A Hybrid Method for Tractography in Neurosurgery Using Artificial Neural Networks and Path Search Algorithms

Mateusz Korycinski

A thesis presented for the degree of
Doctor of Philosophy

Supervisor:

Prof. Ewa Niewiadomska - Szynkiewicz, PhD, DSc

Co-supervisor:

Konrad Andrzej Ciecierski, PhD

Department of Artificial Intelligence Applications in Medicine (AI-MED)

Center of Research and Development

Scientific and Academic Computer Network National Research Institute

Warsaw, 2023

Hybrydowa metoda traktografii w neurochirurgii wykorzystująca sztuczne sieci neuronowe i algorytmy wyznaczania najkrótszej ścieżki

Mateusz Koryciński

Rozprawa doktorska

Promotor:

prof. dr hab. inż. Ewa Niewiadomska - Szynkiewicz

Promotor pomocniczy:

dr inż. Konrad Andrzej Ciecierski

Zakład Zastosowań Sztucznej Inteligencji w Medycynie (AI-MED)

Centrum Badań i Rozwoju

Naukowa i Akademicka Sieć Komputerowa Państwowy Instytut Badawczy

Warszawa, 2023

Abstract

Neurosurgery is one of the youngest and most demanding fields of medicine. It concerns surgical interventions within the central nervous system - the system that coordinates and influences the activity of all parts of the human body. Most neurosurgeries focus on the brain itself. It is a delicate organ of microscopic cellular structures that require extreme precision during the intervention. Despite the risk of impairments, such surgeries are necessary for treating many pathologies, like glioblastoma.

The development of imaging techniques in recent decades, especially MRI, helped profoundly to map important structures non-invasively. Specific modalities, like fMRI or DTI, allow visualization of the near-precise location of functional regions and neural pathways, respectively. These experiments result in a large amount of data that has to be thoroughly analyzed. This task puts much work on radiologists, especially when not equipped with additional tools, like image recognition algorithms. In recent years, artificial intelligence has assisted radiologists and neurosurgeons in their work through decision support systems. They are developed to analyze experimental results landing advice for trained physicians. They can help better plan and perform neurosurgeries by integrating with live neuronavigation systems.

This thesis introduces a novel method for analyzing diffusion data obtained from MRI experiments. A presented hybrid technique comprises a neural network for diffusion data analysis and a path search algorithm computing the topology of nerve fibers based on the analyzed data. This information is suitable for multiple applications, including estimating the topology of the neural pathways near the surgical field or producing maps of connections between different brain functional areas. Neurosurgeons and radiologists can use such knowledge for pre-operative planning and intra-operative navigation.

Streszczenie

Neurochirurgia to jedna z najmłodszych i najbardziej wymagających dziedzin medycyny. Dotyczy interwencji chirurgicznych w obrębie ośrodkowego układu nerwowego - systemu, który koordynuje i wpływa na aktywność wszystkich części ludzkiego ciała. Większość operacji neurochirurgicznych skupia się na samym mózgu. Jest to delikatny organ o mikroskopijnych strukturach komórkowych, które wymagają niezwykłej precyzji podczas interwencji. Pomimo ryzyka upośledzenia, takie operacje są niezbędne w leczeniu wielu patologii, jak np. nowotwory mózgu (glioblastoma).

W ostatnich dekadach rozwój technik obrazowania, zwłaszcza MRI, umożliwił nieinwazyjne mapowanie ważnych struktur mózgowych. Specyficzne rodzaje technik, takie jak fMRI czy DTI, pozwalają na wizualizację niemal dokładnej lokalizacji regionów funkcjonalnych i ścieżek neuronowych. W wyniku tych eksperymentów powstaje duża ilość danych, które muszą być dokładnie przeanalizowane. Zadanie to nakłada wiele pracy na radiologów, zwłaszcza gdy nie są oni wyposażeni w dodatkowe narzędzia, takie jak algorytmy rozpoznawania obrazów. W ostatnich latach sztuczna inteligencja wspomaga radiologów i neurochirurgów w ich pracy poprzez systemy wspomagania decyzji. Wspierają one analizę wyników eksperymentów dając wskazówki wyszkolonym lekarzom. Systemy dedykowane neurochirurgii pomagają lepiej planować i wykonywać operacje neurochirurgiczne. Niektóre rozwiązania integrują się również z systemami neuronawigacji na żywo.

W niniejszej rozprawie prezentowana jest nowatorską metoda analizy danych dyfuzyjnych uzyskanych w wyniku eksperymentów MRI. Przedstawiona technika hybrydowa łączy sieci neuronowe do analizy danych dyfuzyjnych z algorytmem wyszukiwania ścieżek wyznaczającym topologię włókien nerwowych na podstawie analizowanych danych. Proponowana metoda może znaleźć wiele zastosowań, w tym do określania topologii ścieżek neuronowych w pobliżu pola operacyjnego lub tworzenia map połączeń pomiędzy różnymi obszarami funkcjonalnymi mózgu. Neurochirurdzy i radiolodzy mogą wykorzystać pozyskiwaną dzięki niej wiedzę do planowania przedoperacyjnego i nawigacji śródoperacyjnej.

I dedicate this thesis to my beloved wife for her constant support on this journey and unconditional love.

I love you very much.

Acknowledgments

I would like to express my gratitude to my Ph.D. supervisor, Professor Ewa Niewiadomska-Szynkiewicz for her help and guidance throughout the years of my doctoral work. Her insights and guidance cannot be overstated.

I am very grateful to my co-supervisor, Konrad A. Ciecierski, PhD, for his help and patience and many fruitful discussions during this project.

Without encouragement from my supervisors to pursue an academic path this thesis would not be completed.

I would like to thank the medical staff from the National Oncology Institute in Warsaw, Poland. They have introduced me to the peculiarities of their neurosurgeons' work, especially in tumor mass resection surgeries.

My sincere gratitude goes to my colleagues from the Department of Artificial Intelligence Applications in Medicine (AI-MED) and the whole R&D division of the Scientific and Academic Computer Network National Research Institute (NASK PIB). The atmosphere at the institute was very stimulating for conducting research in various areas.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Definition**
5TT	Five-tissue-type segmented tissue image
ANN	Artificial Neural Network
BA	Brodmann Area
BCE	Binary Cross Entropy
CDSS	Clinical Decision Support System
CE	Cross Entropy
CPDG	Closest Peak Direction Getter
CNN	Convolutional Neural Network
CSA	Constant Solid Angle
CSD	Constrained Spherical Deconvolution
CT	Computed Tomography
DDSS	Diagnosis Decision Support System
DMDG	Deterministic Maximum Direction Getter
DTI	Diffusion Tensor Imaging
DWI	Diffusion Weighted Imaging
FC	Fully-connected layer
FFNN	Feedforward neural network
FLIRT	FMRIB's Linear Image Registration Tool
FOD	Fiber Orientation Distribution
fODF	Fiber Orientation Distribution Function
fMRI	Functional Magnetic Resonance Imaging
GFA	Generalized Fractional Anisotropy
GMWMI	Gray matter-white matter interface
HCP	Human Connectome Project
ICBM	International Consortium of Brain Mapping
LGN	Lateral Geniculate Nucleus
LSTM	Long Short-Term Memory model
MGH	Massachusetts General Hospital
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
ODF	Orientation Distribution Function
PET	Positron Emission Tomography
ReLU	Rectified Linear Unit
RF	Radio Frequency
RNN	Recurrent Neural Network
RT	Repetition Time
SDT	Sharpening Deconvolution Transform
SH	Spherical Harmonic
SLF	Superior Longitudinal Fasciculus
SMA	Simple Moving Average
Tanh	Hyperbolic Tangent
TE	Time Echo
WM	White matter

List of Mathematical Symbols

Symbol	**Definition**
y	Neuron or layer output
$\hat{y}$	Trained model output or prediction
$\bar{y}$	Data label, ground truth
x	Input features
φ	Single neuron output
ϱ	Cost of the path
k, K	Number of classes in multi-class prediction
$\mathscr{L}$	Loss function
$L, \quad l$	Number of layers
m, M	Batch size
w, W	Neural network weight, weights matrix
Ψ	Threshold for a voxel value
s	Slope parameter for the negative side of LeakyReLU function
v, v_n, v_t	Graph node, n-th graph node, target graph node
z	Smoothing window size
ζ	Total length of the path, representing a single streamline (neural fiber)

1 Introduction

1.1 Problem statement

Gliomas are one of the most common primary brain tumors. They are classified into four histological grades, with high grades, i.e. III and IV (glioblastoma) being the most frequent [1]. Unfortunately, in the case of high grades, the prognosis for a patient is poor, given the high invasiveness and ability to infiltrate neighboring neural tissue [2]. In addition to radiation therapy, resecting the tumor mass can improve treatment outcomes.

Even though such surgeries are beneficial, they are invasive, particularly when the tumor infiltrates significant cortical regions. Damaging these cortical structures during surgery can impair the patient's motor or cognitive functions [3, 4]. In addition, it is essential to keep their connections intact. Damage to such can have similar effects as damaging cortex regions themselves. To partially solve this problem, resection can be performed in several surgeries. Only part of the mass is removed during the first surgery to keep important structures intact. Thanks to brain plasticity, the rest can be removed in consecutive attempts. It has been shown that such regions can shift slightly on the brain's cortex with time, allowing it to remove more infiltrated tissue [5].

A key aspect of successful treatment is good diagnostics. Good resolution in brain visualization in vivo implies a constant need to improve imaging techniques and image (signal) processing algorithms. Different modalities of Magnetic Resonance Imaging (MRI) can be used to examine the brain structure non-invasively [6]. The brain's anatomy can be investigated via T1 and T2 MRI sequences. In such imaging, however, only a limited set of features can be visualized (e.g. grey and white matter). Other MRI modalities can depict other features, such as functional MRI (fMRI), allowing visualization of functional cortex areas [7, 8]. Diffusion-weighted imaging (dwMRI) [9], particularly diffusion tensor imaging (DTI) [10], can project all the neural connections across various regions. The information provided by these modalities helps to build a comprehensive view of the structural and functional connections between different areas of the brain.

An important part of this view is the precise location of the white matter fibers. Their reconstruction based on the diffusion signal is addressed by tractography. Many methods have been developed to achieve this task. However, they provide a specific probability level. Moreover, insight from the trained radiology specialist is needed. Based on anatomical knowledge and expertise, such an expert can assess the topology of the nerve fibers with higher assurance than a single method. Besides precision being at stake, these methods usually require multiple steps of preprocessing and fine-tuning, which makes them hardly accessible to neurosurgeons.

An accurate understanding of the patient's brain connectivity is crucial when planning and performing neurosurgery. To make tractography more accessible to physicians specializing in areas other than radiology, a hybrid model based on an artificial neural network and a path search algorithm has been proposed. This model can effectively compute the location of nerve fibers with high probability using DTI data. Leveraging the artificial neural network for DTI data analysis reduces the amount of preprocessing steps required. With the successful implementation of this model, the system will be able to project the location of critical cortical regions based on functional MRI data and adjacent neural connections based on the tractography results, facilitating the planning and execution of the surgical intervention.

1.2 Motivation

Despite various methods for delineating nerve fibers, they face problems that make these solutions not readily adaptable for preoperative planning. Tractography methods require data cleaning before analysis. Diffusion data have noise and artifacts created by strong magnetic fields, and their presence can lead to suboptimal, or even erroneous, results. Moreover, these methods tend to produce many false positives. Hence, several tractography methods must be used to compute a fiber arrangement closely reflecting the true one. Existing artificial neural networks proposed for tractography tasks are trained for specific applications, making these methods challenging to adapt to similar tasks, such as searching for crossing points (joining and splitting fibers) and looking for fibers connecting two given functional regions. To properly predict and visualize the white matter near the surgical field, it is necessary to use several different tools, which creates additional work for the surgeon during preoperative planning.

1.3 Aims and thesis statement

Research carried out and described in this dissertation focused on designing an artificial neural network coupled with an effective path search algorithm. Using an artificial neural network for diffusion data processing will allow using data subjected only to preliminary preprocessing steps. In addition, processing the data in small batches allows training of the model on small datasets, given a single scan provides numerous training examples. Adapting a path search algorithm will allow for determining fibers near the surgical site, connecting different brain areas, and differentiating fiber crossing points. The hybrid model adds explainability as the artificial neural network computes the graph fed into the path search algorithm. Therefore, it is possible to analyze the data from which the fiber was computed. In summary, this thesis states the following:

Artificial neural network coupled with a path search algorithm is an effective and efficient method to compute tractograms.

The formulated thesis was proved by carrying out the following research tasks:

- development of a method for creating labels based on tractograms obtained by various techniques from diffusion data,
- designing and training of the artificial neural network (ANN) for diffusion data processing,
- implementing a modified variant of the A* algorithm for a path search in a graph build based on the ANN output,
- implementing an automated pipeline to compute tractograms on real data,
- carrying out a series of experiments, showing the results of the automated pipeline on real data.

An automated pipeline allowing to use hybrid method consisting of ANN coupled with a path search algorithm for tractography was implemented and named HyTract (HT). In the future, this method will be used in real-life preoperative planning as a part of the decision support system for neurosurgery.

1.4 Structure of the dissertation

This thesis is composed of ten chapters:

- The first chapter introduces the problem statement, motivation, aims, and a thesis statement regarding the hybrid model.
- The second chapter describes the medical background - the anatomy of the central nervous system, including the organisation of white matter within the brain. Gliomas are described, together with their treatment. This chapter also characterizes magnetic resonance imaging and preoperative planning with tractography as crucial steps before brain surgery.
- The third chapter focuses on artificial neural networks, describing the principles behind their design and training. This chapter concludes with an overview of available AI methods in medicine, with special emphasis on decision support systems.
- The fourth chapter describes path search algorithms.
- The fifth chapter describes related works in tractography, divided into mathematical models and machine learning approaches.
- The sixth chapter defines the hybrid method for tractography, HyTract. It characterizes the general overview, architecture of the neural network, and adaptation of an A* algorithm for path reconstruction.
- The seventh chapter describes the processing and analysis of the diffusion data. It contains a description of the usual preprocessing steps, as well as a description of how the labels for training and validation of the neural network were prepared.
- The eighth chapter describes the experimental setup and the dataset used for training and validating the HyTract ANN. The results of all experiments carried out are described. The chapter ends with the experimental results discussion.
- The ninth chapter focuses on the HyTract as a whole. Experimental setup and dataset for testing HyTract on real data are described, followed by experimental results. The chapter concludes with a results discussion.
- The tenth chapter concludes the thesis.

2 Medical background and challenges in modern neurosurgery

2.1 Introduction to Neurophysiology

2.1.1 Functional anatomy of the brain

The human nervous system is divided into peripheral (PNS) and central nervous systems (CNS). The peripheral nervous system involves cranial and spinal nerves, which are responsible for communicating all body parts with the CNS. The central nervous system has two main parts: the brain and the spinal cord. The brain resides in the cranial cavity of the skull, while the spinal cord is in the vertebral canal.

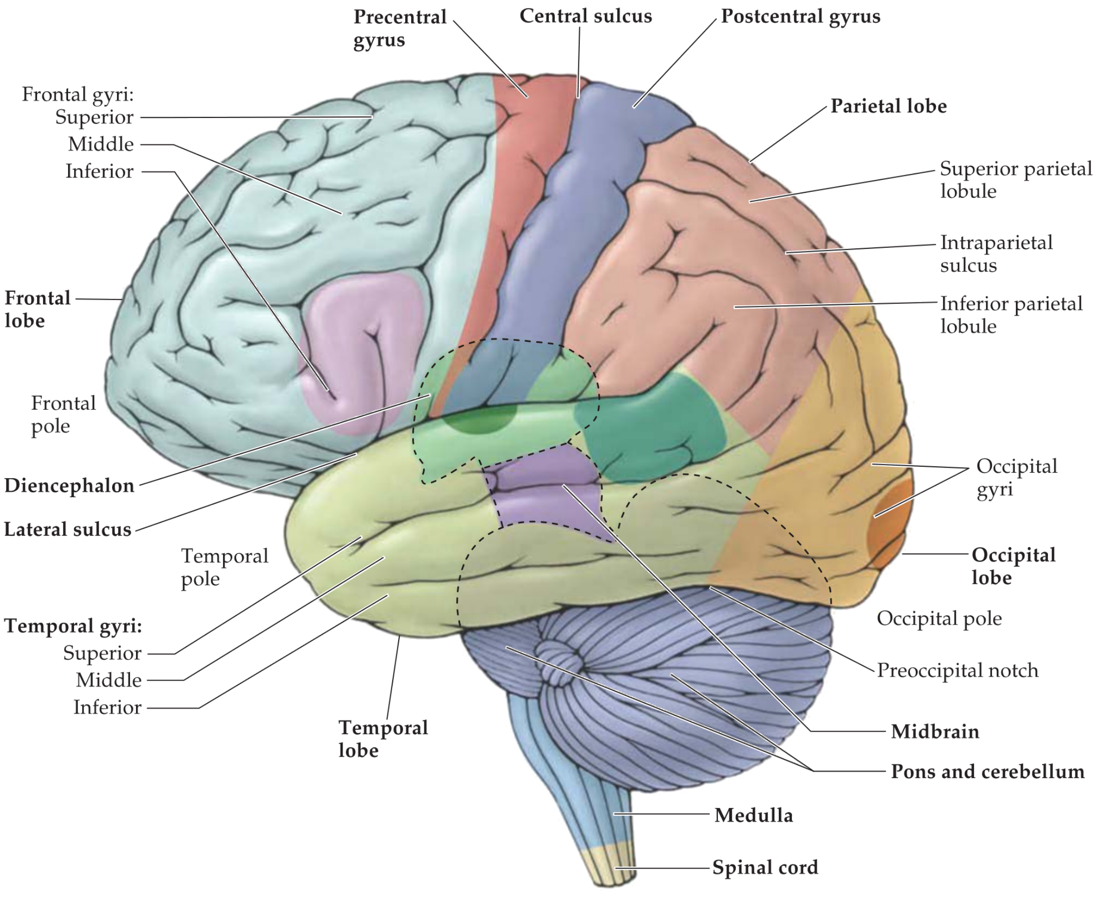


Figure 2.1: Anatomy of the brain [11]

The brain is subdivided into the medulla, pons, cerebellum, midbrain, diencephalon and two cerebral hemispheres (Fig. 2.1) [11]. The latter is the most highly developed part of the CNS. A long cleft-longitudinal fissure and the diencephalon separate the two hemispheres.

