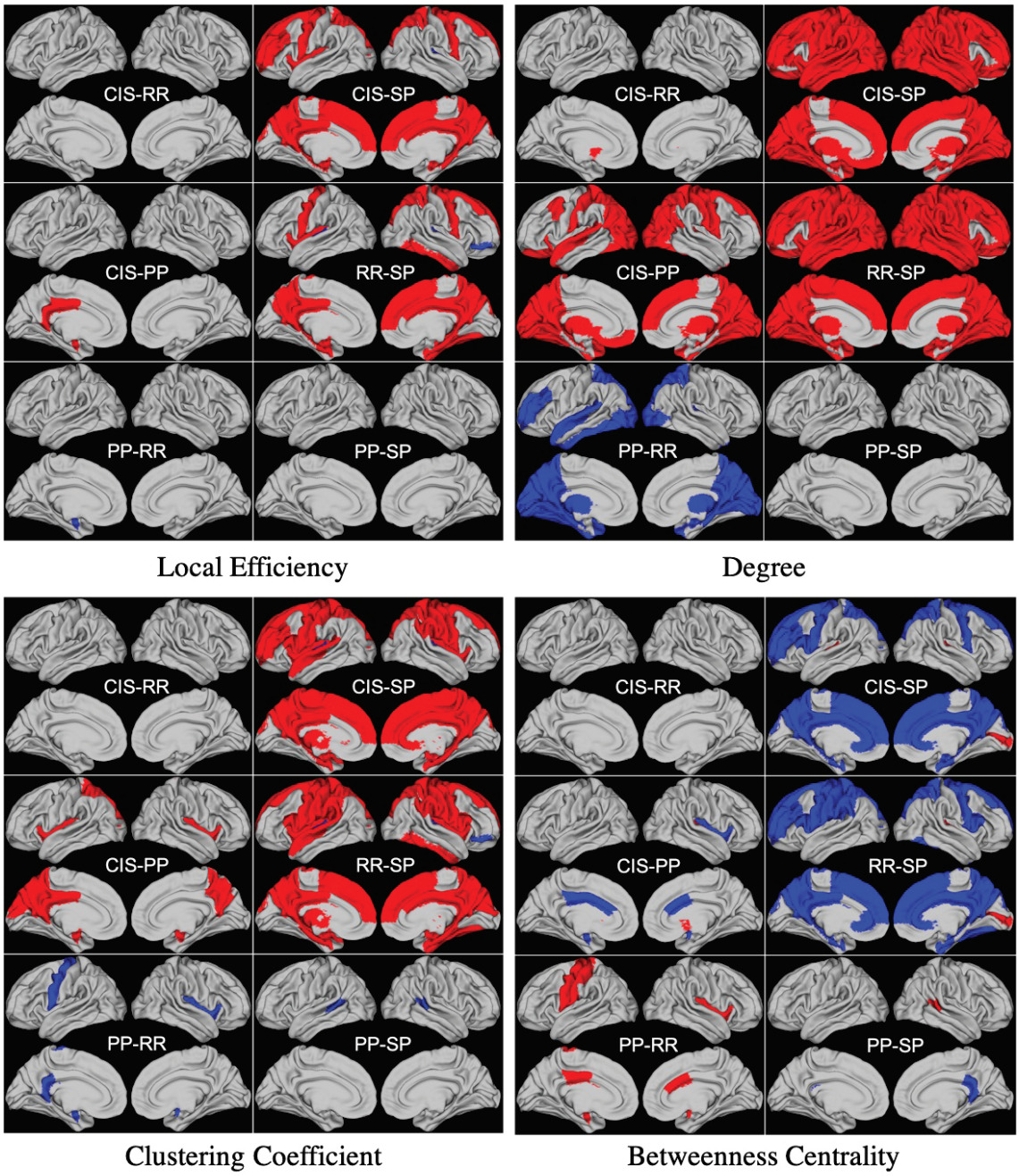


Figure 1.1: Differences between groups found in statistical analysis performed using unweighted local graph metrics. Blue and Red regions represent negative and positive differences, respectively.

3 Results

In the experiments, we trained the GCNN to classify patients given their brain connectivity adjacency matrix representation and the corresponding vector of node descriptors. Furthermore, we trained the GCNN using all the features together (all-graph). Finally, we used a featureless approach, meaning that the no node descriptor is provided.

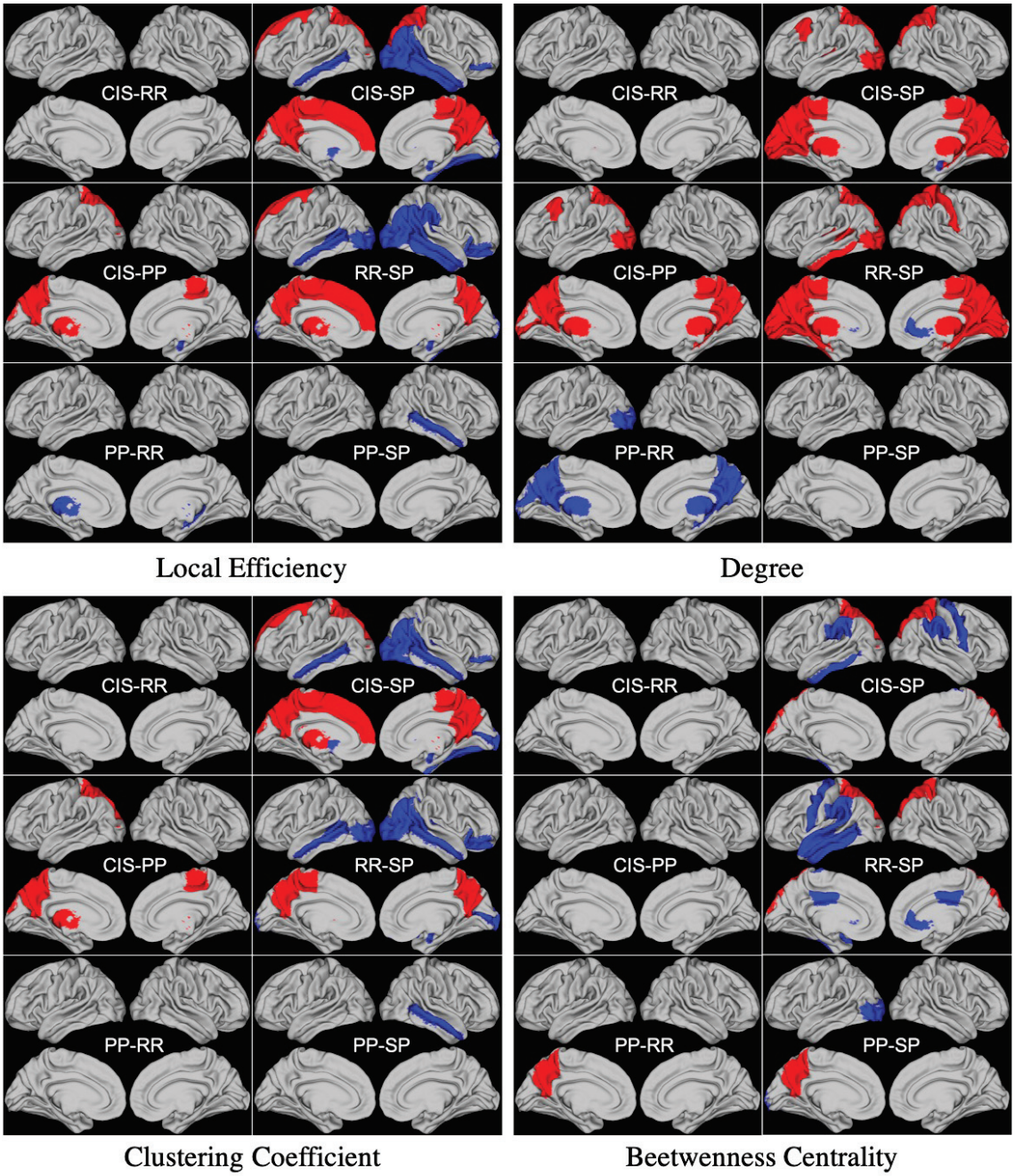


Figure 1.2: Differences between groups found in statistical analysis performed using weighted local graph metrics. Blue and Red regions represent negative and positive differences, respectively.

3.1 Conventional MRI Data Analysis

Before illustrate results of the proposed approach, we analyze conventional MRI information (T2 lesions and grey matter volumes) in order to investigate whether such information are sufficient to discriminate MS clinical forms. Gray matter and lesions were segmented based on T2 FLAIR, using the MSmetrix software developed by icometrix (Leuven, Belgium). Results are illustrated in Table 1.2 and Table 1.3. Significant differences were found comparing grey matter volumes of

Table 1.2: Statistical analysis results using T2 lesions volumes

contrast	estimate	t-ratio	p-value
CIS - PP	-13.85	-3.10	0.0139 *
CIS - RR	-10.73	-2.60	0.0524
CIS - SP	-24.11	-5.76	<0.0001 ***
PP - RR	3.12	0.88	0.8132
PP - SP	-10.26	-2.85	0.0278 *
RR - SP	-13.40	-4.24	0.0003 ***
* p<0.05; ** p<0.01; *** p<0.001			

Table 1.3: Statistical analysis results using grey matter volumes

contrast	estimate	t-ratio	p-value
CIS - PP	49.24	3.32	0.0072 **
CIS - RR	17.24	1.26	0.5910
CIS - SP	53.44	3.85	0.0013 **
PP - RR	-32.01	-2.73	0.0376 *
PP - SP	4.30	0.35	0.9850
RR - SP	36.21	3.46	0.0046 **
* p<0.05; ** p<0.01; *** p<0.001			

CIS with PP $(p < 0.01)$ and SP $(p < 0.01)$ and comparing grey matter volumes of RR with SP $(p < 0.01)$ and PP $(p < 0.05)$. Furthermore, significant differences were found comparing lesions volumes of CIS with PP $(p < 0.05)$ and SP $(p < 0.001)$ and comparing lesions volumes of SP with RR $(p < 0.001)$ and PP $(p < 0.05)$.

Classification using Naïve Approach

Despite statistical differences among MS clinical forms in terms of lesions volumes and grey matter volumes, such information might not be sufficient to discriminate patients for developing an automated classification method. A naïve classifier was defined in order to investigate the discrimination capability of conventional MRI information. A randomly selected seventy percent of the dataset was used to compute the mean of each group. Then, to each sample of the test set is assigned the class corresponding to the closer mean. The experiments was repeated $k = 100$ times and results using lesions volumes and grey matter are reported in Table 1.4 and Table 1.5, respectively.

Table 1.4: Naïve classification results for the multiclass classification task using T2 lesions volumes

	mean ($\pm$ std)
F-Measure	0.37 ± 0.02
Accuracy	0.42 ± 0.03
Precision	0.37 ± 0.03
Recall	0.48 ± 0.02

Table 1.5: Naïve classification results for the multiclass classification task using grey matter volumes

	mean ($\pm$ std)
F-Measure	0.31 ± 0.03
Accuracy	0.32 ± 0.03
Precision	0.32 ± 0.03
Recall	0.38 ± 0.03

Classification using Support Vector Machine

A Support Vector Machine (SVM) was trained to classify among the MS clinical forms using lesions and grey matter volumes. In order to find the optimal input parameters of SVM, namely C and γ, grid search was performed using growing sequences of C and γ. More in detail, we used the range [0.001,10] for C and [0.001, 1] for γ. Both linear and radial basis function (rbf) kernel were tested. Generalization of classification performances was ensured by K-Fold cross validation using 3-folds. Finally, each feature was standardized in order to improve the quality of the classification. Perfomances was evaluated by means of F-Measure, Accuracy, Precision and Recall. Best results were obtained using linear kernel with $C = 0.01$, and $\gamma = 0.001$ and reported in Table 1.6. As observable, conventional MRI data do not provide sufficient information to correctly perform the multiclass classification task. Indeed, despite statistical differences among groups, the number of overlapping values is too high to discriminate among clinical statuses.

Table 1.6: SVM classification results for the multiclass classification task (CIS vs RR vs SP vs PP) using conventional MRI information

	mean ($\pm$ std)
F-Measure	0.27 ± 0.03
Accuracy	0.47 ± 0.04
Precision	0.24 ± 0.02
Recall	0.34 ± 0.03

3.2 Local Graph Metrics Analysis

We report results of the statistical analysis performed using unweighted local graph metrics. Many significant differences were found when comparing the betweenness centrality metric of CIS vs PP and SP as well as when comparing RR vs PP and SP. No differences were found when comparing CIS vs RR and PP vs SP. The same behaviour was observed when comparing clustering coefficient, degree and local efficiency metrics. Moreover, no significant differences were observed when comparing the local efficiency metric of PP vs RR, except for the left amygdala ($p < 0.05$). Finally, important differences in several regions were found when comparing the local degree of CIS vs SP and RR vs SP. An illustration of these results is reported in Figure 1.1.

Concerning the statistical analysis performed using weighted local graph metrics, several significant differences were found, again, when comparing the betweenness centrality, clustering coefficient, degree and local efficiency metrics in the CIS and RR groups with respect to the PP and SP groups. No significant differences were observed between CIS and RR except for the degree of the left-caudate nucleus ($p < 0.05$). Concerning the comparison between PP and SP groups, significant differences were found when comparing the betweenness centrality in the left lateral-occipital region and the left precuneus (parietal lobe) and clustering coefficient and efficiency in the right middle-temporal region ($p < 0.05$). This latter region was also found to be the only one differing between PP and SP groups in terms of clustering coefficient ($p < 0.05$). Statistical analysis results performed using weighted local graph metrics

are reported in Figure 1.2. Detailed results about our statistical analysis can be downloaded at *https://www.frontiersin.org/articles/10.3389/fnins.2019.00594/full#supplementary-material*.

3.3 Classification using Unweighted Adjacency Matrix

We first trained the proposed GCNN model using unweighted brain connectivity adjacency matrix representations. Results obtained for each experiment are reported in Table 1.7 and graphically illustrated in Figure 1.3. As observable, using local graph-derived metrics as node descriptors combined with unweighted graphs, does not provide sufficient information for the classification, achieving an average F-Measure of about 0.50. In particular, the worst performances were obtained for the Degree metric (F-Measure $= 0.39 \pm 0.03$) while slightly better results were observed, on average, with Clustering Coefficient, Betweenness Centrality and Efficiency. All the graph-metrics together provided, on average, better results. Nevertheless, the most remarkable results were observed with the featureless approach. In this case, we obtained a significant improvement of the performances (F-Measure = 0.80 $\pm$0.01), stating that the brain structure itself is highly discriminative for the clinical profiles.

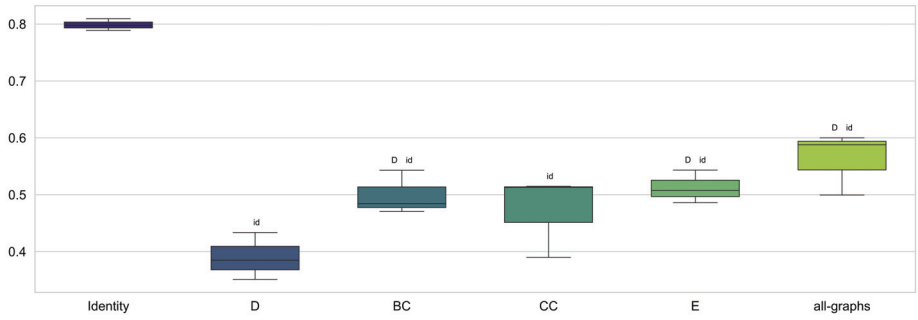


Figure 1.3: Box plot in term of F-Measure for each different unweighted feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

3.4 Classification using Weighted Adjacency Matrix

We trained the proposed model using weighted brain connectivity adjacency matrix representations. Results obtained for each experiment are provided in Table 1.7 and reported graphically in Figure 1.4. Interestingly, good results were achieved (F-Measure > 0.60) with all the studied local graph-metrics. Furthermore, a significant increase in performances was observed when using all the studied features together. Again, the best result was achieved using the featureless approach (F-Measure $= 0.92 \pm 0.02$), with an average F-Measure increasing of 10% with respect to the unweighted version. In particular, a general performance increase can be observed when using weights information, as graphically shown in Figure 1.5. Indeed, for all the proposed experiments, except when we use all the graph metrics together, a significant improvement is achieved.

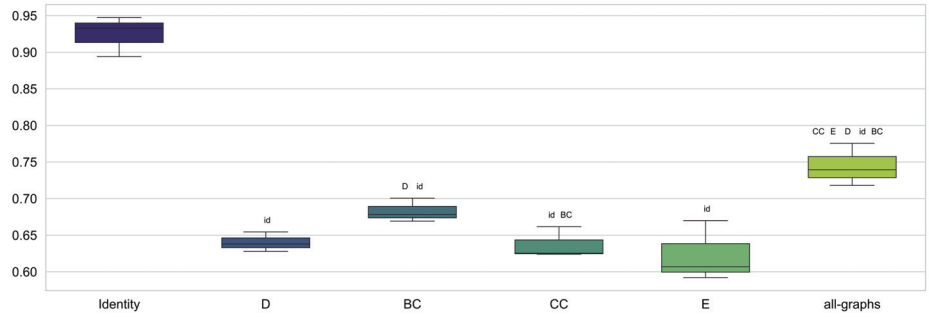


Figure 1.4: Box plot in term of F-Measure for each different weighted feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

Table 1.7: Cross validation results in terms of F-Measure, Precision and Recall (± standard deviation) averaged on three folds. Rows report achieved results using unweighted graphs with unweighted features (upper) and using weighted graphs with weighted features (lower) [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

	Identity	D	BC	CC	E	all-graphs
			unweighted			
F-Measure	0.80 (±0.01)	0.39 (±0.03)	0.50 (±0.03)	0.47 (±0.06)	0.51 (±0.02)	0.56 (±0.04)
Precision	0.81 (±0.01)	0.33 (±0.03)	0.54 (±0.08)	0.47 (±0.11)	0.56 (±0.06)	0.57 (±0.04)
Recall	0.80 (±0.01)	0.48 (±0.04)	0.55 (±0.04)	0.55 (±0.04)	0.56 (±0.03)	0.60 (±0.04)
			weighted			
F-Measure	0.92 (±0.02)	0.64 (±0.01)	0.68 (±0.01)	0.64 (±0.02)	0.62 (±0.03)	0.74 (±0.02)
Precision	0.93 (±0.02)	0.70 (±0.02)	0.69 (±0.01)	0.66 (±0.01)	0.64 (±0.05)	0.76 (±0.02)
Recall	0.93 (±0.02)	0.65 (±0.02)	0.69 (±0.01)	0.64 (±0.02)	0.63 (±0.03)	0.75 (±0.03)

3.5 Classification of Control Subjects vs MS patients

In this section we explore the capability of the proposed models to discriminate between HC and MS patients. More in detail, we performed three main experiments. First, we trained the proposed GCNN model at classifying patients at early stages of the pathology (CIS and RR) from HC. Then, we trained the model at classifying patients at progressive stages (SP vs PP) from HC. Finally, we performed a multiclass classification task including all the clinical forms, i.e. HC vs SP vs PP vs RR vs CIS. As for the other experiments in this study, we compared performances using weighted and unweighted brain connectivity adjacency matrix representations.

HC vs (CIS + RR)

HC vs (CIS + RR) classification task is described in this section. Results obtained using unweighted and weighted connectivity representations are reported in Table 1.8. Results are then compared graphically in Figure 1.6. As observable, both weighted and unweighted local

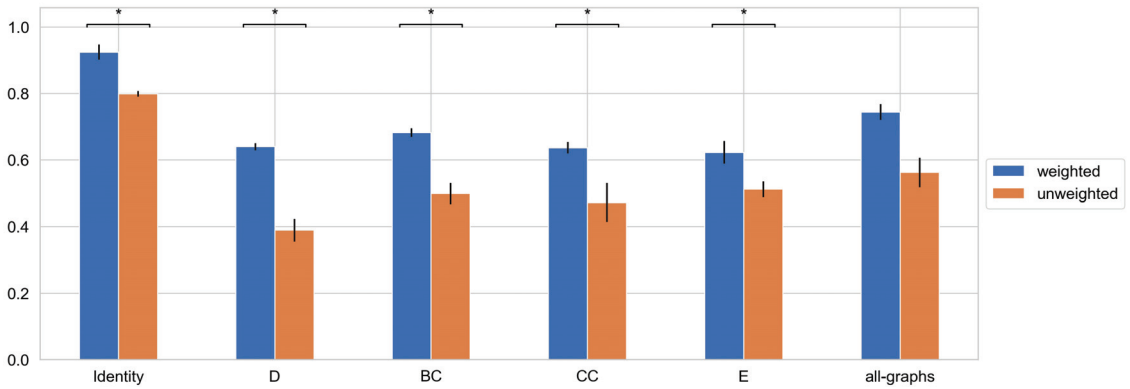


Figure 1.5: average F-Measure comparison for weighted and unweighted approach for each feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

graph-derived metrics provide sufficient information for distinguish between HC and MS patients. However, weighted connectivity matrices provide overall better results. Best performances were achieved using no node descriptions and using all graph-metrics together.

Table 1.8: Cross validation results of HC vs (CIS+RR) in terms of F-Measure, Precision and Recall ($\pm$ standard deviation) averaged on three folds. Rows report achieved results using unweighted graphs with unweighted features (upper) and using weighted graphs with weighted features (lower) [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

	Identity	D	BC	CC	E	all-graphs
			unweighted			
F-Measure	0.96 ($\pm$0.05)	0.93 ($\pm$0.06)	0.87 ($\pm$0.14)	0.89 ($\pm$0.11)	0.89 ($\pm$0.11)	1.0 ($\pm$0.0)
Precision	1.0 ($\pm$0.01)	0.97 ($\pm$0.03)	0.99 ($\pm$0.01)	0.99 ($\pm$0.01)	0.99 ($\pm$0.01)	1.0 ($\pm$0.0)
Recall	0.94 ($\pm$0.08)	0.91 ($\pm$0.09)	0.83 ($\pm$0.16)	0.85 ($\pm$0.14)	0.85 ($\pm$0.14)	1.0 ($\pm$0.0)
			weighted			
F-Measure	1.0 ($\pm$0.0)	0.96 ($\pm$0.01)	0.96 ($\pm$0.04)	0.98 ($\pm$0.01)	0.98 ($\pm$0.01)	1.0 ($\pm$0.0)
Precision	1.0 ($\pm$0.0)	0.98 ($\pm$0.02)	0.97 ($\pm$0.03)	1.0 ($\pm$0.0)	1.0 ($\pm$0.0)	1.0 ($\pm$0.0)
Recall	1.0 ($\pm$0.0)	0.95 ($\pm$0.03)	0.94 ($\pm$0.05)	0.96 ($\pm$0.03)	0.96 ($\pm$0.03)	1.0 ($\pm$0.0)

HC vs (SP + PP)

HC vs (SP + PP) classification task is described in this section, in order to test the capability of the model in in discriminating HC from progressive MS. Results obtained using unweighted and weighted connectivity representations are reported in Table 1.9. Results are then compared graphically in Figure 1.7. Again, high level of accuracy were obtained using weighted and unweighted information and weighted connectivity matrices provide overall better results. As for the HC vs (CIS + RR) task, the highest performances were achieved using no node descriptions and using all graph-metrics together.

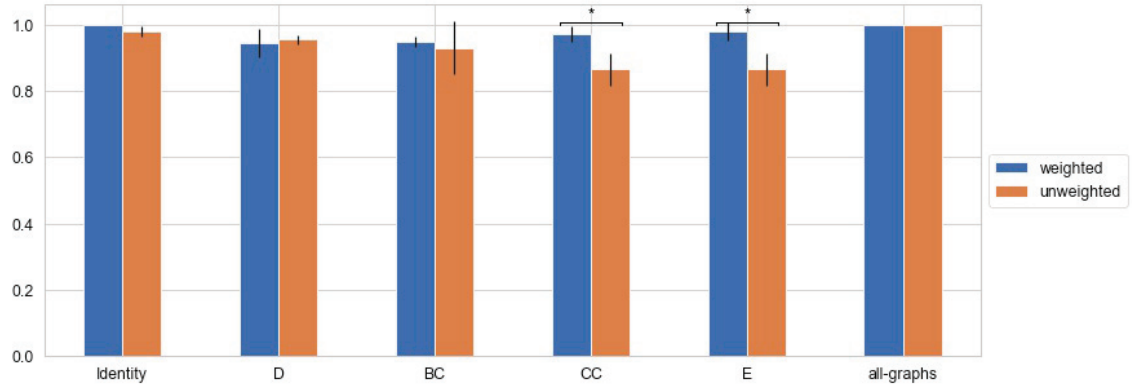


Figure 1.6: average F-Measure comparison for weighted and unweighted approach [HC vs (CIS+RR)] for each feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

Table 1.9: Cross validation results of HC vs (SP+PP) in terms of F-Measure, Precision and Recall (± standard deviation) averaged on three folds. Rows report achieved results using unweighted graphs with unweighted features (upper) and using weighted graphs with weighted features (lower) [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

	Identity	D	BC	CC	E	all-graphs
			unweighted			
F-Measure	0.96 (±0.05)	0.93 (±0.06)	0.87 (±0.14)	0.89 (±0.11)	0.89 (±0.11)	1.0 (±0.0)
Precision	1.0 (±0.01)	0.97 (±0.03)	0.99 (±0.01)	0.99 (±0.01)	0.99 (±0.01)	1.0 (±0.0)
Recall	0.94 (±0.08)	0.91 (±0.09)	0.83 (±0.16)	0.85 (±0.14)	0.85 (±0.14)	1.0 (±0.0)
			weighted			
F-Measure	1.0 (±0.0)	0.96 (±0.01)	0.96 (±0.04)	0.98 (±0.01)	0.98 (±0.01)	1.0 (±0.0)
Precision	1.0 (±0.0)	0.98 (±0.02)	0.97 (±0.03)	1.0 (±0.0)	1.0 (±0.0)	1.0 (±0.0)
Recall	1.0 (±0.0)	0.95 (±0.03)	0.94 (±0.05)	0.96 (±0.03)	0.96 (±0.03)	1.0 (±0.0)

All Classes Classification

Finally a multiclass classification task is performed using all the forms together. Results obtained using unweighted and weighted connectivity representations are reported in Table 1.10. Results are then compared graphically in Figure 1.8. All the graph-metrics together provided, on average, better results. Higher performances were observed performing a featureless approach.

3.6 Early vs Progressive Forms Comparison

In this section we explore the capability of the proposed models to discriminate between CIS and RR and progressive MS clinical forms. The proposed GCNN model was trained at distinguishing between CIS-RR and SP-PP patients, in order to provide a better understanding of the different pathophysiology of patients at an early stage of the disease (CIS and RR) and patients with progressive MS. Furthermore, performances using weighted and unweighted brain connec-

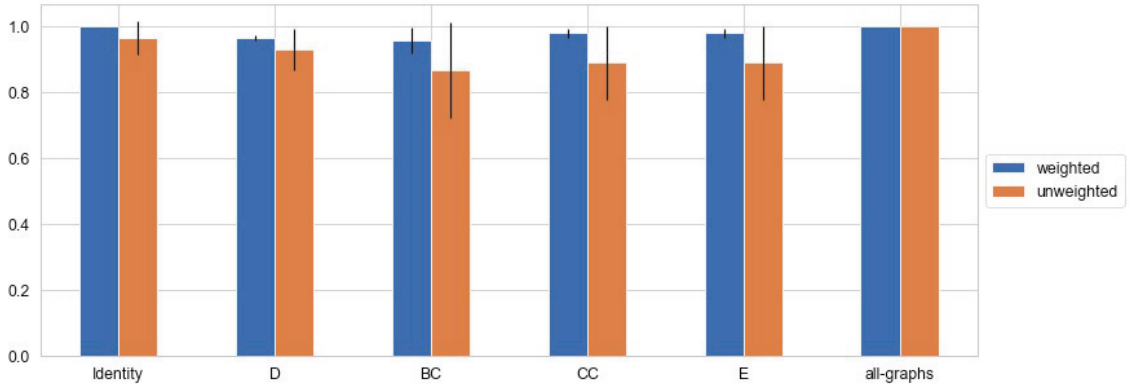


Figure 1.7: average F-Measure comparison for weighted and unweighted approach [HC vs (SP+PP)] for each feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

Table 1.10: Cross validation results of HC vs SP vs PP vs RR vs CIS in terms of F-Measure, Precision and Recall (± standard deviation) averaged on three folds. Rows report achieved results using unweighted graphs with unweighted features (upper) and using weighted graphs with weighted features (lower) [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

	Identity	D	BC	CC	E	all-graphs
			unweighted			
F-Measure	0.82 (±0.03)	0.56 (±0.08)	0.54 (±0.01)	0.52 (±0.02)	0.53 (±0.03)	0.63 (±0.03)
Precision	0.83 (±0.02)	0.72 (±0.04)	0.58 (±0.03)	0.61 (±0.1)	0.59 (±0.03)	0.71 (±0.01)
Recall	0.81 (±0.03)	0.56 (±0.07)	0.53 (±0.01)	0.54 (±0.03)	0.52 (±0.04)	0.63 (±0.02)
			weighted			
F-Measure	0.94 (±0.02)	0.66 (±0.07)	0.7 (±0.03)	0.66 (±0.04)	0.68 (±0.0)	0.81 (±0.02)
Precision	0.94 (±0.02)	0.74 (±0.02)	0.75 (±0.03)	0.68 (±0.04)	0.7 (±0.02)	0.84 (±0.01)
Recall	0.93 (±0.02)	0.66 (±0.06)	0.68 (±0.02)	0.67 (±0.04)	0.68 (±0.02)	0.8 (±0.03)

tivity adjacency matrix representations were compared.

Results obtained using unweighted and weighted connectivity representations are reported in Table 1.11. Results are then compared graphically in Figure 1.9.

An overall increase in performances were observed respect to the intra clinical form classification tasks previously performed. Both weighted and unweighted local graph-derived metrics provide promising results. Interestingly, weighted Degree allow to achieve high level of accuracy; indeed, it is worth to note that graph density decreases along with the progress of the pathology, due to neurodegenerative processes. However, as for previous experiments, the featureless approach provided better and more stable results.

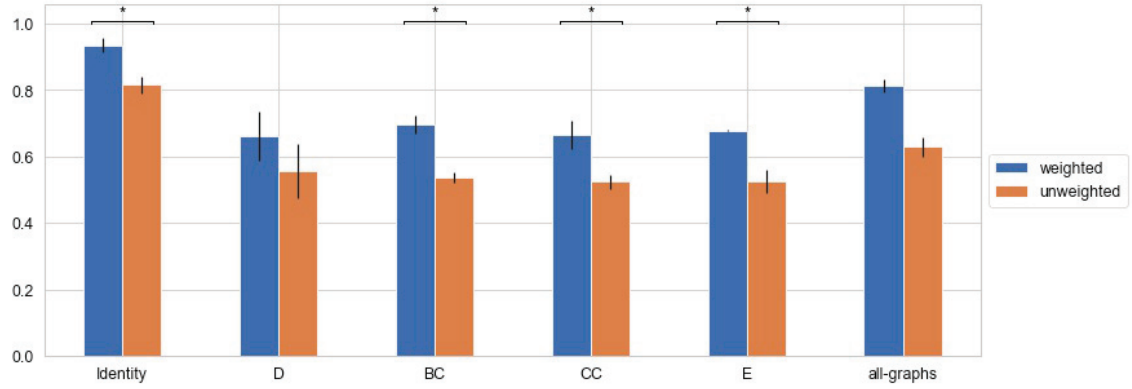


Figure 1.8: average F-Measure comparison for weighted and unweighted approach [HC vs SP vs PP vs RR vs CIS] for each feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

Table 1.11: Cross validation results of (CIS+RR) vs (SP+PP) in terms of F-Measure, Precision and Recall (± standard deviation) averaged on three folds. Rows report achieved results using unweighted graphs with unweighted features (upper) and using weighted graphs with weighted features (lower) [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

	Identity	D	BC	CC	E	all-graphs
			unweighted			
F-Measure	0.91 (±0.1)	0.68 (±0.02)	0.77 (±0.02)	0.78 (±0.02)	0.77 (±0.01)	0.83 (±0.01)
Precision	0.92 (±0.1)	0.7 (±0.01)	0.78 (±0.01)	0.78 (±0.01)	0.78 (±0.02)	0.83 (±0.01)
Recall	0.91 (±0.0)	0.68 (±0.02)	0.77 (±0.02)	0.78 (±0.02)	0.77 (±0.02)	0.83 (±0.01)
			weighted			
F-Measure	0.97 (±0.01)	0.99 (±0.0)	0.85 (±0.01)	0.83 (±0.02)	0.84 (±0.02)	0.92 (±0.01)
Precision	0.97 (±0.01)	0.99 (±0.0)	0.85 (±0.01)	0.84 (±0.02)	0.85 (±0.02)	0.92 (±0.01)
Recall	0.97 (±0.01)	0.99 (±0.0)	0.85 (±0.01)	0.83 (±0.02)	0.84 (±0.02)	0.92 (±0.01)

4 Discussion

In this work, we proposed a novel graph-based neural network method to classify MS patients according to their clinical phenotype using brain structural connectivity information. To this aim, we exploited a peculiar type of neural network architecture designed to handle arbitrarily structured graphs. We compared the impact of local graph metrics to the classification performances, either using weighted and unweighted brain connectivity representation. Furthermore, we performed a statistical analysis on the local graph metrics (weighted and unweighted) computed for each MS clinical form, attempting to characterize differences in the groups, and eventually classify patients. Notice that, conventional MRI information such as lesion and grey matter volumes did not allowed to accurately classify patients into the four clinical forms, thus showing the lack of specificity of these measurements to the pathophysiological effects of the disease. Details about the analysis can be found in the supplementary material.