The brain's surface, the cerebral cortex, is highly convoluted, which is an evolutionary adaptation to accommodate greater surface area in the limited space of the skull. Elevated parts of convolutions are named gyri. They are separated by grooves called sulci or fissures. Beneath the brain's surface are many essential structures, including nerve fibers providing information transfer between different parts of the cortex.

Nervous tissue, building both PNS and CNS, has two major types of cells: neurons and glial cells. Neurons are responsible for the nervous system's primary function and produce electric potentials. Glial cells, on the other hand, do not take an active part in signalling but provide support and protection for neurons [12].

Neural cells consist of the cell body and extensions called processes. An axon is one of the processes that is always elongated and connects a neuron with its target (e.g. another neuron/neurons). A single neural cell always has only one axon, which can branch into axon terminals, allowing a neuron to connect with many target cells. Shorter processes are present in bigger numbers and are called dendrites. They connect with axons of other cells and are responsible for receiving input (Fig. 2.2). In hemispheres, neural cell bodies stacked in layers form a cerebral cortex. Their axons, organized in bundles, form nerve fibers. Due to the myelination of the axon (discussed in more detail in section 2.1.2) they have a whitish color, contrary to the cell bodies, which are grayish. Hence the cortex is made of gray matter with white matter below. This division is noticeable in medical imaging (Fig. 2.9) or on the specimens from the brain dissection.

Different parts of the cerebral cortex correspond to different functions. The surface is split into four lobes named after cranial bones overlying them: frontal, parietal, occipital and temporal. In 1909 German anatomist Korbinian Brodmann published maps of functional cortical areas in humans, monkeys, and other species [14]. He defined those regions based on the cytoarchitectural organization of neurons (Fig. 2.3). Since then, proposed areas have been debated and refined by the field, and nowadays, they are the source of coarse localization of the functional regions in the cortex.

Three of the specified areas were of particular interest to the topic of this thesis. Two of them, Brodmann area 44 (BA44, pars opercularis) and 45 (BA45, pars triangularis), are described as the Broca area (Fig. 2.3). This area is located in the frontal lobe of the dominant hemisphere (usually left) and is linked functionally with speech production. Pierre Paul Broca discovered this link by observing two patients who lost the ability to speak after

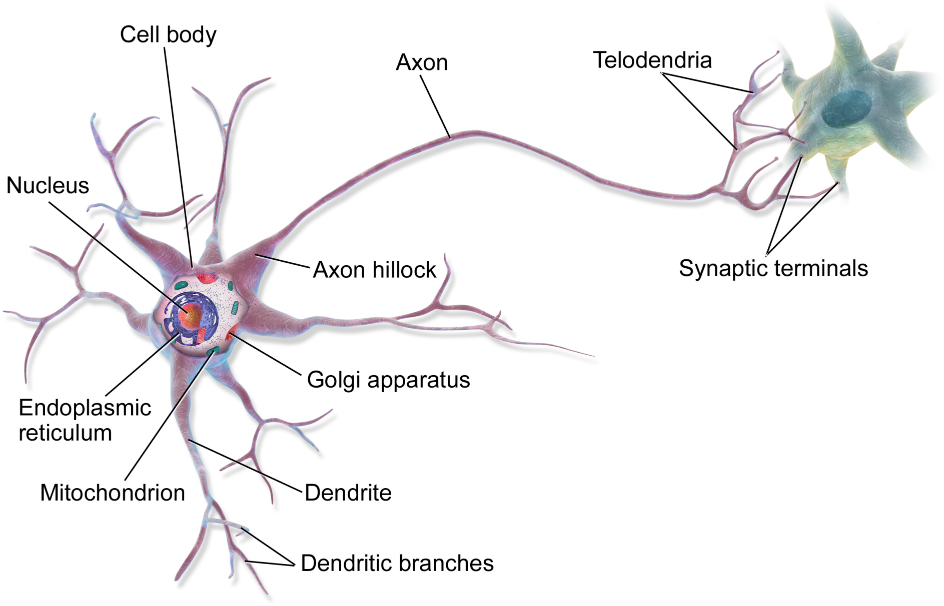


Figure 2.2: Neural cell [13]

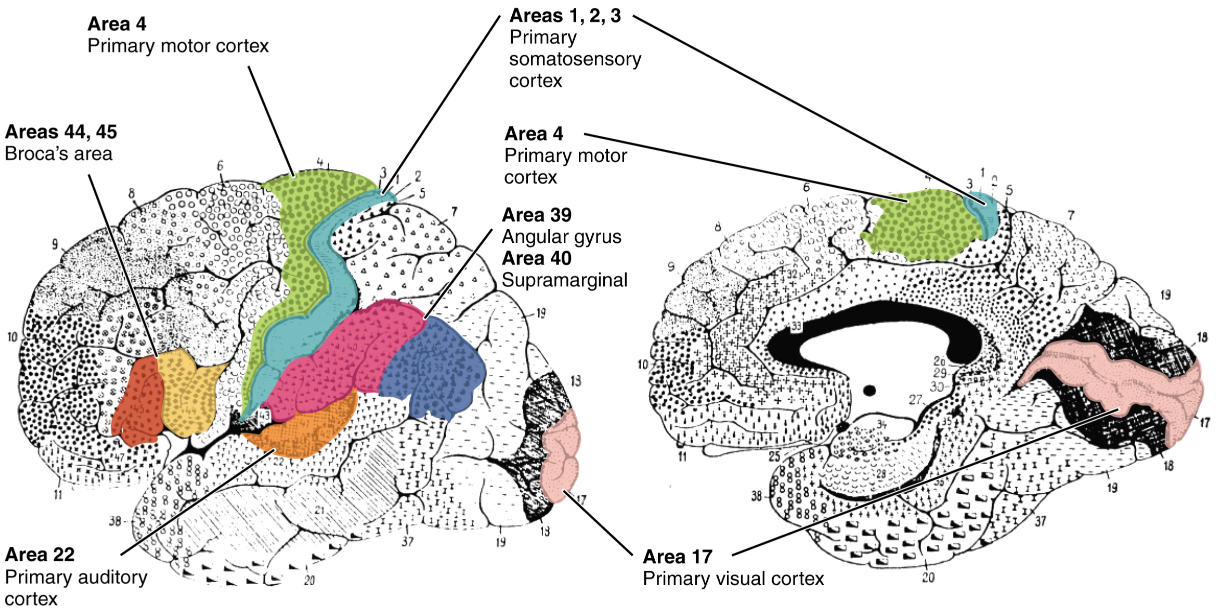
injury to pars triangularis (BA45). Since then, a deficit in language production is known as Broca's aphasia. Recent fMRI studies have shown that speech production can also be linked to other areas, described together as the Broca region [15].

Brodmann area 22, called Wernicke's area, is located in the superior temporal gyrus in the dominant hemisphere (Fig. 2.3). It is functionally and anatomically connected with Broca's area and is responsible for comprehending written and spoken language. Injury to that part of the brain results in fluent aphasia. In this case, a person connects words fluently; however, the sentences lack meaning.

Even though the localization of these areas have been established and debated through experiments throughout the years, their precise location differ among patients. Therefore it is essential to investigate their precise location before any surgical interventions involving nearby areas to avoid unnecessary damage leading to impairments.

2.1.2 White matter and neural circuits within the brain

The inner part of both hemispheres consists predominantly of white matter. It is formed by axons of the neural cells, which transfer the information as action potentials (Fig. 2.4). Information is always conducted from the cell body through the axon to the axon terminals and then to other neurons. Axons are tightly surrounded with myelin sheath made by Schwann cells and oligodendrocytes, two types of glial cells. It is rich in a fatty substance called myelin,



Brodmann's cytotechtonic map (1909):
Lateral surface

Brodmann's cytotechtonic map (1909):
Medial surface

Figure 2.3: Brodmann's map of the brain cortex [14]

giving the axons a whitish color. In addition to its protective function, this envelope allows action potentials to be conducted with increased velocity. It is possible due to constrictions on the surface, called the Nodes of Ranvier (Fig. 2.4). Instead of being carried along the entire length of the axon, the potentials jump between these constrictions.

Neural circuits are formed by connecting axon terminals with dendrites of other neural cells at the synapses. A schematic representation of the synapse is shown in Fig. 2.5. The neuron serving as a source of information is called a presynaptic neuron, and the neuron receiving information through dendrites is named a postsynaptic neuron. Most often,

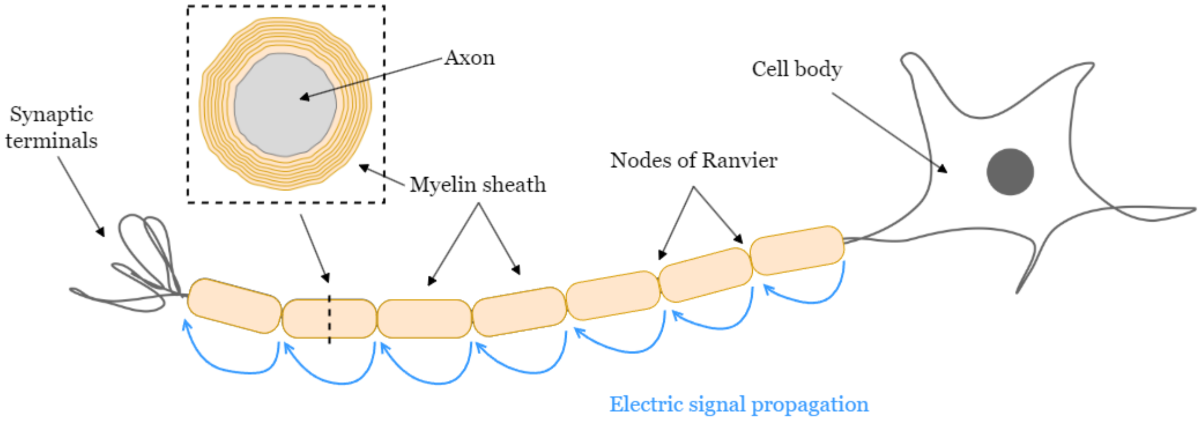


Figure 2.4: Schematic representation of an axon

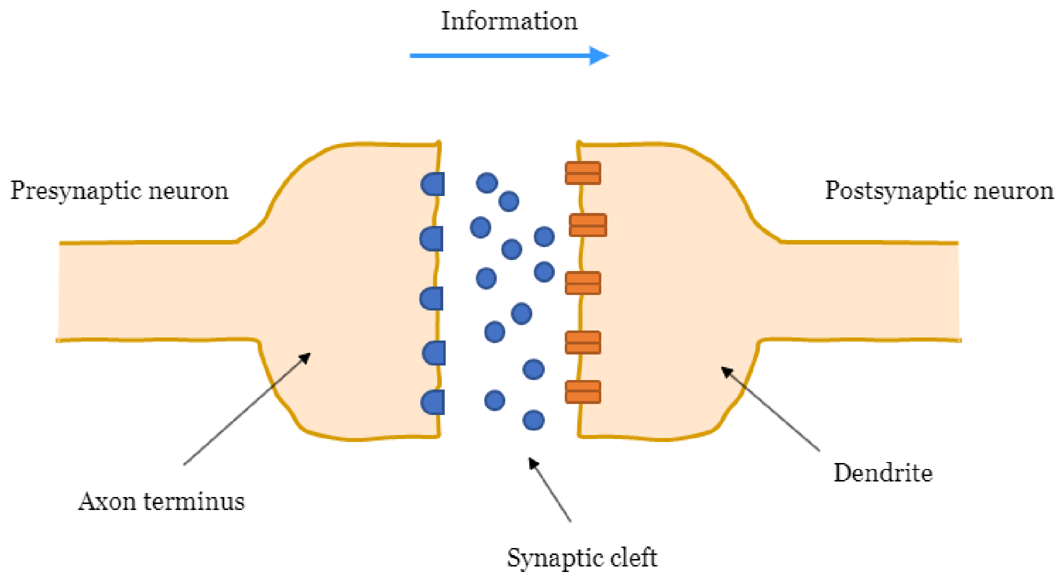


Figure 2.5: Schematic representation of a synapse

a synapse is not a continuous connection between two neurons, and neurotransmitters mediate the transmission of information. These small molecules are secreted by the axonal terminus to the synaptic cleft and recognized by the receptors on the dendrite membrane.

Bundles of axons have unbelievably many names, but the most common is a tract. They serve as a signal transmission medium between different brain parts and the brain, peripheral organs, and tissues. Fig. 2.6 depicts some of the most common neural fibers in the human brain. Based on their role, tracts can be divided into the association, commissural and projection fibers.

Association tracts connect cortex areas located in the same hemisphere. While short fibers connect gyri within the lobe, long ones connect areas in different lobes of one hemisphere [16]. The cingulum and superior longitudinal fasciculus (SLF) are two primary examples of association fibers. The cingulum allows communication between components of the limbic system. It projects from the cingulate gyrus to the entorhinal cortex [11]. The SLF subdivides into FSL I, II, and III. It connects the frontal, occipital, parietal and temporal lobes [17]. All discussed fibers are depicted in Fig. 2.6.

Commissural tracts connect cortical areas in the two hemispheres and allow them to communicate with each other. Connections between the hemispheres are called commissures. Most of the fibers pass through the corpus callosum. It is the biggest commissure and the largest white matter structure at the same time. Two other essential commissures are anterior and posterior commissures [11].

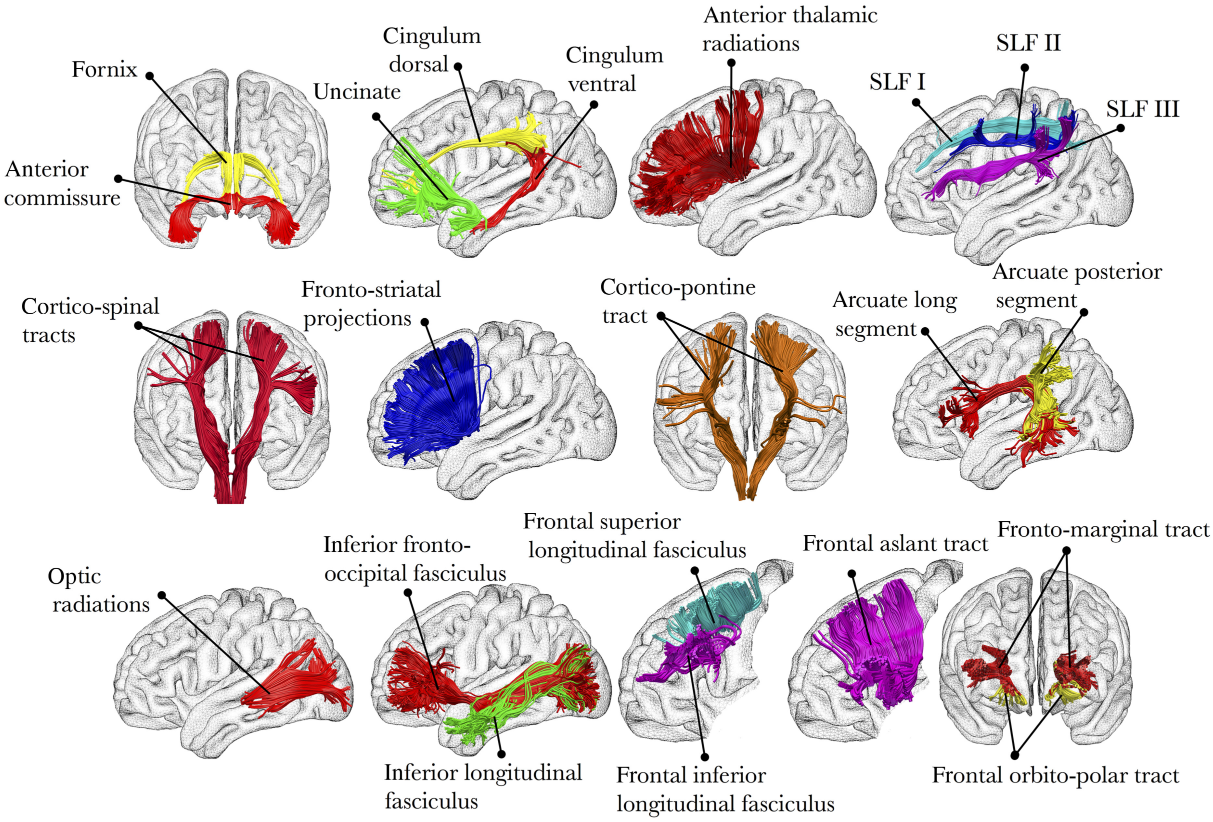


Figure 2.6: Main neural tracts of the human brain [18]

Projection fibers connect the cortex with the corpus striatum, diencephalon, brainstem and spinal cord, allowing communication between the brain and the rest of the body. Ascending tracts communicate sensory information to the brain, while descending tracts communicate motor information from the brain to peripheral organs through the spinal cord [11]. Examples of these can be observed in Fig. 2.6 as cortico-spinal and cortico-pontine tracts.