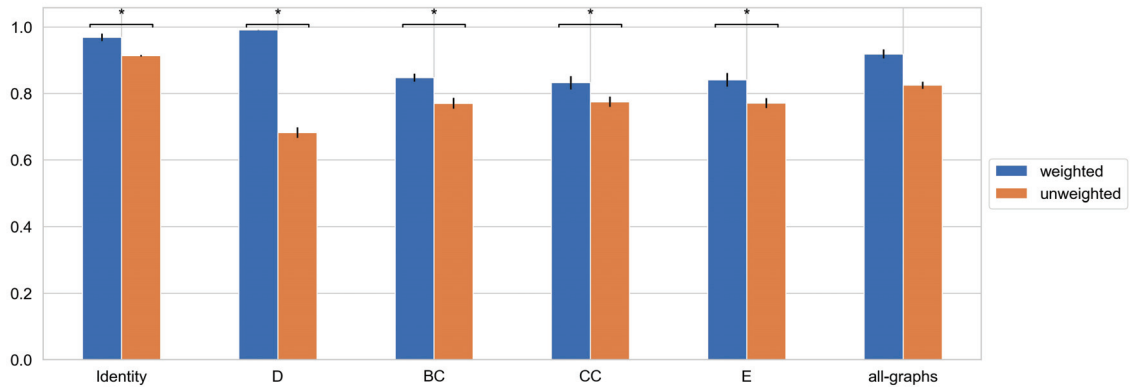


Figure 1.9: average F-Measure comparison for weighted and unweighted approach [(CIS+RR) vs (SP+PP)] for each feature [Degree (D), Betweenness Centrality (BC), Clustering Coefficient (CC), Local Efficiency (E), with all graph-metrics (all-graphs)] and without features (identity).

Our attention focalized on the classification of the MS clinical courses, differently from previous related works which mostly focused on the differentiation between MS patients from HC subjects [Charalambous *et al.* (2019), Zurita *et al.* (2018)a, Zhang *et al.* (2018)c, Maleki *et al.* (2012)]. The task addressed in this work has a strong clinical interest as the early clinical classification and thus prognostic of MS patients is the major challenge for neurologists today. Indeed, it is worth recalling that MS etiology is still unknown and that each MS patient may follow a different clinical course, resulting probably from the variety of the underground pathological mechanisms. More in detail, CIS and RR patients present comparable brain pathological processes, mainly inflammation, while SP and PP patients share neurodegenerative mechanisms. Indeed, as showed by the statistical analysis, few differences were found when comparing local graph metrics between these two pair of groups. As for the CIS vs RR comparison, these differences are mostly localized in the sub-cortical regions, temporal and parietal lobes, highlighting that early pathological processes start in central subcortical structures. Also, these differences are more related to weighted measures, thus showing that inflammation has a stronger impact on large WM fiber bundles.

Concerning the SP vs PP comparison, very few differences were found in the occipital, parietal, and temporal lobes, reflecting the similarity effect of the neurodegenerative process. Significant differences were mainly observed when comparing early stages of the disease (CIS,RR) with more severe clinical forms (SP,PP). As previously observed in literature [Charalambous *et al.* (2019), Kocevar *et al.* (2016)], indeed, a general reduction in network efficiency, density and clustering coefficient was observed in SP relative to RR patients due to severe brain damages. Finally, is interesting to notice the similarity between differences found in local efficiency and clustering coefficient (weighted). This result is in agreement with a recent study [Strang *et al.* (2018)], where these two metrics were found to be asymptotically linearly correlated in functional connectivity and various benchmark graphs.

However, despite difficulties in discriminating among groups using statistical markers, the proposed approaches achieved promising results. The proposed GCNN architecture was able to achieve good results operating with a relatively small number of parameters (about 42000 trainable

weights) compared with classical convolutional networks models working on images. Indeed, it is worth to note that the proposed method works on adjacency matrices of 84 nodes, significantly reducing the number of input units with respect to models directly operating on MR images [Zhang *et al.* (2018)c, Maleki *et al.* (2012)].

Interestingly, the main results were obtained using only the connectivity matrices, without graph-metrics as node descriptor. The brain structure, indeed, seems to contain highly discriminative properties characterizing the clinical profiles. Some of these properties are already mentioned in literature. According to the statistical analysis performed in [Kocevar *et al.* (2016)], for example, significant differences were found when comparing several graph metrics in progressive courses, reflecting the neurodegenerative mechanisms acting in the brain. However, as shown in our experiments, none of these properties used as node descriptors were effectively exploited by the neural network to discriminate the groups. By contrast, interesting improvement in classification performances were observed when using all the graph metrics together. The result is in agreement with a previous study [Kocevar *et al.* (2016)] where the combination of global graph-derived metrics provided the best results in the CIS vs RR, RR vs PP, CIS vs RR vs SP classification tasks. This may be explained considering that each local descriptor can provide useful information for a particular clinical course. Thus, exploiting them all together allows a better characterization of MS pathological alterations.

Interesting results were also observed when evaluating the capability of the network in classifying “early" stages of the pathology vs “progressive" stages. The proposed model was able to perform the binary classification task achieving high level of accuracy. Remarkable results were obtained considering the weighted Degree, which allowed the model to achieve the best performance. Indeed, it is worth to note that graph density decreases along with the progress of the pathology, due to neurodegenerative processes [Charalambous *et al.* (2019), Kocevar *et al.* (2016)].

Experiments including control patients were also reported in this paper. The proposed architecture was trained at discriminating between HC and early stages of the pathology (CIS and RR), and between HC and progressive stages (PP and SP). The model was able to achieve high performances. HC subjects were also used to perform a multiclass classification task using all the forms together. All the graph-metrics together as well as the featureless approach provided, on average, better results, confirming our previous observations. These results are somehow straightforward. Related studies have already shown several differences comparing brain structure representation of control subjects with respect to MS patiens [Charalambous *et al.* (2019), Zurita *et al.* (2018)a], due to pathological alterations. Such effects, cause HC networks to be more dense and well organized compared to MS, thus allowing an accurate discrimination [Kocevar *et al.* (2016)]. However, even if expected, these further analyses confirm the capability of the model to detect and exploit brain structure differences.

Finally, one of the main observation is related to the significant role played by edge weights in the classification task. As shown in our results, weights information allowed significant performance improvements in almost all the experiments. This achievement suggests that, despite comparable alterations in white matter network structure among groups may lead to misclassifi-

cation in some cases, the fiber bundles' strength provides a complementary information helpful to improve the overall accuracy.

5 Conclusion

In this chapter, we proposed a graph-based method to classify MS patients according to their clinical forms. Graph theory has been applied to describe brain network topology and Graph Convolutional Neural Networks have been used for the classification of MS clinical courses.

Thanks to a robust experimental activity, we showed the capability of GCNN to classify MS patients using the whole graph structure. In order to have a clear picture, we also enriched our analysis by combining the graph structure information with local graph-based metrics. Three major results were achieved by this analysis: *(i)* NNs are able to achieve high classification results using only the connectivity matrix *(ii)* local graph metrics do not improve the classification results suggesting that the latent features created by the NN in its layers have a much important informative content *(iii)* graph weights representation of brain connections preserve important information to discriminate between clinical forms. This result suggests that with graph binarization a lot of useful information may be lost.

It is worth to note that a limitation of this study is the small number of patients. However, we minimized these potential biases by using K-Fold cross-validation to generalize classification results. Further, the small number of each patient profile may not reflect the general population and induce biases in graph metrics results. As for future work, we aim at improving our method using a whole trail of longitudinal data collected for each patient as input for the model. In order to carry out this task, novel models proposed in literature may be taken into account. In this context, Recurrent Neural Networks [Medsker and Jain (1999)] have achieved remarkable results in dealing with short-long temporal relations ([Graves and Jaitly (2014), Fragkiadaki *et al.* (2015), Donahue *et al.* (2015)]) and can be effectively extended to handle graph data [Jain *et al.* (2016), Manessi *et al.* (2017), Jin and JaJa (2018)], achieving promising results. Another interesting perspective would be to perform a deep clinical analysis in order to understand the potential interest of such methods to better characterize the disease progression and thus better predict the patient evolution.

Chapter 2

Semisupervised Classification of Multiple Sclerosis Clinical Profiles

Contents

1 Introduction

Multiple Sclerosis (MS) is a chronic disease of the central nervous system that is characterized by inflammation, demyelination and neurodegenerative pathological processes [Polman *et al.* (2011)]. For 85% of the patients disease onset is characterized by a first acute clinical episode (called *clinically isolated syndrome* (CIS)), that includes optic neuritis, paresthesia, paresis and fatigue [McDonald *et al.* (2001)] and evolves first into a relapsing-remitting (RRMS) course, and then, after a delay between 15 to 20 years, into a secondary-progressive (SPMS) course, leading to long-term disability. The remaining 15% of MS patients starts with the primary-progressive course (PPMS) [Miller *et al.* (2005)a, Miller *et al.* (2005)b].

Several Magnetic Resonance Imaging (MRI) strategies have been defined to help establishing diagnosis of MS and monitor the course of the disease [Kocevar *et al.* (2016)]. Magnetization transfer, spectroscopy, and diffusion tensor imaging (DTI [Rovira *et al.* (2013)]) were successfully applied to detect alterations outside visible T2-lesions helping to understand pathological mechanisms occurring in an apparently normal white matter (WM). In particular, novel approaches based on the analysis of WM network by means of graph theory [Rubinov and Sporns (2010)] have achieved promising results in the characterization of either cognitive impairments or pathological alterations caused by different brain diseases, including MS.

In the last decade, the richness and variety of available data sets has opened a new world of findings in the biomedical field. Novel methods have been proposed to classify and delineate concepts, based on different tools and techniques [Gu and Angelov (2018)]. In particular, supervised machine learning approaches have gained strong results in several tasks, including MS clinical forms classification [Kocevar *et al.* (2016), Calimeri *et al.*, Calimeri *et al.* (2018)a]. In this context, a set of samples with known label or result is available and a model is prepared through a training process where it is required to make prediction and is corrected when those predictions are wrong. However, in many real scenarios, obtaining large number of labelled images is expensive and time consuming. Semi-supervised learning methods [Chapelle and Zien (2005), Zhu *et al.* (2003)] overcome this limitation considering both labelled and unlabelled data. These techniques allow to extend datasets by including samples which would have remained unused with traditional supervised approaches.

Several semi-supervised learning algorithms have been developed, mostly exploiting some clustering and manifold assumptions [Chapelle and Zien (2005), Chapelle *et al.* (2009)]. Typical family of algorithms are those built on support vector machines (SVM) [Wang *et al.* (2013), Chapelle *et al.* (2008)] and data graph [Belkin *et al.* (2004), Jiang *et al.* (2017)]. SVM-based methods attempt to balance the estimated maximum-margin hyperplane with a separation of all the data through the low-density regions. Graph-based approaches use the labelled and unlabelled data as vertices in a graph and build pairwise edges between the vertices weighted by similarities. Semi-supervised learning applications are particularly suited in the biomedical domain; indeed, in such context collecting a significant amount of “good" data is not always an easy task, due, for instance, to the high costs in terms of money and time required to perform screenings and analyses, or, as in the case of certain pathologies, to the number of case studies which is too limited for the creation of

data banks large enough to train physicians, experts, or artificial models. Several techniques have been developed to address various tasks, including medical image classification [Filipovych *et al.* (2011)], disease diagnosis [Zhao *et al.* (2014)], brain damages detection and segmentation [Baur *et al.* (2017)]. Another popular approach is based on the idea of *pseudo-label*. Such methods enlarge the training set by including unlabelled data, assigning to each unlabeled sample the class which has the maximum predicted probability and used as if it was the true label [Lee (2013), Wu and Yap (2006)]. Pseudo-labelling methods have recently gained particular attention also thanks to their combination with deep learning techniques [Lee (2013)].

In this chapter, we introduce a semi-supervised method to classify MS patients into four clinical profiles, using structural connectivity information. The main goal is to take advantage from structural brain connectivity and Neural Networks for a better characterization and classification of four MS clinical profiles [Clinical Isolated Syndrome (CIS), Relapsing Remitting (RR), Secondary Progressive (SP), Primary Progressive (PP)]. To the best of our knowledge, this is the first study which attempts to solve the question in a non-supervised manner, and could represent an important step forward on the road to the definition of completely automatic and unsupervised approaches.

2 Material and Methods

In this section we provide a detailed description of each step of the proposed approach, also describing the dataset used for the study.

2.1 Dataset Description

The dataset described in Section 2.1 was used in this study. The processing phase described in 2.2 was used to generate connectivity matrices. Each connectivity matrix is flattened in order to obtain a single 1-D array of edge-weight values. Edge weights were normalized to be in the real interval [0,1].

2.2 Deep Autoencoder Classifier

Given the structural connectivity information and limited number of labelled training samples, we could train a neural network to estimate the corresponding MS clinical form. The proposed architecture is composed by two main parts: *(i)* an *autoencoder* network which learns a latent representation for the input connectivity matrices; *(ii)* a fully-connected neural network which is trained to estimate the MS clinical form by taking as input the latent representation learnt by the autoencoder. During the training phase, the classifier aims at refining the latent space learned by the autoencoder, in order to make it useful for classification tasks. Roughly speaking, we force the autoencoder to learn features that depend on the true classification of the original data. These features hopefully hold some relevant information, useful to discriminate a MS group from another.

Formally, an autoencoder is a neural network trained to attempt to copy its input to its output [Goodfellow *et al.* (2016)]. Internally, it has a hidden layer h that describes a code used to represent the input. The network may be viewed as consisting of two parts: an encoder function $h = f(x)$ and a decoder that produces a reconstruction $r = g(h)$. The proposed autoencoder is composed by three layers. The encoder network encodes the input in a bi-dimensional space. The decoder network is its relative symmetric. We use linear activation functions in all encoder/decoder pairs, meaning that the code h is a linear combination of the input [Sperduti (2013), Pasa and Sperduti (2014)], as well as the reconstruction r, that in turn, ends to be a linear combination of the code:

$$h = W^{\top}\hat{h}_{encoder} + b_{encoder}$$

$$r = W^{\top}\hat{h}_{decoder} + b_{decoder}$$

where $\hat{h}$ is the vector of hidden parameters, and b are the bias values.

The classifier is defined by stacking two fully-connected layers of size 100 with Rectified Linear Units (ReLUs) activation followed by a fully connected layer with *softmax* activation to handle the classification problem. The input of the classifier network is the latent representation learned by the autoencoder. Its output is the probability distribution over c different classes.

2.3 Semisupervised Training

The whole architecture is trained simultaneously using the Adam [Kingma and Ba (2014)a] optimizer with a learning rate of 0.001. More in detail, to force the autoencoder learning a latent representation possibly helpful to perform classification we define a multi-objective optimization.

In particular, we need to train the classifier by minimizing the *categorical cross-entropy* [Goodfellow *et al.* (2016)] while forcing the autoencoder to minimize the Mean Squared Error (MSE) of the input w.r.t. the reconstruction. Formally, we need to minimize the following loss function:

$$\mathscr{L}(x,r,y,\hat{y}) = -\sum_{i}^{n} y_i log(\hat{y}_i) + \alpha \frac{1}{n}\sum_{i}^{n}(x-r)_i^2$$

where the first term is the cross-entropy between the true class label y, and its prediction $\hat{y}$ and the second term is the mean squared error between the input x and the reconstruction r. α is used to weight the reconstruction error.

However, we need to train the architecture in a semisupervised fashion, meaning that only a portion of true class label are available. For this reason we need to modify the loss function in order to handle this setting.

Let D be the set of available samples with $n = |D|$. Let $m \in \{0,1\}^n$ and H be the set of unlabelled instances. m is defined as follows:

$$m_i = \begin{cases} 0 \text{ if } D_i \in H \\ 1 \text{ otherwise} \end{cases}$$

The loss function $\mathscr{L}$ can be modified as follows:

$$\mathscr{L}_{semisup}(x,r,y,\hat{y}) = -\sum_{i}^{n} m_i y_i log(m_i \hat{y}_i) + \alpha \frac{1}{n} \sum_{i} (x-r)_i^2$$

In this case, m act as a mask for y and $\hat{y}$, by introducing a new “unknown" class. During the training, classification scores relative to unlabelled instances are annealed during the computation of the final loss.

2.4 Pseudo-labelling Training

Pseudo-labelling is a method of semi-supervised learning for deep neural networks which has gained particular attention in recent years [Lee (2013)]. It consists in training the architecture in a semi-supervised fashion, where unlabelled data are associated with the class with the maximum predicted probability. In order to improve generalization performances, we propose an iterative approach which acts similarly: as described in Algorithm 1 the network is trained using the semi-supervised method of Section 2.3. Then, at each iteration i, we assign to unlabelled data the class with maximum predicted probability, thus iteratively expanding the set of known labels until a particular stopping criterion is met (e.g. performances do not improve for a certain number of iterations).

```
Data:
X: training instances;
y: training labels;
H: indices of unknown labels
Result: model: the trained model
initialization(model);
while stopping criterion is met do
    m ← create_mask(H);
    model ← semisupervised_train(X,y,m);
    predictions ← model.predict(X);
    best_indices ← sample(predictions,H);
    for idx ∈ best_indices do
        y[idx] = predictions[idx];
        remove(H,idx)
    end
end
```

Algorithm 1: Semisupervised training for MS clinical forms classification

It is worth noting that the quality of pseudo-labelling methods strictly depends on the accuracy of predicted labels. Introducing inappropriately instances, indeed, can degrade instead of improving the performance [Lee (2013), Wu and Yap (2006)]. In this work, we select unlabelled samples according to the scores of confidence of the estimated labels. More in detail, we introduce in the known label set, iteratively, only a portion of maximum predicted probability instances. Two sampling approaches were compared: (i) the balanced approach, in which we select a comparable number of instances from each classes; (ii) the proportional approach, in which the number of instances to select is proportional to the original distribution of true labels. In more detail, using the

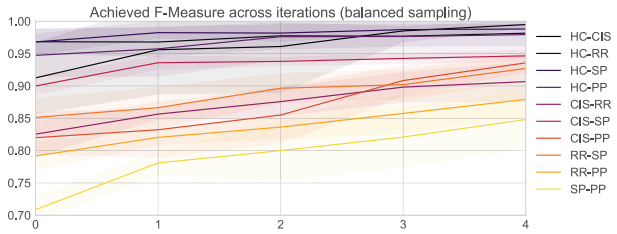


Figure 2.1: Results, in terms of F-Measure for each binary task, obtained at each iteration using balanced sampling.

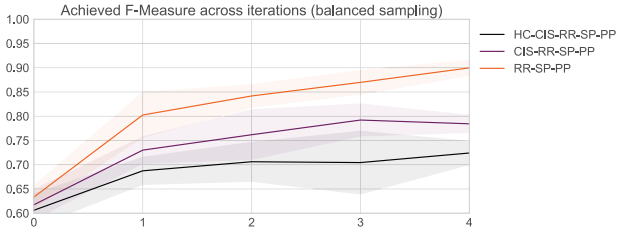


Figure 2.2: Results, in terms of F-Measure for each multiclass task, obtained at each iteration using balanced sampling.

balanced approach, at each iteration, the same percentage of labels from each class is selected to be included in the set of known labels. Using the proportional approach, we compute for each class its relative frequency in the set of known labels. Then, at each iteration, a number of instances proportional to such frequencies is selected to be included in the set of known labels. It is worth to note that the latter method is useful when some prior knowledge on the data distribution is known.

3 Results

3.1 Evaluation of sample selection criteria

In order to evaluate the effect of the different sampling criteria presented in this work, we performed an extensive experimental campaign. More in details, we analyzed and compared the performances obtained using balanced and proportional sampling approaches. Goal of this analysis is to understand how the sample selection criteria can influence our algorithm. All the experiments described in this section were obtained using only the 10% of initial labeled data. Moreover, each experiment was performed 5 times in order to get, for each task, mean and standard deviation of the performances.

The first analysis is described in Figure 2.1 where we show the results obtained with balanced sampling for binary semi-supervised tasks. From the analysis of Figure 2.1, it is possible to see how the classification performances reach high values of F-Measure starting already from the second iteration. Indeed, for the binary classification, all the tasks reach, at iteration 1, a F-Measure > 0.75. For the easiest classification tasks: HC - CIS, HC - RR, HC - SP and HC - PP the algorithm reaches high F-measure values at the first iteration and then stabilizes. On the other side, for the most difficult tasks: SP - PP, RR - PP and RR - SP, the classification performances increase at each

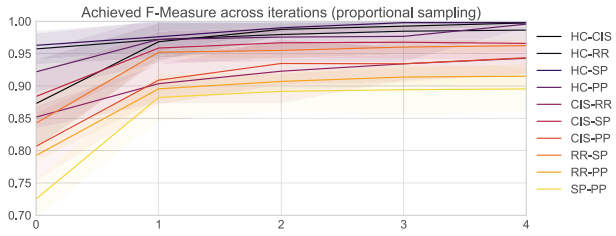


Figure 2.3: Results, in terms of F-Measure for each binary task, obtained at each iteration using proportional sampling.

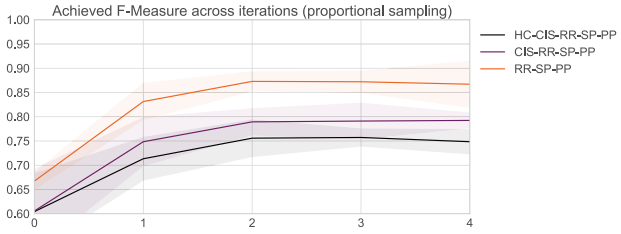


Figure 2.4: Results, in terms of F-Measure for each multiclass task, obtained at each iteration using proportional sampling.

iteration.

For the multi-class task, the behaviour is slightly different as described in Figure 2.2. Task RR-SP-PP reaches a F-Measure > 0.8 already at iteration 1 whereas CIS-RR-SP-PP reaches a similar value at iteration 3. As expected the most complex multi-class task HC-CIS-RR-SP-PP reaches its best F-Measure at the last iteration (0.73).

In Figure 2.3 we show the results obtained with proportional sampling for binary semi-supervised tasks. As observable, performances increase rapidly at iteration 1, reaching high level of F-Measure. Then, they stabilize in the next iterations. A similar behaviour is observable in Figure 2.4, where we show the results obtained with proportional sampling for the multiclass semi-supervised tasks.

Finally, in order to have a fair comparison of the two sample selection criteria, we analyzed the difference, in terms of F-Measure ($\Delta_F(it_i, it_j)$), between two consecutive iterations it_i, it_j. Goal of this analysis is to understand the performances gain obtained in each iteration using the two methods. The results are reported in Table 2.1. From this comparison it is possible to see how the proportional selection criteria (P) allows to get a better improvement from it_0 to it_1. This difference is not visible in the other iterations where the improvements are stable for both the selection criteria.

3.2 Evaluation of initial number of labeled data

The number of initial labeled data was investigate in order to understand the minimum number of labeled samples to use in order to get satisfiable results. In more detail, we performed several experiments using a different percentage of initial randomly selected samples, ranging from 10% to 50% of the total population. Each experiment was performed 5 times for each task. Results, in

Task Δ_F	$\Delta_F(it_1, it_0)$		$\Delta_F(it_2, it_1)$		$\Delta_F(it_3, it_2)$		$\Delta_F(it_4, it_3)$	
	B	P	B	P	B	P	B	P
HC-CIS	0.04	**0.1**	0.01	0.02	0.02	0.0	0.01	0.01
HC-PP	0.01	**0.05**	0.02	0.0	0.0	0.0	0.0	0.02
HC-RR	0.0	**0.01**	0.01	0.01	0.0	0.01	0.0	0.0
HC-SP	0.01	0.01	0.0	0.01	0.01	0.01	0.0	0.0
CIS-PP	0.01	**0.1**	0.02	0.03	0.05	0.0	0.03	0.01
CIS-RR	0.03	**0.05**	0.02	0.02	0.02	0.01	0.01	0.01
CIS-SP	0.04	**0.07**	0.0	0.01	0.0	0.0	0.0	0.0
RR-PP	0.03	**0.1**	0.02	0.01	0.02	0.01	0.02	0.0
RR-SP	0.02	**0.11**	0.03	0.0	0.01	0.01	0.02	0.0
SP-PP	0.07	**0.16**	0.02	0.01	0.02	0.0	0.03	0.0
RR-SP-PP	**0.17**	0.16	0.04	0.04	0.03	0.0	0.03	0.01
CIS-RR-SP-PP	0.11	**0.14**	0.03	0.04	0.03	0.0	0.01	0.0
HC-CIS-RR-SP-PP	0.08	**0.11**	0.02	0.04	0.0	0.0	0.02	0.01

Table 2.1: Delta of F-Measure ($\Delta_F(it_i, it_j)$), between two consecutive iterations it_i, it_j for balanced sampling (B) and proportional (P) selection criteria.

	0	1	2	3	4
HC - CIS - RR - SP - PP	0.60 (0.08)	0.71 (0.05)	0.76 (0.04)	0.76 (0.02)	0.75 (0.03)
CIS - RR - SP - PP	0.61 (0.09)	0.75 (0.05)	0.79 (0.03)	0.79 (0.04)	0.79 (0.02)
RR - SP - PP	0.67 (0.02)	0.83 (0.04)	0.87 (0.02)	0.87 (0.02)	0.87 (0.05)

Table 2.2: Mean and standard deviation (in parenthesis) of F-Measure per iteration. With 10% of ground truth.

terms of F-Measure, are reported in Figure 2.5. Obviously, the best results were achieved using 50% of the labelled data, reaching a mean F-Measure $98\% \pm 1\%$ for all the binary tasks. From the analysis of these results it comes out that already with 10% of labeled data good results were achieved with a mean F-Measure $94\% \pm 5\%$.

3.3 Subsample iteration selection

According to our algorithm, the semi-supervised procedure assigns, at each iteration, a label to all the samples. We then evaluated the number of samples to select at each iteration to be included as labeled data for the next one. We performed different experiments using the proportional selection criteria and changing the percentage of new labeled data to select. Each experiment was performed 5 times.

Results, in terms of F-Measure, for difference percentages (ranging from 5% to 20%) are reported in Figure 2.6. For all the binary tasks, as expected, best results were achieved using the 20% of the new labelled data with a mean F-Measure of $97\% \pm 2\%$. It is worth noting that good results are already achieved using only 5% of the new labeled data with a mean of $93\% \pm 3\%$. Indeed performances from 10% to 20% do not show large improvements.

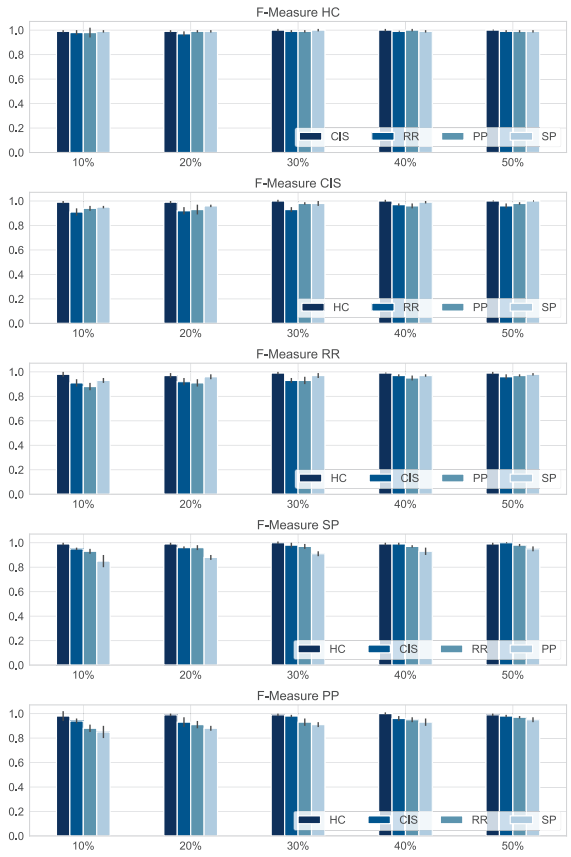


Figure 2.5: Results, in terms of F-Measure for each binary task using different percentages of initial labeled samples.

	0	1	2	3	4
HC - CIS - RR - SP - PP	0.61 (0.04)	0.69 (0.03)	0.71 (0.04)	0.70 (0.07)	0.72 (0.02)
CIS - RR - SP - PP	0.62 (0.03)	0.73 (0.03)	0.76 (0.05)	0.79 (0.03)	0.78 (0.02)
RR - SP - PP	0.63 (0.02)	0.80 (0.05)	0.84 (0.02)	0.87 (0.03)	0.90 (0.02)

Table 2.3: Mean and standard deviation (in parenthesis) of F-Measure per iteration. Balanced sampling 10%.

Figure 2.6: Results, in terms of F-Measure for each binary task using different percentages of selected sample in each iteration.

	0	1	2	3	4
ACC	0.69	0.84	0.88	0.93	0.93
AMI	0.08	0.34	0.45	0.61	0.64
Fm	0.69	0.84	0.88	0.93	0.93
NMI	0.08	0.35	0.46	0.62	0.66
Prec	0.70	0.84	0.88	0.93	0.94
Rec	0.69	0.84	0.88	0.93	0.93

Table 2.4: Performances of the encoded representation computed at each iteration for the SP - PP task.

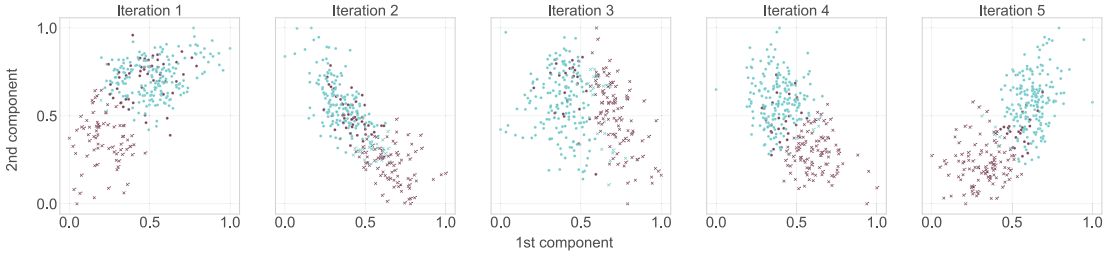


Figure 2.7: Visualization of bidimensional encoding space generated by autoencoder from iteration 1 (left) to iteration 5 (right) in SP (blue) vs PP (violet) task.

3.4 Encoding space analysis

A final evaluation was performed on the encoding space generated by the encoding layer of the proposed NN. In more detail, we analyzed, at each iteration, the latent representation of all graphs generated by the NN. In order to understand if the latent representation showx difference between groups, the following performance measures were computed: Normalized Mutual Information (NMI), Adjuster Mutual Information (AMI), Accuracy (ACC), Precision (Prec), Recall (Rec) and F-Measure (Fm). In Table 2.4 we reported the performance metrics computed, at each iteration, for the SP - PP task. Moreover, we reported in Figure 2.7 the evolution of bidimensional encoding space representation. It is interesting to observe a clearer separation between classes as the algorithm moves forward.