Before the invention of diffusion-weighted imaging, tracking neural fibers was possible only with *post mortem* dissection of the brain. The development of MRI techniques allowed the investigation of these structures *in vivo*. Details of these techniques will be discussed in the following sections.

2.1.3 Gliomas and their treatment

Malignant gliomas are the most common primary brain tumor, accounting for about 80% of cases. Due to poor prognosis and deterioration in the standard of living, including worsening cognitive function, it is a very harmful and invasive kind of cancer [2]. They easily infiltrate nearby parenchyma but are limited to CNS and do not metastasize. The World Health Organization (WHO) classifies gliomas into four histological grades. From grade I to grade IV, the degree of undifferentiation, anaplasia and aggressiveness increases. [1]. The most common are grade IV and grade III. Grade II tumors are less differentiated but may progress into higher-grade tumors.

The most common symptom is a unilateral headache associated with the same side of the head as the tumor location. Nevertheless, it can still be mistaken for benign headaches. Similarly, other symptoms like cognitive dysfunctions and personality changes might be mistaken for psychiatric disorders or dementia, especially if observed in older patients. The best diagnostic tool is an MRI scan, with CT reserved for people who cannot undergo MRI scanning (e.g. patients with pacemakers). Even though gliomas are easily identifiable with an MRI, wide screening is not advisable since early diagnosis and treatment do not improve outcomes [2].

Corticosteroids, e.g. dexamethasone, may reduce the severity of neurological symptoms. However, they might cause side effects, especially when CNS lymphoma is present. In patients with seizures, antiepileptics are required. Regardless, there is no evidence of benefit from the prophylactic use of these drugs. Patients with suspected cancer are advised to undergo tumor resection surgery. That way, more detailed mass characterization is possible, including grading [19]. Resection of such a mass is a difficult task. Especially considering the possibility of infiltration into the surrounding functional cortical regions. Imprecise surgical intervention may cause damage to functional regions or nerve pathways adjacent to these areas, leading to patient impairments - motor or cognitive.

Therefore, detailed planning should be performed before intervention occurs, including studies using functional MRI and diffusion MRI.

2.2 Magnetic Resonance Imaging

2.2.1 Introduction to MRI

Magnetic resonance imaging (MRI) allows for imaging of the anatomy and physiology of the body. It uses a very strong magnetic field and radio frequency waves during the scan, making it completely non-invasive. In contrast to other radiology techniques, such as X-ray, computed tomography (CT) or positron emission tomography (PET), it does not use radiation or ionization, making it completely safe for the patient. MRI was initially used for anatomical depiction. However, with the development of different MRI modalities, the clinical applications nowadays are very wide, including neurology, cardiology and psychiatry.

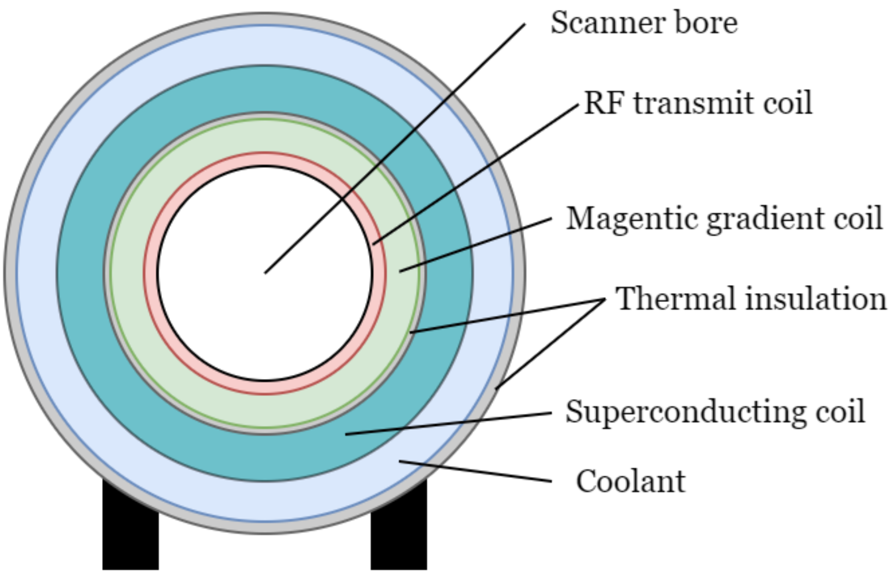


Figure 2.7: Cross-section of an MRI scanner

The MRI scanner consists primarily of superconducting wire introducing a huge magnetic field (Fig. 2.7). It is kept at a very low temperature with liquid helium to allow superconductivity. The magnetic gradient coil is closer to the scanner bore where the subject is placed. It allows for the introduction of a variation in the local magnetic field. This process is crucial for the proper signal acquisition from the whole subject volume. Characteristic noise heard during the MRI scan is caused by these gradient coils switching very rapidly. Closest to the patient is the radio frequency (RF) transmission coil, which excites protons in the body (the working principles explained later in the text). Coils receiving emitted energy are placed close to the body and are not attached to the magnet's bore.

Hydrogen atoms in the human body have random alignment when no strong external magnetic field is applied (Fig. 2.8 A). These nuclei align with the lines of the magnetic field introduced by the scanner and thus are in a low-energy state (Fig. 2.8 B). Pulses of radio frequency (RF) energy are then emitted and absorbed by the hydrogen nuclei, which effectively change into a high-energy state. When radio frequency emission ceases, the hydrogen atoms return to the low-energy state through various relaxation processes and emit back the energy captured by the device as an echo signal (Fig. 2.8 C). The difference between the excitation time and the echo signal is the relaxation time used to create the contrast using

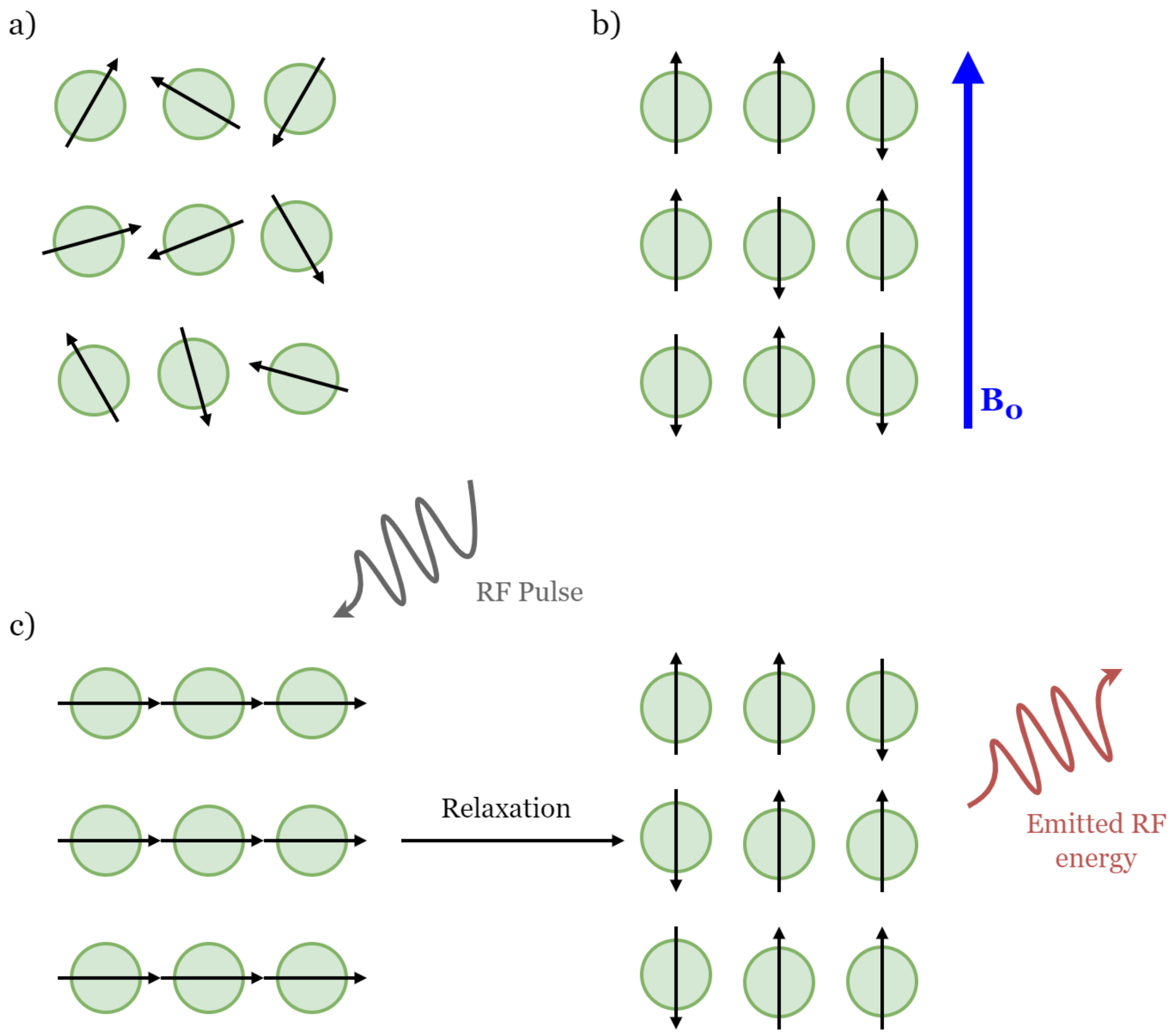


Figure 2.8: Protons behaviour in MRI

Fourier transformation to convert frequency information to intensity levels displayed on the final images as a grey gradient. Different tissues in the body have different relaxation times, which makes it possible to observe individual anatomical structures with MRI. The high sensitivity of these measurements allows distinguishing even between soft and similar tissues, such as the grey and white matter of the brain.

Distinct images can be obtained by tuning the appropriate parameters of the experiment. The two main are repetition time (RT) and time to echo (TE). Repetition time describes the time interval between separate pulses of RF energy emitted toward the sample. Time to echo is the delay between delivering the radio frequency pulse in a particular location and receiving the echo signal.

The most common are T1 and T2 modalities, used for anatomical depiction (Fig. 2.9). T1-weighted images are constructed by using short TE and TR times to measure spin-lattice relaxation (the time it takes nuclei to realign with the magnetic field). T2-weighting uses longer TE and TR times to measure spin-spin relaxation transverse to the magnetic field. Certain tissues appear different on T1 and T2 images. Cerebrospinal fluid (CSF), present in the ventricles and surrounding the cortex, appears dark on T1 and bright on T2; gray matter is much darker on T2 than on T1; the white matter is bright on T1 and dark on T2 (Fig. 2.9). In addition to differences in color, each of these modalities provides benefits in diagnosing various abnormalities like infections, edema, or tumors.

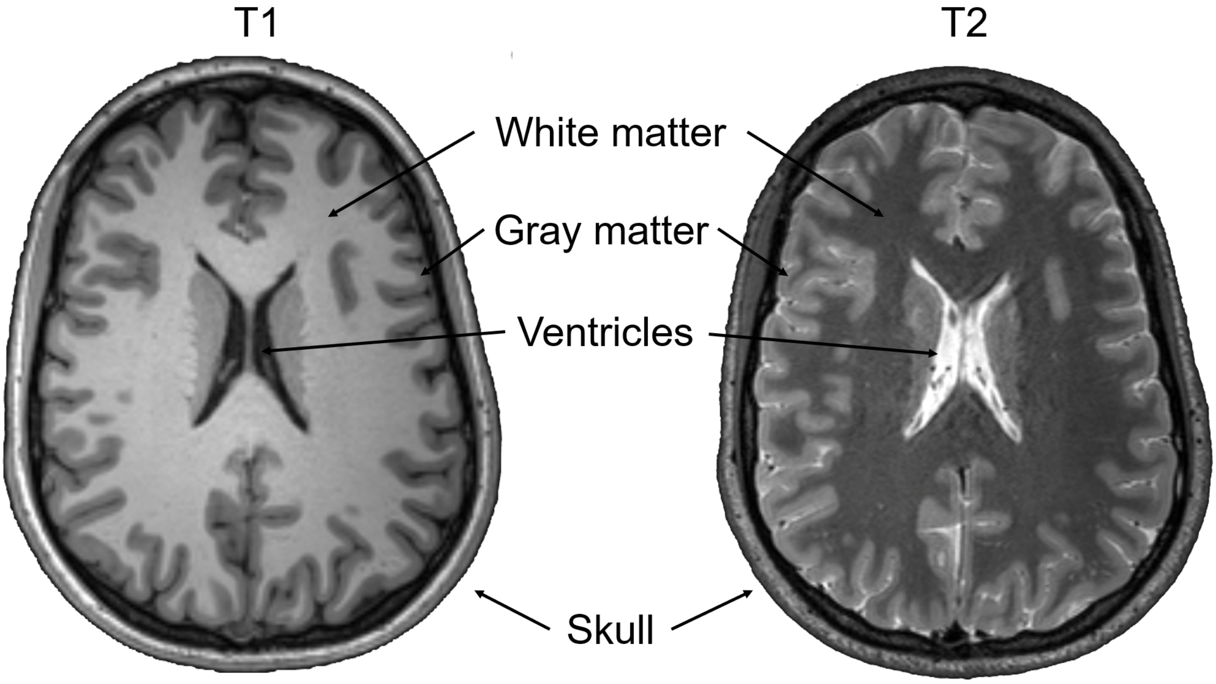


Figure 2.9: Exemples of T1 and T2 images

In addition to T1 and T2 images, two other modalities are widely used in neurosurgery: functional MRI (fMRI) and diffusion-weighted MRI (diffusion-weighted imaging, DWI). The latter is the basis for tractography and will be discussed in more detail in the following section 2.2.3.

2.2.2 Functional MRI

Functional MRI allows radiologists and researchers mapping of different tasks to brain regions in a given individual. As neurons performing tasks require more oxygen and nutrients, the active brain region requires increased blood flow. Oxygen is transported in the blood by hemoglobin, a protein containing an iron atom. Oxygen binds to the iron atom, making it 'rust' temporarily. Since normal and rusted iron influence the magnetic field differently, we can measure the difference between oxygenated and non-oxygenated blood using MRI. Because the contrast is obtained due to blood oxygenation level, this technique is called blood-oxygen-level-dependent imaging (BOLD). Mapping a particular activity, like speaking, hearing, tapping finger, etc., is challenging as this activity has to be separated from others (e.g. hearing). It can be achieved using special paradigms prepared by neuropsychologists. These contain instructions for the patient to follow to activate brain regions responsible for the task under investigation, silencing other parts of the brain simultaneously.

The activity in a given cortex area is coupled with increased blood flow as active neurons have higher requirements for oxygen and nutrients [20, 7]. Since neural cells do not retain any energy reserves in the form of sugar or oxygen, these compounds must be delivered quickly. Therefore, it is possible to measure brain activity by detecting changes associated with blood flow [21]. The coupling between neural activity and an increased blood flow is called hemodynamic response. A special sequence of pulses in MRI allows using blood-oxygen-level-dependent (BOLD) as a contrast to map functional regions. As shown in figure 2.10, the BOLD signal increases about 5 seconds after neuronal activity occurs. This time is needed for the vascular system to respond to the brain's need for glucose and oxygen. When the activity ceases, it falls below the original level, what is described as the post-stimulus undershoot, and with time recovers to the baseline [22].

Presurgery studies aim to identify regions related to critical functions like speech production and comprehension, moving limbs, or sensing. Clinical use is much harder given that brain pathology, and the use of drugs can lead to changes in the blood flow that are not related to neuronal activity [23]. In this type of diagnostics, it is extremely important to separate the functionality under investigation from the background signal and other involuntary activity. For this particular purpose, neuropsychologists design special studies paradigms which help to achieve this goal.

2.2.3 Diffusion-weighted MRI

Diffusion-weighted imaging (DWI) is an MRI technique using special sequences of pulses to obtain image contrast from the movement of different molecules in the body [24, 25]. The most common molecule to track with this technique is water, present in all human body cells. Water movement in tissue is not completely random due to natural obstacles such as cell membranes and organelles. Therefore, patterns of water diffusion can unveil microscopic details of tissue architectures.

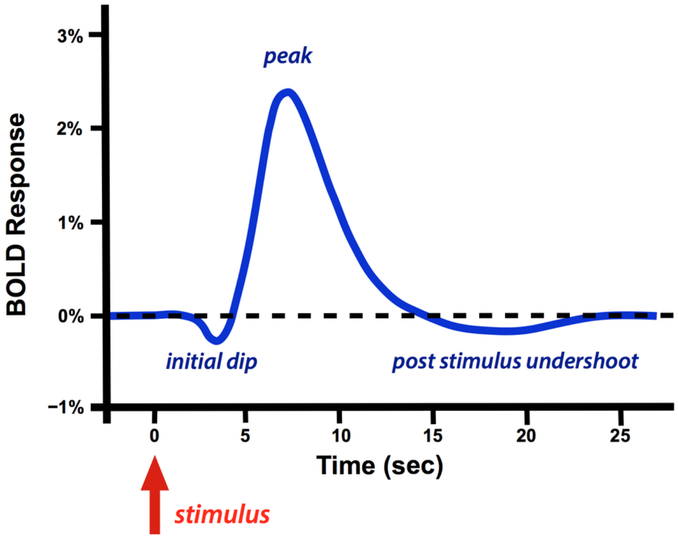


Figure 2.10: BOLD response [26]

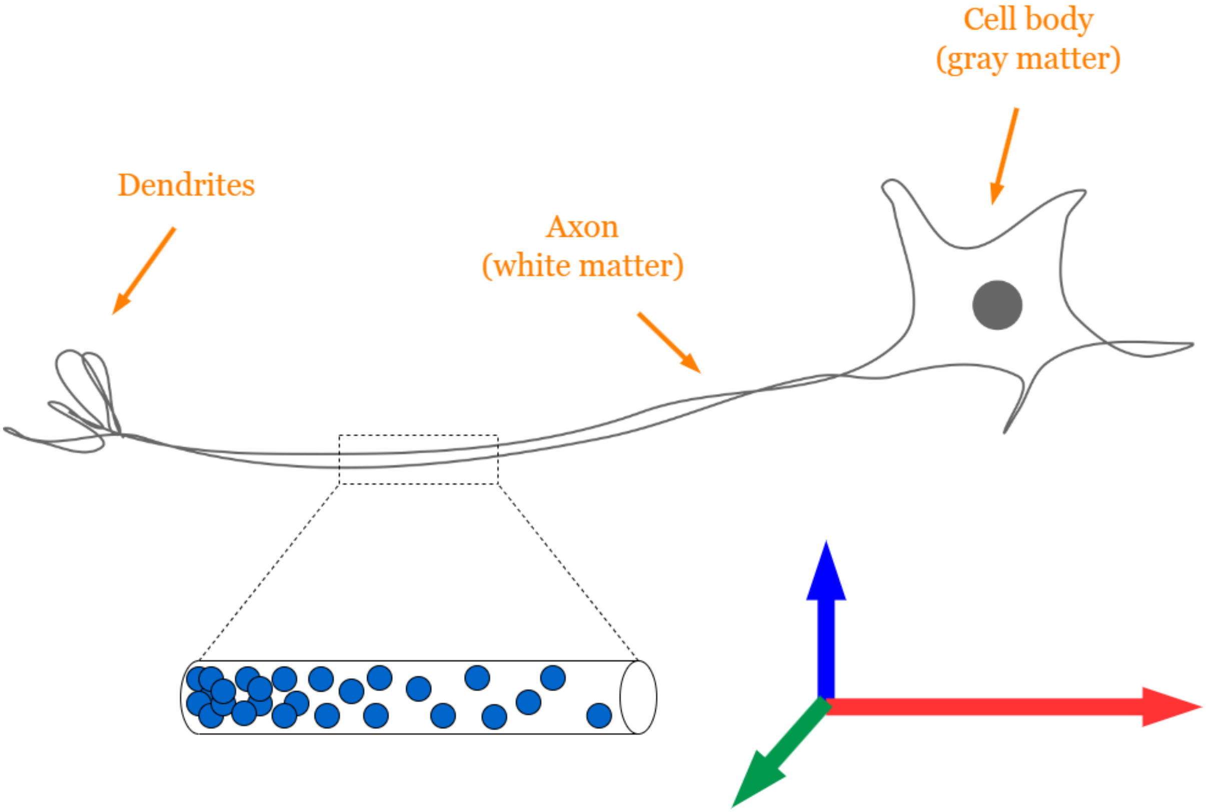


Figure 2.11: Water movement in neural cell

In the case of cells with elongated shapes, the major direction of water movement is in line with the longest axis. Thus tracking the movement of water molecules allows for determining the microscopic details of the tissue. In such an image, the intensity of a particular voxel reflects the rate of water diffusion and can lead to many applications. One of the most interesting ones is the trajectory of neural fibers (Fig. 2.11). A special kind of diffusion imaging, Diffusion Tensor Imaging (DTI), aims not only to produce contrast from the diffusion but also to estimate the trajectory of neural tracts [10]. In practice, a DTI study contains a series of measurements obtained with distinct orientations of diffusion-sensitizing gradients. This information can be used by tractography methods developed to follow neural tracts within the brain.

2.3 Preoperative tractography

Every neurosurgical intervention is preceded by detailed planning. At this stage, an optimal entry site and the scope of intervention are determined. A procedure prepared in this way reduces the risk of complications, including motor or cognitive impairments.

Planning begins with a detailed patient examination using various MRI imaging techniques. First, anatomical imaging (T1 and T2 scans) is performed to determine the precise location of the tumor mass. In the next step a study using functional imaging (fMRI) is performed, considering the subject variance in the location of eloquent cortical regions. Such a study allows to map functional regions proximate to the tumorous mass. Based on the coarse localization derived from functional atlases, the clinician can narrow down the list of tasks to be examined. The narrowing is essential for a fast diagnosis, given quite lengthy experiments.

Another modality used before the surgery is diffusion tensor imaging (DTI). This data can be used to elucidate the topology of the nerve fibers localized in close proximity to the potential site of intervention. A procedure aimed at determining neural fiber arrangement is called tractography, and the bundles of neural fibers are often referred to as tracts. Tracking (reconstructing fibers) starts with a seed. Tractography itself answers many interesting questions regarding the functioning of the human brain. However, its application in neurosurgery has very specific aims.

Simply determining the location of nerve fibers near the surgical field is not enough. Without knowing their origin or destination, one cannot make a prognosis about side effects caused by eventual damage. Hence, preoperative tractography shall be coupled with fMRI experiments.

With this detailed information, the neurosurgeon can precisely plan an entry site and the scope of the intervention that will impose the smallest risk of damaging critical tissue. Furthermore, neuronavigation systems help to achieve this goal by visualizing all the necessary information during surgery. They are fed with anatomical images but lack information about functional areas or topology of the nerve fibers.

3 Artificial neural networks and their applications in medicine

3.1 Introduction to Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the brain's neural networks. Such ANN consists of connected nodes based upon biological neurons. The output of the single neuron results from executing a nonlinear function on the sum of its inputs. Artificial neurons (nodes) are connected with others by edges. Both nodes and edges have weights that change during the learning process and thus constitute the model's parameters.

Models achieve their skills in the training process. Training is a process in which the network is presented with input data (the sample) and corresponding expected output (the label). After the network processes the input data, its output is compared with the expected result. This comparison is made using the loss function, which maps the difference between the two onto a real number. Mathematically, ANN training is a problem of optimizing the loss function. It is achieved through backpropagation and usage of the optimization technique. Backpropagation and different loss functions will be discussed in the following sections.

Many different ANN architectures have been proposed to tackle different tasks. The most general ones are perceptrons and feedforward neural networks (FFNN). Problems related to image recognition and classification are usually solved using convolution layers, where the network learns the weights of filters used on the input images. Sequential data, like text or video, require architectures to extract information encoded in a sequence of inputs. These architectures are usually a sub-type of recurrent neural networks (RNN), best suited for continuous signal or prognosis problems.

3.1.1 Perceptron and feedforward networks

Designing a neural network for a given task is non-trivial, as many parameters must be considered. Neural networks consist of stacked layers processing input information into the desired output envisioned by the training label. Before discussing different kinds of layers, it is crucial to focus on their main building block - an artificial neuron.

A function of an artificial neuron can be easily related to the biological neuron (Fig. 2.2, and Fig. 2.4). It receives input information (a vector of features) and produces a scalar value output. The most basic neural network is a perceptron that can be trained for a binary clas-

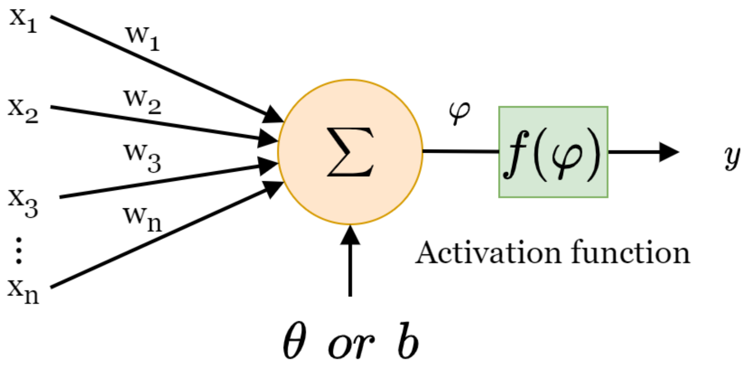


Figure 3.1: Architecture of McCulloch-Pitts neuron and modern artificial neuron

sification task [27]. Figure 2.2 represents a diagram of such architecture, a single McCulloch-Pitts neuron, otherwise known as a threshold neuron. Input data vector $x_1, x_2, \ldots, x_n$, and a set of weights $w_1, w_2, \ldots, w_n$ are used to compute an output (y). For binary classification with unipolar activation function ($f(\varphi)$), output values take

$$y = f(\varphi) = \begin{cases} 1, & \varphi > 0, \\ 0, & \varphi \leq 0, \end{cases} \tag{1}$$

where

$$\varphi = \sum_{n=1}^{n} w_i x_i - \theta, \tag{2}$$

where θ is the threshold value of the activation function and n number of inputs. A single perceptron can be used for binary classification tasks, whereas additional neurons must be used for classification with more than two classes. Moreover, a single-layer perceptron is suitable for learning patterns that can be separated linearly, as Marvin Minsky and Seymour Papert showed that a single-layer perceptron cannot learn XOR function [28]. Neurons used today take a slightly different form than the McCulloch-Pitts neuron (Fig. 3.1). In this model, θ is replaced by bias b, added to the result of multiplying the input data and weights, resulting in the neuron transfer function:

$$\varphi = \sum_{n=1}^{n} w_i x_i + b. \tag{3}$$

The unipolar activation function is replaced by a nonlinear activation function, such as a rectified linear unit (ReLU) [29].

Such artificial neurons are organized into stacked neural layers building modern artifi-

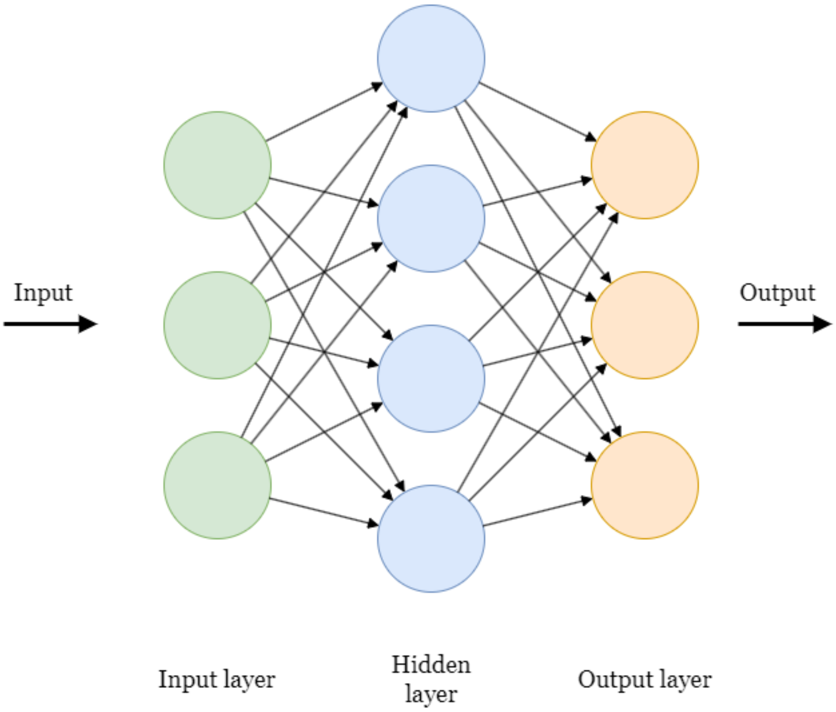


Figure 3.2: Three basic layers of a feedforward neural network

cial neural networks. Figure 3.2 depicts a schematic representation of three types of neural layers building a fully connected feedforward network, with circles representing single neurons within the layer and arrows depicting weighted connections between them [30]. The first is the input layer, accepting the input data passed to the other layers. With one or more hidden layers, neural networks can perform complex analyses of the data they receive. They are responsible for data transformation and feature creation. The last layer in a neural network is the output layer. It processes an output of the last hidden layer, called the last hidden state, to the desired output. An indispensable feature of a feedforward network is that connections between the nodes do not form a cycle. Such cycles can be formed in recurrent neural networks (discussed in the following section). Fully-connected layers are the most abstract types, having the potential to learn the same tasks as any other layers. However, designing and training such a network is tedious, and specialized neural layers shall be used for particular input data.

3.1.2 Convolutional neural networks

Convolutional neural networks (CNN) were developed for processing data with grid-like topology. They are most often used for image processing since an image can be viewed as a two-dimensional grid of pixels [31].

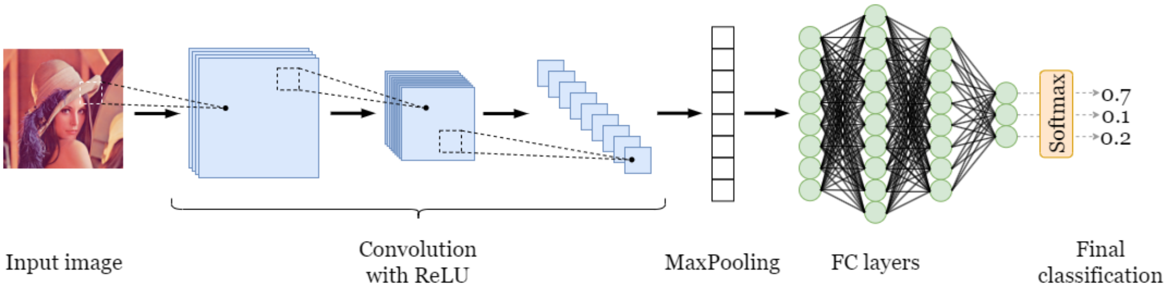


Figure 3.3: An example of convolutional network architecture

In a typical convolution network, convolution layers (or blocks of such layers) are typically followed by a pooling layer and fully connected layers. An input image data is of shape $N \times C_{in} \times H \times W$, where N is the batch size, C_{in} is the number of channels, and H and W are height and width of the image respectively. The convolution operation performed by the convolution layer is a dot product of the convolution filter with the input tensor. As the filter slides through the input, the dot product of the operation adds to the feature map produced by the layer. The filter has a specific, predefined size, e.g. 3 x 3 pixels, called the kernel size. The amount of movement between the filter and the input, in the width and height dimensions, is called the stride. The default stride is $(1,1)$ and indicates movement by one pixel in both directions. The resulting tensor has dimensions $N \times C_{out} \times H_{out} \times W_{out}$ and is the input tensor of the next layer.

The pooling layer can be used globally to process the final feature map produced by the last convolution layer or between two convolution layers. It reduces the dimensionality of a tensor by combining the outputs of neuron clusters from one layer to the single neuron of the next. Clusters commonly used are small, usually with 2×2 or 3×3 tilling size [32, 33], and the either maximum value is used (max pooling), or the average (average pooling) [34, 35].

The final output is produced by fully connected layers, which process a feature map produced by convolution layers into final classification. To obtain a vector of probabilities for each class, a softmax function is used. It turns an output of the last dense layer into a probability distribution of K outcomes, where K is the number of target classes.

Convolutional layers can be used in more complex architectures as well. An example is an autoencoder developed for masking image data. The encoder part of the model allows to mask image so it can no longer be identified visually without using a decoder. Encoded version however, still carries information allowing for classification [36].

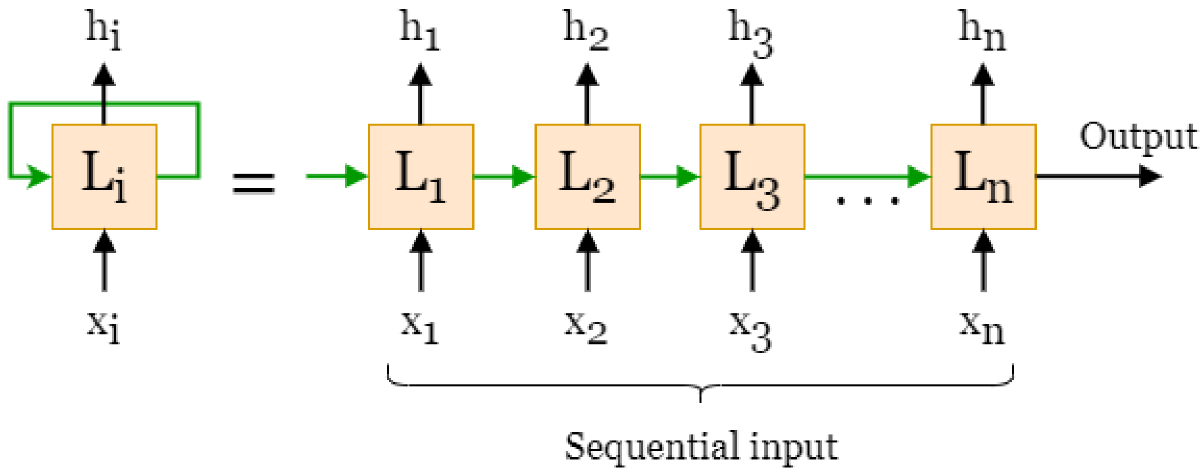


Figure 3.4: Recurrent neural network

3.1.3 Recurrent neural networks

Many types of data are sequential. One of the best examples is speech, where individual words have little meaning, and the entire message of the utterance is hidden in their proper sequencing. Traditional neural networks cannot analyze such data effectively as they process information one sample at a time. For proper reasoning from sequential data, a memory mechanism is required. Recurrent neural networks (RNN) achieve this by introducing cycles to the graph of node connections between layers [37]. Hence, they can reason about sequential data in tasks such as recognition of speech [38, 39] or unsegmented handwriting [40]. Figure 3.4 describes the idea behind creating cycles within the network. Given a set of sequential data $x_1, x_2, \ldots, x_n$, each layer of the network computes its hidden state, denoted by $h_1, h_2, \ldots, h_n$, which is saved and passed as a second input to the next layer (green arrows). In the case of the first layer, a "hidden state" input is a vector of random numbers from a particular distribution.