4 Discussion

In particular contexts, such as the biomedical domain, large amount of labelled data are not easy to obtain. Semisupervised classification approaches constitute an important resource to address this issue. In this study, we proposed a novel method based on neural networks, to automatically classify MS patients using brain structural connectivity information. For the first time, the problem is addressed in a semisupervised manner, including unlabelled data that could not be used in the previously proposed methods.

Several binary classification tasks were performed, including HC-CIS, CIS-RR, RR-PP, RR-SP, SP-PP. In addition, multiclass classification tasks were included in the study. High levels of accuracy and F-Measure were achieved both for easy and more difficult classification tasks,

showing the capability of the method to exploit pathological alterations to differentiate MS clinical profiles.

The highest level of F-Measure were obtained in the HC vs MS classification tasks. These results were expected. Indeed, related studies have already shown several differences comparing brain structure representation of control subjects with respect to MS patients [Charalambous *et al.* (2019), Zurita *et al.* (2018)a], due to pathological alterations. Such effects, cause HC networks to be more dense and well organized compared to MS, thus allowing an accurate discrimination [Kocevar *et al.* (2016)].

However, high classification performances were also achieved in the more difficult tasks, such as SP-PP and CIS-RR. Indeed, despite comparable white matter alterations were observed in those two pairs of clinical profiles, neural networks were able to detect important information to discriminate between clinical forms.

In machine learning, the composition of the training set may affect the performance of the model. This is also true in the proposed semisupervised method, where the initial distribution of labeled data may impact the final classification accuracy. We evaluated the performance variations according to the initial number of labeled data. Interestingly, good performances were achieved using a moderate amount of labeled data, demonstrating the capability of the algorithm of generalizing predictions on unseen data even with a small set of training samples.

In the proposed method, the initial number of labelled data is iteratively increased pseudo-labelling data using the network predictions. Two sampling criteria were proposed in this study, namely balanced and proportional, to iteratively augment the set of known labels. However, other sampling approaches could be implemented to address different tasks. Comparable final performances were obtained in all the binary classification tasks and in the multiclass classification tasks when using the two sampling criteria. However, an important difference was observed by evaluating the gain obtained in each iteration using the two methods. As observable from Table 2.1, the proportional sampling outperforms the balanced sampling at the beginning of the process, resulting in faster convergence of the algorithm. Indeed, in the balanced approach, the number of new labels included in the process may not reflect the true data distribution. By contrast, when some prior knowledge of such distribution is known, it can be exploited by the proportional method to improve the learning process.

Finally, the evolution of the encoding space was analyzed to better understand the behaviour of the classifier. We noticed that clusters were better organized as performances increase, that is, the encoding part of the network learned similar latent representations for objects belonging to the same class. This result could be interpreted as common latent features are shared among instances of the same class.

5 Conclusion

In this work we proposed a semi-supervised approach to classify MS patients into four clinical profiles. Pseudo-labelling was used to iteratively train a deep neural network in a semi-supervised manner. The experimental activities illustrated the robustness of our approach to variations respect

to the initial training set size and with respect to the new knowledge to include in the further steps of the process. Two main results achieved by this analysis: *(*i) unlabelled data could be effectively used in the training process to classify MS patients *(*ii) good classification performance could be achieved by using a moderate amount of labelled data. Achieved results could represent an important step forward on the road to the definition of completely automatic and unsupervised approaches.

Chapter 3

Tensor Factorization for Unsupervised Multiple Sclerosis Detection

Contents

1 Introduction

In recent years, several studies were conducted to identify MS from healthy controls with the help of MRI. In particular, quantify brain damages by analyzing the white matter network is a promising technique to detect pathological alterations useful to delineate biomarkers to characterize the disease. indeed, network metrics can provide clinically relevant information about MS pathology. In [Charalambous *et al.* (2019)], several structural changes were identified between subjects' groups. Compared with controls, network efficiency and clustering coefficient were reduced in MS. Furthermore, in [Kocevar *et al.* (2016)], author observed greater assortativity, transitivity, and characteristic path length as well as a lower global efficiency when comparing MS patients to HC subjects, probably corresponding to transient damages caused by inflammatory and demyelinating processes.

On the other hand, several methods based on computer vision and image processing techniques to accomplish MS automatic identification tasks were developed. Such methods mostly exploit recent advances in machine learning. Zurita *et al.* [Zurita *et al.* (2018)b] developed classifiers based on DTI and fMRI data able to distinguish between patients and healthy subjects, reaching accuracy levels as high as $89\% \pm 2\%$. Based on a simple DTI acquisition associated with structural brain connectivity analysis, Kocevar *et al.* proposed a method based on support vector machines which allowed an accurate classification of HC subject and MS patients' clinical profiles achieving high level of accuracy (F-Measure 91.8%). Whang *et al.* exploited complex Convolutional Neural Networks (CNN) to identify MS from healthy control with an accuracy greater than 98% starting from Brain MRI slices [Wang *et al.* (2018)]. The same task was addressed by Maleki *et al.* and Zhang *et al.* where CNN achieved similar results [Maleki *et al.* (2012), Zhang *et al.* (2018)c].

As we already shown in Chapter 2, good results can be achieved by semi-supervising the training of complex deep learning models. However, obtaining large number of labelled images remains a relevant issue, since the task is often expensive and time consuming. For this reason, adopting completely unsupervised (i.e. without any human intervention) methods constitutes one of the main direction of our researches. In particular, in this study we propose a combination of the above mentioned approaches: we take advantages from structural connectivity and machine learning methods to develop a fully automatic method to classify MS patients and HC subjects. Furthermore, we proposed a completely unsupervised method. To the best of our knowledge, this is the first attempt to solve this question in a fully unsupervised automated manner, which constitutes an important improvement.

2 Material and Methods

The proposed method is composed by five main steps: (i) structural connectivity information are extracted from the images in order to produce a graph representation of the MRI; (ii) the adjacency matrix representation of the produced graphs is used to create a third order tensor (iii) the tensor is decomposed to create a latent representation of the dataset (iv) unsupervised clustering on the latent space was performed using the *k-means* algorithm.In this section we provide a detailed

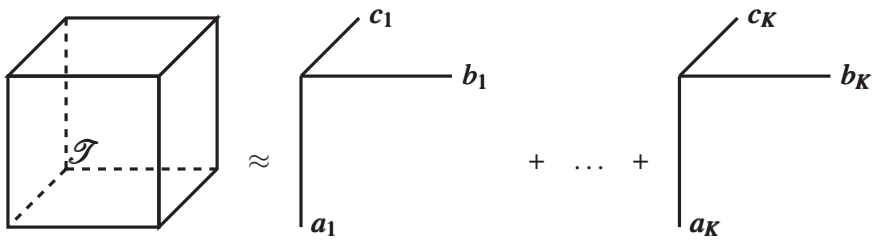


Figure 3.1: Graphic representation of the canonical polyadic decomposition (CPD).

description of each step of the proposed approach, also describing the dataset used for the study.

2.1 Notation

In this paper, to better defining the proposed method, we denote scalar values with small letters (e.g., a), 1-dimensional vectors with bold small letters (e.g., $\boldsymbol{a}$), matrices with boldface capital letters (e.g., $\boldsymbol{A}$). Tensors are denoted with boldface Euler script letters (e.g., $\boldsymbol{\mathscr{A}}$). $\mathscr{G} = (V,E)$ is an undirected graph, where V is the set of vertices and E is the set of edges. If not differently specified, given a graph $\mathscr{G} = (V,E)$ we denote by $N = |V|$ the number of nodes of the graph.

2.2 Dimensionality Reduction using Tensor Factorization

A *tensor* is a multidimensional array. Formally, a tensor is an element of the tensor product of D vector spaces each defined in its own coordinate system [Kolda and Bader (2009)].

In recent years, tensor factorization (decomposition) became a widely used method for identifying relations and correlations in high dimensional data. Many decomposition techniques have been developed and found their application in many research fields such as signal processing, computer vision, numerical analysis, data mining, neuroscience, graph analysis, and more. In particular a number of different factorization methods have been proposed in the literature, such as Higher Order SVD (HOSVD) [Hunyadi *et al.* (2014)], Tucker decomposition [Tucker (1966)], Parallel Factor (also known as PARAFAC or CANDECOMP or CP) [Harshman (1977)] and Non-negative Tensor Factorization (NTF) [Shashua and Hazan (2005)]. These techniques are mainly generalizations of the Singular Value Decomposition (SVD) [Golub and Reinsch (1971)]. In this study, we refer to the canonical polyadic decomposition (CPD) method [Asta and Özcan (2015)]. In particular, given a third-order tensor $\boldsymbol{\mathscr{X}} \in \mathbb{R}^{i \times j \times z}$ the aim is to write it as a sum of component rank-one tensors:

$$\boldsymbol{\mathscr{X}} = \sum_{r=1}^{K} \boldsymbol{a_r} \circ \boldsymbol{b_r} \circ \boldsymbol{c_r} + \boldsymbol{\mathscr{E}} \tag{3.1}$$

where $K \in \mathbb{N}_+, \boldsymbol{a_r} \in \mathbb{R}^i, \boldsymbol{b_r} \in \mathbb{R}^j, \boldsymbol{c_r} \in \mathbb{R}^z \; \forall \; 1 \leq r \leq K$ are the component vectors and $\boldsymbol{\mathscr{E}} \in \mathbb{R}^{i \times j \times z}$ is denoted as the error tensor. The symbol "$\circ$" represents the vector outer product.

In an exact CP decomposition, where "exact" means that the residual tensor $\boldsymbol{\mathscr{E}}$ in equation 3.1 is a zero-element tensor, the smallest number of rank-one tensors that generate $\boldsymbol{\mathscr{X}}$ as their sum

Table 3.1: Clustering performance comparison with and without dimensionality reduction.

	Accuracy			F-Measure			Precision			Recall		
	HC vs CIS	HC vs PP	HC vs RR	HC vs CIS	HC vs PP	HC vs RR	HC vs CIS	HC vs PP	HC vs RR	HC vs CIS	HC vs PP	HC vs RR
No reduction	0.97	0.81	0.50	**1.00**	0.88	0.62	1.00	0.79	0.54	1.00	0.99	0.95
PCA	**0.99**	0.81	0.53	**1.00**	0.69	0.61	1.00	0.79	0.54	1.00	0.54	0.89
Tensor	0.94	**0.99**	**0.99**	0.95	**1.00**	**0.99**	0.97	1.00	0.99	0.91	1.00	0.96
TSNE	0.70	0.55	0.61	0.58	0.59	0.73	0.65	0.54	0.58	0.52	0.81	0.98

is called the rank of $\mathscr{X}$, denoted $rank(\mathscr{X})$. An exact CP decomposition with $K = rank(\mathscr{X})$ components is referred to as the rank decomposition.

The CPD problem can be formalized as follows:

$$\min_{\hat{\mathscr{X}}} \|\mathscr{X} - \hat{\mathscr{X}}\| \quad \text{with} \quad \hat{\mathscr{X}} = \sum_{r=1}^{K} \boldsymbol{a_r} \circ \boldsymbol{b_r} \circ \boldsymbol{c_r} = [\![\boldsymbol{A};\boldsymbol{B};\boldsymbol{C}]\!] \tag{3.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{i \times K}$, $\boldsymbol{B} \in \mathbb{R}^{j \times K}$, $\boldsymbol{C} \in \mathbb{R}^{z \times K}$.

The factorization is then obtained by solving the following optimization problem:

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \|\mathscr{T} - \boldsymbol{A} \circ \boldsymbol{B} \circ \boldsymbol{C}\| \tag{3.3}$$

The formulation of the tensor factorization using CPD is graphically illustrated in Figure 3.1.

Tensor factorization techniques feature interesting properties that turn out to be of use in many contexts. Indeed, factorization helps to represent data in a more concise and generalizable manner, which can be of specific use according to various criteria and applications.

Our approach is based on the intuition that tensor factorization allows to identify correlations and relations among high dimensional data. A third-order tensor $\mathscr{T}$ is created, which encodes the t graphs generated from the MR images (in their adjacency matrix form). Then, the tensor $\mathscr{T}$ is factorized in its basic factors, with $K = 2$, producing the approximation $\hat{\mathscr{T}}$ as described the equation 3.1, where the length of each vectors $\boldsymbol{a}_r, \boldsymbol{b}_r$ and $\boldsymbol{c}_r \; \forall \; 1 \leq r \leq K$, are N, N and t respectively.

It is worth to note that the two components $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ represent a projection of each graph in the euclidean space. Thus, we used this latent representation to perform the unsupervised clustering task.

2.3 Unsupervised Clustering

K-means clustering is a type of unsupervised learning algorithm used on unlabelled data to find K groups (i.e. clusters). The algorithm works iteratively to assign each data point to one of the groups based on the features that are provided. Data points are clustered based on feature similarity. In the proposed approach the latent representation of the dataset was used as input to the algorithm to perform perform to detect 2 groups, eventually corresponding to the MS and HC classes. The output of the k-means clustering is a vector of labels, specifying for each voxel whether it belongs to one or the other cluster by assigning it the values 0 or 1.

3 Results

3.1 MS Identification Using Clustering Analysis

First, we analyze whether the clustering accuracy benefits from a latent representation of the brain connectivity graphs. In the proposed experiments, the latent representation was used as input for the k-means to compute two clusters. Additionally to the proposed method based on tensor factorization, we compared performances using two widely used dimensionality reduction algorithms: Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (TSNE). Three different binary classification tasks were performed (HC vs CIS, HC vs RR, HC vs PP). Quality of the classification was assessed by means of Precision, Recall, F-Measure and Accuracy. Results are reported in Table 3.1.

High level of accuracy were observed already when using the k-means algorithm without dimensionality reduction. However, PCA and Tensor factorization allow to achieve better results. In particular, the best results are achieved using PCA for the for the HC vs CIS task while the best results are achieved using tensor factorization for HC vs RR and HC vs PP. However, the proposed method allow to achieve the most stable results all over the experiments. Worse performances are observed using TSNE.

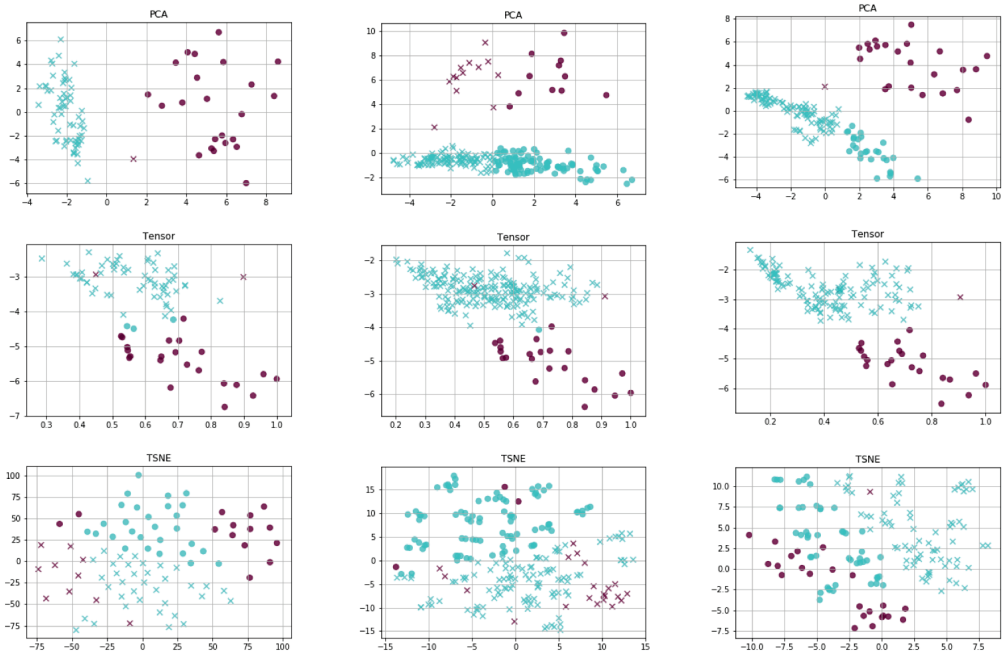
The latent representations computed for each task and for each dimensionality reduction algorithms are reported in Figure 3.2. A well defined separation between the two clusters can be observed for PCA and tensor factorization. Interestingly, when using PCA for HC vs RR and HC vs PP, two well defined areas can be observed, which, however, are not related to the real classes. This is probably due to common structural similarities among the patients, not directly related to the pathology.

3.2 Unsupervised Classification of MS Clinical Profiles

In this section, a preliminary study concerning the unsupervised classification of MS clinical profiles. As shown in the previous sections, dimensionality reduction using tensor factorization of structural connectivities provides a stable method for distinguish MS patients from HC subjects. For the sake of completeness, we performed preliminary experiments to investigate whether such method can be useful to discriminate the various forms of MS.

MS Forms Clustering Using Brain Connectivity

The proposed dimensionality reduction approach based on Tensor Factorization was applied for each MS patient. The obtained latent representation was used as input for the k-means to compute two clusters. Additionally, the k-means algorithm was feed using directly the adjacency matrix representation, for comparison. For this experiment, we performed pairwise comparison of clinical interest, namely RR vs CIS, RR vs PP, RR vs SP and PP vs SP. Table 3.4 and Table 3.5 illustrate obtained results using unweighted and weighted graphs, respectively. As observable, the proposed approach is not able to achieve suitable results in many cases. When using



(a) HC vs CIS (b) HC vs RR (c) HC vs PP

Figure 3.2: Illustration of the latent space computed by tensor factorization for each experiment (HC vs CIS , HC vs RR, HC vs PP) and for PCA, Tensor factorization and TSNE. Different shapes represent the two clusters assigned by the k-means algorithm while different colors represent the real classes.

Table 3.2: pairwise MS clinical profile clustering performance comparison using unweighted graph metrics

	F-Measure	Precision	Recall	Accuracy
RR vs CIS	0.48	0.63	0.50	0.51
RR vs PP	0.69	0.78	0.66	0.66
RR vs SP	0.75	0.80	0.74	0.74
PP vs SP	0.59	0.64	0.59	0.59

Table 3.3: pairwise MS clinical profile clustering performance comparison using weighted graph metrics

	F-Measure	Precision	Recall	Accuracy
RR vs CIS	0.45	0.55	0.48	0.48
RR vs PP	0.66	0.67	0.66	0.66
RR vs SP	0.70	0.69	0.70	0.70
PP vs SP	0.57	0.58	0.58	0.57

either weighted or unweighted graphs, the worst results were obtained comparing RR vs CIS. Furthermore, using unweighted graphs, low level of accuracy was obtained comparing SP vs PP. By contrast, better results were obtained when comparing RR vs SP (using weighted and unweighted graphs). Furthermore, we observed that unweighted graphs provided better results, overall. Moreover, dimensionality reduction did not provide better results.

MS Forms Clustering Using Brain Connectivity Metrics

In this preliminary study, we did not use directly the adjacency matrix representing the brain structure. Instead, the matrix $A \in \mathscr{R} \times d$, $(d = 4)$ of local graph metrics already presented in Section 2.5 (Node Degree, Clustering Coefficient, Local Efficiency, Betweenness Centrality) was computed for each subject. Then, a third-order tensor $\mathscr{T}$ is created, which encodes the feature matrices, and factorized in its basic factors, with $K = 2$, as previously illustrated. Table 3.2 and Table 3.3 illustrate obtained results using unweighted and weighted graphs, respectively. Obtained results are comparable with the approach using directly connectivity matrices: low levels of accuracy were observed comparing RR vs CIS as well as PP vs SP. The best results were obtained comparing RR vs SP. Even in this case, dimensionality reduction did not provide better results.

Table 3.4: Pairwise MS clinical forms clustering performance comparison with and without dimensionality reduction using unweighted graphs.

	Accuracy		F-Measure		Precision		Recall	
	No reduction	Tensor	No reduction	Tensor	No reduction	Tensor	No reduction	Tensor
RR vs CIS	0.53	0.53	0.45	0.45	0.67	0.67	0.47	0.47
RR vs PP	0.65	0.66	0.40	0.40	0.61	0.61	0.62	0.60
RR vs SP	0.75	0.76	0.71	0.71	0.60	0.62	0.81	0.83
SP vs PP	0.65	0.65	0.64	0.66	0.67	0.66	0.65	0.65

Table 3.5: Pairwise MS clinical forms clustering performance comparison with and without dimensionality reduction using weighted graphs.

	Accuracy		F-Measure		Precision		Recall	
	No reduction	Tensor	No reduction	Tensor	No reduction	Tensor	No reduction	Tensor
RR vs CIS	0.55	0.55	0.42	0.42	0.53	0.53	0.44	0.46
RR vs PP	0.63	0.63	0.53	0.52	0.51	0.52	0.53	0.53
RR vs SP	0.70	0.70	0.68	0.68	0.66	0.65	0.70	0.71
SP vs PP	0.51	0.52	0.51	0.51	0.47	0.50	0.62	0.62

4 Discussion

Graph representation of the brain structure offers a sensitive tool to detect pathological alterations respect to healthy subjects. As already suggested by previous work, this kind of representation enhances several structural changes in MS brain networks compared with controls. As shown in our experiments, such structural variations are easily captured by clustering algorithms, and can be exploited to create a more readable representation in a bi-dimensional space. However, as observed, dimensionality reduction allowed to achieve more stable and better results. In particular, the proposed approach based on tensor factorization of structural connectivities provided an effective way to extract suitable features to improve clustering results for all the binary tasks. It is worth to note that the proposed method requires a number of parameters significantly lower with respect to previous approaches based on complex deep learning structures.

Concerning the MS clinical forms comparison, we observed how the proposed unsupervised algorithm did not provide suitable performances. The CIS and RR patients comparison provided the worst results in all the experiments. These two preliminary MS stages, indeed, presents several structure similarities which, probably, prevent the algorithm to find a proper separation criteria. Similar results were observed when comparing SP and PP patients, which are characterized by similar degenerative brain damages, are associated to different disability values. For this reason, the pathological mechanisms differentiating these two stages are too complex to be automatically modelled by simply looking at the brain structure in a whole. Differently, the RR vs SP comparison provided interesting results. RR patients are characterized by inflammation processes while SP patients are more subjected to neurodegeneration, leeading the clustering algorithm to find more suitable structural difference between the two classes.

5 Conclusion

In this chapter, we presented a preliminary study aimed at solving the MS identification and classification problem in a completely unsupervised manner. As observed in the experiments, graph representation of the brain structure allow unsupervised algorithm to detect and exploit differences between HC and MS. By contrast, the same approach were not successful in detecting the complex pathological mechanisms of MS. For this reason, more sensitive tools need to be investigated, including innovative unsupervised clustering approaches. In this context, Graph Neural Networks could play a key role, given their ability to model complex non-linear representations

of structured data. Furthermore, less general approach should be explored: since pathological mechanisms act differently in each stages, more “class-oriented" methods could be defined, which could take into account the clinical background knowledge about MS.

Chapter 4

Beyond Classification: A Logic-Based Framework Leveraging Neural Networks for Studying the Evolution of Neurological Disorders

Contents

1 Introduction

After a number of works on automatic theorem proving and artificial intelligence, the work on logic programming took actually off in the 1970's, with the aim of obtaining automated deduction systems. Answer Set Programming (ASP) [Gelfond and Lifschitz (1991), Niemelä (1999), Lonc and Truszczynski (2006)] is one of the several formalisms that stemmed out of such research efforts, and during the years it turned out to be a powerful declarative formalism for knowledge representation and reasoning (KRR).

After more than twenty-five years of scientific research, the theoretical properties of the language are considered to be well understood, and even if the community is still very active on several extensions, the solving technology, as witnessed by the availability of a number of robust and efficient systems [Gebser *et al.* (2017), Gebser *et al.* (2018)], is mature for practical applications [Brooks *et al.* (2007), Terracina *et al.* (2008), Gebser *et al.* (2011)b, Manna *et al.* (2015)]. In the latest years, ASP has been indeed employed in many different domains, and used for the development of enterprise and industrial-level applications [Calimeri and Ricca (2013), Leone and Ricca (2015), Erdem *et al.* (2016)], fostered by the release of a variety of proper development tools and interoperability mechanisms for allowing interaction and integration with external systems [Ricca (2003), Calimeri *et al.* (2007)]

Nevertheless, even though Answer Set Programming, and deductive approaches via logic formalisms in general, are of wide use for Artificial Intelligence (AI) applications, they are not the ultimate, comprehensive solution to AI, as some kind of problems can be hardly encoded by logic rules. This can be due to several reasons: the nature of the problem, the lack of proper development tools, and severe performance issues even for the best performing systems. Bioinformatics is a prime example, as both related data (think, for instance, of biomedical images, temporal data, etc.) and relevant tasks (think, for instance, of classification) are not naturally approachable with deductive strategies. In such research area, and similar ones, different AI-based strategies have been employed. Lately, approaches relying on Machine Learning (ML) and Artificial Neural Networks (ANNs) [Haykin (1998), Goodfellow *et al.* (2016)] are on the rise, both because of the great results achieved by the research community in the latest years and because they better deal with the data and the nature of the tasks of interest.

Basically, with ANNs the problem is not actually modelled, and its structure remains almost unknown; rather, approaches progressively learn, usually by examples, the best answers to provide in presence of specific inputs. ANNs can learn and model non-linear complex relationships, and even generalize acquired knowledge in order to infer new properties over unseen data; moreover, once trained, a neural network can be extremely fast in providing answers to instances of complex problems. Unfortunately, obtained results have only statistical significance; it is noteworthy, indeed, that the main weakness of ANNs is in general their incompleteness, since their precision may strongly depend on the training phase and on the quality of the training data. The logic they use can be sound, yet proven incorrect by further observations. Hence, clearly, as for deductive reasoning, not all problems can be properly solved by ANNs.

This chapter focuses on some opportunities provided by a combined use of ASP and ANN.

As a motivating scenario, we concentrate on a relevant bioinformatics problem, namely the study of neurological disorders and, specifically, on the Multiple Sclerosis (MS) disease. To the best of our knowledge, computer science research on this topic mainly focused on the identification of the disease and, possibly, on the detection of its severity. Actually, in order to clinically analyze disease evolution, long periods of observations, even decades, are needed. Obviously, a tool supporting physicians in accelerating the analysis process, e.g. via simulations, would be of great benefit. Notably, it has been demonstrated by several independent studies that there is a strong correlation between the variations of the structure of the connections among neurons (also called *connectome*) with possible insurgence of several neurological disorders [Bargmann and Marder (2013)]. Hence, it would be of high interest to simulate the course of the disease by simulating brain connections degradation, in order to understand which kind of modifications might mostly determine an evolution of the disease into a worst state, or which recovery processes might induce a remission state. Unfortunately, this simulation represents a non-trivial challenge due to various reasons, including the fact that the actual mechanisms guiding the evolution of the pathology are still largely unknown.

As we have seen in previous chapters, the brain structure can be fruitfully represented by means of a graph; hence, a possible solution would be to simulate the progress of the pathology by means of a set of custom-defined rules for modelling the evolution of the brain structure, which may involve a certain background knowledge. An effective tool for the experts could consist of a comprehensive environment which allows to dynamically detect minimal alterations of brain connections, based on specific guidelines, that induce a change of state in the disease. The possible change of state can, in turn, be detected by neural networks, exploring latent relations learnt from samples. It is worth pointing out that, to the best of our knowledge, a tool providing these features is not available yet; moreover, simulating *manually* brain alterations is not an easy task and it is almost impossible to manually detect significant brain substructures.

In this context, ASP can play a relevant role. In fact, simulation tasks could be in principle designed with any programming paradigm; nevertheless, since it is still not clear which changes to which graph properties are likely to impact on the evolution of the disease, a try-and-check methodology is necessary. This implies to write ad-hoc simulation machineries for each of them. Clearly, a declarative methodology allows for compact and clear definitions as well as fast prototyping: ASP paves the way to easy definition of rules for the identification of brain substructures that can be of interest for the analyst.

Interestingly, it has been shown [Kocevar *et al.* (2016)] that it is important to look at minimal graph changes allowing to reach a certain goal in some graph variations, such as density or assortativity; this implies that most of the tasks to be carried out would actually involve optimization. The use of ASP for optimization problems is still relatively less popular than its use for decision problems, even though it has been proved to be perfectly suited for them; moreover, the formulas for the computation of graph metrics can be in general not easy to be written by rules only. This increases the interest in the application of recent extensions of ASP systems, such as I-DLV/DLV2 [Calimeri *et al.* (2017), Adrian *et al.* (2018)], DLVHEX [Eiter *et al.* (2016), Calimeri *et al.* (2016), Eiter *et al.* (2018)a], etc., which allow both to solve optimization problems and the

integration of external computation sources within the ASP program. The potentialities of ASP in these two areas are pointed out in the rest of the paper.

The main contribution illustrated in this chapter consists of a framework for studying the evolution of neurological disorders by simulating variations in the connectome. The framework relies on two main modules: an ANN that classifies a given connectome with respect to disorder stages, and a logic program (specified via ASP) that allows to perform non-deterministic variations on a given connectome that guarantee fixing some graph parameters under study. The framework iterates between the ANN and the ASP modules to simulate possible evolutions of a neurological disorder.

The remainder of the chapter is structured as follows. In Section 2 we discuss related work, and in Section 2.1 we recall some preliminary notions both from the biomedical and ASP contexts. In Section 3 we introduce the general framework and discuss the role ASP and ANN play therein, while Section 4 presents three actualizations of the framework. Section 5 describes the experiments we carried out to assess potentiality and applicability of the approach. Section 6 introduces the web tool we developed based on the presented work, whereas Section 7 surveys some other applications possibly benefiting of the presented framework. Eventually, in Section 8, we draw our conclusions and outline some future work.

2 Related Works

In this section we outline related literature. Since, to the best of our knowledge, there is no proposal available, to date, that simulates the evolution of neurological disorders by a combined use of ASP and ANN, we concentrate our attention on works exploiting together, to different extents, declarative formalisms and ML solutions.

Some works have been carried out to integrate data-driven solutions into declarative systems with the aim of increasing performance; for instance, such solutions are used for inductively choosing configurations, algorithms selection, and proper coupling of subsystems [Gebser *et al.* (2011)a, Maratea *et al.* (2014), Fuscà *et al.* (2017)]. In some proposals, see for instance SMT [Cok *et al.* (2015), Barrett *et al.* (2016), Barrett and Tinelli (2018), Barrett *et al.* (2013)] or CASP [Baselice *et al.* (2005), Mellarkod *et al.* (2008), Balduccini and Lierler (2017), Lierler and Susman (2017), Shen and Lierler (2018), Arias *et al.* (2018)], the logic solver can select statements that should be checked by external theory/numerical solvers, so that the next steps carried out by the logic solver depend on the answers produced by the external ones. Some works mix statistical analysis and ASP [Gelfond (2010), Nickles and Mileo (2014), Beck *et al.* (2015)]: here, the aim is to extend logic programs with probabilistic reasoning, either by a direct integration or by embedding external predicates. Other approaches are related to the use of methods that "guide" the reasoning, the generation of logic programs or other optimizations [Law *et al.* (2015), Law *et al.* (2016), Chabierski *et al.* (2017), Dodaro and Ricca (2018)]; most of them are still at a preliminary stage.

In the context of ASP, some proposals allow ASP systems to access external sources of compu-

tation and even value invention [Calimeri *et al.* (2007), Redl (2016), Kaminski *et al.* (2017), Calimeri *et al.* (2017), Eiter *et al.* (2018)b, Alviano *et al.* (2019), Gebser *et al.* (2019)a], making them to impact on semantics computation to different extents. In particular, thanks to extended built-in constructs, it is possible to invoke external functions and define custom constraints; via such invocations, one might in principle place a call even to an ANN from an ASP program.

In the last decades, ANNs have become one of the most powerful machine learning tools for solving complex problems. Especially in visual domains, they achieved impressive results in object detection, recognition and classification [Goodfellow *et al.* (2016)]. An ANN is typically represented as a composition of functions, each computing a nonlinear weighted combination of its input, which constitute the neural structure, and are organized in layers. Starting from samples, the learning algorithm iteratively refines the network parameters θ in order to approximate a target function f. For example, in the particular case of classification, the ANN learns how to approximate a function $y = f(x, \theta)$ which maps an input x to a category whose label is y. However, the approximation given by the ANN does not provide any insight on the form of f, meaning that there is no interpretable connection between the parameters and the target function. This is one of the main causes for the interpretation of ANN being an open problem.