RNNs work very well when the task requires context only from the recent elements of the sequence. In theory, they are capable of learning long-term dependencies. However, when RNN is trained with backpropagation, the long-term gradients can easily vanish (when tending to zero) or explode (when tending to infinity). It is caused by computations that use finite-precision numbers. When the gap between context and task increases, performance deteriorates. A special kind of RNNs, the Long short-term memory (LSTM) network, partially solves a vanishing gradient problem, as they allow gradients to flow unchanged [41]. They also use special mechanisms allowing them to learn long-term dependencies. At their core is the cell state which is protected and controlled by three gates: forget, input/update, and output. In this arrangement, a forget gate decides what portion of the information is unnecessary for the task and shall be forgotten. The input gate decides what shall be updated and how. The last gate decides what information the layer shall output.

3.1.4 Attention mechanism

Many artificial neural networks include attention mechanisms in their architectures. This technique was developed to mimic cognitive attention [42, 43, 44]. It enhances some parts of the input data while reducing the importance of other parts. It follows the notion that not all features are equally relevant to the task, and the neural network should focus mainly on the crucial ones.

In recent years, attention gained popularity when Transformer architecture was introduced [45]. It relies on attention for capturing global dependencies between input and output. Authors achieve this by replacing completely recurrent layers with multi-headed self-attention. Such architecture has better parallelization capabilities and trains faster than the standard approach, achieving state-of-the-art performance at the same time.

3.1.5 Training neural networks with backpropagation

Originally backpropagation (backprop) was an algorithm for training feedforward neural networks. Nowadays, many algorithms for training different ANN architectures fall under this generalized term. Backprop calculates a gradient of the loss function during training, considering all of the model's weights. Such an approach allows greater efficiency than the classical one, where the gradient is computed individually for each weight. This makes backprop suitable for training multilayer networks with ease [31].

In the case of the multilayer network, the derivatives of the layer weights cannot be simply calculated with respect to the loss function as in a single-layer network. This is especially important given that all architectures nowadays include nonlinear transfer functions called activation functions. To calculate derivatives, backpropagation uses a chain rule from calculus and is generalized by automatic differentiation - a special case of reverse accumulation as in (). A gradient is computed for each layer separately, iterating backwards from the last layer to the first (hence the term backward propagation). This prevents redundant computing of the intermediate terms in the chain rule.

Following the chain rule a derivative for the loss function $\mathscr{L}(\hat{y}, \bar{y})$ in respect to weight w_n in case of a single neuron (Fig. 3.1) can be calculated as follows:

$$\frac{\partial \mathscr{L}}{\partial w_n} = \frac{\partial \mathscr{L}}{\partial y}\frac{\partial y}{\partial w_n} = \frac{\partial \mathscr{L}}{\partial y}\frac{\partial y}{\partial \varphi}\frac{\partial \varphi}{\partial w_n}, \tag{4}$$

where φ is the output of the transfer function as in (3), and y is the output of an activation function. The derivative of a neuron output y with respect to its input is the derivative of an activation function, which for backpropagation to work has to be differentiable:

$$\frac{\partial y}{\partial \varphi} = \frac{\partial f(\varphi)}{\partial \varphi} \tag{5}$$

Many optimization algorithms can be used to train neural networks. Among them, two are the most common: Stochastic Gradient Descent (SGD) [46] and Adam [47]. SGD is a stochastic approximation of gradient descent, where the gradient is not computed for the whole dataset. SGD uses its approximation calculated on the randomly selected data subset. Such computation takes less time than the traditional method. Hence the time required for training is much smaller.

Adam optimizer is another method for efficient stochastic optimization. Based on the first and second moments of the gradients, it computes individual adaptive learning rates for different parameters. This method is memory efficient as it only requires first-order gradients. Nowadays, it is the most commonly used algorithm for training neural networks.

Foret et al. propose a Sharpness-Aware Minimization (SAM) method to minimize both the loss value and the sharpness of the loss [48, 49]. It focuses on the set of weights in the neighbourhood of uniformly low loss. With experiments on the benchmark datasets and models, authors argue that using their optimization method can lead to state-of-the-art performance with label noise robustness.

3.1.6 Activation functions

If we stack multiple fully-connected layers together, we could achieve the same goal with one fully-connected layer with more neurons and weights. Adding non-linearity in the form of activation functions allows for representing more complex functions. One of the core, broadly used activation functions is a sigmoid function, accepting as an input, neuron out-

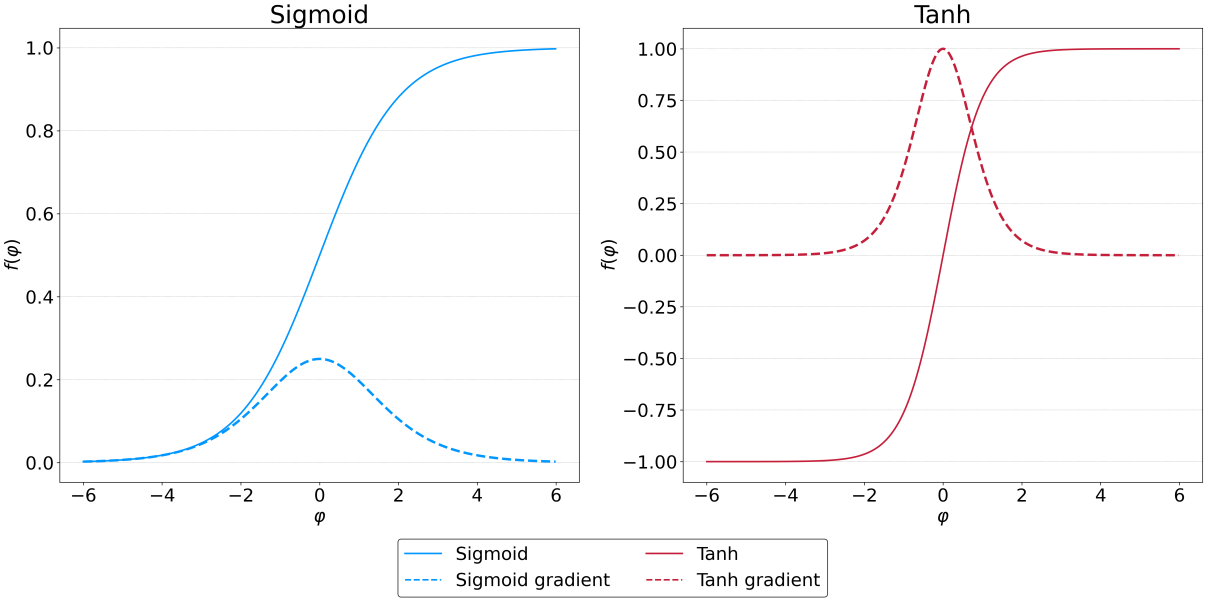


Figure 3.5: The Sigmoid and Hyperbolic Tangent activation functions

put φ as denoted in Fig. 3.1. Sigmoid function $f(\varphi)$ is expressed as

$$f(\varphi) = \frac{1}{1+e^{-\varphi}}, \tag{6}$$

which plot is presented in Figure 3.5. It assumes values in the range of $[0,1]$, hence it is often used to transform the real value output of the layer into a likelihood.

Another example is a hyperbolic tangent function (Tanh):

$$Tanh(\varphi) = \frac{\exp(\varphi) - \exp(-\varphi)}{\exp(\varphi) + \exp(-\varphi)}, \tag{7}$$

which has a bigger range of output values, $Tanh(\varphi) \in [-1,1]$ (Fig. 3.5). Due to the negative values of the output this function could not be used as a probability projection.

The commonly used is Rectified Linear Unit (ReLU), which is a simple maximum function:

$$ReLU(\varphi) = \max(0,\varphi). \tag{8}$$

This method essentially activates only those neurons that output values equal or greater than 1 (Fig. 3.6). Leaky ReLU is a very similar function, however in that case also neurons producing small negative values are activated:

$$LeakyReLU(\varphi) = \max(0,\varphi) + s \cdot \min(0,\varphi), \tag{9}$$

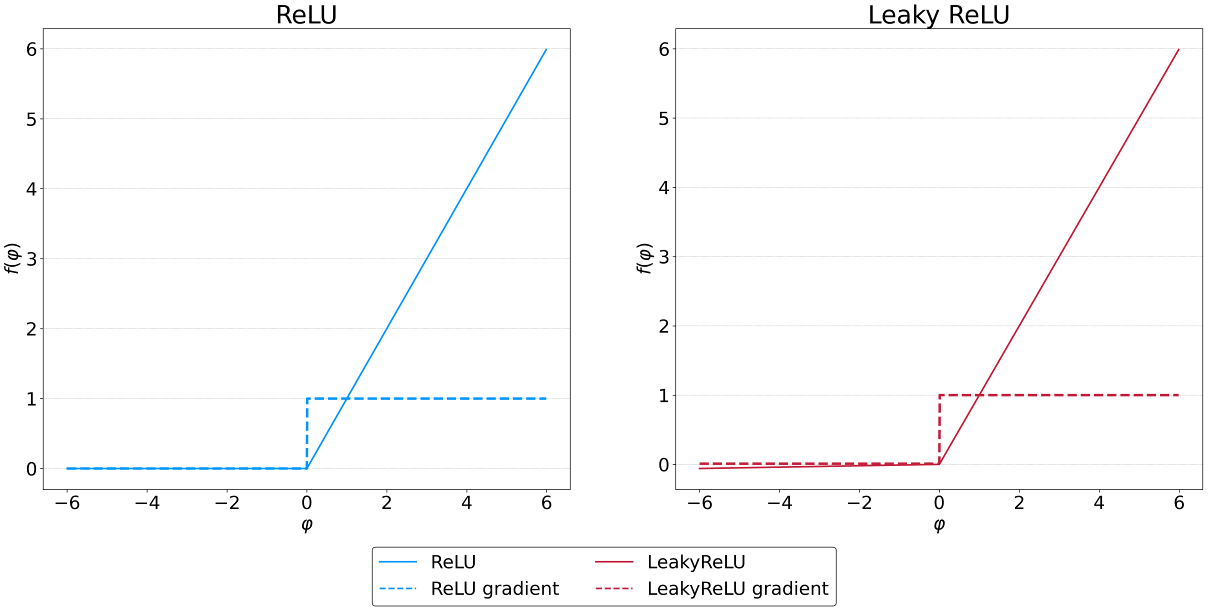


Figure 3.6: The ReLU and Leaky ReLU activation functions

where an s parameter controls the slope of the negative side and, by default, is set to $s = 1e{-}2$ (Fig. 3.6).

When choosing the right activation function for the network, it is important to consider the range of returned values and their derivate. Due to backpropagation and the chain rule, the latter profoundly impacts how the network learns from the data. In the case of the sigmoid function, its derivate values are in the range $[0, 0.25]$ and with an φ tending to $-\infty$ and $+\infty$, the gradient converge to 0 (Fig. 3.5). This can contribute to the vanishing gradient problem. Such a situation occurs when the input values from the earlier layers increase, and the gradient becomes so small that it cannot involve the correction. The gradient of the loss function $\nabla\mathscr{L}$ at a given layer l, where $l = 1 \dots L$ is the gradient at its subsequent layer $\mathscr{L}_{l-1}$ multiplied by the gradient of the activation function $f(\varphi_l)$:

$$\nabla\mathscr{L}_l = \nabla\mathscr{L}_{l-1} \cdot \nabla f(\varphi_l). \tag{10}$$

If the gradient value is less than 1, the gradient on the layers distant from the output will tend to 0. Layers with a gradient converging to 0, will stop the gradient propagation to the farthest layers. Tanh function counters this problem by increasing the maximal gradient value. However, it produces gradient values greater than 0.25 for the small range of φ values (Fig. 3.5). The vanishing gradient problem is solved for ReLU and LeakyReLU functions, as all activated neurons returning positive values have a gradient value of 1 (Fig. 3.6). Despite that, ReLU introduces another problem known as the dead neuron or the dying ReLU problem. Gradients of all inactive neurons are equal to 0, which can result in the consistently inactive neuron.

3.1.7 Loss functions

Choosing the right loss function is extremely important in training a neural network. This function compares the expected result (a label, denoted with $\bar{y}$) with the network's actual output denoted by $\hat{y}$. The result mapped onto the real number directly impacts how the optimization algorithm updates the model weights. The wrong choice can result in a suboptimal or incorrect representation of the differences between the output and the label, pushing the model weights toward achieving a different goal.

The most simple loss function is 0-1 loss:

$$\mathscr{L}(\hat{y},\bar{y}) = \begin{cases} 1 & \text{for} \quad \hat{y} \neq \bar{y}, \\ 0 & \text{for} \quad \hat{y} = \bar{y}, \end{cases} \tag{11}$$

where $\hat{y}$ is the output of the network, and $\bar{y}$ is the label. In practice, this indicator function returns either 0 or 1, which is not very useful in learning with optimization algorithms, as it does not provide information about how far the current solution is from the expected one.

A more informative loss function is quadratic loss:

$$\mathscr{L}(\hat{y},\bar{y}) = C(\hat{y}-\bar{y})^2. \tag{12}$$

In this equation, C is the constant value and is usually ignored by setting $C = 1$, resulting in the basis for calculating Mean Squared Error (MSE). Before defining an MSE loss function in (15), we will define a loss function for a batched input. Given the batch size is M, the loss function can be described as:

$$\mathscr{L}(\hat{y},\bar{y}) = \mathscr{L} = \{l(\hat{y}_1,\bar{y}_1),\ldots,l(\hat{y}_M,\bar{y}_M)\}. \tag{13}$$

If we consider (12) and set $C = 1$ for batched input as in (13), we end up with MSE loss for a batched network output:

$$l(\hat{y}_m,\bar{y}_m) = (\hat{y}_m-\bar{y}_m)^2, \quad m \in [1,M]. \tag{14}$$

The MSE is defined as the mean of squared errors of all examples:

$$MSE = \frac{1}{M}\sum_{m=1}^{M}(\hat{y}_m - \bar{y}_m)^2, \quad m \in [1, M]. \tag{15}$$

Even though MSE loss is much more informative in comparison to 0-1 loss, many approaches use Cross Entropy (CE) to calculate the loss between input features and desired labels. Given the classes indices are in the range $k \in [0, K]$, where K is the number of classes, and M is the batch size as in (13), a CE loss function can be described as follows

$$\mathscr{L} = -\sum_{k=1}^{K} \bar{y}_k \log(\hat{y}_k), \tag{16}$$

where $\bar{y_k}$ is the true label, and $\hat{y}_k$ is the Softmax probability for the k-th class. A CE loss can be used for multi-class and binary classification problems. However, for the latter, a Binary Cross Entropy (BCE) is more suitable

$$\mathscr{L} = -\frac{1}{M}\sum_{m=1}^{M} \bar{y}_m \cdot \log \hat{y}_m + (1 - \bar{y}_m) \cdot log(1 - \hat{y}_m), \quad m \in [1, M]. \tag{17}$$

A softmax or sigmoid function usually has to be applied to the network output before using these loss functions. Programming libraries, such as PyTorch, provide versions of the loss functions mentioned above, operating directly on logit probabilities obtained from the last layer. Some evidence from the field suggests that using these lays better results.

3.2 Application of machine learning in medicine

3.2.1 Genomics and proteomics

Even though genomics and proteomics are considered basic sciences, their developments have a direct effect on medicine. Due to the massive amounts of data produced by next-generation sequencing, its analysis is impossible without advanced algorithms, including machine learning and deep learning. Such methods helped to sequence SARS-CoV-2 genome, identify its variants, as well as design treatments, including vaccines [50]. Lowering the costs of genome sequencing allows the inclusion of genetic information in the diagnosis and treatment process. As shown by Sun et al. [51], SVM [52] is one of the available methods that can be used to seek genes causing diseases such as cancer or diabetes. The availability of large-scale data allows for studying the evolution and structure of proteins [53, 54, 55]. This is crucial to understanding the biochemical basis behind diseases and developing new drugs [56, 57].