Several attempts have been made for interpreting the behavior of Neural Networks, also using declarative approaches [Zhang and Zhu (2018)], and to incorporate symbolic knowledge into neural networks, resulting in a class of networks known as knowledge-based neural networks [Towell and Shavlik (1993)]. In the context of network interpretation, Zhang *et al.* used explanatory graphs and decision trees to create interpretable rules describing convolutional neural network (CNN) features [Zhang *et al.* (2018)b, Zhang *et al.* (2018)a]. Furthermore, based on a semantic And-Or representation, Zhang *et al.* also proposed a method to use active question-answering to assign a semantic meaning to neural patterns in convolutional layers of a pre-trained CNN and built a model for hierarchical object understanding [Zhang *et al.* (2017)b]. Hu *et al.* proposed a framework using logic rules to obtain more meaningful network representations by constructing an iterative distillation method that transfers the structured information of logic rules into the weights of neural networks [Hu *et al.* (2016)].

2.1 Background on Answer Set Programming

The term "Answer Set Programming" was introduced by Vladimir Lifschitz to denote a declarative programming methodology [Lifschitz (1999)]; concerning terminology, ASP is sometimes used in a broader sense, referring to any declarative formalism which represents solutions as sets. However, the more frequent understanding is the one adopted in this article, which dates back to [Gelfond and Lifschitz (1991)]. For a more thorough introductory material on ASP, we refer the reader to [Baral (2003), Gelfond and Leone (2002), Lifschitz (1999), Marek and Truszczyński (1999)]; in the following, we briefly recall syntax and semantics of the formalism.

The language of ASP is based on rules, allowing (in general) for both disjunction in rule heads and nonmonotonic negation in the body. It is worth recalling that a significant amount of work has been carried out by the scientific community for extending the basic language, in order to increase

the expressive power and improve usability of the formalism. This has led to a variety of ASP "dialects", supported by a corresponding variety of ASP systems[1] that only share a portion of the basic language; notably, the community relatively recently agreed on the definition of a standard input language for ASP systems, namely ASP-Core-2 [Calimeri *et al.* (2012)a], which is also the official language of the ASP Competition series [Calimeri *et al.* (2012)b];it features most of the advanced constructs and mechanisms with a well-defined semantics that have been introduced and implemented in the latest years.

For the sake of simplicity, we next focus on the basic aspects of the language; for a complete reference to the ASP-Core-2 standard, and further details about advanced ASP features, we refer the reader to [Calimeri *et al.* (2012)a] and the vast literature.

A variable or a constant is a *term*. Variables are denoted by strings starting with some uppercase letter, while constants can either be integers, strings starting with some lowercase letter or quoted strings. An *atom* is $a(t_1,\ldots,t_n)$, where a is a *predicate* of arity n and $t_1,\ldots,t_n$ are terms. A *literal* is either a *positive literal* p or a *negative literal* `not` p, where p is an atom. A *disjunctive rule* (or simply *rule*, for short) r is a formula of the form: $a_1 \mid \cdots \mid a_n$:– $b_1,\cdots,b_k$, `not` b_{k+1}, $\cdots$, `not` b_m., where $a_1,\cdots,a_n,b_1,\cdots,b_m$ are atoms and $n \geq 0, m \geq k \geq 0$. The disjunction $a_1 \mid \cdots \mid a_n$ is the *head* of r, while the conjunction $b_1,...,b_k$, `not` $b_{k+1},...,$ `not` b_m is the *body* of r. A rule without head literals (i.e. $n = 0$) is usually referred to as an *integrity constraint*. If the body is empty (i.e. $k = m = 0$), it is called a *fact*. $H(r)$ denotes the set $\{a_1,...,a_n\}$ of the head atoms, and $B(r)$ the set $\{b_1,...,b_k,$`not` $b_{k+1},\ldots,$`not` $b_m\}$ of the body literals. $B^+(r)$ (resp., $B^-(r)$) denotes the set of atoms occurring positively (resp., negatively) in $B(r)$. A rule r is *safe* if each variable appearing in r appears also in some positive body literal of r.

An *ASP program* $\mathscr{P}$ is a finite set of safe rules. An atom, a literal, a rule, or a program is *ground* if no variables appear in it. According to the database terminology, a predicate occurring only in *facts* is referred to as an *EDB* predicate, all others as *IDB* predicates; the set of facts of $\mathscr{P}$ is denoted by $EDB(\mathscr{P})$.

The *Herbrand Universe* and the *Herbrand Base* of $\mathscr{P}$ are defined in the standard way and denoted by $U_{\mathscr{P}}$ and $B_{\mathscr{P}}$, respectively. Given a rule r occurring in $\mathscr{P}$, a *ground instance* of r is a rule obtained from r by replacing every variable X in r by $\sigma(X)$, where σ is a substitution mapping the variables occurring in r to constants in $U_{\mathscr{P}}$; $ground(\mathscr{P})$ denotes the set of all the ground instances of the rules occurring in $\mathscr{P}$.

An *interpretation* of $\mathscr{P}$ is a set of ground atoms, that is, an interpretation is a subset I of $B_{\mathscr{P}}$. A ground positive literal A is *true* (resp., *false*) w.r.t. I if $A \in I$ (resp., $A \notin I$). A ground negative literal `not` A is *true* w.r.t. I if A is false w.r.t. I; otherwise `not` A is false w.r.t. I. Let r be a ground rule in $ground(\mathscr{P})$. The head of r is *true* w.r.t. I if $H(r) \cap I \neq \emptyset$. The body of r is *true* w.r.t. I if all body literals of r are true w.r.t. I (i.e., $B^+(r) \subseteq I$ and $B^-(r) \cap I = \emptyset$) and is *false* w.r.t. I otherwise. The rule r is *satisfied* (or *true*) w.r.t. I if its head is true w.r.t. I or its body is false w.r.t. I. A *model* of $\mathscr{P}$ is an interpretation M of $\mathscr{P}$ such that every rule $r \in ground(\mathscr{P})$ is true w.r.t. M. A model

[1] During the years, the scientific community has been very active, and many ASP systems have been released relying on different algorithms and solving technologies; we refer the reader to the latest available report on the ASP competition series [Gebser *et al.* (2019)b], the therein reported references and the vast literature.

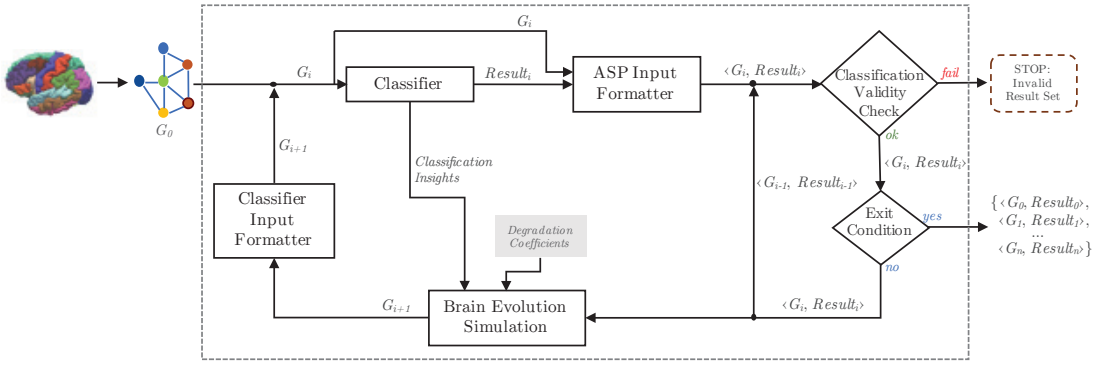


Figure 4.1: Architecture of the proposed framework

M of $\mathscr{P}$ is *minimal* if no model N of $\mathscr{P}$ exists such that N is a proper subset of M. The set of all minimal models of $\mathscr{P}$ is denoted by $\mathrm{MM}(\mathscr{P})$.

Given a ground program $\mathscr{P}$ and an interpretation I, the *reduct* of $\mathscr{P}$ w.r.t. I is the subset $\mathscr{P}^I$ of $\mathscr{P}$, which is obtained from $\mathscr{P}$ by deleting rules in which a body literal is false w.r.t. I. Note that the above definition of reduct, proposed in [Faber *et al.* (2004)], simplifies the original definition of Gelfond-Lifschitz (GL) transform [Gelfond and Lifschitz (1991)], but is fully equivalent to the GL transform for the definition of answer sets [Faber *et al.* (2004)]. Let I be an interpretation of a program $\mathscr{P}$. I is an *answer set* (or stable model) of $\mathscr{P}$ if $I \in \mathrm{MM}(\mathscr{P}^I)$ (i.e., I is a minimal model of the program $\mathscr{P}^I$) [Przymusinski (1991), Gelfond and Lifschitz (1991)]. The set of all answer sets of $\mathscr{P}$ is denoted by $ANS(\mathscr{P})$.

Example 1. *In order to appreciate declarativity and expressiveness of ASP, let us consider the well-known NP-complete* 3*-Coloring problem. Given a graph, we must decide whether there exists an assignment of one out of three colors (red, green, or blue, for instance) to each node such that adjacent nodes always have different colors. If we suppose that the graph is represented by a set of facts F consisting of instances of the unary predicate $node(X)$ and of the binary predicate $arc(X,Y)$, then the following ASP program (in combination with F) describes all* 3*-colorings (as answer sets) of that graph.*

r_1: `color(X,red) | color(X,green) | color(X,blue) :- node(X).`

r_2: `:- color(X1, C), color(X2, C), arc(X1, X2).`

Rule r_1 expresses that each node must either be colored red, green, or blue[2]*; due to minimality of answer sets, a node cannot be assigned more than one color. The subsequent integrity constraint checks that no pair of adjacent nodes (connected by an arc) is assigned the same color.*

Thus, there is a one-to-one correspondence between the solutions of the 3-Coloring problem and the answer sets of $F \cup \{r_1, r_2\}$: the graph is 3-colorable if and only if $F \cup \{r_1, r_2\}$ has some answer set.

3 Framework and Methodology

In this section we present a framework to support the analysis of neurological disorders evolution. It is worth noting that the framework is intended to be rather general, and can be adapted to several kind of disorders and, as it will be shown in Section 7, it can be even extrapolated to other scenarios. Here, we focus on MS disorder. We first introduce the general workflow of the framework, and then illustrate its components in abstract terms. In Section 4 we consider different use cases, and provide some specializations of the various modules.

The general workflow is presented in Figure 4.1. Intuitively, it takes a brain representation as input, classifies the current stage of the disease, and then simulates the effects of the disease course by "damaging the brain"; the newly obtained brain representation is then classified, and used for the next steps. These steps are iterated until some condition holds or until a misclassification is detected; in the first case, the overall set of results is provided as output; in the latter case, the execution of the framework is aborted.

More in detail, the framework takes as input a graph representation of the brain of a patient, which can be obtained starting from a combination of MRI acquisition methods. This is expected to be a weighted graph $G_0 = (V, E_0, \omega_0)$ representing the brain connectome of the patient; this graph is then processed by the *Classifier* module.

The *Classifier* module can be formally modelled as a function $\chi : \mathscr{G} \to \mathscr{R}^4$. In particular, it takes as input the graph G_i of the i-th iteration of the framework, which represents the possibly modified brain connectome of a patient, and outputs four real values, each indicating the probability of the input graph of expressing a specific MS stage, namely $Result_i = (P_{CIS_i}, P_{RR_i}, P_{PP_i}, P_{SP_i})$. Here, P_{CIS_i} (resp., P_{RR_i}, P_{PP_i}, P_{SP_i}) corresponds to the probability of G_i of representing a CIS (resp., a RR, a PP, a SP) MS stage (see Section 2.1). The stage associated with G_i is implicitly the one showing the highest probability.

As previously pointed out, the classification task can be carried out in several ways. One possibility is to make use of results from graph theory [Rubinov and Sporns (2010), Kocevar *et al.* (2016), Shovon *et al.* (2017)]; however, despite network analysis applied on brain connectivity represents a powerful tool, it is not possible yet to define precise biomarkers to classify subjects, especially in the MS context. As far as current classification methods for MS are concerned, there is a clear distinction between approaches that are based on ANN and those that use different learning methods, such as Support Vector Machines (SVM). ANN demonstrated to be one of the most promising tools for the analysis and classification of images, and has been used in a wide range of applications even if, to date, image analysis via ANN in the context of MS has been exploited mostly for the identification of MS lesions rather than MS profile classification. As usually done in similar contexts, in the approaches studying MS via ANN the ground truth for training purposes is obtained by annotations on the data directly provided by human experts. The ANN training phase is a pre-processing step propaedeutic to the application of the framework.

Depending on the actual implementation of the classifier, this module may also provide some insights on the classification process that could be of use in the subsequent steps. This case will

[2]The same piece of knowledge can be equivalently expressed by means of choice rules, see [Calimeri *et al.* (2012)a].

be detailed in Section 4. Intuitively, the specific ANN we adopt in the specialization of our framework allows to leverage the topology of the graph for the classification process and to compute a meaningful coefficient for each graph edge, in the form (x, y, imp_{xy}), representing the importance of the edge (x, y) in the classification task. These insights can be exploited in the simulation of brain evolution and are expressed in Figure 4.1 by the direct connection between the *Classifier* module and the *Brain Evolution Simulation* module.

The output of the classification is verified by a *Classification Validity Checker*. One of the recently arising research issues in classification tasks is the automatic check of validity of results. Indeed, verification of ANN results is receiving increasing interest, and it is currently a very active research area [Pulina and Tacchella (2010),Kouvaros and Lomuscio (2018),Leofante *et al.* (2018)]. However, given the nature of the problem we are addressing in this chapter, we can simplify this task by resorting to rules and constraints that model domain knowledge.

In particular, if certain clinical evidences are available for a patient, they can be used to limit classification alternatives. As an example, it is well known that a brain model obtained by a severe disruption of its previous structure cannot induce a remission of the pathology. In other words, given a brain structure G_{i-1} classified, e.g., as RR, and given its modified version G_i obtained by a strong reduction of arcs and/or weights, if G_i is classified as CIS, i.e., a remission is hypothesized from G_{i-1} to G_i, the classification is evidently wrong and must be discarded. This kind of checks can be easily carried out through suitable sets of rules and constraints; these will be presented in detail in Section 4. As a side note, if the starting input graph G_0 is annotated with the ground truth provided by an expert on the right classification of the initial MS stage, this step can stop the process at the very first iteration if the classification result disagrees with the ground truth. In order to provide the input to the checker in the proper formalism, an *ASP Input Formatter* module stands between the classifier and the checker. It is in charge of translating the output provided by the classifier and the current graph into ASP facts.

The *Classification Validity Checker* can be modelled in abstract terms as a function $\nu : \mathscr{G} \times \mathscr{R}^4 \times \mathscr{G} \times \mathscr{R}^4 \rightarrow \{\text{“OK”},\text{“FAIL”}\}$ which takes as input two graphs, G_{i-1} and G_i and the corresponding classification results $Result_i = (P_{CIS_i}, P_{RR_i}, P_{PP_i}, P_{SP_i})$ and $Result_{i-1} = (P_{CIS_{i-1}}, P_{RR_{i-1}}, P_{PP_{i-1}}, P_{SP_{i-1}})$; it provides as output one among the two possible values “OK” or “FAIL”, depending on the outcome of the check. If the Classification Validity Checker returns “FAIL”, the iterative process is immediately stopped and current and previous results are invalidated. On the contrary, if the checker returns “OK”, the execution of the framework proceeds to the next steps.

In particular, a generic *Exit Condition* is subsequently checked in order to verify whether it is necessary to proceed with the next iteration of the framework or not. The definition of such condition strongly depends on the objective of the analysis. As an example, it could be interesting to check if a certain target probability is reached for a certain MS stage, or if a certain degree of disruption of the original graph has been induced in the last step, or simply if the required number of iterations has been carried out. If the exit condition is verified, the execution is stopped and the set $\{\langle G_0, Result_0\rangle, \langle G_1, Result_1\rangle, \ldots \langle G_n, Result_n\rangle\}$ of graphs and corresponding classification results are provided as output. Otherwise, the execution proceeds with the next brain evolution

simulation step.

The *Brain Evolution Simulation* module can be formally modelled as a function $\mu : \mathscr{G} \times 2^E \times 2^E \rightarrow \mathscr{G}$; here, $\mathscr{G}$ is the set of all possible graphs, whereas 2^E represents all possible sets of triples of the form (x, y, c_{xy}), where x and y are nodes of the graph and c_{xy} is a label. In particular, μ takes as input the current graph G_i, and two sets of triples (x, y, imp_{xy}), and (x, y, dc_{xy}) which convey, respectively, information on the importance of each edge for the classification task, whenever provided by the classifier, and information on how to modify edge weights during the simulation. We formally introduce and specialize these sets is Section 4. μ provides as output a new graph G_{i+1} which represents a simulated evolution of brain structure.

Some metrics over graphs representing brain structures have been considered in previous studies on MS; however, it is still unclear how these metrics influence the progress of the disease. In our framework, the *Brain Evolution Simulation* module is used to explore a wide variety of graph modification criteria. In this context, ASP plays a very relevant role as a fast and effective tool for the definition and identification of subgraphs satisfying some predefined property that could be involved in MS course; in some cases, the identification of such subgraphs may involve the solution of optimization problems, whose coding can be significantly time-consuming in other programming paradigms. In our framework, the *Brain Evolution Simulation* module is composed of an ASP program of choice that, given the graph G_i, first defines a connectome modification criterion described by a set of edges to modify and then produces the new graph G_{i+1}. The corresponding ASP program(s) enjoy the nice properties of such a declarative formalism, resulting very flexible and easy to adapt to small changes in the desiderata.

Each ASP program is coupled with an extensional knowledge base consisting of a set of facts representing nodes and edges of G_i, and identifies a set of atoms which represent the set of edges E' to be modified. Given that, in our context, edge weights are related to the number of fibers linking two points in the brain, an edge (x, y, w) with $w = 0$ is considered inactive and not contributing to the network, i.e., the corresponding nodes are considered not connected. If available, the choice of E' can also be guided by the information about the importance of each edge for the classification task; this information is expressed by the first set of triples (x, y, imp_{xy}) provided as input along G_i.

As already noted, different ways of altering the brain structure using E' can be devised. A basic altering method could consist in simply removing these arcs; this corresponds to set the weight of each edge in E' to 0. However, edge weights play an important role for the structure itself and, from a biological point of view, the strength of the connections (expressed by edge weights in our model) progressively decreases while the brain degenerates. As a consequence, possible evolution strategies might include progressive variations of selected edge weights, as expressed by the second set of triples (x, y, dc_{xy}) provided as input. All the edges not included in E' are simply copied into G_{i+1}.

Examples of interesting criteria for identifying the set of edges to be modified, and that will be detailed in Section 4, are reported next:

- (i) *Max Clique*: contains the greatest subset of vertices in G such that every two distinct vertices in the clique are connected by an edge;
- (ii) *k-hub*: the set of k nodes having the highest degree;

(*iii*) *Min Vertex Cover*: the smallest set of vertices *MVC* such that each edge of the graph is incident onto at least one vertex of the set;

(*iv*) *Density reduction*: the minimal set of edges that, if removed, allows a reduction of the graph density by a given amount;

(*v*) *Assortativity increase*: the minimal set of edges that, if removed, allows an increase of the graph assortativity by a given amount.

It is important to note that, as already mentioned and as it will be clearer in the following, switching between these properties or slightly modify the criteria in ASP requires just to change a few rules; on the contrary, using a classical imperative programming scheme, it would require to rewrite and adapt source code that can be significantly harder to maintain.

The newly obtained graph G_{i+1} is given as input back to the *Classifier*; the ASP representation of G_{i+1} is translated back into the format required by the *Classifier* by the *Classifier Input Formatter* module.

4 Specializations of the framework

In this section we specialize the framework to three biologically relevant settings, upon which we also carry out some experiments in Section 5. Some of the modules are implemented in the same way for all the three specializations, and are hence described first.

Before the actual description, it is worth pointing out, once again, that the goal of the present work is not to provide clinical validation of some kind of results; rather, we want to show the potential of the herein proposed framework that, thanks to the combined use of ASP and ANN, can help experts in studying the evolution of the disease from different perspectives.

4.1 From MRI to Graphs

To recall, the connectivity matrix $A \in \mathbb{R}^{q \times q}$ $(q = 84)$ is generated for each subject. We recall that A represents the adjacency matrix of the weighted undirected graph $G = (V, E, \omega)$ where V is the set containing the segmented GM brain regions (with $|V| = q$), E is the set of graph edges defined as:

$$E = \{\{i, j\} \mid \omega(i, j) > 0,\ 1 \leq i, j \leq q\}$$

and $\omega : \mathbb{N}^2 \to [0, 1]$ is a function that measures the strength of the connection between a pair of nodes by summing the number of streamlines connecting them and scaling this number in the range $[0, 1]$[3]

4.2 Specialization of the Classifier

The Classification of MS patients in their respective clinical forms is achieved by means of a slightly modified version of BrainNetCNN [Kawahara *et al.* (2017)] we specifically defined for

[3]It is worth observing that, since current ASP systems do not support real numbers, the ASP Input Formatter scales values in the real interval $[0, 1]$ into integer values between 0 and 100.

this work. BrainNetCNN is a convolutional neural network (CNN) framework which operates on brain connectivity. Differently from the spatially local convolutions done by traditional CNN, BrainNetCNN is designed to exploit topological locality of structural brain networks. This result is achieved by using two novel operators: the *Edge2Edge* layer, which performs a convolution over the weights of edges that share nodes together, and the *Edge2Node* layer, which computes, for each node i, a weighted combination of the incoming and outgoing weights of edges connected to i.

The BrainNetCNN architecture parameters originally proposed by Kawahara *et al.* were kept unchanged since they already showed promising results in predicting clinical neurodevelopmental outcomes from brain networks. The model takes as input the adjacency matrix representation of a graph G (the brain connectivity matrix) and outputs a probability value for each MS stage. The architecture is composed of two *Edge2Edge* layers, which process the input using 32 filters, followed by two *Edge2Node* layers with 64 and 256 filters. Then, two *fully connected* layers of size 128 and 30 are applied. Both convolutional and fully connected layers use *Leaky ReLU* ($\alpha = 0.33$) activation function. Finally, a fully connected layer of size 4 (the output layer) with *softmax* activation is used to perform the classification.

4.3 Specialization of the Classification Validity Checker

The Classification Validity Checker is implemented via an ASP encoding. Due to pace reasons, we refrain from discussing in detail this and the following encodings, but we point out that they comply with the ASP-Core-2 standard; the interested reader can refer to Section 2.1 and the vast literature on Answer Set Programming for more details.

For the purpose of this work we implement a very basic checker that can be easily enriched with more domain specific knowledge; this is shown in Figure 4.2. In particular, it takes as input two graphs G_i and G_{i-1} encoded by a set of facts of the form `edge(X,Y,W)` and `edge_1(X,Y,W)`, respectively; it takes as input also a threshold T, used to determine whether a severe disruption occurred and the results obtained by the classifier at steps i and $i-1$, encoded as facts of the form `result(STAGE, P)` and `result_1(STAGE, P)`, where `STAGE` $\in$ {"CIS", "RR","PP", "SP"} and `P` represents the probability computed for the corresponding stage by the classifier.

The encoding first determines whether a severe disruption occurred in the brain model between G_{i-1} and G_i by counting the number of arcs that have been removed from G_{i-1} and comparing it with a threshold. If this happens, it checks whether a known impossible transition has been inferred by the classifier between step $i-1$ and step i. As an example, in presence of a severe disruption, it is well known that a transition from RR (resp., from PP, or SP) to CIS is biologically implausible [Lublin *et al.* (2014)]. If a known impossible transition is detected, the computed answer set contains `check("FAIL")`; otherwise, the answer set contains `check("OK")`.

4.4 Specialization of the Exit Condition

The exit condition can be specialized in several ways. As an example, it can stop iterations as soon as it detects that the classifier predicts a transition from one MS stage to another for the current

```
% input: facts of the form "edge(X,Y,W)" and "edge_1(X,Y,W)" as the input graphs
% input: facts of the form "result(STAGE,P)",
%    e.g., result("CIS", 90). result("RR", 25). result("PP", 15). result("SP", 10).
%        representing the classification results for the current graph
% input: facts of the form "result_1(STAGE,P)",
%        representing the classificatoin results for the previous graph
% input: a fact of the form "Th(T)" indicating the minimum number of removed arcs
%        representing a severe disruption according to domain knowledge

% determine if a severe disruption occurred in the last iteration
severedisruption :- #count{X,Y: edge_1(X,Y,W1), edge(X,Y,W), W1>0, W=0}>T, Th(T).

% check for validity of the classification step
check("FAIL") :- severedisruption, result("CIS",R_CIS), result_1(S,R_1), S!="CIS",
                 #max{ R : result(_,R)} = R_CIS,  #max{ R : result_1(_,R)} = R_1.

check("OK") :- not check("FAIL").
```

Figure 4.2: An ASP encoding for the Classification Validity Checker.

patient, for instance when a patient starts from a CIS stage and, after some modifications to the brain structure, she is classified as RR. This would imply that modifications simulated to the brain structure are sufficient to simulate a transition in the pathology of the patient. Analogously, the exit condition can stop the iterations of the framework whenever the difference between predicted probabilities become very low (i.e., below a certain threshold) from one iteration to the other; this may imply that the last modifications simulated on the brain structure are no more informative.

In our framework, we exploited a simple exit condition which stops the execution after a certain number of iterations.

4.5 Specializations for three different use cases

Specialization for studying structural properties

A first interesting use case for our framework is the study of the impact of graph structural properties, and their modification, in the evolution of MS. The aim is to determine whether there is a latent relation between the presence/absence of particular graph structures in the connectome and the stage of the MS disease. In particular, we are interested in verifying if and how modifications on the connectome of a patient, simulated by modifications on the graph representing it, can modify the classification returned by the ANN. Observe that, as a side effect, understanding these relations could provide at least partial motivations for ANN classifications; this is still an open issue in ANN.

In this use case, the Brain Evolution Simulation module must be specialized to detect the structural property of interest, and in particular the set of edges representing it, and to generate a new connectome by modifying selected edges.

Interesting criteria and returned edges are the following: *(i) Max Clique*, i.e., the greatest subset of vertices in G such that every two distinct vertices in the set are adjacent; in this case, the module modifies the edges E' linking the vertices in the clique. *(ii) Independent Set*, i.e., the greatest subset of vertices in G such that no two vertices in the set are adjacent; in this case the

module modifies the edges E' having exactly one vertex in the independent set. *(iii) Max-degree node*, i.e., the node showing the maximum degree in G; in this case the module modifies the edges connected to it. *(iv) k-hub*, i.e., the set of k nodes having the highest degree; in this case the module modifies the edges connected to the *k-hub*. *(v) Min Vertex Cover*, i.e., the smallest set of vertices *MVC* such that each edge of the graph is incident onto at least one vertex of the set; in this case the module modifies the edges E' such that both vertices of the edge are in *MVC*.

Each version of the Brain Evolution Simulation module is then obtained by a suitable ASP encoding detecting the property and the corresponding edges. As an example, Figure 4.3 reports an ASP encoding for the *Max Clique* problem.

Intuitively, the program "guesses" the nodes that belong to a clique in the graph G_i by means of the choice rule:

```
{clique(X)} :- node(X)
```

and then checks, by means of the strong constraint:

```
:- clique(X), clique(Y), X < Y, not activeEdge(X,Y)
```

that the inclusion of two unconnected nodes in the candidate clique set is forbidden. Cardinality of the clique is maximized using the weak constraint

```
:~ node(X), not clique(X). [1@1,X]
```

that penalizes the exclusion of a node in the candidate clique set.

The set of the edges connecting the nodes within the resulting clique is represented by the extension of predicate `e(X,Y,W)`, which is built according to the rule `e(X,Y,W) :- edge(X,Y,W), clique(X), clique(Y)`. The new modified graph G_{i+1} is built with the last two rules appearing in the encoding. In particular, all the edges that must not be modified are just copied in the new graph G_{i+1} (see the last but one rule). The last rule simulates the progressive disrupting process of the MS disease on the portion of brain connectome identified by the extension of predicate `e(X,Y,W)`; specifically, we designed it in order to act as a degradation function on the weights of selected edges; this simulates a degradation in the strength of the connections. In particular, given the initial graph G_0 a *degradation coefficient* is computed for each edge (x,y,w_{xy}) in G_0 as $d_{xy} = w_{xy} \times p$, where p is a percentage of degradation, set as a parameter for the experimentation. Degradation coefficients are given as input to the program as facts of the form `dc(X,Y,D)` and computed as a preprocessing step before starting framework iterations. Then, each edge `e(X,Y,Wxy)` generates in G_{i+1} an edge `edge1(X,Y,max{Wxy-Dxy,0})` (see the last rule). Here, a weight set to 0 means a deletion of the edge from the resulting graph (it will no longer be an `activeEdge`), and consequently a complete disruption of the corresponding connection; in this case, the subsequent iterations and the corresponding ASP programs will no longer consider this edge as belonging to the graph. In our experiments, we considered both $p = 25\%$ and $p = 50\%$.

The value of p heuristically determines the intensity of degradation applied to the strength of the connections in one iteration and, consequently, the velocity of degradation of the connectome through the iterations of the framework. Thus, the choice of p determines the velocity of the simu-

```
% input: facts of the form "node(X)" and "edge(X,Y,W)" as the input graph
% input: facts of the form "dc(X,Y,D)" as degradation coefficients for all edges
% input: support atom "zero(0)"

% guess the clique
{ clique(X) } :- node(X).

% take into account only active edges
activeEdge(X,Y) :- edge(X,Y,W), W>0.
:- clique(X), clique(Y), X < Y, not activeEdge(X,Y).

% maximize clique cardinality
:~ node(X), not clique(X). [1@1,X]

% edges to be modifed in the new graph
e(X,Y,W) :- edge(X,Y,W), clique(X), clique(Y).

% define the new graph
edge1(X,Y,W) :- edge(X,Y,W), not e(X,Y,W).
edge1(X,Y,NW) :- e(X,Y,W),
                 #max{K : e(X,Y,W), dc(X,Y,D), K=W-D ; 0 : zero(0)} = NW.
```

Figure 4.3: An ASP encoding for the *Max Clique* problem.

lation. In particular, if we assume that $p = 25\%$ and that the same edge is chosen at each iteration[4], it takes four iterations to virtually remove it from the connectome. Analogously, when $p = 50\%$, and the same edge is chosen at each iteration, two iterations of the framework are sufficient to virtually remove it. As a consequence, the choice of p strictly depends on the granularity of degradations one wants to study and on the maximum number of iterations of the framework that one wants to carry out.

All ASP encodings for considered criteria can be found at https://www.mat.unical.it/calimeri/material/mix-lp-nn/.

Specialization for studying graph metrics

In [Kocevar *et al.* (2016)] a relationship between some graph metrics, e.g., Density and Assortativity, and the MS stage of a patient has been clinically demonstrated (see Figure 4.4 for a summary). It is then interesting to evaluate whether the evolution of the disease could be also related to progressive modifications (decrease/increase) of such metrics obtained by modifications in the corresponding graphs. It is worth pointing out that a change in these metrics is not always related to specific substructures and, generally, depends on presence/absence of edges. As a consequence, the setting introduced in the previous section cannot be applied in this new context. Moreover, while searching for a variation in a metric, as an example a decrease in density, it is important to avoid trivial modifications such as the removal of all the edges. It is also worth noting that, for some metrics such as assortativity, the removal of edges may induce either a decrease or an increase in the property. As a consequence, it is crucial to look at minimal graph changes

[4]Observe that this is not obvious, and it strongly depends on both the studied property and the configurations of the other edges.

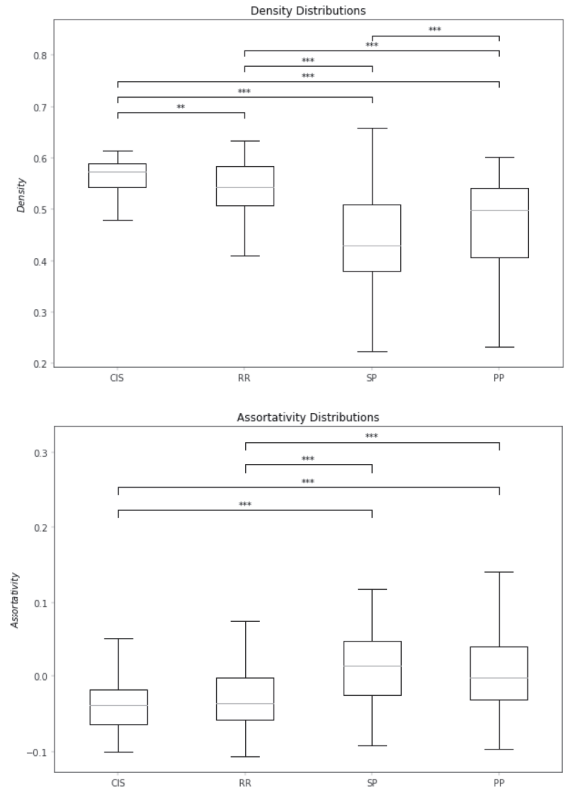


Figure 4.4: Summary of results on the correlation between Density and Assortativity and MS stages.

allowing to reach a certain goal in metric variation. Also in this context, the expressiveness and compactness of ASP and its extensions allow for very elegant and readable encodings even for optimization problems, i.e., the identification of minimal changes involving the computation of complex metrics.

Among the metrics considered in [Kocevar *et al.* (2016)], we focus on Density and Assortativity. In order to keep the paper self contained, we next recall the basic definition of these metrics.

The density d of a graph $G = \langle V, E \rangle$ is defined as:

$$d = \frac{|E|}{|V|(|V|-1)}$$

where $|E|$ is the number of edges in G and $|V|$ represents the number of nodes in G.

Assortativity measures the similarity of connections in the graph with respect to the node degree. In particular, the formula for computing the assortativity degree of a graph G is defined as [Newman (2002)]:

$$r = \frac{\sum_{xy}(xy(e_{xy} - a_x b_y))}{\sigma_a \sigma_b}$$

where x and y are values of node degrees for G, e_{xy} is the fraction of all edges in G joining vertices having degree values x and y, and $a_x = \sum_y e_{xy}$, $b_y = \sum_x e_{xy}$. Moreover, σ_a and σ_b are the standard deviations of the distributions a_x and b_y.

```
% input: facts of the form "node(X)" and "edge(X,Y,W)" as the input graph
% compute the starting value of the metric and the corresponding threshold
target(X) :- &computeMetric[activeEdge](Y), X=Y/90.

% take into account only active edges
activeEdge(X,Y) :- edge(X,Y,W), W>0.

% guess the new graph
0 < { in(X,Y): activeEdge(X,Y) }.

% check that the goal is reached
:- &computeMetric[in](X), target(K), X > K.

% minimize removed edges
:~ activeEdge(X,Y), not in(X,Y). [1@1,X,Y]

% define the new graph
edge1(X,Y,W) :- in(X,Y), edge(X,Y,W).
edge1(X,Y,0) :- edge(X,Y,W), not in(X,Y).
```

Figure 4.5: An ASP encoding for analyzing the decrease of a graph metric.

The formulas above, especially the one for assortativity, show that the computation of graph metrics can be in general not easy to be carried out using only rules in an ASP program. As a consequence, we resort to recent extensions of ASP systems, such as I-DLV/DLV2 [Calimeri *et al.* (2017), Adrian *et al.* (2018)], DLVHEX [Eiter *et al.* (2016), Calimeri *et al.* (2016), Eiter *et al.* (2018)a], etc., which allow the integration of external computation sources within the ASP program. In particular, the problem at hand requires to send a (possibly guessed) entire graph, i.e., a set of edges, to an external source of computation. The ASP standardization group has not released standard language directives yet for such features; here, we make use of syntax and semantics of DLVHEX [Eiter *et al.* (2016)], while a slightly different formulation must be used to comply with I-DLV/DLV2 [Calimeri *et al.* (2017), Adrian *et al.* (2018)] or clingo [Gebser *et al.* (2019)a] syntax.

Figure 4.5 shows the specialization of the Brain Evolution Simulation module for deriving the minimal changes to perform on a graph G_i in order to obtain a decrease in the measure of a certain property (by 10% by edge removal in the example).

In particular, the program of Figure 4.5 first defines the current value of the metric on the input graph; this is done with the support of a call (via "external" atom `&computeMetric`) to an external function written in an imperative programming language, e.g., Python. Then, it defines the target value for the metric. Hence, the program reports the guess for a subgraph that satisfies the goal, where the number of removed edges, expressed by the edges not `in` the subgraph, is minimized by the weak constraint. Finally, the new graph G_{i+1} is generated by the last two rules. Specifically, edges of G_i in the guessed subgraph are copied into G_{i+1}, whereas all the other edges are removed by setting their weight to 0. The changes to be performed on the program for analyzing an increase of the metric are straightforward.

Specialization for exploiting ANN insights

As previously pointed out, the specific ANN presented in Section 4.2, leverages the topology of the graph for the classification process. Furthermore, the particular structure of the network allows to estimate a meaningful coefficient for each graph edge representing the importance of that edge in the classification task. Specifically, in order to define an importance coefficient for each input edge of the brain connectivity, we used the method formerly proposed by Simonyan *et al.* [Simonyan *et al.* (2013)] and already applied to BrainNetCNN by Kawahara *et al.* [Kawahara *et al.* (2017)].

Let $p_c(G)$ be the score assigned to the class c by the trained classification layer of the ANN for a given graph $G = (V, E, \omega)$. Then, the contribution of each edge $e \in E$ can be estimated based on its influence to the score $p_c(G)$. More in detail, the edge importance is computed by the partial derivative $\frac{\delta p_c(G)}{\delta e}$ for each edge $e \in E$ by backpropagation [Simonyan *et al.* (2013)].

Then, given a graph G, let us assume we can derive from the output of the Classifier module a set of facts of the form `imp(X,Y,P)`, where P is the importance in the ANN of the edge (X,Y) given by the formula introduced above. It is strongly interesting to study the impact over MS clinical course of both structural properties over graphs and graph metrics also taking into account these insights from the ANN. Indeed, determining that there are sets of edges that could be ignored without losing important information or, conversely, that it is possible to focus on a small subset of edges, could both significantly help experts in their analyses and reduce computational requirements.

The specialization of the framework taking into account edge importance is quite straightforward. In particular, let us use a threshold T to distinguish between important ($P \geq T$) and not important ($P \leq T$) edges. Consider the ASP program shown in Figure 4.6; this is substantially the same as the one shown in Figure 4.3 for studying cliques, except for the second rule, which forces important edges (if $P \geq T$ is used) or not important edges (if $P \leq T$ is used) to belong to the clique. The threshold T can be dynamically set. Observe that putting not important edges in the clique, means that the graph will be modified on not important parts only (as judged by the ANN). Differently from what we have shown in Figure 4.3, the last two rules state that, in the new graph, the weights of edges identified by the clique are set to 0 and, hence removed.

Analogously, Figure 4.7 shows the specialization for graph metrics by minimizing/maximizing the use of important edges in reaching the goal for graph metrics variation. Again, there is a minimal difference consisting in the introduction of a weak constraint that minimizes the use of important/not important edges.

5 Experiments

This section reports about the experiments we carried out in order to assess the proposed framework and its specializations. The section is organized in three parts. The aim of the first part is to show the flexibility of the framework, and to adapt the analysis to different perspectives. In particular, we focus on analyzing how classification probabilities vary during the brain evolution

```
% input: facts of the form "node(X)" and "edge(X,Y,W)" as the input graph
% input: facts of the form "imp(X,Y,P)" denoting the importance of edge(X,Y,W)
% input: a fact of the form Th(T) used to discriminate between important and
%        not important edges
% guess the clique
{ clique(X) } :- node(X).

% take into account only active edges
% if P <= T not important edges are in the clique,
% if P >= T important edges are in the clique
activeEdge(X,Y) :- edge(X,Y,W), W>0, imp(X,Y,P), Th(T), P <= T.
:- clique(X), clique(Y), X < Y, not activeEdge(X,Y).

% maximize clique cardinality
:~ node(X), not clique(X). [1@1,X]

% edges to be modifed in the new graph
e(X,Y,W) :- edge(X,Y,W), clique(X), clique(Y).

% define the new graph
edge1(X,Y,W) :- edge(X,Y,W), not e(X,Y,W).
edge1(X,Y,0) :- e(X,Y,W).
```

Figure 4.6: An ASP encoding for the *Max Clique* problem maximizing the use of important/not important edges.

simulation, in order to verify appropriateness of the approach in the biomedical context herein discussed. The second part specifically focuses on performance, while the third part is devoted to the discussion of obtained results and to outline some lessons learned thanks to experiments outcome.

5.1 Experiments on the application of the framework to simulate MS evolution

In the following, we first introduce the dataset and the preprocessing steps. Then, we report and discuss obtained results for each specialization introduced in Section 4. It is worth noting that we developed also a web tool for supporting experts in carrying out their analyses online through our framework; such tool will be discussed in Section 6.

Dataset Description and Preprocessing Steps

Structural connectivity matrices were extracted for each subject. A total of 578 samples (distributed into the four aforementioned categories as 63 CIS, 199 RR, 190 SP, 126 PP, respectively) were considered for the experiments overall, and for each sample the corresponding graph G has been extracted as explained in Section 4.1. Ground truth on the correct MS stage of all the samples is available, as it has been provided by expert physicians. Each graph consists of 84 vertices with an average of 2036.31 ± 139.19 edges for the samples in CIS, 1951.25 ± 235.43 in RR, 1634.56 ± 315.27 in SP and 1760.96 ± 293.58 in PP.

The ANN introduced in Section 4.2 has been trained before starting the experiments on the framework, by cross validation with 3 folds, using 70% of the samples in each fold for training

```
% input: facts of the form "node(X)" and "edge(X,Y,W)" as the input graph
% input: facts of the form "imp(X,Y,P)" denoting the importance of edge(X,Y,W)
% input: a fact of the form Th(T) used to discriminate between important and
%        not important edges
% compute the starting value of the metric and the corresponding threshold
target(X):- &computeMetric[activeEdge](Y), X=Y/90.

% take into account only active edges
activeEdge(X,Y):- edge(X,Y,W), W>0.

% guess the new graph
0 < { in(X,Y): activeEdge(X,Y)}.

% check that the goal is reached
:- &computeMetric[in](X), target(K), X > K.
:~ activeEdge(X,Y), not in(X,Y). [1@2,X,Y]

% In the next constraint,
% if P >= T maximizes removal of not important edges,
% if P <= T maximizes removal of important edges
:~ activeEdge(X,Y), not in(X,Y), imp(X,Y,P), Th(T), P >= T. [1@1,X,Y]

% define the new graph
edge1(X,Y,W) :- in(X,Y), edge(X,Y,W).
edge1(X,Y,0) :- edge(X,Y,W), not in(X,Y).
```

Figure 4.7: An ASP encoding for analyzing the decrease of a graph metric maximizing the use of important/not important edges.

the model and the remaining 30% as test set for validation. The quality of the classification was evaluated by means of the average Precision, Recall and F-Measure achieved during the cross validation, as usual in the literature. The proposed ANN was trained using Adam [Kingma and Ba (2014)b] with learning rate 0.001. Early Stopping was used to prevent overfitting. Average evaluation of the cross validation is shown in Table 4.1. It can be observed how the ANN we designed for this work is particularly effective in determining the right stage of the pathology under consideration. This is a crucial factor in the framework, as studying the impact of the variations in the connectome on the course of the disease requires a very high precision in the classification step. Observe that, even if augmenting the classification accuracy is beyond the scope of this work, the use of well-performing networks allows to obtain more reliable results.

Notably, the new classification model adopted in this work allows to reach an average F-Measure of 88%, which represents a significant improvement with respect to the 80% reached in [Calimeri *et al.* (2018)c] for the same quality measure. It is important to point out that results shown in Table 4.1 are slightly lower than previous results obtained in [Calimeri *et al.* (2018)b]; nevertheless, we consider these slightly lower values definitely acceptable, as the new ANN used in the present work allows a meaningful and more interpretable representation of the classification process, and this is particularly useful in the new version of the framework. Indeed, BrainNetCNN leverages the topological locality of structural brain networks, thus performing more meaningful operations on the graph structure with respect to the previous approach [Calimeri *et al.* (2018)b] and allowing to compute edge importance. Furthermore, high-level features learned by Brain-

Table 4.1: Average Precision, Recall and F-Measure (± standard deviation) achieved during cross validation (3 folds). Results are computed per class (CIS, PP, RR, SP) and with respect to all the classes (Tot).

	Precision	Recall	F-Measure
CIS	0.76 (±0.12)	0.88 (±0.13)	0.81 (±0.10)
PP	0.91 (±0.04)	0.69 (±0.17)	0.78 (±0.10)
RR	0.89 (±0.03)	0.94 (±0.06)	0.91 (±0.02)
SP	0.90 (±0.06)	0.93 (±0.02)	0.92 (±0.03)
Tot	0.89 (±0.01)	0.88 (±0.01)	0.88 (±0.02)

NetCNN have been already discussed in the literature in the context of the anatomy and function of the developing pre-term infant brain.

After the training phase, in order to keep experiments on the overall framework coherent, before starting the tests on the framework we filtered out some input samples, relying on the ground truth provided by physicians. In particular, we filtered out input samples misclassified by the trained ANN at the very first classification; this way, we avoid to propagate initial classification errors in the framework. As a consequence, a total of 55 CIS, 189 RR, 187 SP and 109 PP correctly classified samples have been actually fed as input to the framework for the tests.

Experiments for studying structural properties

In the context of this analysis, we are interested in studying the possible variations of each stage of the MS clinical course by modifying the connectome of a patient, according to the structural properties introduced in Section 4.5. For the sake of presentation and space constraint, we discuss in detail the results of a subset of the experiments we carried out, namely *Max Clique*, *Min Vertex-Cover*, and *k-hub*, with $p = 50\%$. Nevertheless, the complete set of results is available at https://www.mat.unical.it/calimeri/material/mix-lp-nn/.

Figures 4.8, 4.9, and 4.10 report the overall results for this specialization. For each starting stage of the pathology, we report the probability values (indicated by a group of four vertical bars, for each iteration of the framework), computed by the ANN. In particular, from left to right, one can observe the variation of the average probability values for each class. As an example, the first bar in the leftmost group of the first bar chart in Figure 4.8 represents the probability associated with the CIS stage for a CIS classified patient. The same bar in subsequent groups shows variations of this probability through iterations 1–4 with the ASP program shown in Figure 4.3; the same bar in the chart below shows variations of the same probability for patients formerly classified as RR.

In order to evaluate the significance of obtained results, we considered also a *random test* for each test case, designed as follows: at each iteration, given the number n of edges identified by the *Brain Evolution Simulation* module, we generate a parallel modified graph choosing n *random* edges to be modified that are not related to the structure under consideration. In other words, we are interested in evaluating whether the variations in classification results depend on the structure

of the modified portion of the connectome, or simply on the number of varied edges. Outcomes of random tests are reported on the right side of Figures 4.8, 4.9, and 4.10.

Results show interesting variations, when testing the framework with the *Max Clique* criterion (Figure 4.8). In particular, it is worth noting that *Max Clique* seems to affect mostly the CIS stage, as CIS probability values significantly decrease. Interestingly, this behaviour seems not to be simply related to the amount of modified edges: random tests show substantially constant probability values across iterations. More interestingly, the aforementioned behaviour is not observable for the other stages RR, SP and PP, where the alteration of cliques does not actually induce significant changes. This absence of variations is not related to the absence of cliques to change, or to their different cardinalities; indeed, the number of edges modified in all stages are comparable being on average 258.37 ± 34.30 from iteration 1 to iteration 2, and 130.38 ± 5.34 from iteration 3 to iteration 4. The results for the CIS starting stage also show that probabilities of PP actually increase through iterations, even if not sufficiently enough to allow a guess over a change of state.

As far as *Min Vertex Cover* is concerned (Figure 4.9), significant variations can be observed in all stages. However, we observe in this case also a significant decrease of the probability of election in random tests, especially in the first iterations. This is mainly due to the high number of modified edges, being on average 1490.45 ± 147.12 from iteration 1 to iteration 2, i.e. 41.59% of the total. In the subsequent iterations, very small minimum vertex cover could be identified in the modified graphs due to the low number of remaining edges; as a consequence, very few edges (about 0.003% of the total) are modified and, thus, very small variations on probabilities are detected.

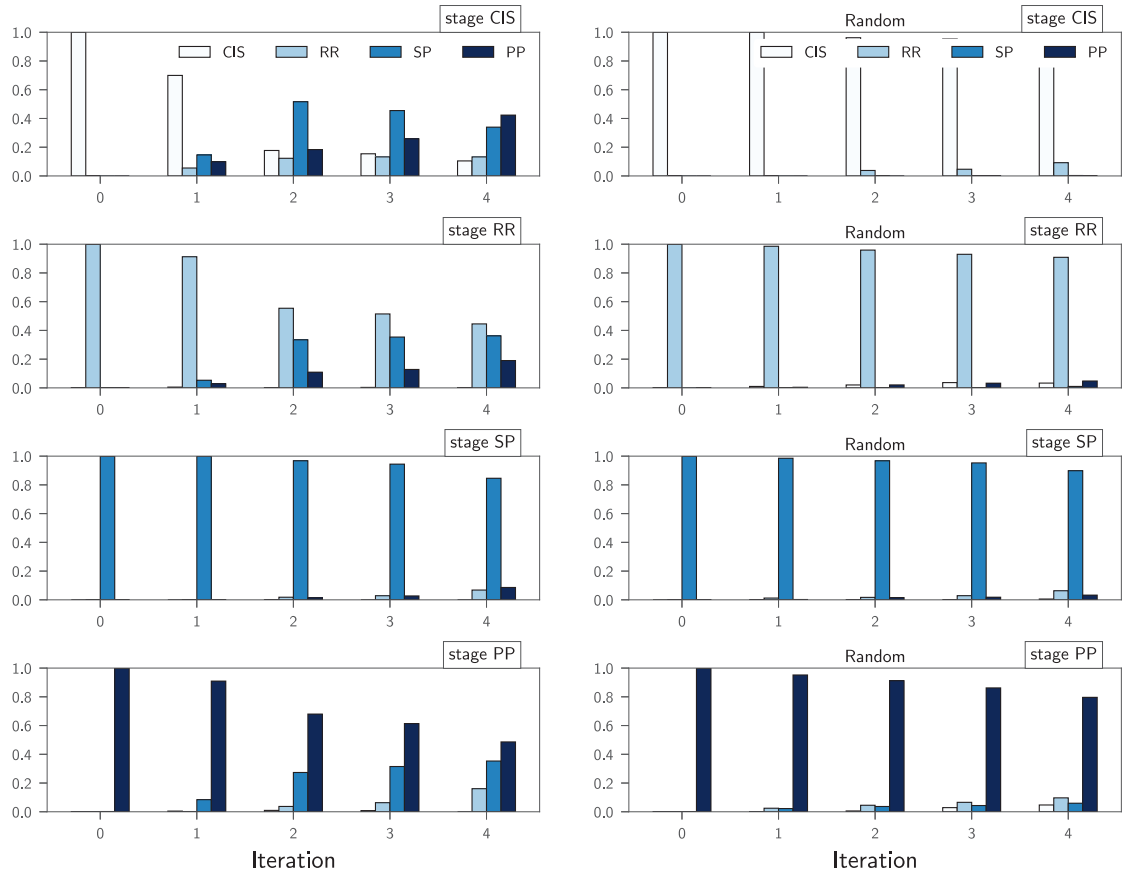
However, if we concentrate on non random tests, we observe a quite different and interesting behaviour. In particular, at iteration 2 the probability of RR is always the highest suggesting that (the absence of) this sub-structure might characterize the RR class, and calls for further studies.

Finally, the *k-hub* sub-structure (see Figure 4.10) can be considered as a counter-example of previous results. In fact, even if the number of modified edges is significant and comparable with *Max Clique*, i.e., $301.47 \pm 22,82$ from iteration 1 to iteration 2, and $202,99 \pm 26,47$ from iteration 3 to iteration 4, probability values across iterations are almost constant and very similar to the random tests. This leads us to hypothesize that *k*-hub sub-structures are not characterizing any stage of the disease.

Experiments for studying graph metrics

In the context of this analysis, we are interested in studying the possible variations of each stage of the MS clinical course by modifying the connectome of a patient, according to the metrics introduced in Section 4.5. In particular, in this section, we discuss the results obtained for *Density* and *Assortativity*. It is worth pointing out that current versions of state-of-the-art ASP systems have not been able to reasonably scale over the graphs involved in the following tests; as a consequence, after proving the viability of the approach over small examples, we simulated the behavior of the ASP-based module via ad-hoc heuristic algorithms.

Figure 4.11 shows the results obtained for *Density*; recall that, at each iteration, we reduce

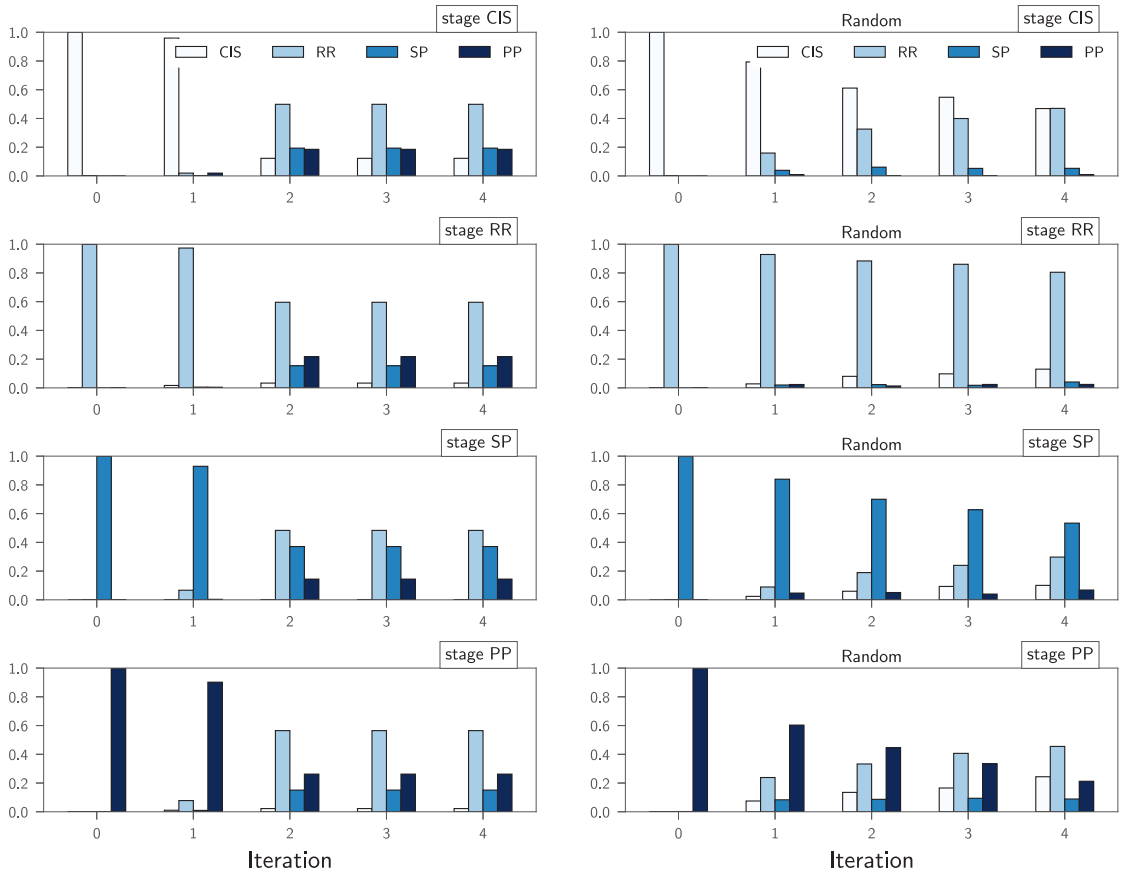


(a) *Clique (normal)* (b) *Clique (random)*

Figure 4.8: Results for *Clique* (iterations $i = 0..4$).

the density of each graph by 10% by removing edges. From the analysis of Figure 4.11(a) it is possible to observe a generalized reduction of the probability of the initial classification through the iterations. Only for the CIS stage, however, this is more remarkable, even if a clear transition cannot be observed from this figure. In order to better analyze this result, we computed (see Figure 4.11(b)) the average probabilities obtained by isolating the cases in which a transition from CIS to PP is observed through the iterations (these are 39.53% of the total) and the cases in which a transition from CIS to RR is observed through the iterations (these are 62.79% of the total). We point out that some of the graphs presented both transitions through the iterations; they have been considered in both cases. The analysis of Figure 4.11(b) shows indeed that there is quite an interesting evidence of these transitions in the last iterations; first of all, these results partially confirm the results obtained from studies presented in [Kocevar *et al.* (2016)], pointing out that the *Density* of a graph characterizes the stage of MS. Moreover, these results provide more insight in the potential *evolution* of the disease. In fact, depending on the area of the brain that is altered, the transition from CIS can evolve in RR stage. This calls for further studies on this aspect.

In order to better analyze these results, we computed the average number of edges removed at each iteration for the two transitions separately (CIS-PP and CIS-RR). Results, reported in Table



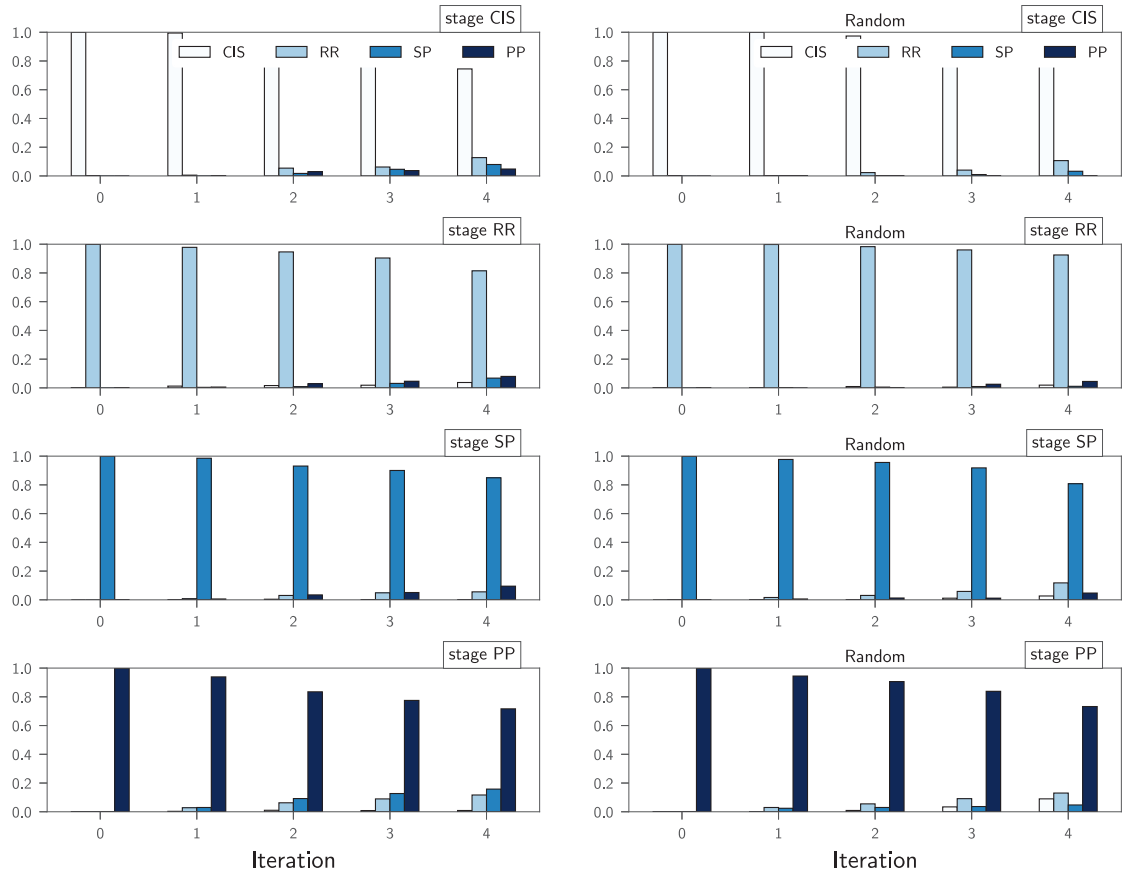
(a) *Min Vertex-Cover (normal)*

(b) *Min Vertex-Cover (random)*

Figure 4.9: Results for *Min Vertex-Cover* (iterations $i = 0..4$).

4.2, show that even a low number of removed edges, if properly selected, may induce a significant change in MS stage.

Results obtained for *Assortativity* are shown in Figure 4.12. Recall that, in this case, we modified the graphs in order to obtain an increase in the property of 10% at each iteration by removing edges. As far as these results are concerned, we observe an almost complete independence of the computed probabilities on this property through the iterations. Observe that the final variation of assortativity at the last iteration is about 40% of its initial value; as a consequence, this result cannot be motivated by a low variation in the property itself. As a matter of fact, the number of edges to be removed in order to reach the variation goal on assortativity was indeed extremely small. As an example, only 27.67 ± 15.94 edges have been removed on average through the four iterations for CIS patients, and only 24.26 ± 16.07 edges for PP patients. This result can be linked to the ones obtained for *Density* where the specificity of removed edges is probably more important than the overall properties of the corresponding graphs. And this calls for a deeper analysis on the role of specific subsets of edges in the classification process, which is precisely what we analyze in the next section.



(a) *k-hub (normal)* (b) *k-hub (random)*

Figure 4.10: Results for *k-hub* (iterations $i = 0..4$).

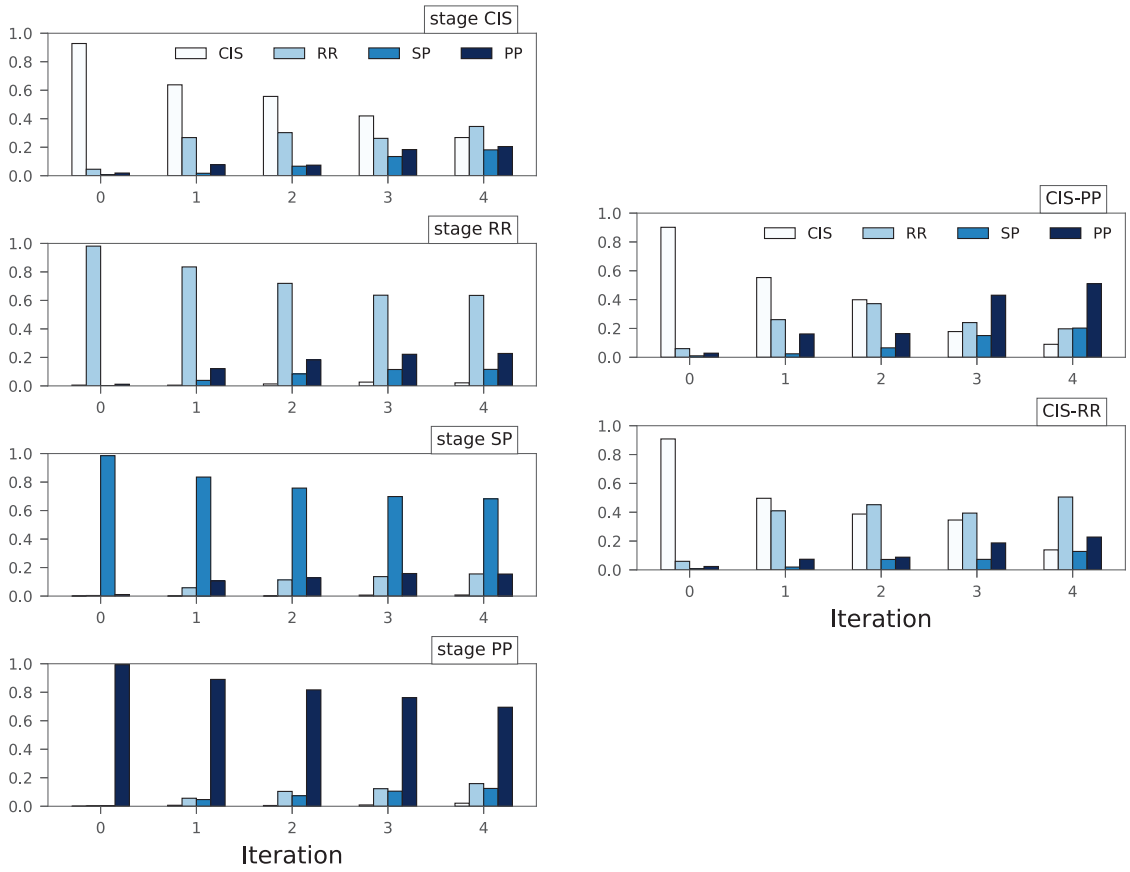
Table 4.2: Average number of removed edges ($\pm$ standard deviation) for reducing *Density* of 10% at each iteration.