3.2.2 Clinical decision support systems

Computerized clinical decision support systems (CDSS) provide significant assistance in diagnosis and treatment [58]. Such a system can use provided and computerized clinical knowledge to interpret patient characteristics. It makes it easier to integrate the clinician's experience, the patient's value and scientific knowledge in evidence-based medicine [59]. When CDSS are applied in the diagnosis domain, they are called diagnostic decision support systems (DDSS). They were developed to mimic a natural process of differential diagnosis performed by clinicians [60]. This procedure involves analyzing the patient's history and the results of other tests (laboratory, physical, imaging) to make an accurate diagnosis. Because of the many factors that go into making a decision, this is a time-consuming process with a risk of error, especially if it involves a rare condition with symptoms similar to other conditions. DDSSs are developed as a diagnostic aid because computer systems can accurately infer from large data sets, revealing correlations that a human expert might overlook. They are not designed to replace phycisians, but merely provide with a list of possible diagnoses given the patient information. Despite hopes of improving the diagnostic process, the reception of the methods was not positive, given the poor accuracy and system integration [61]. More specialized DDSSs have achieved greater success in application. For instance, Kunhimangalam et al. proposed a system for diagnosing peripheral neuropathy [62]. Using fuzzy logic, they integrate information about symptoms and results of the diagnostic tests. Comparing the results of the model with the opinion of the experts, they achieved satisfactory accuracy.

The CDSSs are also used during surgical procedures. A system described in [63, 64] helps to localize a subthalamic nucleus (STN) in deep brain stimulation surgeries aimed at the treatment of Parkinson's disease. The location of the nucleus is established based on the microelectrode recordings placed within the brain during the surgery. When close to the STN, surgeon moves the electrodes 1 mm at a time and take 10 s of recording to analyze, till reaching the STN structure. Described algorithm cleans and analysis the data by removing high frequencies, sporious spikes and other artifacts using wavelet transformations [65] and power spectral analysis. Recordings analyzed in this way can be passed to an a machine learning algorithm, which assesses whether the footage is from an STN or another structure. Such analysis are necessary as this structure cannot be visualized on CT or MRI. The system relieves the surgeon of the labor-intensive interpretation of the recordings, improving the outcomes and decreasing the time of the surgery significantly.

Such systems are also used in aortic valve replacement surgeries [66]. These surgeries remain complex and technically challenging given the limited workspace and restrictive surgical field. The system indicates an entry site optimal for the patient's morphology based on anatomical measurements from the preoperative CT images and the surgical guidelines. An initial step is skin and bone structure segmentation using 3D region growing method [67]. This process allows to detect sternum using image processing methods. Once the right border of sternum is identified, an intercostal spaces (ICS) can be detected from the ribs countur. Using clinical guidelines, this information can be easily used to propose surgical technique best for the patient morphology. With interactive three-dimensional visualization of all relevant information, it makes planning the surgery much easier.

Many DDSS systems are designed for radiology applications. These systems are described in more detail in the following section.

3.2.3 Radiology

Radiology is a branch of medicine that focuses on the analysis of images obtained by various techniques, such as X-ray, CT, MRI, and ultrasound. The successes achieved by neural networks, particularly convolutional networks, in image classification have led researchers to develop methods to support the diagnostic process of radiologists [68].

Ismael and Sengur developed and described in [69] a variety of models to classify chest X-ray images into two classes: healthy and COVID-19 disease. They used well-known and tested convolutional neural network (CNN) architectures (ResNet18, ResNet50, ResNet101, VGG16 and VGG19) described in [70, 71], to extract features from X-ray images, which were then classified by the support vector machine (SVM). CNN models were fine-tuned in a separate experiment to classify the original images. In both approaches, authors achieved accuracy above 90%. Chest X-rays were also used by Sogancioglu et al. [72] to detect cardiomegaly by anatomical segmentation and image-level classification. Models proposed by the authors achieve great performance with a segmentation approach reaching an AUC of 0.977 and a classification approach with an AUC of 0.941.

The use of deep learning in radiology is not limited to X-ray image analysis. ANNs were also applied to a challenging task of brain tumor classification and grading [73] based on MRI images. CNN architecture used by the authors can detect three types of tumors: meningioma, glioma and pituitary tumor, as well as differentiate between three grades: II, III, and IV. The accuracy achieved in both tasks is above 96%, making it a feasible diagnostic aid.

Many methods were developed for the diagnosis of primary liver cancers and their metastases [74]. A typical screening method for liver abnormalities is a periodical abdomen ultrasound. Guo et al. proposed and described in [75] a two-stage multi-view learning framework for diagnosing liver tumors using contrast ultrasound images. The first stage corresponds to the deep canonical correlation analysis on ultrasound images resulting in the multi-view features. These features are fed in the second stage to a multiple kernel learning classifier providing the final diagnosis.

Contrast-enhanced ultrasound cine clips were used by Ta et al. in [76] for classifying benign and malignant liver tumors. They achieved performance with AUCs of 0.829 and 0.883 using artificial neural networks and support vector machines (SVM).

Various approaches are described in [77], which provide invaluable assistance in diagnosing neurodegenerative diseases. An example can be given by Payan and Montana in [78]. They employed sparse autoencoders and 3D convolutional neural networks to detect Alzheimer's disease from MRI scans. Another example is using rather simple Alexnet architecture [33] to diagnose Parkinson's disease based on T2 MRI. A system proposed by Sivaranjini and Sujatha, and described in [79], achieves accuracy of 88.9%.

4 Path search algorithms

Pathfinding is the process of calculating the shortest route between two given points using a computer algorithm. It is a practical approach to solving maze-like problems and is highly influenced by Dijkstra's algorithm [80] for finding the shortest path on a weighted graph. Pathfinding is related to the shortest path problem from graph theory, which focuses on identifying the most suitable path between two points in a large network based on specific criteria, such as length, speed, or cost.

The pathfinding method involves searching a graph by starting at a given vertex and exploring adjacent nodes until reaching the target node. The objective is usually to find the most cost-effective route. Two widely used algorithms allowing to achieve this task are Dijkstra's algorithm and its variant - an A* algorithm, described in this chapter.

4.1 Dijkstra algorithm

Dijkstra's algorithm was designed and published in [80]. It exists in many variants, while the original version was developed for finding the shortest path between two nodes in a weighted graph. An example of such an application is to find the shortest route between two cities on a network of roads. In such a case, the nodes represent the cities connected with weighted edges. The weights of the edges reflect the driving distance between the two given cities. The algorithm computes the shortest path between the source node and every other node in a graph, stopping when the shortest path between the source and target is found.

The first step of the algorithm is to mark all the nodes as unvisited and store them in a set. Each node is assigned a distance value, 0 for the source and infinity for all other nodes. The procedure computes the tentative distance value for all neighbouring nodes through the currently investigated one. This new distance is compared to the presently stored value, and the smaller is assigned. After all the neighbours are considered, the current node is marked as visited and removed from the unvisited set. The process repeats by setting as the current node, a node from the unvisited set, having the smallest distance value. The algorithm finishes when the destination node is marked as visited, or the target node has the smallest distance value among the unvisited ones.

4.2 A* algorithm

A* algorithm was designed and published in [81]. It is a variant of Dijkstra's algorithm, which assigns a weight to each unvisited node equal to the weight of that node increased by the approximate distance between the inspected node and a target node. This distance is approximated by the heuristic function $h(v_n, v_t)$, which distinguishes this approach from the original idea. Hence, for a given node v_n an A* algorithm minimizes:

$$f(v_n) = g(v_n) + h(v_n, v_t), \tag{18}$$

where $g(v_n)$ is the cost of the path so far, and $h(v_n, v_t)$ is the heuristic function, estimating a cost of a path from the current point v_n to the target v_t.

A* is equivalent to Dijkstra when the heuristic evaluates to 0. As the estimates increase and come close to the true distance, the algorithm runs faster, still finding the optimal paths. This expedited computation time is caused by inspecting a smaller number of nodes. The smallest number is inspected when the value of the heuristic method equals the true distance.

Due to the use of a heuristic measure computed between the source and target node, the use of this algorithm is restricted to finding the shortest path between two nodes (a goal-oriented pathing). Hence, an A* algorithm does not allow to compute the shortest path-tree between the source and all possible targets, as in the case of Dijkstra.

5 Related works in tractography

One of the applications of computational methods in medicine is tractography. Its main purpose is to compute the topology of the neural fibers, based on the DTI sequences of MRI imaging. In preoperative planning it allows to visualize the organisation of white matter near the planned surgery site, providing invaluable aid for neurosurgeons (see section 2.3).

5.1 Methods taxonomy

Figure 5.1 depicts the taxonomy of tractography methods. Because of their convergent purpose, these methods are best classified by their approach. Hence, they can be divided into classical mathematical models and learning models. Mathematical models are the results of methods predicting the orientation of the neural fibers without support from machine learning methodology. These methods can be subdivided into deterministic and probabilistic approaches. The deterministic methods focus on elucidating the deterministic connectivity between regions of interest. However, they are prone to errors due to the noisy nature of MRI. Probabilistic methods take into account the uncertainty of orientation estimates. When creating a streamline in each propagation step, an orientation of a fiber is drawn randomly from the orientation distribution [82]. Even though they are computationally more expensive, they are better suited for high uncertainty regions (e.g. crossing fibers) and where noise is present [83]. Learning models include both classical machine learning approaches and deep learning methods. With the development of these techniques, they have been successfully applied in many areas, such as image recognition and classification [33, 71, 84], the design of drugs [85], or patient's scans [68] and genomes [86].

5.2 Mathematical models

Basser, one of the inventors of DTI imaging itself, proposed one of the first approaches to tracking neural fibers [87]. His method is based on the fact that in each voxel, a trajectory of the tract is parallel to the eigenvector associated with the largest eigenvalue of the local diffusion tensor [10]. The question, though, is how to follow the trajectory of the neural path across multiple voxels. As the author states using the eigenvectors is prone to fail as these are inherently discrete and noisy and are just estimating the true direction of the water diffu-

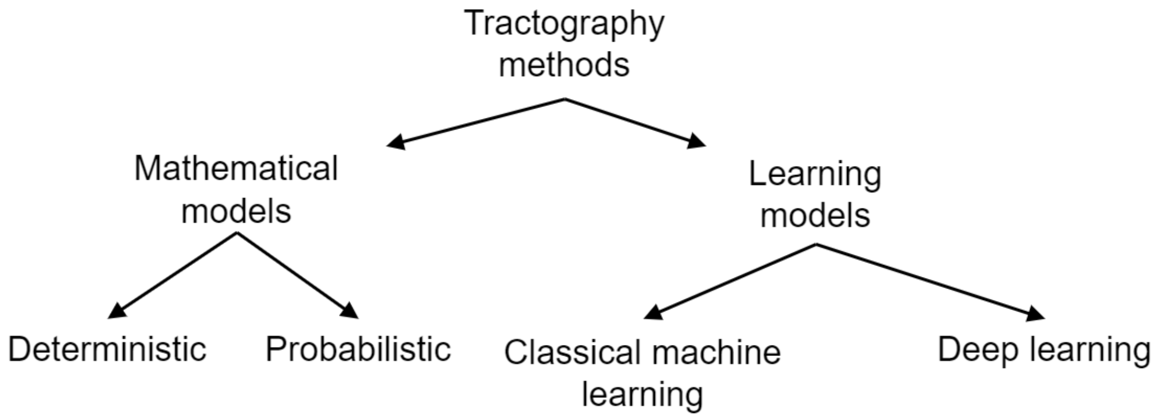


Figure 5.1: Taxonomy of tractography methods

sion [10]. His method calculates a continuous diffusion tensor field. In the next step, based on that field, an eigenvector-field map is computed, which allows using the linear forced vector differential equation to compute the trajectory of a tract, as the trajectory vector is parametrized by arc length [87].

Lazar et al. [88] embarked on using not only the eigenvector of the biggest eigenvalue, but to utilize the information carried by the whole diffusion tensor. This algorithm, called TEND, uses the tensor deflection technique. Tracking is done in a stepwise fashion. The tract direction from the previous step is considered the incoming vector, which is then deflected towards the major eigenvector direction at a given position. The curvature of deflection is limited to achieve smoother tracts. The resulting deflected vector depicts the directions of the neural path in a voxel under consideration. This method introduces stopping criteria allowing to block tract expansion when fractional anisotropy drops below a certain value or when the change of the followed direction changes by more than 45°.

A nerve tract can be conceptualized as a path in the very dense graph representing all anatomical connections within the brain's white matter. Therefore, a group of researchers decided to use path search algorithms, more specifically, some modifications of Djikstra's algorithm [80], to track nerve fibers.

Andrew Zalesky, in his work from 2008, describes fiber tracking as a problem in computing shortest paths in a weighted digraph [89]. In such an approach, a single voxel becomes a vertex, and nodes are placed between vertices representing neighbouring voxels. The weights are computed with a Bayesian framework and reflect the alignment with fiber trajectories in the vicinity. More precisely, it quantifies how likely a given edge is tangential to a small segment of the genuine fiber. A probability for the whole tract is computed as the product of probabilities assigned to the edges. To build such a graph and track fibers this method requires two points as the input: the seed and the target. According to the authors, this method produces the same fiber trajectory between two given points regardless of which one is which.

A similar graph representation of a tracking problem was used by Sotiropoulos et al. [82]. This method is designed for probabilistic tracking through crossing fibers, based on orientation distribution function (ODFs) derived from Q-ball imaging [90]. The main aim of this work is to deal better with crossing regions. In such areas, voxels contain multiple fiber orientations. This poses a challenge to propagate the current trajectory in the direction following the underlying anatomical truth. The algorithm proposed by the authors considers multiple fiber populations occurring in a single voxel (if partial volume exists). Hence the image is treated as a multigraph, allowing to distribute the connectivities in a weighted manner with the most probable tract obtaining the highest weight. Similarly to other methods, vertices represent voxels from the diffusion data, connected with neighbours by edges. Partial paths are constructed in the close vicinity of a voxel under consideration (cube sizes of $3 \times 3 \times 3$ or $5 \times 5 \times 5$) and scored based on the weights of edges. A trajectory with the highest score is then selected as the elongation of the current one.

Work by Aronis et al. [91] proposes a novel cost definition for the graph edges, allowing to take into account tract curvature and its alignment with the diffusion vector field. The only modification to the Dijkstra algorithm, except for the cost function defining the edge weights, is keeping the set of vertices with a determined distance from the source and the distance map holding the current estimation of the distance of each voxel from the seed. Calculating the cost function considers three terms calculated from the vector field. Minimizing these terms favours the smoothness of the calculated tracts and ensures that the transition between the current node and the candidate nodes is as parallel as possible with the diffusion eigenvectors. This method was tested on 2D and 3D synthetic data and the clinical MRI-DTI study to show it can calculate known brain tracts.

MRtrix3 [92] is a freely available software package for medical image processing and visualization. Among many useful tools, it implements global tractography using a multi-tissue spherical convolution model, introduced by Christiaens et al. [93]. It is an approach which extends the method proposed by Reisert et al. [94] to be used with multi-shell response functions. It also adopts a multi-tissue model proposed by Jeurissen et al. [95] to differentiate between white matter (WM), gray matter (GM) and cerebrospinal fluid (CSF). Although interesting, global tractography are not of interest in neurosurgery applications. MRtrix package implements, however, constrained spherical deconvolution [96], allowing to estimate fiber Orientation Distribution Function (fODF). fODF can then be used by tracking algorithms to compute tractograms. MRtrix implements both deterministic and probabilistic approaches [97]. The deterministic algorithm computes a single fiber along its local orientation. The Newton-Raphson gradient ascent algorithm is used to identify the nearest fODF peak iteratively. In the probabilistic approach, a future streamline direction is sampled from the fODF. Sampled directions are constrained to a certain angle from the current location. These methods can produce smooth results and good resolution due to using different data sources (such as HARDI imaging) and step sizes smaller than the voxel size. Tracking is stopped when no satisfying fODF peak can be found or the area is outside the predefined ROI.

The last mathematical model described here was proposed by Descoteaux et al. [98]. In this method, fODF is obtained by applying sharpening deconvolution transform (SDT) of the diffusion ODF from Q-ball imaging [90]. The transform shows new insights into the relation between the HARDI signal, diffusion ODF and the fODF. The sparsity of the fODF and the difference between Q-ball and real diffusion direction compel the use of sharpening operation. Similarly to MRtrix, this method offers deterministic and probabilistic variants. The deterministic approach extends classical streamline generation by considering multiple fODF maxima at each step and the tract is elongated into one from 1281 possible directions. The probabilistic approach extends a random walk method [99, 100], and it uses the information contained in multidirectional fODF. For calculating the topology of the tract, particles are used that moves freely from the seed point, based on the local fODF information. Each voxel is scored based on the number of particles that reached it. Elongation direction is chosen from 120 discrete directions computed from the voxel scores with a step size of half the voxel size.

5.3 Machine learning models

The first machine learning (ML) model for tractography was proposed by Neher et al. [101, 102]. Each streamline is elongated stepwise, similar to the traditional approaches. However, local tissue propensities are not derived from mathematical models. Instead, a random forest classifier lays the directional proposals based on the raw diffusion data. An algorithm considers information from the closest neighbourhood when extending from a given point. The model decides upon a new orientation and outputs probabilities for each direction together with non-fiber probability. Tracking stops when non-fiber probability exceeds the cumulative sum of other probabilities. This study revealed that machine learning helps avoid errors caused by MRI noise and local signal ambiguities.

The random forest classifier described above is the only tractography method using the classical machine learning approach. All the other methods implement neural networks.

Poulin et al. describe two approaches to tractography, using feedforward (FFNN) and recurrent neural network (RNN) [103]. While FFNN returns a three-dimensional vector describing fiber orientation for each point within the diffusion data, the RNN takes advantage of the previously seen voxels by remembering features relevant to the entire streamline orientation. The authors raise the issue of learning a proper stopping criterion. It requires careful engineering and balancing of the loss function. Nonetheless, the presented method achieved high spatial coverage on a given test set, controlling the number of false positives at the same time. Another study carried out by the group postulated using RNN in a bundle-wise manner. It results in improved tracking efficiency, a higher number of valid streamlines, and better volume coverage in comparison to other methods [104].

RNNs were also used by Benou et al. in a method called DeepTract [105]. Like other ML methods, it is suitable to work on various types of raw diffusion data. It estimates the orientation of local fibers as a discrete probability density function, which allows to sample directions at a given point. Choosing the right direction is treated as a classification task, where the model outputs the probability of each orientation.

The methods available to solve the tractography problem are not directly applicable to preoperative planning. Given the noise and artefacts that occur, they usually require a tedious data-cleaning process. Without this, there is a risk of obtaining suboptimal or even erroneous results. Expertise is needed from the user not only in anatomy but also in the DTI imaging, as well as methods used to infer directionality information on the level of voxels. Successful fiber tracking is often possible only when using several available techniques, as their results may differ. Moreover, individual methods are usually prepared for a specific application, such as global tracking, local tracking or connectomics. In preoperative planning, the ability to study fibers in all these aspects is a significant advantage. Thus, there is a need for a method that can work on data that has undergone only basic processing and that does not require substantial domain knowledge. An indispensable advantage would be the ability to determine various structural features of the white matter, such as crossing fibers, or computing only the fibers connecting two specific regions. Such a method could be successfully used by neurosurgeons providing additional knowledge about the white matter near the surgical field.

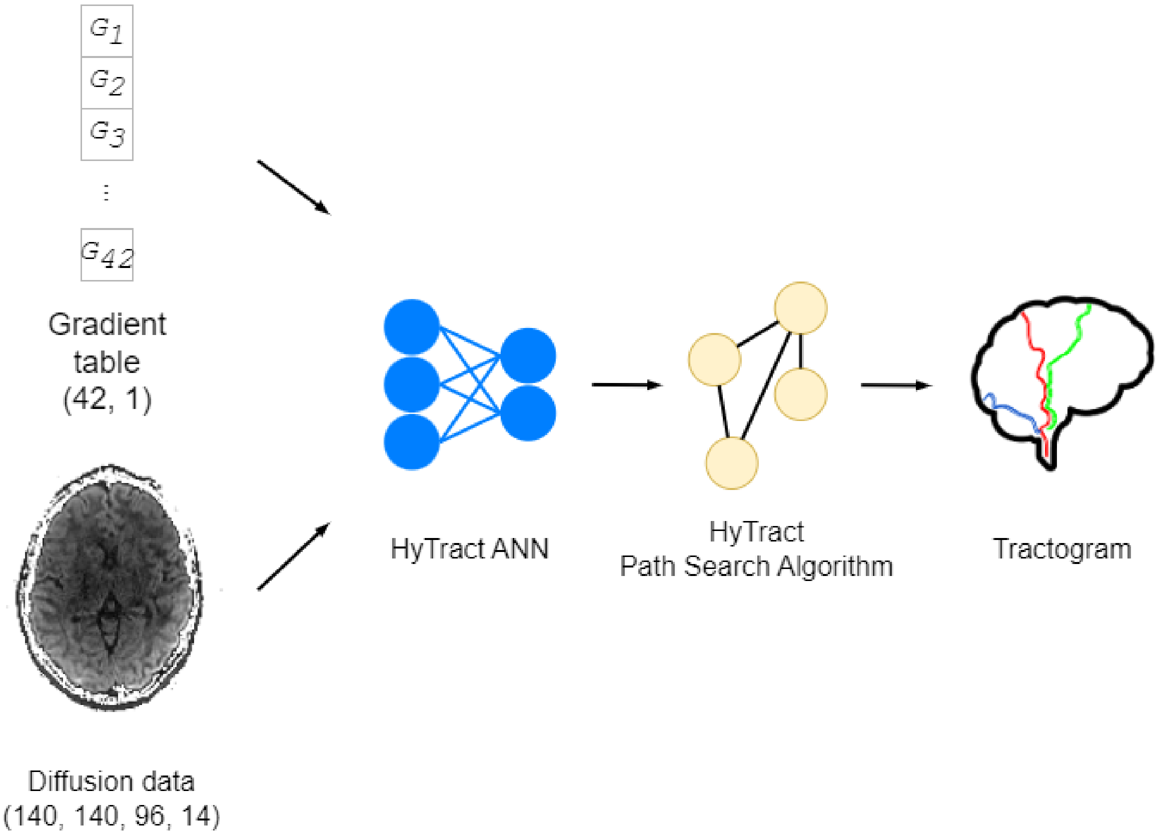


Figure 6.1: HyTract method for tractography

6 Hybrid method for tractography

6.1 Overview

A hybrid method for tracking neural fibers, HyTract (HT), was designed and developed. It combines an artificial neural network (ANN) and a path search algorithm to calculate the topology of the nerve fibers. Preoperative planning does not require a tractogram of the whole brain, hence analyzing the whole scan is unnecessary. Therefore, the artificial neural network processes small samples picked from the entire study. As a result, the amount of analyzed data is small, allowing quick calculations. The neural network model accepts as the input a sample of a predefined size, e.g. $5 \times 5 \times 5$ voxels. It outputs a tensor of likelihoods of the same shape as the input, containing scalar values used by a search path algorithm to calculate the topology of the nerve tract.

Details of the methods and their implementation are discussed in the following sections.

6.2 Automated tracking pipeline

The HyTract method assumes that the neural network processes diffusion data in small portions - cubes with a size of e.g. $(5 \times 5 \times 5$ voxels. Hence, the automated pipeline guides tracking through the white matter.

Algorithm 1 Automated tracking

```
Require: $S$, $s \in S$, set of tracking seed points
Require: $M(D)$, neural network model operating on a three-dimensional diffusion tensor $D$
Require: $A(T)$, Path search algorithm operating on a three-dimensional tensor $T$
Require: $\Upsilon(x)$, Function sampling diffusion data, where $x$ is the central point
Require: $\chi(x)$, Stopping criterion function
  while $S \neq \emptyset$ do
      $Tracts = \{s_i,\}$
      while $\chi(s_i) \neq 1$ do
          $D \leftarrow \Upsilon(s_i)$
          $T \leftarrow M(D)$
          $P \leftarrow A(T)$
          if $|P| > 1$ then
              Append existing tract to each, tracts will have multiple paths
          else if $|P| = 1$ then
              Extend current tract
          end if
      end while
  end while
```

Tracking starts with a set of tracking seed points. These are defined by a user performing tracking and can be picked depending on the aim of the tracking. For instance, if someone wants to compute tracts originating in a given functional cortex area, such tracking seeds can be picked from the interface between the gray and white matter in that area. When attempting neurosurgery, seed points can represent the area adjacent to the planned site of the intervention.

Each seed point is treated as the beginning of the tract. The first step is to draw the desired portion of the diffusion data surrounding the seed point. In all of the descriptions, a cube of the size of $5 \times 5 \times 5$ voxels is used (5 voxels long in each dimension), with the seed point as the central voxel. This small sample of the diffusion data is processed by the ANN, which outputs a three-dimensional tensor of probabilities for each voxel. These values describe how likely the voxels are to contain the same neural fiber as the central one.

In the next step, the path search algorithm is employed to calculate the paths within the cube. The process of tracking within the cube is described in the section 6.4. This algorithm

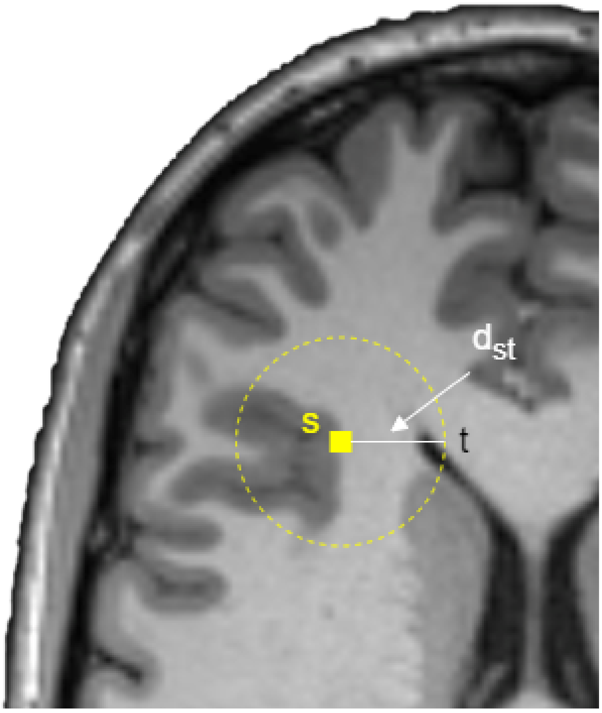


Figure 6.2: Euclidean distance from the seed point as the stopping criterion

returns either a single path or a set of paths if fiber split is detected. If the result is a single path, a current path is elongated. If more than one path is found, the existing path is multiplied, and each copy is extended with the paths found. Consequently, the approach described here is probabilistic, computing all possible paths originating at a given point.

Tracts for a given tracking seed are continuously extended until no other paths are found in a drawn cube (see target selection in section 6.4) or the stopping criterion is met. Considering this system is designed for use in planning neurosurgery, an adequate stopping criterion is a Euclidean distance measured from the beginning of the tract. Figure 6.2 visualizes a seed point (yellow square) and their respective tracking limits (dashed line) at a predefined distance from the origin d_{st}.

6.3 Neural network for diffusion data processing

A neural network processes the diffusion data. The output of this network can be used in the following steps to calculate the streamlines. It is common to design a neural network in one of two ways, either by growing or pruning. Due to the growing method's greater popularity, it has also been adopted in this work.

In each case, we consider input containing two vectors. The first item is the gradient table, containing information about the magnetic field gradient used in the acquisition of each sample during the MRI experiment. The size of the gradient table is $N \times 3$, where N is the number of samples from the diffusion data. Gradient table is the product of row-wise multiplication between the gradient vector matrix and gradient values. The second element of the input data is a series of three-dimensional matrices with diffusion intensities. The number of matrices depends on the specified sample count used or acquired during MRI experiment.

Neural network models designed in this work allow configurable cube size (a portion of diffusion intensity 3D matrix) and the number of samples. For a cube size of M, the input diffusion intensities matrix is of size $M \times M \times M \times N$. Cubes of small size (e.g. $5 \times 5 \times 5$ voxels) can be iteratively drawn from the data, and neural network results are reassembled afterwards. This procedure allows limiting the area for which computation shall be made. The location from which the cube is drawn is not random. The central voxel of the cube always corresponds to a seed point used to start tracking or to a voxel containing part of an already computed streamline (see section 6.2).

The number of samples used can be limited as well. Each sample in the experiment portrays a water diffusion in a specific direction (described by the gradient table) for all voxels. Intuitively, it is possible to compute diffusion intensity for each voxel in all possible directions in the three-dimensional space, using just several selected samples. Each sample measures the strength of the water diffusion in a particular direction described by the gradient vector.

Limiting the size of the part of the diffusion matrix considered (drawing cubes) and the number of samples analyzed by a neural network in a single run allows to decrease the number of neurons in the model without deteriorating its performance. The model's output is a matrix of size $M \times M \times M \times 1$. Each value in the matrix describes how likely does the corresponding voxel contain a continuation of the streamline from the central voxel.

Each architecture presented in this work is initially trained with a cube size of $M = 5$ and a sample size of $N = 14$, i.e. 14 measurements. This means that the size of a gradient table is 14×3 or 42×1 when flattened. A fragment of diffusion data (a cube) providing input to the model will have the size of $5 \times 5 \times 5 \times 14$, or 1750×1 when flattened. Architectures are depicted for $M = 5$ and $N = 14$.

6.3.1 Minimal architecture

The simplest deep-learning neural network contains at least three layers: an input layer, a hidden layer, and an output layer. Such architecture has been used as the first attempt in this work. Figure 6.3 depicts this model in more detail, with the input-output size of each layer. B_2^3 denotes the second value of the gradient vector for 3rd sample, V_3 a gradient value for the 3rd sample, G elements of a gradient table vector, I elements of the diffusion input vector, and F elements of the input feature vector. The gradient table and diffusion data sample are flattened to a vector of 42 and 1750 scalar values. Both input vectors are concatenated to produce a feature vector of 1792 values.

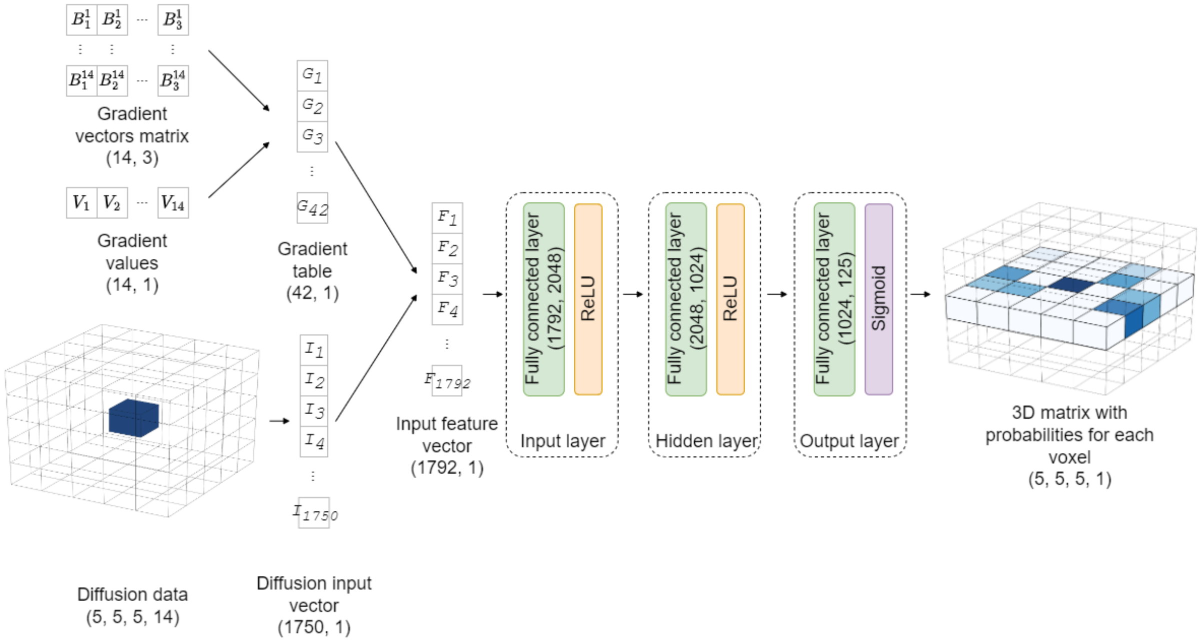


Figure 6.3: Minimal model

The input layer is the fully-connected (FC) layer accepting 1792 features and is activated by the ReLU function. The output of this layer is a 2048 × 1 vector, matching the input of the hidden layer, which also has ReLU activation. The output of the hidden layer is 1024 and is passed to the model output layer, followed by a sigmoid function. The model's output is the three-dimensional matrix of probabilities for each voxel, as described in the introduction to this section.

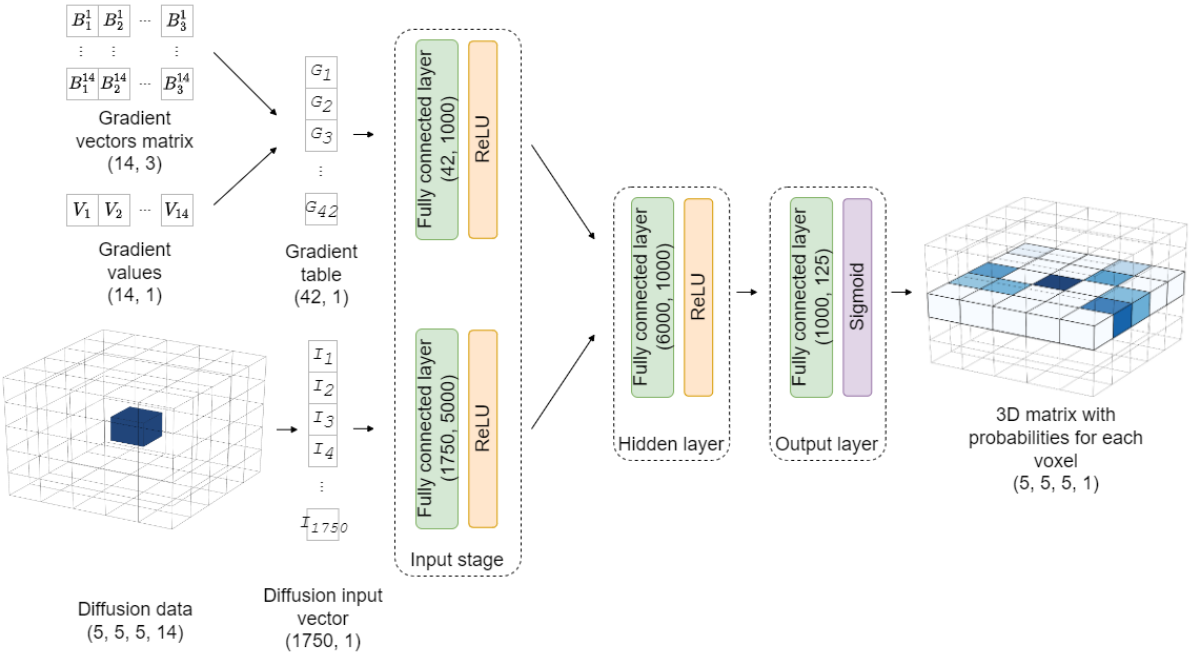


Figure 6.4: Two-fold-input model

6.3.2 Two fold input model

The Two-fold-input model is an extension of the minimal network model. Architecture of this model is depicted in figure 6.4. The input layer is replaced with an input stage. It consists of two FC layers, processing the input data in parallel. The gradient table part results in a vector of size 1000 × 1. The diffusion data part results in the vector of size 5000 × 1. Both vectors are concatenated to create a single vector of size 6000 × 1, passed to the hidden layer with ReLU activation. The output layer accepts a vector of size 1000 × 1. It is followed by a sigmoid function to compute a three-dimensional matrix of probabilities for each voxel as in the minimal model.