Iteration	CIS-PP	CIS-RR
1	190.65 ($\pm$15.82)	199.48 ($\pm$11.15)
2	171.47 ($\pm$14.19)	179.52 ($\pm$9.96)
3	154.47 ($\pm$12.87)	161.67 ($\pm$9.00)
4	138.94 ($\pm$11.48)	145.41 ($\pm$8.14)
Tot	655.53 ($\pm$54.35)	686.07 ($\pm$38.23)

Experiments for studying ANN insights

In Section 4.5 we introduced the concept of importance of an edge for the classification purposes, computed by exploiting the peculiar properties of the adopted ANN. In particular, we introduced a specialization of our framework dealing with edge importance.

Before analyzing our tests coupling graph structures or graph metrics with edge importance, we first consider the impact of the importance degree through the following two simple tests. Assume edges are ordered by importance, in descending order: *(i)* remove, one by one, the edges



(a) *Results for all stages*

(b) *Detail for CIS transitions*

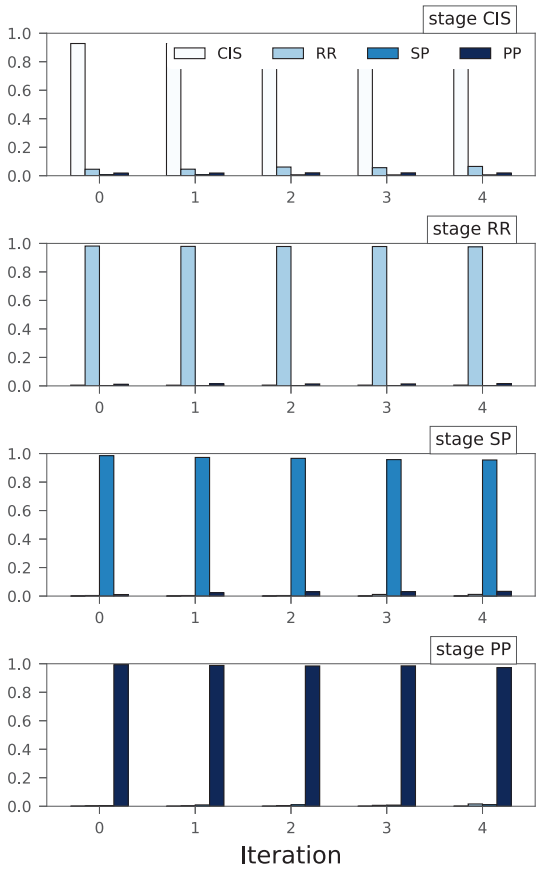
Figure 4.11: Results for *Density* (iterations $i = 0..4$).

starting from the most important ones; *(ii)* remove, one by one, the edges starting from the less important ones. Each time an edge is removed, the classification task is carried out again and results plotted for each stage. Results are shown in Figure 4.13. Consider the results in Figure 4.13(b) first; from the analysis of this figure, it is manifest that removing not important edges is completely irrelevant for the classification result at any stage. And this is true up to about 1500 removed edges. On the contrary, removing even very few important edges may strongly affect classification results (see Figure 4.13(a)).

These considerations call for a deeper analysis of the two settings tested above. In particular, it would be of great interest for an expert both knowing that there are edges he/she may completely disregard in the analysis, and that there are edges that need more attention for stage variation analyses.

In particular, given the specializations for the Brain Evolution Simulation module introduced in Section 4.5, we set a threshold such that the top 40% of edges having the highest importance are considered as important, whereas the remaining edges are considered non important.

Figure 4.14(a) shows results for *Max Clique* when not important edges only are allowed in the cliques. The number of altered edges is comparable to the one obtained for the previous tests on *Max Clique*. Figure 4.14(a) clearly confirms that altering graph sub-structures using not important



(a) *Assortativity*

Figure 4.12: Results for *Assortativity* (iterations $i = 0..4$).

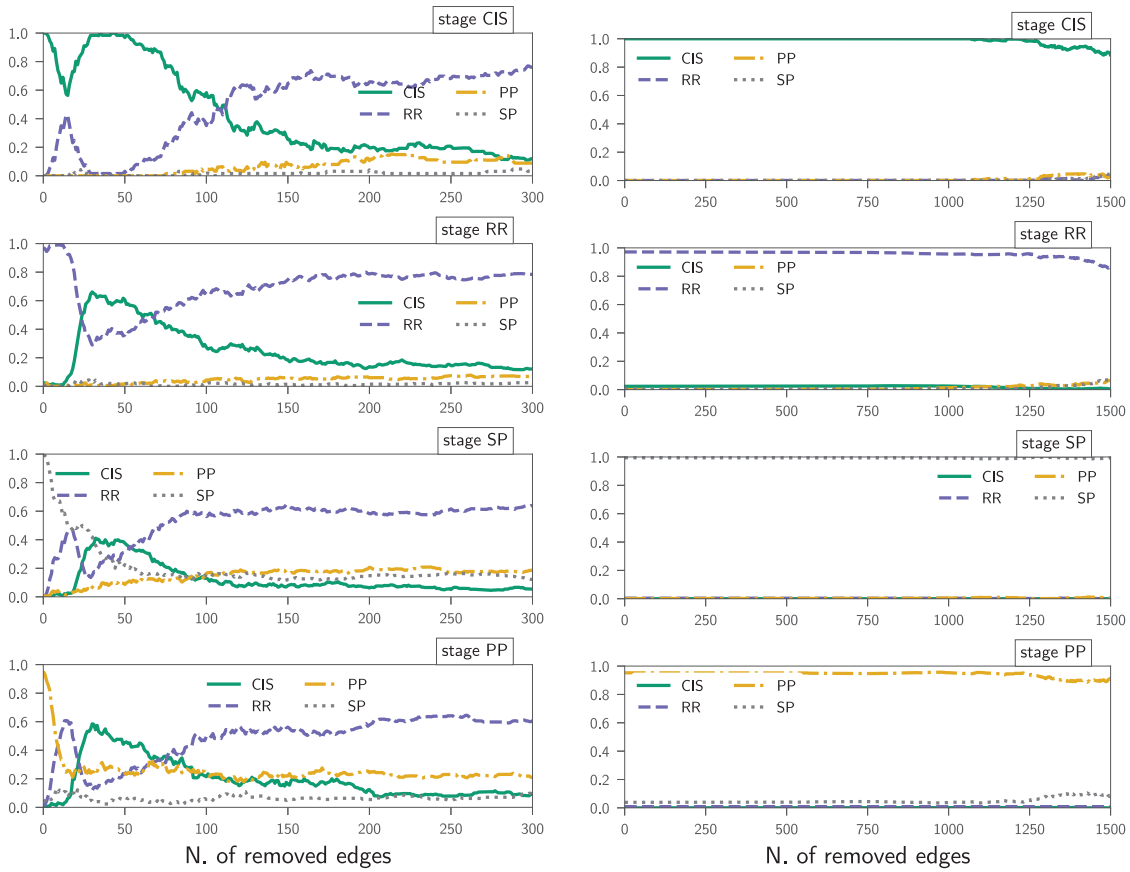
edges only, provides no apparent modifications in the classification. As a consequence, it confirms the fact that experts may completely disregard at least 60% of edges in their analyses.

On the contrary, Figure 4.14(b) reports results on *Density* when only important edges are removed. In this case, results show that when a huge amount of important edges is removed, the ANN becomes almost unable to perform a reliable classification. As a consequence, this reinforces the intuition that there are very few important, and let's say *critical*, edges guiding transitions between MS stages. In Section 6 we show how we took these preliminary results into account in order to provide experts with a powerful analysis tool.

5.2 Experiments on performances

In this section we present the results of a series of tests aimed at providing support for a performance analysis of the system. We first analyze execution times of the entire framework, then we single out the role of the main modules in the overall execution times. Eventually, we focus on the ASP-based *Brain Evolution Simulation* module.

All tests have been carried out on a Linux machine $4.15.0-20$-generic #21-Ubuntu, with an Intel(R) Core(TM) i7-4770 CPU @3.40Ghz and 15.6 GB of RAM. As for the grounder and the



(a) *Starting from the important edges*

(b) *Starting from the less important edges*

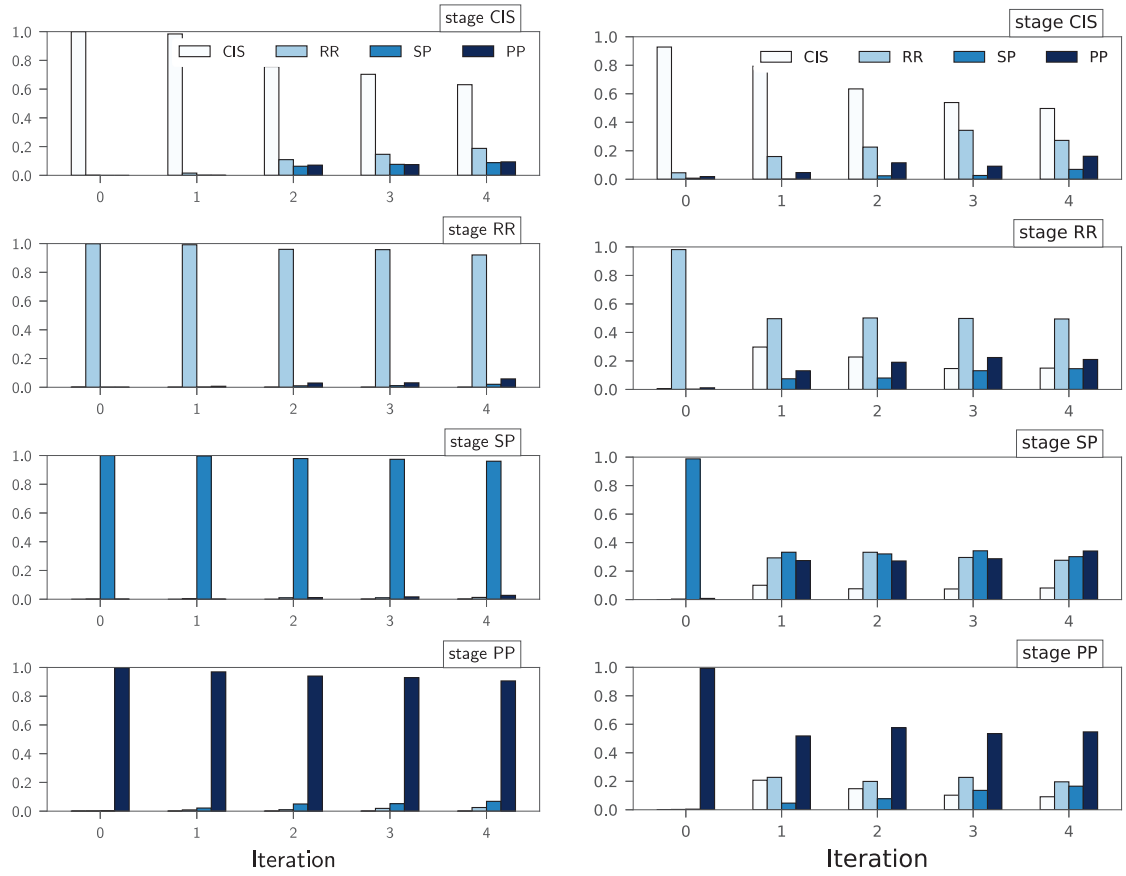
Figure 4.13: Variation of classification results removing important/not important edges.

solver, we coupled I-DLV (version 1.1.0) and WASP (version 2.0). The solver parameters have been set to `-silent` and `-printonlyoptimum`; this means, that the computation is stopped at the first optimum found for optimization problems.

When not differently specified, we used the dataset introduced in Section 5. It is worth pointing out that the aim of these tests is not an assessment of ASP solvers and their performance (as extensively done in related literature [Calimeri *et al.* (2012)b, Gebser *et al.* (2016), Gebser *et al.* (2017), Gebser *et al.* (2019)b]), but rather to assess the applicability of the proposed framework to the herein considered context and related ones.

In a first series of experiments, we measured the execution times of one iteration of the framework, i.e., involving one step in the brain evolution simulation. We separately considered the three structural properties *Max Clique*, *Min Vertex-Cover*, and *k-hub* discussed above and, as for *Max Clique*, we tested also the ASP programs altering both not important and important edges[5]. Results are shown in Figure 4.15. In order to verify whether the starting stage influences performance or not, we highlighted running times for each property and for each stage; times are averaged over

[5]Recall that in Section 5.1 we set a threshold such that the top 40% of edges having the highest relevance are considered as important, whereas the remaining edges are considered as not important. It is worth mentioning again that information about importance of edges is directly provided by the Classification module.



(a) *Clique - Altering not important edges*

(b) *Density - Removing important edges*

Figure 4.14: Analyzing structural properties and graph metrics considering important/not important edges.

all the samples grouped by MS stage and standard deviation is also shown in the figure. From the analysis of Figure 4.15, we can observe that (a) the execution time of one iteration is particularly small for all properties, always significantly lower than one second; (b) there is no actual correlation between starting stages and performance; (c) interestingly, considering important/not important edges reduces average execution times, as the dimension of the graphs the ASP program works on is reduced, in terms of edges.

We then evaluated the potential impact on running times due to subsequent iterations. Figure 4.16 reports execution times averaged over all the stages for four subsequent iterations; it is easy to see that fluctuations of running time among iterations are negligible except for *Min Vertex-Cover* where, as discussed in Section 5.1, at the third iteration a very low number of edges remains in the modified graphs.

In a further series of experiments, we considered the impact of each module of the framework in the running time of one iteration. We take into account the three main modules, namely the *Classifier*, the *Classifier Validity Checker* and the *Brain Evolution Simulation* modules. Results shown in Figure 4.17 clearly point out that the main load of computation is on the *Brain*

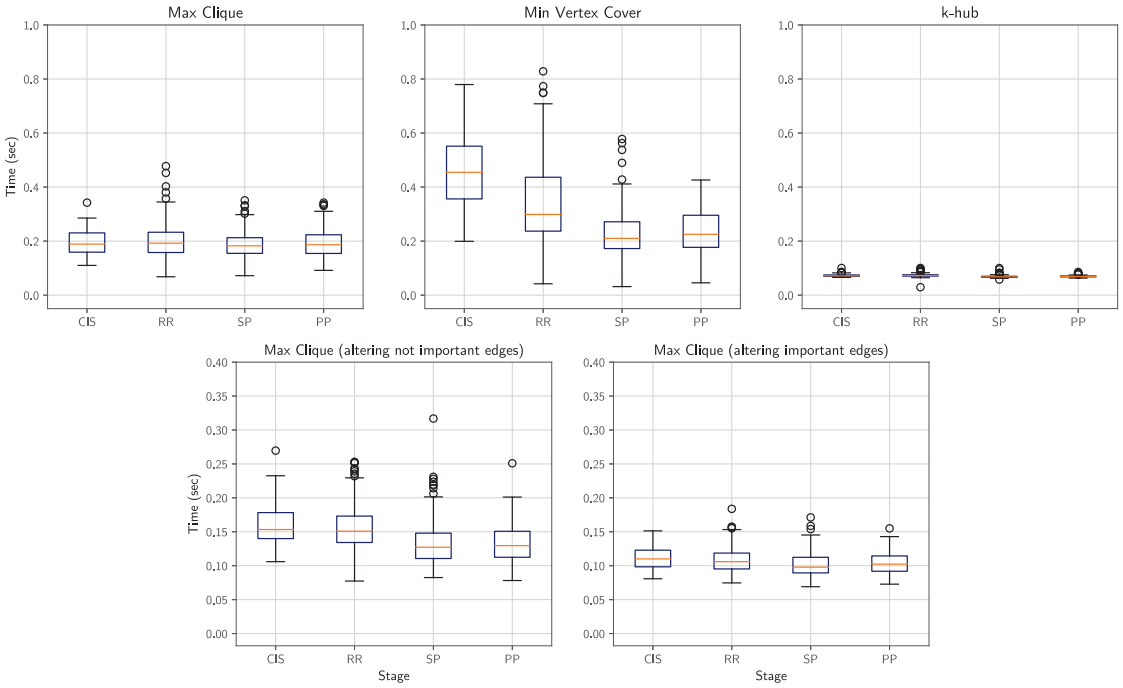


Figure 4.15: Execution times for one iteration of the framework, considering the three structural properties *Max Clique*, *Min Vertex-Cover*, and *k-hub*. Bottom graphs show execution times for one iteration of the framework on *Max Clique* considering either important or not important edges.

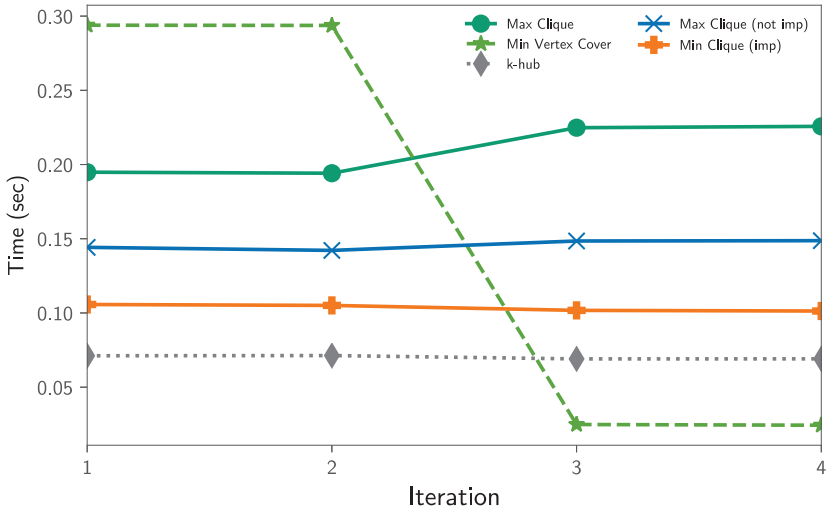


Figure 4.16: Execution times for four iteration of the framework.

Evolution Simulation module. Obviously, both *Classifier* and *Classification Validity Checker* execution times are independent from the graph property under examination; interestingly, they are both significantly faster than the simulation task. Higher execution times for *Max Clique* and *Min Vertex-Cover* with respect to *k-hub* depend on the deterministic nature of the encoding for *k-hub*. The same considerations carried out in the previous tests when including important/not important edges for *Max Clique* are still valid in this test.

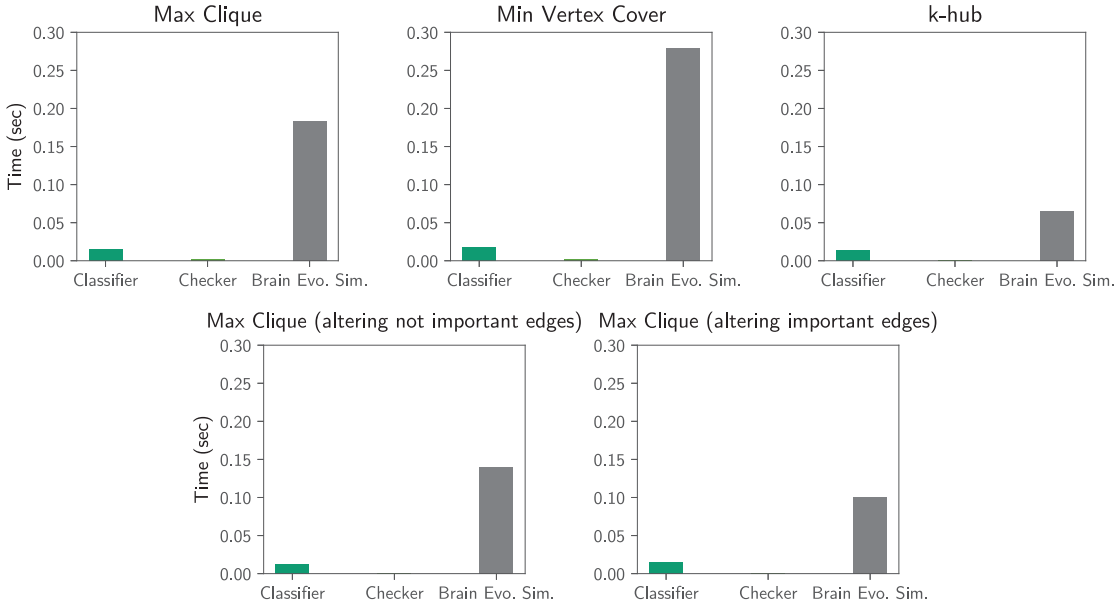


Figure 4.17: Impact of each module of the framework in the running time of one iteration.

The scalability of the ASP part of the system has been tested over graphs of increasing size. First of all, we measured running times of the *Brain Evolution Simulation* module over a set of simulated graphs possibly representing connectome; in particular, we fixed the number of nodes (84 in our tests, coherently with the technique described in Section 4.1) and we randomly generated graphs with increasing number of edges up to a complete graph. It is worth recalling that, as pointed out in Section 5, the average number of edges in graphs corresponding to real connectome is around 2000. Results are reported in Figure 4.18; each data point is the average running time of 10 different executions over random graphs having the same number of nodes and edges. Via this figure, it is possible to observe that all the tested ASP programs for studying structural properties are solvable with execution times always below one second on any potential graph representing a connectome. There are obviously small variations between different samples and properties; nonetheless, the figure shows that any connectome can be easily managed by our approach in order to study structural properties.

As a further scalability test, we generated graphs with increasing number of nodes; as for the number of edges in these graphs, we measured the average number of edges in the graphs representing real connectome and we kept the same proportion of edges for each generated graph. Results are shown in Figure 4.19 for graphs up to 700 nodes; again, each data point is the average running time of 10 different executions over random graphs having the same number of nodes and edges. In this case, it is possible to observe that, while the number of nodes is around 100, running times for all the problems are reasonable; when the number of nodes grows further, the combinatorial explosion of programs including non-deterministic choice rules is reflected in rapidly increasing running times (indeed, as an example, *Max Clique* is reported as an hard problem in the ASP Competition series [Gebser *et al.* (2016), Gebser *et al.* (2017), Gebser *et al.* (2019)b]); *Min Vertex-Cover* is affected first by this issue. In fact, we observed that, on the machine used, it may

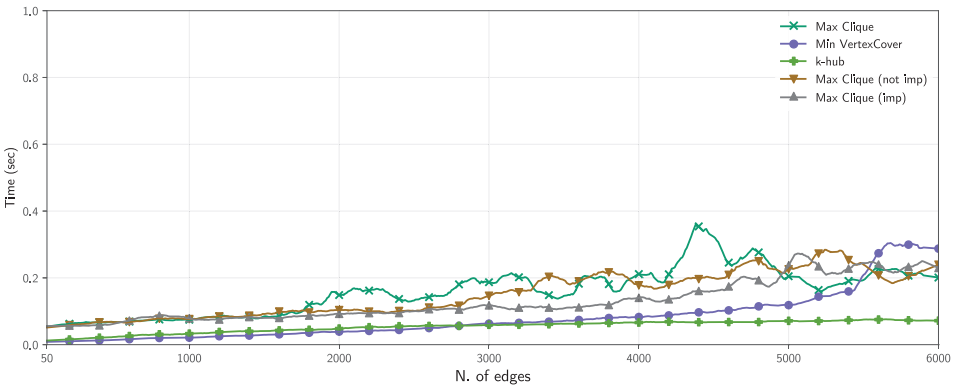


Figure 4.18: Running time of the Brain Evolution Simulation module on simulated graphs with increasing number of edges.

require more than one hour of computation for determining the *Min Vertex-Cover* on a graph with around 150 nodes or finding the *Max Clique* on a graph with around 500 nodes. It is interesting to observe that, again, considering only important/not important edges allows to move forward the limit of computation. Indeed, the lower number of considered edges simplifies the graph and the execution time is faster; as it was expected, considering important edges (40% of the total) allows a further improvement with respect to the not important edges (60% of the total). Notably, *k-hub* scales very well on tested graphs.

These results pose some questions on the applicability of ASP solutions, and generally of exact solutions for optimization problems, in contexts different from the one studied in this work, where the graphs to be handled become very large and non-deterministic reasoning over the graph is needed. In these contexts, heuristic algorithms, not spanning the entire search space, might be more efficient. However, as previously pointed out, compactness, versatility, and declarative nature of ASP allow for a fast prototyping, and make it an excellent tool for testing numerous alternative graph properties; in those contexts where input is represented by large graphs, one may think of applying ASP based solutions on small sample graphs, in order to identify the most promising properties and, then, implementing them with other ad-hoc, more efficient, solutions in order to study the problem on real graphs. The adoption of ANN insights on important edges shown in this chapter may be of significant help in this task; indeed, leveraging the most important edges only allows to work on smaller but still significant graphs.

Furthermore, we do believe that applications like the one herein described can significantly motivate the scientific community, especially the one working on ASP, at improving performance of systems for their use in real-world applications.

5.3 Discussion

The tests presented in this section allow us to draw some interesting considerations. First of all, all tests presented in Section 5.1 actually proved the appropriateness of the approach for studying the evolution of MS; also, simplicity and high versatility in defining, setting up and carrying out a wide variety of tests showed how crucial is the role played by ASP. We also provided some experts

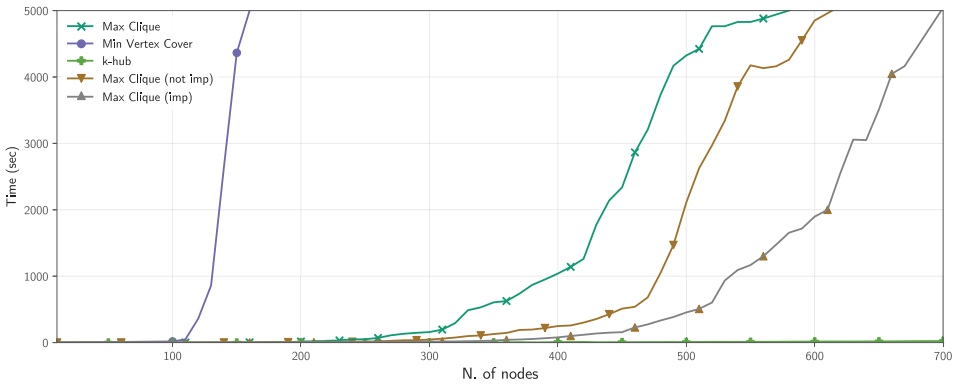


Figure 4.19: Running time of the Brain Evolution Simulation module on simulated graphs with increasing number of nodes.

with the system for a first view, and they have been quite impressed by the possibility of simulating brain evolution so easily.

On the ASP side, we can also say that what discussed in Section 5.2 confirms that expressiveness and compactness of the language make it perfectly suitable to address a wide variety of problems on graphs. Moreover, language extensions significantly expanded the range of applicability of ASP; as an example, weak constraints allowed us to easily express optimization problems and, analogously, the recently introduced possibility of placing external function calls into a logic program, with a predicate as function parameter instead of a single variable (see, e.g., the encoding in Figure 4.5), allowed us to keep the encoding simple and elegant even with the inclusion of complex graph metrics computation. Furthermore, when the ASP program does not include choice rules, actual ASP implementations can deal with very large graphs, and scale definitely well. Unfortunately, on the downside, the major weakness of current ASP systems becomes apparent when a combinatorial explosion of the problem occurs. In particular, while we have shown that the system is fully capable of addressing structural properties on connectome, we experienced that addressing large graphs is possible only to some extent.

Moreover, as pointed out in Section 5, when dealing with graph metrics, since the nondeterministic choice is carried out on edges instead of nodes, and since the metrics need to be computed on the entire guessed graph, current versions of state-of-the-art ASP systems do not reasonably scale over the connectome graphs. In particular, we observed that systems incur in out-of-memory or exceed time limit (more than one hour) much earlier with respect to the tests focusing on structural graph properties. Intuitively, the problem is that the systems need to generate all possible guesses on potential graphs, before computing the corresponding metrics. In order to exclude external function calls as the potential bottleneck in this case, we also checked a version of the program for controlling graph density variations using aggregates only; while avoiding external function calls allowed us to reduce memory issues, we encountered similar scalability issues on connectome.

The experience above calls for the need of some extra features of ASP systems, e.g., extending their solving capabilities with custom heuristics and propagators; some work in this direction is

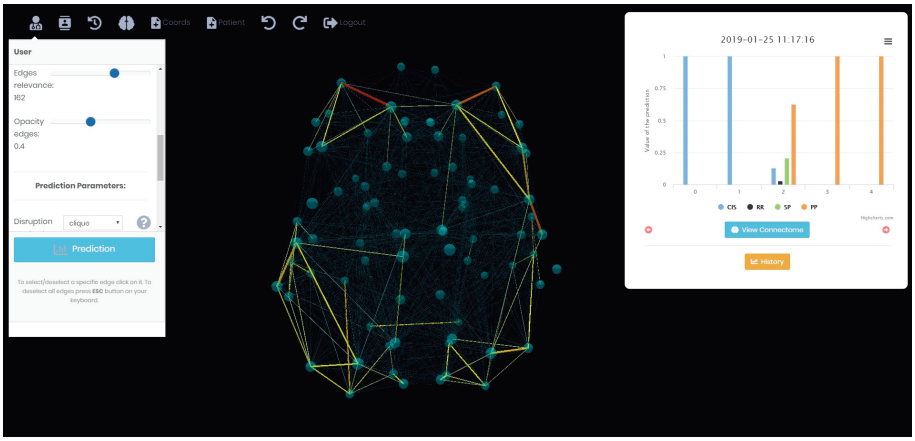


Figure 4.20: Screenshot of the integrated web environment

currently ongoing (see [Dodaro and Ricca (2018)] and references therein). However, the applicability of these approaches in our context is not straightforward. As a matter of fact, even taking the possibility of specifying suitable propagators into account, it is not always possible to easily define model generation guiding rules; let us think, for instance, to assortativity, where it is not clear how to guide the edge selection in order to imply a decrease in the property. Moreover, the problem is even more complex if we consider that the combinatorial explosion of this problem is coupled with an optimization task.

However, it is worth stressing again the potential role of important/not important edges in encompassing, to some extent, scalability issues in our general framework. In fact, we have first shown in Sections 5 (see specifically Figure 4.13) that removing even a high percentage of not important edges does not affect the classification quality. We have then shown in Section 5.2 (see specifically Figure 4.19) that limiting the computation on (a small number of) important edges only significantly reduces performance issues, thus extending the dimensions of graphs that can be managed.

To the best of our knowledge, this is the first work showing how to exploit the importance of an edge in graph-based classification tasks in order to boost reasoning capabilities over graphs. In our opinion, this result deserves further investigations in future works, and can stimulate the research community in looking for new ASP program evaluation optimizations.

6 Integrated Web Tool

In this section we present an integrated web tool that has been developed in order to implement the framework introduced in this chapter and make it actually usable. The tool is available online at `https://brainmsa.mat.unical.it`. The integrated environment provides a user-friendly interface that shows analysis results in real time. The main objective of the tool is to help physicians, typically neurologists that are not likely to be ANN/ASP experts, to study the evolution of MS through the application of the proposed framework, but also by manual inspection of brain modifications.

The input to the tool is expected to be a graph representation of the brain, obtained as described in Section 4.1. A 3D environment showing the connectome is then generated, as shown in Figure 4.20. The graph accurately reflects the shape of a human brain, so that it is possible to identify which nodes belong to a specific brain area. Usual rotation, translation and zooming operations are available for inspecting the brain structure. Edge colors depend on the corresponding weight, so as to provide a visual representation of connection strength.

The tool provides pre-defined specializations for the various modules of the framework. For some of them, such as the *Brain Evolution Simulation* module, the user can choose one among different ASP programs already available, or she/he can provide other programs personally developed for specific purposes. It allows also to choose an empty program, in order to apply only manual modifications as explained below.

By launching the classification and asking for one single iteration of the framework, the user can immediately check (see Figure 4.20) the new prediction on the right panel, and the 3D brain representation is updated with the applied modifications. If the number of required iterations is more than one, the right panel shows the graphs for each iteration (similarly to the ones presented in Section 5), whereas only the last 3D brain representation is shown.

Once the user carried out several runs, it is possible to have a general overview of obtained results by clicking the *History* button. In this case, the page shown in Figure 4.21 is presented; it first shows a *boxplot* for each stage of MS that summarizes the overall probability values returned by the classifier during the current test session. Moreover, the detailed history of prediction results computed on the current connectome is also provided. Finally, for each prediction, the system provides also a heat map representing the adjacency matrix of the corresponding graph; this can be useful to see distribution of edges and weights at a glance.

Besides the implementation of the framework, the tool provides also some more functionalities helping experts to carry out manual and more refined analyses. Specifically, first of all, the user can manually select the set of edges to modify in a brain evolution step. This selection can be either exploited in substitution of the Brain Evolution Simulation module (if the user chooses the empty ASP program) or it can be seen as a pre-processing step on the connectome, if one of the specializations of the Brain Evolution Simulation module is chosen.

Furthermore, the user can modify the visualization of the connectome, based on edge importance. In particular, if at least one classification has been carried out on the connectome, edge importance, as introduced in Section 4.5, is available. Then, users can hide or show edges, based on their importance, by using a slider. As a consequence, manual inspection on the connectome can be greatly simplified, allowing the user to concentrate her/his attention on important edges only.

7 Specialization of the framework to other scenarios

In order to show the generality of the proposed framework, in this section we present some additional application scenarios it can be specialized to in a quite straightforward way. In particular, we first overview some additional biomedical contexts and then we show the application of the

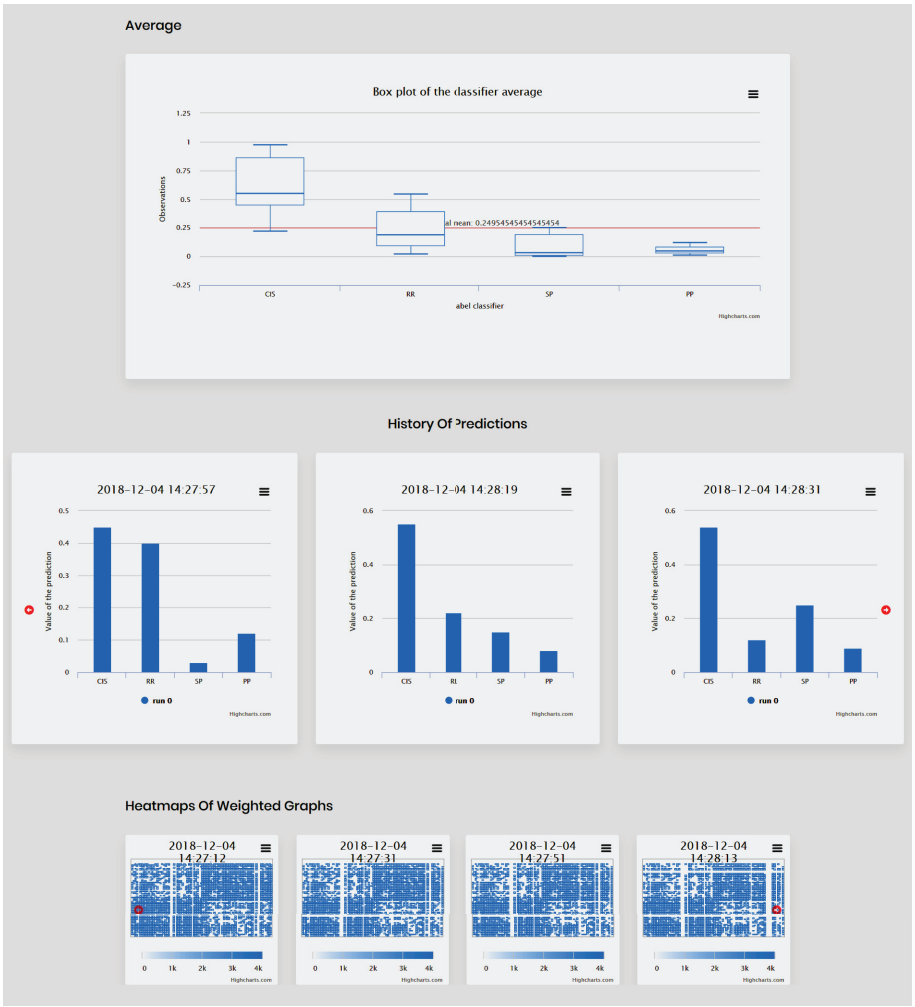


Figure 4.21: An example of how the environment encompasses results.

framework to a very different application scenario, namely influence prediction in social networks.

7.1 Specialization of the framework to other neurological disorders

It has been proved by several independent studies that there is a strong correlation between the variations of connections among neurons and the function of the brain and, consequently, with possible insurgence of several neurological disorders [Bargmann and Marder (2013)]. As an example, in the Alzheimer Disease researchers observed a drecrease in the connectivity, associated with changes in the hippocampus [Lenka *et al.* (2015)]; altered connectivity has been observed also in the Parkinson disease [Lenka *et al.* (2015)]; similarly, an increased connectivity associated with changes in the amygdala have been associated with anxiety disorder [Stein *et al.* (2007)]. In all such contexts, the analysis of brain connections and their variations can provide significant insights in the knowledge of disease evolution.

Let us concentrate on the Alzheimer Disease (hereafter, AD); it is well known that, at early stages, this disease appears as Mild Cognitive Impairment (hereafter, MCI) but not all patients with MCI subsequently develop AD [Petersen (2004)]. However, it is also known that in MCI

progression towards AD a key role is played by the loss of connectivity among the different cortical areas. Thus, a large variety of approaches aiming at characterizing both MCI and AD have been proposed in the literature [Hornero *et al.* (2009), Jeong (2004)]. Some of them are based on the analysis of electroencephalograms (EEG) data; this is a less invasive observation method than MRI. In this case, the analysis can be based on a graph, where each node represents an electrode, and the weight of each edge expresses the similarity degree between the signals registered by the corresponding electrodes (see [Cauteruccio *et al.*]).

Our framework can be specialized quite straightforwardly to the analysis of MCI evolution in order to study key factors determining its progress to AD. In fact, it is sufficient to specialize the *Classifier* module with any approach developed to classify patients in one of the *Healthy*, *MCI*, or *AD* stages. The *Brain Evolution Simulation* module can be then specialized with a proper ASP program that identifies subgraphs corresponding to (variations of) some property of interest; as an example, in [Cauteruccio *et al.*] it has been shown that interesting graph properties related to AD are network density and clustering coefficient. Here, again, the *Classification Validity Check* can refute the classification outcome if a significant degradation in the graph corresponds to a predicted remission of the disease; indeed, this situation is not biologically relevant.

7.2 Specialization of the framework in the context of Social Networks

The work presented in [Wu *et al.* (2019)] surveys several contexts where ANNs are applied to graph data; indeed, there is increasing interest in extending deep learning approaches for this kind of data. Interesting applications include, but are not limited to, e-commerce and recommender systems, citation networks, social networks, traffic analysis, drug discovery, adversarial attack prevention, and event detection. All of these problems can benefit from the application of our framework in the identification of potentially relevant graph properties.

In order to provide an example, we focus next on one of them, namely influence prediction in social networks [Qiu *et al.* (2018)]. A social network can be represented as a graph $G = (V, E)$, where V denotes the set of users and E denotes the set of relationships between them. Each user in a social network performs social actions towards other users; these can be suitably summarized as edge labels between the nodes corresponding to the involved users. Social influence commonly refers to the phenomenon that the opinions and actions of a user are affected by others. In many applications, such as advertising and recommendation, it is crucial to predict the social influence of each user.

In [Qiu *et al.* (2018)] a neural network-based approach is proposed to predict the action status of a user given the action statuses of both near neighbors and local structural information. The list of action statuses strictly depends on the kind of social network under analysis. As an example it can be a "retweet" action in *Twitter* or a citation action in academic social networks. The input network is then fed to the ANN which outputs a two-dimension representation for each user indicating the action status prediction, which is then exploited for the social influence computation.

In this contexts it would be of great relevance to study the evolution of social influence with respect to modifications on the social graph. As an example, it would be interesting to find the

minimal graph modifications required to increase the social influence in the network. Our framework can be specialized quite straightforwardly also to this context. In particular, the *Classifier* module can be specialized with the approach presented in [Qiu *et al.* (2018)] in order to predict action statuses and, thus, social influence. The *Brain Evolution Simulation* module, which in this context could be more appropriately renamed as *Graph Evolution Simulation*, can be specialized with the ASP program of choice identifying the minimal changes to be applied on the input graph in order to reach some target value of social influence. Even if the complete development of this specific application is out of the scope of this paper, it is easy to see that the *Classification Validity Checker* can be easily encoded with specific rules allowing to detect wrong classifications in action statuses.

8 Conclusion

This chapter introduced a general and extensible framework pointing out opportunities provided by a combined use of ASP and ANN. In particular, we grounded the framework in order to provide an effective support for neurologists in studying the evolution of neurological disorders.

We have shown that a mixed use of ASP and ANNs can be of significant impact both in bioinformatics and other research fields. Indeed, logic-based modules greatly simplify the exploration of different, possibly complex, variations in the structure of the connectome, and ANNs allow to immediately check the potential impact of such variations on the course of the disease. We provided three specializations of the general framework and tested them on real data to show the effectiveness of the proposed approach. Extensive tests proved the potential impact of the framework on the discovery process and some limitations of current ASP solvers. Based on this experience, we developed a web tool allowing even non experts to explore the connectome and test the impact of its variations on the course of the disease.

We believe that the results are encouraging; moreover, they further motivate the already running research activities for optimizing ASP program evaluations. Finally, obtained results provide us with a solid basis for encouraging the communities of both ASP and ANN areas to identify more contexts where a mixed use of these tools can lead to significant benefits.

As far as future work is concerned, we plan to specialize the presented framework on the application contexts outlined in Section 7.

IV Conclusions and Perspectives

Chapter 1

Conclusions

Processing MRI data is crucial to better understand the pathological mechanisms of complex brain diseases such as MS. In this context, graph theory is a powerful approach for the analysis of the WM network. These techniques, combined with powerful machine learning and deep learning algorithms, open new perspectives for the neuroimaging field.

Deep learning techniques have represented in recent years a powerful tool for supporting various biomedical analysis. Concerning our contribution in this field, Generative Adversarial Networks were used to automatic generate artificial Magnetic Resonance Images of slices of the human brain; both quantitative and human-based evaluations of generated images have been carried out to assess the effectiveness of the method. This work represented one of the first applications of these techniques to the neuroimaging field. Furthermore, Deep Learning methods were also developed to solve the Optic Disc Detection task. In this context, we showed how transfer learning can be easily applied to different domains, thus allowing us to overcome the problems in presence of big data sets and high requirements in terms of computational power, and laying the foundation for interesting future works.

Various approaches for the automatic classification of MS patients in their respective clinical profiles based on their brain structural connectivity were presented. Various deep learning techniques were defined and compared, gradually trying to reduce human intervention in any step of the process. In these studies, we have shown the role of local graph metrics in the characterization of the different MS forms. However, was interesting to notice that local graph metrics do not improve the classification results suggesting that the latent features created by the NN in its layers have much important informative content. Furthermore, graph weights represent an important source of information to discriminate between clinical forms.

Finally, beyond the patients' classification itself, one of the main objectives of our studies relies

on the interpretation of the defined models, with the primary goal of detecting and characterize the presence of pathological mechanisms in the brain networks. In this context, a generic framework showing the impact of mixing rule-based systems and neural networks to investigate neurological disorders were proposed. The ambitious goal is to boost the interest of the research community in developing a more tight integration of these two approaches. Interesting substructures of the connectome were identified, which could be analyzed from a clinical perspective.

Chapter 2

Perspectives

The various methods explored in this work can be seen as a starting point for a more general and robust pipeline to investigate pathological mechanisms. It is worth to note that the methods and results reported in this manuscript represented a portion of the work developed under this project. Several other researches were conducted using cutting edge approaches, which still need to be further investigated.

Generative approaches in the biomedical field, for example, have been just started to be studied in the literature, and a large variety of applications are still open. An interesting perspective is to improve the quality of the generated images, to be more and more similar to real MRI scans; to this aim, a comparison with alternatives models will be of clear interest, as they are currently emerging in literature. Furthermore, the generation may be improved by allowing the network to add pathological symptoms and provide unseen data of synthesized patients; this might also improve the study of rare diseases. Furthermore, Generative models are widely used to extract latent features of the input in a compressed dimensional space. Investigating such latent space would be interesting to reduce the complexity of the problem, thus isolating discriminative biomarkers or to characterizing pathological alterations. Also, longitudinal data can be exploited in this context to simulate the evolution of the pathology.

Novel graph-based techniques are also emerging, which could provide better and more interpretable results. In this context, another interesting perspective is to combine functional and structural connectivity to analyze the progression of the disease. Such a multiview approach could be applied for predicting the short term evolution of the patient disability, which represents one of the more disturbing aspects of MS.

Bibliography

[Adrian *et al.* (2018)] Adrian, W. T., Alviano, M., Calimeri, F., Cuteri, B., Dodaro, C., Faber, W., Fuscà, D., Leone, N., Manna, M., Perri, S., Ricca, F., Veltri, P., and Zangari, J. (2018). The ASP system DLV: advancements and applications. *KI*, 32(2-3):177–179.

[Albers (1998)] Albers, G. W. (1998). Diffusion-weighted mri for evaluation of acute stroke. *Neurology*, 51(3 Suppl 3):S47–S49.

[Alviano *et al.* (2019)] Alviano, M., Amendola, G., Dodaro, C., Leone, N., Maratea, M., and Ricca, F. (2019). Evaluation of disjunctive programs in WASP. In Balduccini, M., Lierler, Y., and Woltran, S., editors, *Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019, Philadelphia, PA, USA, June 3-7, 2019, Proceedings*, volume 11481 of *Lecture Notes in Computer Science*, pages 241–255. Springer.

[Aquino *et al.* (2010)] Aquino, A., Gegúndez-Arias, M. E., and Marín, D. (2010). Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques. *IEEE transactions on medical imaging*, 29(11):1860–1869.

[Arias *et al.* (2018)] Arias, J., Carro, M., Salazar, E., Marple, K., and Gupta, G. (2018). Constraint answer set programming without grounding. *TPLP*, 18(3-4):337–354.

[Asta and Özcan (2015)] Asta, S. and Özcan, E. (2015). A tensor-based selection hyper-heuristic for cross-domain heuristic search. *Information Sciences*, 299:412–432.

[Aytar (2014)] Aytar, Y. (2014). *Transfer learning for object category detection*. PhD thesis, Oxford University, UK.

[Azevedo *et al.* (2018)] Azevedo, C. J., Cen, S. Y., Khadka, S., Liu, S., Kornak, J., Shi, Y., Zheng, L., Hauser, S. L., and Pelletier, D. (2018). Thalamic atrophy in multiple sclerosis: a magnetic resonance imaging marker of neurodegeneration throughout disease. *Annals of neurology*, 83(2):223–234.

[Balashov and Lindzen (2012)] Balashov, K. E. and Lindzen, E. (2012). Acute demyelinating lesions with restricted diffusion in multiple sclerosis. *Multiple Sclerosis Journal*, 18(12):1745–1753.

[Balduccini and Lierler (2017)] Balduccini, M. and Lierler, Y. (2017). Constraint answer set solver EZCSP and why integration schemas matter. *TPLP*, 17(4):462–515.

[Baral (2003)] Baral, C. (2003). *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, New York, NY, USA.

[Bargmann and Marder (2013)] Bargmann, C. and Marder, E. (2013). From the connectome to brain function. *Nature Methods*, 10:483. Nature Publishing Group.

[Barkhof (2002)] Barkhof, F. (2002). The clinico-radiological paradox in multiple sclerosis revisited. *Current opinion in neurology*, 15(3):239–245.

[Barkhof *et al.* (1997)] Barkhof, F., Filippi, M., Miller, D. H., Scheltens, P., Campi, A., Polman, C. H., Comi, G., Ader, H. J., Losseff, N., and Valk, J. (1997). Comparison of mri criteria at first presentation to predict conversion to clinically definite multiple sclerosis. *Brain*, 120(11):2059–2069.

[Barrett *et al.* (2013)] Barrett, C. W., Deters, M., de Moura, L. M., Oliveras, A., and Stump, A. (2013). 6 years of SMT-COMP. *J. Autom. Reasoning*, 50(3):243–277.

[Barrett *et al.* (2016)] Barrett, C., Fontaine, P., and Tinelli, C. (2016). The Satisfiability Modulo Theories Library (SMT-LIB). `www.SMT-LIB.org`.

[Barrett and Tinelli (2018)] Barrett, C. W. and Tinelli, C. (2018). Satisfiability modulo theories. In Clarke, E. M., Henzinger, T. A., Veith, H., and Bloem, R., editors, *Handbook of Model Checking.*, pages 305–343. Springer.

[Baselice *et al.* (2005)] Baselice, S., Bonatti, P. A., and Gelfond, M. (2005). Towards an integration of answer set and constraint solving. In Gabbrielli, M. and Gupta, G., editors, *Logic Programming, 21st International Conference, ICLP 2005, Sitges, Spain, October 2-5, 2005, Proceedings*, volume 3668 of *Lecture Notes in Computer Science*, pages 52–66. Springer.

[Basser *et al.* (1992)] Basser, P., Mattiello, J., and LeBihan, D. (1992). Diagonal and off-diagonal components of the self-diffusion tensor: their relation to and estimation from the nmr spin-echo signal. *Proc. 11th Annu. Meet. SMRM, Berlin*, 1:1222.

[Basser *et al.* (1994)] Basser, P. J., Mattiello, J., and LeBihan, D. (1994). Estimation of the effective self-diffusion tensor from the nmr spin echo. *Journal of Magnetic Resonance, Series B*, 103(3):247–254.

[Bassett and Bullmore (2009)] Bassett, D. and Bullmore, E. (2009). *Human brain networks in health and disease*, pages 340–347. Curr Opin Neurol.

[Bates *et al.* (2015)] Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4, 1–51. computation.

[Baur *et al.* (2017)] Baur, C., Albarqouni, S., and Navab, N. (2017). Semi-supervised deep learning for fully convolutional networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 311–319. Springer.

[Beck *et al.* (2015)] Beck, H., Dao-Tran, M., Eiter, T., and Fink, M. (2015). LARS: A logic-based framework for analyzing reasoning over streams. In *AAAI*, pages 1431–1438. AAAI Press.

[Behrens *et al.* (2003)] Behrens, T., Johansen-Berg, H., Woolrich, M., Smith, S., Wheeler-Kingshott, C., Boulby, P., Barker, G., Sillery, E., Sheehan, K., Ciccarelli, O., *et al.* (2003). Non-invasive mapping of connections between human thalamus and cortex using diffusion imaging. *Nature neuroscience*, 6(7):750–757.

[Belkin *et al.* (2004)] Belkin, M., Matveeva, I., and Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pages 624–638. Springer.

[Breuleux *et al.* (2011)] Breuleux, O., Bengio, Y., and Vincent, P. (2011). Quickly generating representative samples from an rbm-derived process. *Neural computation*, 23(8):2058–2073.

[Brooks *et al.* (2007)] Brooks, D. R., Erdem, E., Erdogan, S. T., Minett, J. W., and Ringe, D. (2007). Inferring phylogenetic trees using answer set programming. *J. Autom. Reasoning*, 39(4):471–511.

[Brosch *et al.* (2015)] Brosch, T., Yoo, Y., Tang, L. Y. W., Li, D. K. B., Traboulsee, A., and Tam, R. C. (2015). Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, pages 3–11.

[Brown (1828)] Brown, R. (1828). A brief account of microscopical observations made in the months of june, july and august 1827 on the particles contained in the pollen of plants.

[Brück *et al.* (2003)] Brück, W., Kuhlmann, T., and Stadelmann, C. (2003). Remyelination in multiple sclerosis. *Journal of the neurological sciences*, 206(2):181–185.

[Bullmore and Sporns (2009)] Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10:186–198.

[Calimeri *et al.*] Calimeri, F., Marzullo, A., Stamile, C., and Terracina, G. Graph based neural networks for automatic classification of multiple sclerosis clinical courses. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 18)(2018, forthcoming)*.

[Calimeri *et al.* (2007)] Calimeri, F., Cozza, S., and Ianni, G. (2007). External sources of knowledge and value invention in logic programming. *Ann. Math. Artif. Intell.*, 50(3-4):333–361.

[Calimeri *et al.* (2012)a] Calimeri, F., Faber, W., Gebser, M., Ianni, G., Kaminski, R., Krennwallner, T., Leone, N., Ricca, F., and Schaub, T. (2012a). ASP-Core-2: Input language format.

[Calimeri *et al.* (2012)b] Calimeri, F., Ianni, G., Krennwallner, T., and Ricca, F. (2012b). The answer set programming competition. *AI Magazine*, 33(4):114–118.

[Calimeri *et al.* (2016)] Calimeri, F., Fink, M., Germano, S., Humenberger, A., Ianni, G., Redl, C., Stepanova, D., Tucci, A., and Wimmer, A. (2016). Angry-hex: An artificial player for angry birds based on declarative knowledge bases. *IEEE Trans. Comput. Intellig. and AI in Games*, 8(2):128–139.

[Calimeri *et al.* (2017)] Calimeri, F., Fuscà, D., Perri, S., and Zangari, J. (2017). I-DLV: the new intelligent grounder of DLV. *Intelligenza Artificiale*, 11(1):5–20.

[Calimeri *et al.* (2018)a] Calimeri, F., Cauteruccio, F., Marzullo, A., Stamile, C., and Terracina, G. (2018a). Mixing logic programming and neural networks to support neurological disorders analysis. In *International Joint Conference on Rules and Reasoning*, pages 33–47. Springer.

[Calimeri *et al.* (2018)b] Calimeri, F., Cauteruccio, F., Marzullo, A., Stamile, C., and Terracina, G. (2018b). Mixing logic programming and neural networks to support neurological disorders analysis. In *RuleML+RR*, volume 11092 of *Lecture Notes in Computer Science*, pages 33–47. Springer.

[Calimeri *et al.* (2018)c] Calimeri, F., Marzullo, A., Stamile, C., and Terracina, G. (2018c). Graph based neural networks for automatic classification of multiple sclerosis clinical courses. In *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*.

[Calimeri and Ricca (2013)] Calimeri, F. and Ricca, F. (2013). On the Application of the Answer Set Programming System DLV in Industry: a Report from the Field. *Book Reviews*, 2013(03):1–16.

[Cauteruccio *et al.*] Cauteruccio, F., Lo Giudice, P., Terracina, G., Ursino, D., Mammone, N., and Morabito, F. A new network-based approach to investigating neurological disorders. *International Journal of Data Mining, Modelling and Management*. Inderscience. Forthcoming - To appear.

[Cavallari *et al.* (2015)] Cavallari, M., Stamile, C., Umeton, R., Calimeri, F., and Orzi, F. (2015). Novel method for automated analysis of retinal images: results in subjects with hypertensive retinopathy and cadasil. *BioMed research international*, 2015.

[Chabierski *et al.* (2017)] Chabierski, P., Russo, A., Law, M., and Broda, K. (2017). Machine comprehension of text using combinatory categorial grammar and answer set programs. In *COMMONSENSE*, volume 2052 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Chapelle *et al.* (2008)] Chapelle, O., Sindhwani, V., and Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb):203–233.

[Chapelle *et al.* (2009)] Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

[Chapelle and Zien (2005)] Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer.

[Charalambous *et al.* (2019)] Charalambous, T., Tur, C., Prados, F., Kanber, B., Chard, D. T., Ourselin, S., Clayden, J. D., Wheeler-Kingshott, C. A. G., Thompson, A. J., and Toosy, A. T. (2019). Structural network disruption markers explain disability in multiple sclerosis. *J Neurol Neurosurg Psychiatry*, 90(2):219–226.

[Chen *et al.* (2015)] Chen, X., Xu, Y., Yan, S., Wong, D. W. K., Wong, T. Y., and Liu, J. (2015). Automatic feature learning for glaucoma detection based on deep learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 669–677. Springer.

[Cok *et al.* (2015)] Cok, D. R., Stump, A., and Weber, T. (2015). The 2013 evaluation of SMT-COMP and SMT-LIB. *J. Autom. Reasoning*, 55(1):61–90.

[Compston and Coles (2008)] Compston, A. and Coles, A. (2008). Multiple sclerosis. *Lancet (London, England)*, 372:1502–1517.

[Confavreux *et al.* (2003)] Confavreux, C., Vukusic, S., and Adeleine, P. (2003). Early clinical predictors and progression of irreversible disability in multiple sclerosis: an amnesic process. *Brain*, 126(4):770–782.

[Confavreux and Vukusic (2006)] Confavreux, C. and Vukusic, S. (2006). Natural history of multiple sclerosis: a unifying concept. *Brain*, 129(3):606–616.

[Costa *et al.* (2017)] Costa, P., Galdran, A., Meyer, M. I., Niemeijer, M., Abràmoff, M., Mendonça, A. M., and Campilho, A. (2017). End-to-end adversarial retinal image synthesis. *IEEE transactions on medical imaging*, 37(3):781–791.

[Degenhardt *et al.* (2009)] Degenhardt, A., Ramagopalan, S. V., Scalfari, A., and Ebers, G. C. (2009). Clinical prognostic factors in multiple sclerosis: a natural history review. *Nature Reviews Neurology*, 5(12):672.

[Deloire *et al.* (2011)] Deloire, M. S., Ruet, A., Hamel, D., Bonnet, M., Dousset, V., and Brochet, B. (2011). Mri predictors of cognitive outcome in early multiple sclerosis. *Neurology*, 76(13):1161–1167.

[Denton *et al.* (2015)] Denton, E. L., Chintala, S., Fergus, R., *et al.* (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494.

[Disanto *et al.* (2012)] Disanto, G., Morahan, J., Barnett, M., Giovannoni, G., and Ramagopalan, S. (2012). The evidence for a role of b cells in multiple sclerosis. *Neurology*, 78(11):823–832.

[Dodaro and Ricca (2018)] Dodaro, C. and Ricca, F. (2018). The external interface for extending wasp. *Theory and Practice of Logic Programming*, pages 1–24.

[Donahue *et al.* (2015)] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

[Dosovitskiy *et al.* (2015)] Dosovitskiy, A., Tobias Springenberg, J., and Brox, T. (2015). Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546.

[Dosovitskiy and Brox (2016)] Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666.

[Duffner and Garcia (2007)] Duffner, S. and Garcia, C. (2007). An online backpropagation algorithm with validation error-based adaptive learning rate. In *International Conference on Artificial Neural Networks*, pages 249–258. Springer.

[Durand-Dubief *et al.* (2012)] Durand-Dubief, F., Belaroussi, B., Armspach, J., Dufour, M., Roggerone, S., Vukusic, S., Hannoun, S., Sappey-Marinier, D., Confavreux, C., and Cotton, F. (2012). Reliability of longitudinal brain volume loss measurements between 2 sites in patients with multiple sclerosis: comparison of 7 quantification techniques. *American Journal of Neuroradiology*.

[Einstein (1956)] Einstein, A. (1956). *Investigations on the Theory of the Brownian Movement*. Courier Corporation.

[Eiter *et al.* (2016)] Eiter, T., Fink, M., Ianni, G., Krennwallner, T., Redl, C., and Schüller, P. (2016). A model building framework for answer set programming with external computations. *TPLP*, 16(4):418–464.

[Eiter *et al.* (2018)a] Eiter, T., Germano, S., Ianni, G., Kaminski, T., Redl, C., Schüller, P., and Weinzierl, A. (2018a). The DLVHEX system. *KI - Künstliche Intelligenz*, 32(2-3):187–189.

[Eiter *et al.* (2018)b] Eiter, T., Germano, S., Ianni, G., Kaminski, T., Redl, C., Schüller, P., and Weinzierl, A. (2018b). The DLVHEX system. *KI*, 32(2-3):187–189.

[Erdem *et al.* (2016)] Erdem, E., Gelfond, M., and Leone, N. (2016). Applications of answer set programming. *AI Magazine*, 37(3):53–68.

[Eshaghi *et al.* (2018)] Eshaghi, A., Marinescu, R. V., Young, A. L., Firth, N. C., Prados, F., Jorge Cardoso, M., Tur, C., De Angelis, F., Cawley, N., Brownlee, W. J., *et al.* (2018). Progression of regional grey matter atrophy in multiple sclerosis. *Brain*, 141(6):1665–1677.

[Faber *et al.* (2004)] Faber, W., Leone, N., and Pfeifer, G. (2004). Recursive aggregates in disjunctive logic programs: Semantics and complexity. In Alferes, J. J. and Leite, J., editors, *Proceedings of the 9th European Conference on Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes on Artificial Intelligence (LNAI)*, pages 200–212. Springer Verlag.

[Filipovych *et al.* (2011)] Filipovych, R., Davatzikos, C., Initiative, A. D. N., *et al.* (2011). Semi-supervised pattern classification of medical images: application to mild cognitive impairment (mci). *NeuroImage*, 55(3):1109–1119.

[Fillard *et al.* (2011)] Fillard, P., Descoteaux, M., Goh, A., Gouttard, S., Jeurissen, B., Malcolm, J., Ramirez-Manzanares, A., Reisert, M., Sakaie, K., Tensaouti, F., *et al.* (2011). Quantitative evaluation of 10 tractography algorithms on a realistic diffusion mr phantom. *Neuroimage*, 56(1):220–234.

[Fischl *et al.* (2004)] Fischl, B., Van Der Kouwe, A., Destrieux, C., Halgren, E., Ségonne, F., Salat, D. H., Busa, E., Seidman, L. J., Goldstein, J., Kennedy, D., *et al.* (2004). Automatically parcellating the human cerebral cortex. *Cerebral cortex*, 14(1):11–22.

[Fox and Monette (2002)] Fox, J. and Monette, G. (2002). *An R and S-Plus companion to applied regression*. Sage.

[Fragkiadaki *et al.* (2015)] Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J. (2015). Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354.

[Fuscà *et al.* (2017)] Fuscà, D., Calimeri, F., Zangari, J., and Perri, S. (2017). I-DLV+MS: preliminary report on an automatic ASP solver selector. In *RCRA@AI*IA*, volume 2011 of *CEUR Workshop Proceedings*, pages 26–32. CEUR-WS.org.

[Gebser *et al.* (2011)a] Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T., Schneider, M. T., and Ziller, S. (2011a). A portfolio solver for answer set programming: Preliminary report. In *LPNMR*, volume 6645 of *LNCS*, pages 352–357. Springer.

[Gebser *et al.* (2011)b] Gebser, M., Schaub, T., Thiele, S., and Veber, P. (2011b). Detecting inconsistencies in large biological networks with answer set programming. *Theory and Practice of Logic Programming*, 11(2-3):323âĂŞ360.

[Gebser *et al.* (2016)] Gebser, M., Maratea, M., and Ricca, F. (2016). What's hot in the answer set programming competition. In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 4327–4329. AAAI Press.

[Gebser *et al.* (2017)] Gebser, M., Maratea, M., and Ricca, F. (2017). The sixth answer set programming competition. *J. Artif. Intell. Res.*, 60:41–95.

[Gebser *et al.* (2018)] Gebser, M., Leone, N., Maratea, M., Perri, S., Ricca, F., and Schaub, T. (2018). Evaluation techniques and systems for answer set programming: a survey. In Lang, J., editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 5450–5456. ijcai.org.

[Gebser *et al.* (2019)a] Gebser, M., Kaminski, R., Kaufmann, B., and Schaub, T. (2019a). Multi-shot ASP solving with clingo. *TPLP*, 19(1):27–82.

[Gebser *et al.* (2019)b] Gebser, M., Maratea, M., and Ricca, F. (2019b). The seventh answer set programming competition: Design and results. *CoRR*, abs/1904.09134.

[Gelfond (2010)] Gelfond, M. (2010). Knowledge representation language p-log - A short introduction. In *Datalog*, volume 6702 of *LNCS*, pages 369–383. Springer.

[Gelfond and Leone (2002)] Gelfond, M. and Leone, N. (2002). Logic Programming and Knowledge Representation – the A-Prolog perspective . 138(1–2):3–38.

[Gelfond and Lifschitz (1991)] Gelfond, M. and Lifschitz, V. (1991). Classical Negation in Logic Programs and Disjunctive Databases. 9:365–385.

[Goh *et al.* (2001)] Goh, K. G., Hsu, W., Li Lee, M., and Wang, H. (2001). Adris: an automatic diabetic retinal image screening system. *Studies in Fuzziness and Soft Computing*, 60:181–210.

[Goldschmidt *et al.* (2009)] Goldschmidt, T., Antel, J., König, F., Brück, W., and Kuhlmann, T. (2009). Remyelination capacity of the ms brain decreases with disease chronicity. *Neurology*, 72(22):1914–1921.

[Golub and Reinsch (1971)] Golub, G. H. and Reinsch, C. (1971). Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer.

[Goodfellow *et al.* (2014)] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Goodfellow *et al.* (2016)] Goodfellow, I. J., Bengio, Y., and Courville, A. C. (2016). *Deep Learning*. Adaptive computation and machine learning. MIT Press.

[Graves and Jaitly (2014)] Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772.

[Gu and Angelov (2018)] Gu, X. and Angelov, P. P. (2018). Semi-supervised deep rule-based approach for image classification. *Applied Soft Computing*, 68:53–68.

[Hagmann *et al.* (2007)] Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Wedeen, V. J., Meuli, R., and Thiran, J.-P. (2007). Mapping human whole-brain structural networks with diffusion mri. *PloS one*, 2(7):e597.

[Hamilton *et al.* (2017)] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

[Hannoun *et al.* (2012)] Hannoun, S., Bagory, M., Durand-Dubief, F., Ibarrola, D., Comte, J.-C., Confavreux, C., Cotton, F., and Sappey-Marinier, D. (2012). Correlation of diffusion and metabolic alterations in different clinical forms of multiple sclerosis. *PLoS One*, 7(3):e32525.

[Harshman (1977)] Harshman, R. A. (1977). Parafac: Methods of three-way factor analysis and multidimensional scaling according to the principle of proportional profiles.

[Haykin (1998)] Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.

[Hilgetag and Kaiser (2004)] Hilgetag, C. and Kaiser, M. (2004). *Clustered organization of cortical connectivity*, volume 2, pages 353–360. Neuroinformatics.

[Hornero *et al.* (2009)] Hornero, R., Abásolo, D., Escudero, J., and Gómez, C. (2009). Nonlinear analysis of electroencephalogram and magnetoencephalogram recordings in patients with Alzheimer's disease. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1887):317–336. The Royal Society.

[Hu *et al.* (2016)] Hu, Z., Ma, X., Liu, Z., Hovy, E. H., and Xing, E. P. (2016). Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[Hunyadi *et al.* (2014)] Hunyadi, B., Camps, D., Sorber, L., Paesschen, W. V., Vos, M. D., Huffel, S. V., and Lathauwer, L. D. (2014). Block term decomposition for modelling epileptic seizures. *EURASIP J. Adv. Sig. Proc.*, 2014:139.

[Ion-Mărgineanu *et al.* (2017)] Ion-Mărgineanu, A., Kocevar, G., Stamile, C., Sima, D. M., Durand-Dubief, F., Van Huffel, S., and Sappey-Marinier, D. (2017). A comparison of machine learning approaches for classifying multiple sclerosis courses using mrsi and brain segmentations. In *International Conference on Artificial Neural Networks*, pages 643–651. Springer.

[Jain *et al.* (2016)] Jain, A., Zamir, A. R., Savarese, S., and Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317.

[Jbabdi and Johansen-Berg (2011)] Jbabdi, S. and Johansen-Berg, H. (2011). Tractography: where do we go from here? brain connectivity 1 (3), 169–183.

[Jenkinson *et al.* (2012)] Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W., and Smith, S. M. (2012). FSL. *NeuroImage*, 62(2):782–790.

[Jeong (2004)] Jeong, J. (2004). EEG dynamics in patients with Alzheimer's disease. *Clinical neurophysiology*, 115(7):1490–1505. Elsevier.

[Jiang *et al.* (2017)] Jiang, B., Chen, H., Yuan, B., and Yao, X. (2017). Scalable graph-based semi-supervised learning through sparse bayesian model. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2758–2771.

[Jin and JaJa (2018)] Jin, Y. and JaJa, J. F. (2018). Learning graph-level representations with gated recurrent neural networks. *arXiv preprint arXiv:1805.07683*.

[Kaminski *et al.* (2017)] Kaminski, R., Schaub, T., and Wanko, P. (2017). A tutorial on hybrid answer set solving with clingo. In Ianni, G., Lembo, D., Bertossi, L. E., Faber, W., Glimm, B., Gottlob, G., and Staab, S., editors, *Reasoning Web. Semantic Interoperability on the Web - 13th International Summer School 2017, London, UK, July 7-11, 2017, Tutorial Lectures*, volume 10370 of *Lecture Notes in Computer Science*, pages 167–203. Springer.

[Kawahara *et al.* (2017)] Kawahara, J., Brown, C. J., Miller, S. P., Booth, B. G., Chau, V., Grunau, R. E., Zwicker, J. G., and Hamarneh, G. (2017). Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146:1038–1049.

[Kingma and Ba (2014)a] Kingma, D. P. and Ba, J. (2014a). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kingma and Ba (2014)b] Kingma, D. P. and Ba, J. (2014b). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[Kipf and Welling (2016)] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

[Kocevar *et al.* (2016)] Kocevar, G., Stamile, C., Hannoun, S., Cotton, F., Vukusic, S., Durand-Dubief, F., and Sappey-Marinier, D. (2016). Graph Theory-Based Brain Connectivity for Automatic Classification of Multiple Sclerosis Clinical Courses. *Frontiers in Neuroscience*, 10:478.

[Kouvaros and Lomuscio (2018)] Kouvaros, P. and Lomuscio, A. (2018). Formal verification of cnn-based perception systems. *CoRR*, abs/1811.11373.

[Kriege *et al.* (2019)] Kriege, N. M., Johansson, F. D., and Morris, C. (2019). A survey on graph kernels. *arXiv preprint arXiv:1903.11835*.

[Krizhevsky *et al.* (2012)] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kugler and Deppe (2018)] Kugler, A. V. and Deppe, M. (2018). Non-lesional cerebellar damage in patients with clinically isolated syndrome: Dti measures predict early conversion into clinically definite multiple sclerosis. *NeuroImage: Clinical*, 19:633–639.

[Kurtzke (1980)] Kurtzke, J. F. (1980). Geographic distribution of multiple sclerosis: an update with special reference to europe and the mediterranean region. *Acta Neurologica Scandinavica*, 62(2):65–80.

[Kurtzke (2000)] Kurtzke, J. F. (2000). Multiple sclerosis in time and space-geographic clues to cause. *Journal of neurovirology*, 6(2):S134.

[Law *et al.* (2015)] Law, M., Russo, A., and Broda, K. (2015). Learning weak constraints in answer set programming. *TPLP*, 15(4-5):511–525.

[Law *et al.* (2016)] Law, M., Russo, A., and Broda, K. (2016). Iterative learning of answer set programs from context dependent examples. *TPLP*, 16(5-6):834–848.

[LeCun *et al.* (1999)] LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

[Lee (2013)] Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2.

[Lenka *et al.* (2015)] Lenka, A., Naduthota, R., Jha, M., R, R. P., Prajapati, A., Jhunjhunwala, K., Saini, J., Yadav, R., Bharath, R., and Pal, P. (2015). Freezing of gait in parkinsonâĂŹs disease is associated with altered functional brain connectivity. *Parkinsonism & Related Disorders*, 24:100–106. Elsevier.

[Lenth *et al.* (2016)] Lenth, R. V. *et al.* (2016). Least-squares means: the r package lsmeans. *Journal of statistical software*, 69(1):1–33.

[Leofante *et al.* (2018)] Leofante, F., Narodytska, N., Pulina, L., and Tacchella, A. (2018). Automated verification of neural networks: Advances, challenges and perspectives. *CoRR*, abs/1805.09938.

[Leone and Ricca (2015)] Leone, N. and Ricca, F. (2015). Answer set programming: A tour from the basics to advanced development tools and industrial applications. In *Web Reasoning and Rule Systems - 9th International Conference, RR 2015, Berlin, Germany, August 4-5, 2015, Proceedings*, Lecture Notes in Computer Science (LNCS), pages 308–326. Springer Verlag.

[Leray *et al.* (2016)] Leray, E., Moreau, T., Fromont, A., and Edan, G. (2016). Epidemiology of multiple sclerosis. *Revue neurologique*, 172(1):3–13.

[Lierler and Susman (2017)] Lierler, Y. and Susman, B. (2017). On relation between constraint answer set programming and satisfiability modulo theories. *TPLP*, 17(4):559–590.

[Lifschitz (1999)] Lifschitz, V. (1999). Answer Set Planning. In Schreye, D. D., editor, *Proceedings of the 16th International Conference on Logic Programming (ICLP'99)*, pages 23–37, Las Cruces, New Mexico, USA. The MIT Press.

[Lim *et al.* (2015)] Lim, G., Cheng, Y., Hsu, W., and Lee, M. L. (2015). Integrated optic disc and cup segmentation with deep learning. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 162–169. IEEE.

[Lonc and Truszczynski (2006)] Lonc, Z. and Truszczynski, M. (2006). Computing minimal models, stable models and answer sets. *TPLP*, 6(4):395–449.

[Lublin *et al.* (2014)] Lublin, F. D., Reingold, S. C., Cohen, J. A., Cutter, G. R., Sørensen, P. S., Thompson, A. J., Wolinsky, J. S., Balcer, L. J., Banwell, B., Barkhof, F., *et al.* (2014). Defining the clinical course of multiple sclerosis: the 2013 revisions. *Neurology*, 83(3):278–286.

[Lucchinetti *et al.* (2000)] Lucchinetti, C., Bruck, W., Parisi, J., Scheithauer, B., Rodriguez, M., Lassman, H., *et al.* (2000). Heterogeneity of multiple sclerosis lesions: implications for the pathogenesis of demyelination. *Annals of neurology*, 47(6):707–717.

[Maier *et al.* (2010)] Maier, S. E., Sun, Y., and Mulkern, R. V. (2010). Diffusion imaging of brain tumors. *NMR in biomedicine*, 23(7):849–864.

[Maleki *et al.* (2012)] Maleki, M., Teshnehlab, M., and Nabavi, M. (2012). Diagnosis of multiple sclerosis (ms) using convolutional neural network (cnn) from mris. *Global Journal of Medicinal Plant Research*, 1(1):50–54.

[Manessi *et al.* (2017)] Manessi, F., Rozza, A., and Manzo, M. (2017). Dynamic graph convolutional networks. *arXiv preprint arXiv:1704.06199*.

[Mangin *et al.* (2013)] Mangin, J.-F., Fillard, P., Cointepas, Y., Le Bihan, D., Frouin, V., and Poupon, C. (2013). Toward global tractography. *NeuroImage*, 80:290–296.

[Manna *et al.* (2015)] Manna, M., Ricca, F., and Terracina, G. (2015). Taming primary key violations to query large inconsistent data via ASP. *Theory and Practice of Logic Programming (TPLP).. Cambridge University Press, UK.*, 15 (4-5):696–710.

[Maratea *et al.* (2014)] Maratea, M., Pulina, L., and Ricca, F. (2014). A multi-engine approach to answer-set programming. *Theory and Practice of Logic Programming*, 14(6):841–868.

[Marek and Truszczyński (1999)] Marek, V. W. and Truszczyński, M. (1999). Stable Models and an Alternative Logic Programming Paradigm. In Apt, K. R., Marek, V. W., Truszczyński, M., and Warren, D. S., editors, *The Logic Programming Paradigm – A 25-Year Perspective*, pages 375–398. Springer Verlag.

[McCulloch and Pitts (1943)] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

[McDonald *et al.* (2001)] McDonald, W. I., Compston, A., Edan, G., Goodkin, D., Hartung, H.-P., Lublin, F. D., McFarland, H. F., Paty, D. W., Polman, C. H., Reingold, S. C., *et al.* (2001). Recommended diagnostic criteria for multiple sclerosis: guidelines from the international panel on the diagnosis of multiple sclerosis. *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, 50(1):121–127.

[Medsker and Jain (1999)] Medsker, L. and Jain, L. C. (1999). *Recurrent neural networks: design and applications*. CRC press.

[Mellarkod *et al.* (2008)] Mellarkod, V. S., Gelfond, M., and Zhang, Y. (2008). Integrating answer set programming and constraint logic programming. *Ann. Math. Artif. Intell.*, 53(1-4):251–287.

[Miller *et al.* (2005)a] Miller, D., Barkhof, F., Montalban, X., Thompson, A., and Filippi, M. (2005a). Clinically isolated syndromes suggestive of multiple sclerosis, part 1: natural history, pathogenesis, diagnosis, and prognosis. *The Lancet Neurology*, 4(5):281–288.

[Miller *et al.* (2005)b] Miller, D., Barkhof, F., Montalban, X., Thompson, A., and Filippi, M. (2005b). Clinically isolated syndromes suggestive of multiple sclerosis, part 2: non-conventional mri, recovery processes, and management. *The Lancet Neurology*, 4(6):341–348.

[Mitchell *et al.* (1997)] Mitchell, T. M. *et al.* (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.

[Mori *et al.* (1999)] Mori, S., Crain, B. J., Chacko, V. P., and Van Zijl, P. (1999). Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of neurology*, 45(2):265–269.

[Mukherjee *et al.* (2008)] Mukherjee, P., Berman, J., Chung, S., Hess, C., and Henry, R. (2008). Diffusion tensor mr imaging and fiber tractography: theoretic underpinnings. *American journal of neuroradiology*, 29(4):632–641.

[Nesterov (1983)] Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady AN SSSR (translated as Soviet. Math. Docl.)*, 269:543–547.

[Newman (2002)] Newman, M. E. J. (2002). Assortative mixing in networks. *Phys. Rev. Lett.*, 89:208701.

[Newman and Girvan (2004)] Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E. 69:026113*.

[Nickles and Mileo (2014)] Nickles, M. and Mileo, A. (2014). Web stream reasoning using probabilistic answer set programming. In *RR*, volume 8741 of *LNCS*, pages 197–205. Springer.

[Nie *et al.* (2017)] Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q., and Shen, D. (2017). Medical image synthesis with context-aware generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 417–425. Springer.

[Niemelä (1999)] Niemelä, I. (1999). Logic Programming with Stable Model Semantics as Constraint Programming Paradigm. *Annals of Mathematics and Artificial Intelligence*, 25(3–4):241–273.

[Noseworthy *et al.* (2000)] Noseworthy, J. H., Lucchinetti, C., Rodriguez, M., and Weinshenker, B. G. (2000). Multiple sclerosis. *New England Journal of Medicine*, 343(13):938–952. PMID: 11006371.

[O'Donnell and Westin (2011)] O'Donnell, L. J. and Westin, C.-F. (2011). An introduction to diffusion tensor image analysis. *Neurosurgery Clinics*, 22(2):185–196.

[Pallawala *et al.* (2004)] Pallawala, P., Hsu, W., Lee, M. L., and Eong, K.-G. A. (2004). Automated optic disc localization and contour detection using ellipse fitting and wavelet transform. In *European conference on computer vision*, pages 139–151. Springer.

[Pan and Yang (2009)] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[Pasa and Sperduti (2014)] Pasa, L. and Sperduti, A. (2014). Pre-training of recurrent neural networks via linear autoencoders. In *Advances in Neural Information Processing Systems*, pages 3572–3580.

[Perozzi *et al.* (2014)] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

[Petersen (2004)] Petersen, R. (2004). Mild cognitive impairment as a diagnostic entity. *Journal of internal medicine*, 256(3):183–194. Wiley Online Library.

[Phang *et al.* (2019)] Phang, C.-R., Ting, C.-M., Noman, F., and Ombao, H. (2019). Classification of eeg-based brain connectivity networks in schizophrenia using a multi-domain connectome convolutional neural network. *arXiv preprint arXiv:1903.08858*.

[Polman *et al.* (2011)] Polman, C. H., Reingold, S. C., Banwell, B., Clanet, M., Cohen, J. A., Filippi, M., Fujihara, K., Havrdova, E., Hutchinson, M., Kappos, L., *et al.* (2011). Diagnostic criteria for multiple sclerosis: 2010 revisions to the mcdonald criteria. *Annals of neurology*, 69(2):292–302.

[Powers (2011)] Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

[Przymusinski (1991)] Przymusinski, T. C. (1991). Stable Semantics for Disjunctive Programs. 9:401–424.

[Pulina and Tacchella (2010)] Pulina, L. and Tacchella, A. (2010). An abstraction-refinement approach to verification of artificial neural networks. In Touili, T., Cook, B., and Jackson, P., editors, *Computer Aided Verification*, pages 243–257, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Qiu *et al.* (2018)] Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., and Tang, J. (2018). Deepinf: Social influence prediction with deep learning. In *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2110–2119. ACM. New York, NY, USA.

[Radford *et al.* (2015)] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[Redl (2016)] Redl, C. (2016). The dlvhex system for knowledge representation: recent advances (system description). *TPLP*, 16(5-6):866–883.

[Reyes *et al.* (2015)] Reyes, A. K., Caicedo, J. C., and Camargo, J. E. (2015). Fine-tuning deep convolutional networks for plant recognition. *CLEF (Working Notes)*, 1391.

[Ricca (2003)] Ricca, F. (2003). A java wrapper for DLV. In Vos, M. D. and Provetti, A., editors, *Answer Set Programming, Advances in Theory and Implementation, Proceedings of the 2nd Intl. ASP'03 Workshop, Messina, Italy, September 26-28, 2003*, volume 78 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Rovira *et al.* (2013)] Rovira, A., Auger, C., and Alonso, J. (2013). Magnetic resonance monitoring of lesion evolution in multiple sclerosis. *Therapeutic advances in neurological disorders*, 6(5):298–310.

[Rubinov and Sporns (2010)] Rubinov, M. and Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059–1069.

[Rumelhart *et al.* (1986)] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

[Russakovsky *et al.* (2015)] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.* (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[Salimans *et al.* (2016)] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242.

[Sbardella *et al.* (2013)] Sbardella, E., Tona, F., Petsas, N., and Pantano, P. (2013). Dti measurements in multiple sclerosis: evaluation of brain damage and clinical implications. *Multiple sclerosis international*, 2013.

[Scalfari *et al.* (2010)] Scalfari, A., Neuhaus, A., Degenhardt, A., Rice, G. P., Muraro, P. A., Daumer, M., and Ebers, G. C. (2010). The natural history of multiple sclerosis, a geographically based study 10: relapses and long-term disability. *Brain*, 133(7):1914–1929.

[Scarselli *et al.* (2009)] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80.

[Sermanet *et al.* (2013)] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

[Sevetlidis *et al.* (2016)] Sevetlidis, V., Giuffrida, M. V., and Tsaftaris, S. A. (2016). Whole image synthesis using a deep encoder-decoder network. In *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 127–137. Springer.

[Shashua and Hazan (2005)] Shashua, A. and Hazan, T. (2005). Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799. ACM.

[Shen and Lierler (2018)] Shen, D. and Lierler, Y. (2018). Smt-based constraint answer set solver EZSMT+ for non-tight programs. In Thielscher, M., Toni, F., and Wolter, F., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.*, pages 67–71. AAAI Press.

[Shovon *et al.* (2017)] Shovon, M. H. I., Nandagopal, N., Vijayalakshmi, R., Du, J. T., and Cocks, B. (2017). Directed connectivity analysis of functional brain networks during cognitive activity using transfer entropy. *Neural Processing Letters*, 45(3):807–824.

[Simonyan *et al.* (2013)] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

[Sinthanayothin *et al.* (1999)] Sinthanayothin, C., Boyce, J. F., Cook, H. L., and Williamson, T. H. (1999). Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images. *British journal of ophthalmology*, 83(8):902–910.

[Soldán *et al.* (2015)] Soldán, M. M. P., Novotna, M., Zeid, N. A., Kale, N., Tutuncu, M., Crusan, D. J., Atkinson, E. J., Siva, A., Keegan, B. M., Pirko, I., *et al.* (2015). Relapses and disability accumulation in progressive multiple sclerosis. *Neurology*, 84(1):81–88.

[Sperduti (2013)] Sperduti, A. (2013). Linear autoencoder networks for structured data. In *the 9th International Workshop on Neural-Symbolic Learning and Reasoning*.

[Sporns (2013)] Sporns, O. (2013). Structure and function of complex brain networks. *Dialogues in clinical neuroscience*, 15(3):247.

[Sporns *et al.* (2005)] Sporns, O., Tononi, G., and Kotter, R. (2005). *The human connectome: a structural description of the human brain*, volume 1, pages 245–251. PLoS Comput Biol.

[Sporns *et al.* (2007)] Sporns, O., Honey, C., and Kotter, R. (2007). Identification and classification of hubs in brain networks. *PLoS ONE*.

[Stamile *et al.* (2015)] Stamile, C., Kocevar, G., Hannoun, S., Durand-Dubief, F., and Sappey-Marinier, D. (2015). A graph based classification method for multiple sclerosis clinical forms using support vector machine. In *Machine Learning Meets Medical Imaging - First International Workshop, MLMMI 2015, Held in Conjunction with ICML 2015, Lille, France, July 11, 2015. Revised Selected Papers*, pages 57–64.

[Stein *et al.* (2007)] Stein, M., Simmons, A., Feinstein, J., and Paulus, M. (2007). Increased amygdala and insula activation during emotion processing in anxiety-prone subjects. *The American journal of psychiatry*, 164 2:318–327. Psychiatry Online.

[Stejskal and Tanner (1965)] Stejskal, E. O. and Tanner, J. E. (1965). Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient. *The journal of chemical physics*, 42(1):288–292.

[Stewart *et al.* (2016)] Stewart, R., Andriluka, M., and Ng, A. Y. (2016). End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333.

[Strang *et al.* (2018)] Strang, A., Haynes, O., Cahill, N. D., and Narayan, D. A. (2018). Generalized relationships between characteristic path length, efficiency, clustering coefficients, and density. *Social Network Analysis and Mining*, 8(1):14.

[Szegedy *et al.* (2016)a] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016a). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[Szegedy *et al.* (2016)b] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016b). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[Tamura *et al.* (1988)] Tamura, S., Okamoto, Y., and Yanashima, K. (1988). Zero-crossing interval correction in tracing eye-fundus blood vessels. *Pattern recognition*, 21(3):227–233.

[Terracina *et al.* (2008)] Terracina, G., Leone, N., Lio, V., and Panetta, C. (2008). Experimenting with recursive queries in database and logic programming systems. *Theory and Practice of Logic Programming (TPLP). Available on-line at http://arxiv.org/abs/0704.3157. Cambridge University Press, UK*, 8(2):129–165.

[Thompson *et al.* (2018)] Thompson, A. J., Banwell, B. L., Barkhof, F., Carroll, W. M., Coetzee, T., Comi, G., Correale, J., Fazekas, F., Filippi, M., Freedman, M. S., *et al.* (2018). Diagnosis of multiple sclerosis: 2017 revisions of the mcdonald criteria. *The Lancet Neurology*, 17(2):162–173.

[Thomsen *et al.* (1987)] Thomsen, C., Henriksen, O., and Ring, P. (1987). In vivo measurement of water self diffusion in the human brain by magnetic resonance imaging. *Acta Radiologica*, 28(3):353–361.

[Tieleman and Hinton (2012)] Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

[Tournier *et al.* (2012)] Tournier, J.-D., Calamante, F., and Connelly, A. (2012). MRtrix: Diffusion tractography in crossing fiber regions. *Int. J. Imaging Systems and Technology*, 22(1):53–66.

[Towell and Shavlik (1993)] Towell, G. G. and Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13:71–101.

[Tucker (1966)] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

[Upadhyaya (2013)] Upadhyaya, S. R. (2013). Parallel approaches to machine learning - a comprehensive survey. *Journal of Parallel and Distributed Computing*, 73(3):284–292.

[Valverde *et al.* (2017)] Valverde, S., Cabezas, M., Roura, E., González-Villà, S., Pareto, D., Vilanova, J. C., Ramio-Torrenta, L., Rovira, À., Oliver, A., and Lladó, X. (2017). Improving automated multiple sclerosis lesion segmentation with a cascaded 3d convolutional neural network approach. *NeuroImage*, 155:159–168.

[Wang *et al.* (2013)] Wang, J., Jebara, T., and Chang, S.-F. (2013). Semi-supervised learning using greedy max-cut. *Journal of Machine Learning Research*, 14(Mar):771–800.

[Wang *et al.* (2018)] Wang, S.-H., Tang, C., Sun, J., Yang, J., Huang, C., Phillips, P., and Zhang, Y.-D. (2018). Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling. *Frontiers in neuroscience*, 12.

[Weiss *et al.* (2016)] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):9.

[Wilcoxon (1945)] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83.

[Wong *et al.* (2004)] Wong, T. Y., Knudtson, M. D., Klein, R., Klein, B. E., Meuer, S. M., and Hubbard, L. D. (2004). Computer-assisted measurement of retinal vessel diameters in the beaver dam eye study: methodology, correlation between eyes, and effect of refractive errors. *Ophthalmology*, 111(6):1183–1190.

[Wu *et al.* (2019)] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. (2019). A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596.

[Wu and Yap (2006)] Wu, K. and Yap, K.-H. (2006). Fuzzy svm for content-based image retrieval: a pseudo-label support vector machine framework. *IEEE Computational Intelligence Magazine*, 1(2):10–16.

[Xu *et al.* (2013)] Xu, Y., Lin, S., Wong, D. W. K., Liu, J., and Xu, D. (2013). Efficient reconstruction-based optic cup localization for glaucoma screening. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 445–452. Springer.

[Yin *et al.* (2012)] Yin, F., Liu, J., Wong, D. W. K., Tan, N. M., Cheung, C., Baskaran, M., Aung, T., and Wong, T. Y. (2012). Automated segmentation of optic disc and optic cup in fundus images for glaucoma diagnosis. In *2012 25th IEEE international symposium on computer-based medical systems (CBMS)*, pages 1–6. IEEE.

[Zhang *et al.* (2017)a] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017a). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915.

[Zhang *et al.* (2017)b] Zhang, Q., Cao, R., Zhang, S., Edmonds, M., Wu, Y. N., and Zhu, S. (2017b). Interactively transferring CNN patterns for part localization. *CoRR*, abs/1708.01783.

[Zhang *et al.* (2018)a] Zhang, Q., Wu, Y. N., and Zhu, S. (2018a). Interpretable convolutional neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8827–8836. IEEE Computer Society.

[Zhang *et al.* (2018)b] Zhang, Q., Yang, Y., Wu, Y. N., and Zhu, S. (2018b). Interpreting cnns via decision trees. *CoRR*, abs/1802.00121.

[Zhang *et al.* (2018)c] Zhang, Y.-D., Pan, C., Sun, J., and Tang, C. (2018c). Multiple sclerosis identification by convolutional neural network with dropout and parametric relu. *Journal of computational science*, 28:1–10.

[Zhang and Zhu (2018)] Zhang, Q. and Zhu, S. (2018). Visual interpretability for deep learning: a survey. *Frontiers of IT & EE*, 19(1):27–39.

[Zhao *et al.* (2014)] Zhao, M., Chan, R. H., Chow, T. W., and Tang, P. (2014). Compact graph based semi-supervised learning for medical diagnosis in alzheimer's disease. *IEEE signal processing letters*, 21(10):1192–1196.

[Zhu *et al.* (2003)] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.

[Zou *et al.* (2004)] Zou, K. H., Warfield, S. K., Bharatha, A., Tempany, C. M., Kaus, M. R., Haker, S. J., Wells III, W. M., Jolesz, F. A., and Kikinis, R. (2004). Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology*, 11(2):178–189.

[Zurita *et al.* (2018)a] Zurita, M., Montalba, C., Labbé, T., Cruz, J. P., da Rocha, J. D., Tejos, C., Ciampi, E., Cárcamo, C., Sitaram, R., and Uribe, S. (2018a). Characterization of relapsing-remitting multiple sclerosis patients using support vector machine classifications of functional and diffusion mri data. *NeuroImage: Clinical*, 20:724–730.

[Zurita *et al.* (2018)b] Zurita, M., Montalba, C., Labbé, T., Cruz, J. P., da Rocha, J. D., Tejos, C., Ciampi, E., Cárcamo, C., Sitaram, R., and Uribe, S. (2018b). Characterization of relapsing-remitting multiple sclerosis patients using support vector machine classifications of functional and diffusion mri data. *NeuroImage: Clinical*, 20:724–730.

[van den Heuvel and Sporns (2011)] van den Heuvel, M. and Sporns, O. (2011). *Rich-club organization of the human connectome*, volume 41, pages 15775–15786. Neuroscience.

[Kolda and Bader (2009)] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

List of Publications

Journal Papers

1. **A. Marzullo**, G. Kocevar, C. Stamile, F. Calimeri, G. Terracina, F. Durand-Dubief, & D. Sappey-Marinier, “Classification of Multiple Sclerosis Clinical Profiles via Graph Convolutional Neural Networks," Frontiers in Neuroscience, vol. 13, 2019.

2. F. Calimeri, F. Cauteruccio, L. Cinelli, **A. Marzullo**, C. Stamile, G. Terracina, F. Durand-Dubief, D. Sappey-Marinier, “A Logic-Based Framework Leveraging Neural Networks for Studying the Evolution of Neurological Disorders," Forthcoming in Theory and Practice of Logic Programming, 2019.

Authors in alphabetic order contributed equally to these work

Peer Reviewed Conference Papers

1. G. Melissari, **A. Marzullo**, C. Stamile, F. Calimeri, F. Durand-Dubief, & D. Sappey-Marinier, “Inducing Clinical Course Variations in Multiple Sclerosis White Matter Networks," In Intelligent Computing-Proceedings of the Computing Conference, pp. 900-917. Springer, Cham., 2019.

2. **A. Marzullo**, G. Kocevar, C. Stamile, F. Calimeri, G. Terracina, F. Durand-Dubief, & D. Sappey-Marinier, “Prediction of Multiple Sclerosis Patient Disability from Structural Connectivity Using Convolutional Neural Networks," Forthcoming in 41th tInternational Engineering in Medicine and Biology Conference (EMBC), 2019.

3. F. Calimeri, P. Bruno, **A. Marzullo**, “Classification and survival prediction in Diffuse Large B-cell Lymphoma by gene expression profiling," In International Conference on Machine Learning, Optimization, and Data Science, pp. 166-178. Springer, Cham., 2018.

4. F. Calimeri, F. Cauteruccio, **A. Marzullo**, C. Stamile, G. Terracina, “Mixing Logic Programming and Neural Networks to Support Neurological Disorders Analysis," 2nd International Joint Conference on Rules and Reasoning (RuleML+RR 2018), Luxembourg, 2018, Lecture Notes in Computer Science, Springer-Verlag.

5. F. Calimeri, **A. Marzullo**, C. Stamile, G. Terracina, “Graph Based Neural Networks for Automatic Classification of Multiple Sclerosis Clinical Courses," forthcoming in European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2018.

6. F. Calimeri, **A. Marzullo**, C. Stamile, G. Terracina, “Biomedical Data Augmentation using Generative Adversarial Neural Networks," 26th International Conference on Artificial Neural Networks (ICANN), 2017.

7. **A. Marzullo**, C. Stamile, G. Terracina, F. Calimeri, S. Van Huffel, “A tensor-based mutation operator for Neuroevolution of Augmenting Topologies (NEAT)," IEEE Congress of Evolutionary Computation (CEC), pp. 681-687, 2017

8. F. Calimeri, **A. Marzullo**, C. Stamile, G.Terracina, “Optic Disc Detection using Fine Tuned Convolutional Neural Networks," In IEEE 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp. 69-75, 2016.

9. F. Calimeri, **A. Marzullo**, C. Stamile, G.Terracina, “Blood Vessel Segmentation using Hypercube NeuroEvolution of Augmenting Topologies (HyperNeat)," In Italian Workshop on Neural Nets, pp. 173-183. Springer, Cham, 2017.

10. F. Calimeri, M. Caracciolo, **A. Marzullo**, C. Stamile, “BioHIPI: Biomedical Hadoop Image Processing Interface," The Third International Conference on Machine Learning, Optimization and Big Data (MOD), 2017.

Authors in alphabetic order contributed equally to these work

Peer Reviewed Conference Abstracts

1. **A. Marzullo**, C. Stamile, G. Kocevar, F. Calimeri, G. Terracina, F. Durand-Dubief, D. Sappey-Marinier, “Exploiting Heterogeneous Data for Automatic Classification of Multiple Sclerosis Clinical Forms through Neural Networks," ISMRM, Montréal, QC, Canada, 2019.

2. **A. Marzullo**, C. Stamile, G. Kocevar, F. Calimeri, G. Terracina, F. Durand-Dubief, D. Sappey-Marinier, “Exploitation de données hétérogènes d’IRM de diffusion pour la classification automatique des formes cliniques de sclérose en plaques par une nouvelle approche de réseaux de neurones," SFRMBM, Strasbourg, France, 2019.