Replacing a single input layer with two independent layers allows the network to have two sets of unconnected weights to process the input data. This stage's output size was increased compared to the minimal model. However, due to efficient parallelisation, it does not influence efficiency. This procedure allows using more neurons in the hidden layer. As a result, this layer intuitively has a higher capacity to learn the relationship between two types of data.

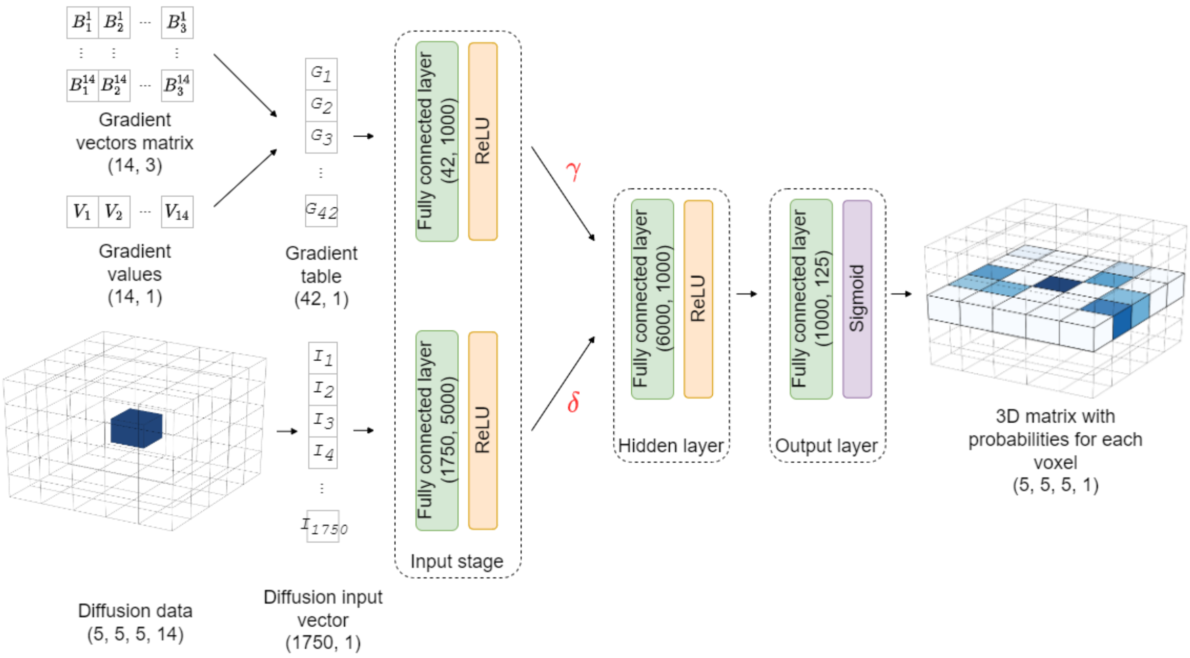


Figure 6.5: Two-fold-input model with attention weights

6.3.3 Two fold input model with input attention weights

The two-fold-input model with attention weights extends the two-fold-input model described in the previous section. When concatenating the two vectors from the input stage, each is multiplied by a scalar value, γ and δ respectively (Fig. 6.5). Both are model parameters, and their values change during the learning process. These values will reflect the importance of individual data in the learning process, increasing the model's performance. Due to their ability to weaken or strengthen the signal, they were named input attention weights.

6.3.4 Two-fold input model with convolution

Even though fully-connected layers have the potential to simulate all the other, more specialized layers, the other architectures were tested as well. The two-fold input model with convolution, presented in figure 6.6, is very similar to other architectures presented above. The main difference is in the diffusion data input layer, which was replaced with a one-dimensional convolution layer. It accepts 125 channels on input and outputs 250 channels. The kernel size for convolution is set to 5 and stride to 2. The flattened output of this layer is a vector of size 1250×1. The gradient processing layers' output has decreased and is left the same as its input. After merging the two vectors, the input to the hidden layer has a size of 1292×1. The output of the hidden layer, as well as the output layer of the network itself,

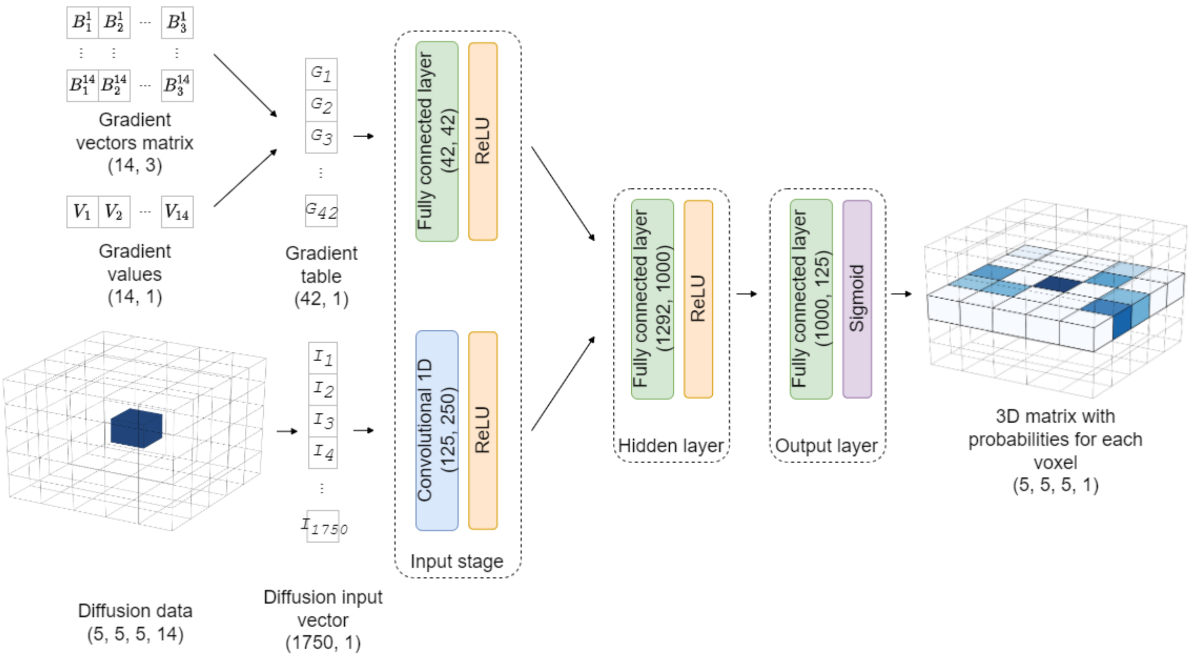


Figure 6.6: Two-fold-input model with convolution layer

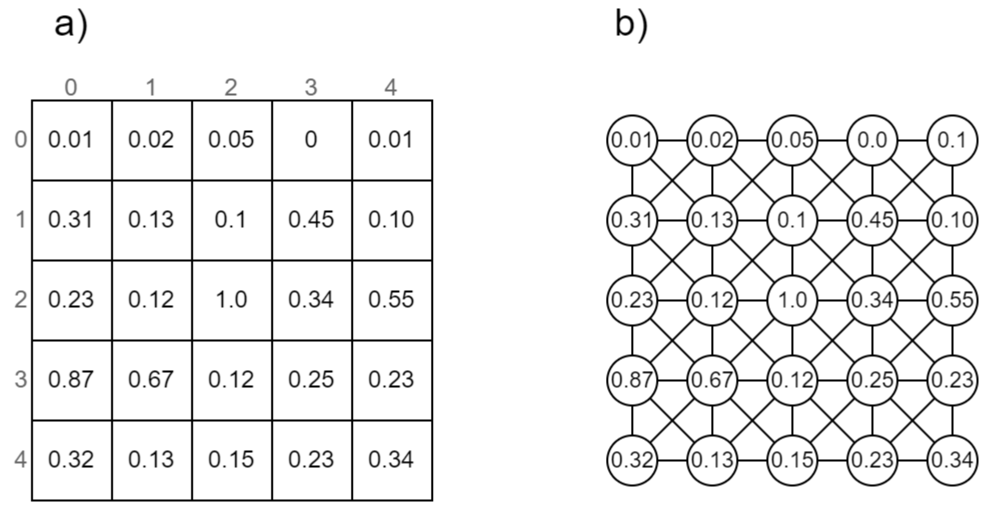
Architecture	Number of layers	Number of parameters
Minimal model	3	5 898 365
Two-fold input model	4	14 938 125
Two-fold input model with weights	4	14 938 127
Two-fold input model with convolution	4	1 576 431

Table 6.1: Number of trainable parameters in architectures

remains unchanged in comparison to the architectures described previously.

6.4 Modified A* algorithm

The output of an artificial neural network is a three-dimensional matrix of specific dimensions (e.g., 5 x 5 x 5 voxels). Each element of this matrix corresponds to a voxel from the input sample, and each value is the probability with which a given voxel has the same nerve fiber as the central voxel. This data is used to build a graph in which the vertices correspond to individual voxels and, consequently, to particular values of the resulting network matrix. Edges connect the vertices to map the voxels' immediate vicinity. Figure 6.7 shows the graph built from the network's output. For simplicity, the concept is shown in two dimensions. Each neighbour of a given node has a weight assigned to it. This weight equals a probability value given by the neural network, describing how likely it is that a given voxel (node) contains the same neural path as the central voxel at position $[2, 2, 2]$. By design, a central voxel is treated



	0	1	2	3	4
0	0.01	0.02	0.05	0	0.01
1	0.31	0.13	0.1	0.45	0.10
2	0.23	0.12	1.0	0.34	0.55
3	0.87	0.67	0.12	0.25	0.23
4	0.32	0.13	0.15	0.23	0.34

Figure 6.7: Constructing a graph from neural network output. a) Example of the neural network output; b) A graph representation of the neural network output

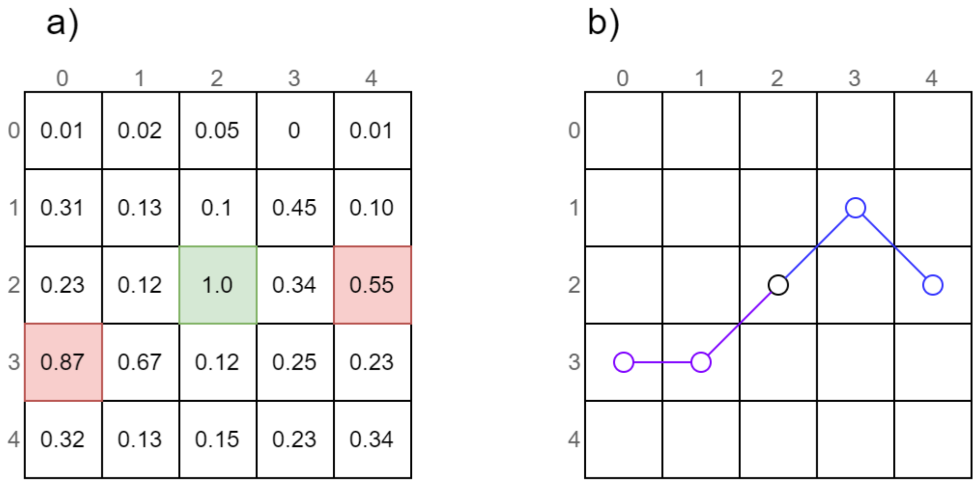
as containing a neural path. Hence other voxels in a cube are considered in this respect.

The HyTract method aims to build a tractogram - the topology of the nerve fibers. To achieve this a path search algorithm is required to find the most probable paths in the graph, representing neural fibers. Some of the classical methods described in section 5.2 use a modified Dijkstra algorithm to reconstruct the trajectory of a neural path. In this work Dijkstra algorithm was also tested (data not shown). However, in this case nodes represent voxels, which have well-defined position in the three-dimensional space. In such a case an Euclidian distance can be used as a good approximation of the target distance between two nodes, allowing to take the advantage of an A* variant of Dijkstra.

In HyTract implementation of the path search, an A* algorithm starts at a given starting node (seed point) and aims to find a path to the given goal node having the smallest cost. This objective is achieved by keeping a tree of paths that originate at the seed point and are extended one node at a time until the stopping criterion is met. At each step, an algorithm must estimate which path to extend. It is done based on the cost of the path so far and an estimate of the cost required to extend the path to the target node. It can be achieved by minimizing the following

$$f(v_n) = g(v_n) + h(v_n, v_t), \tag{19}$$

where v_n is the current node, v_t is the target node, $g(v_n)$ is the cost of the path so far, and $h(v_n, v_t)$ is the heuristic function, estimating a cost of a path from the current point to the target. To use neural network output probabilities for minimizing $f(v_n)$, the cost for a given



	0	1	2	3	4
0	0.01	0.02	0.05	0	0.01
1	0.31	0.13	0.1	0.45	0.10
2	0.23	0.12	1.0	0.34	0.55
3	0.87	0.67	0.12	0.25	0.23
4	0.32	0.13	0.15	0.23	0.34

Figure 6.8: Source and target points for a path search in a grid with possible paths found by an algorithm

node v_n have to be computed as a probability subtracted from 1:

$$\varrho_{v_n} = 1 - \hat{y}_n, \quad \hat{y}_n \in [0, 1], \tag{20}$$

where $\hat{y}_n$ is the neural network output for a node n. The cost for a path so far becomes a simple sum of all the nodes in the path:

$$g(n) = \sum_{n=1}^{\zeta} \varrho_{v_n}, \quad \zeta \in \mathbb{N}, \tag{21}$$

where ζ is the total length of the path so far (streamline). The heuristic function used is a Euclidean distance from the current node to the target

$$h(v_n, v_t) = \sqrt{(v_{n1} - v_{t1})^2 + (v_{n2} - v_{t2})^2 + (v_{n3} - v_{t3})^2} \tag{22}$$

where $v_n = (v_{n1}, v_{n2}, v_{n3})$ is the current node, and $v_t = (v_{t1}, v_{t2}, v_{t3})$ is the target node.

A path search is performed in small samples, as described in the HyTract definition. For described grid size of (5×5) the central voxel $[2, 2]$ is the starting point (marked green, Fig. 6.8, a). The targets, $[2, 4]$ and $[3, 0]$ are nodes on the border of the cube with probability value equal to or greater than the predefined threshold, e.g. $\Psi = 0.5$ (marked red, Fig. 6.8, a). This threshold has to be picked based on the mean prediction value for a positive class, calculated from predictions during the training. It indicates the value Ψ above which predictions are correctly classified as containing neural fiber.

Two paths can be computed for an example grid shown in figure 6.8 a. One of them, marked blue, contains voxels $([2,2],[1,3],[2,4])$. The second, marked purple, contains voxels $([2,2],[3,1],[3,0])$ (Fig. 6.8, b). Depending on the paths found in other cubes, a path marked with blue can continue the purple one if purple has already been marked. Such a situation would occur if a purple path in the previous steps had been selected as the connection between tracking seed and current $[2,2]$ central voxel. If another path leads from the tracking seed to the central voxel, the situation depicted in figure 6.8 can be interpreted as a splitting fiber.

6.4.1 Splitting fibers

Splitting fibers are typical in brain organization, where a single fiber bundle splits into two, connecting the source area with two or more target areas. It is easy to detect such cases by using a path search with A* on a set of cubes. If we consider the situation depicted in figure 6.9 in the first two grids (a and b), a tract from point A (seed point) to point B (border target on the second grid, b) has been computed. Drawing the next grid (c), we see three paths originating from the central voxel (point B, the target in the previous grid). Gray path has already been marked in the previous cube. The green path is identified as connecting central new voxel B with border target C. The blue path connects central voxel B with another border target D. In that case, none of the paths ($B \rightarrow C$ and $B \rightarrow D$) was seen previously, indicating splitting fibers. In such a case, there is no certainty about extending the current streamline into a single direction. Both paths, $A \rightarrow C$ and $A \rightarrow D$, are probable and are be included in the results.

6.4.2 Tract smoothing

Tracks computed on the resolution of voxels can have sharp turns. This situation can be observed in an example on the upper left pane in figure 6.10. Neural fibers in the body do not arrange themselves in such a way. A simple moving average method (SMA) described in [106], can smooth out the fibre's topology.

For each node v, average coordinates in the three-dimensional space can be calculated for a given window size z, where the window size corresponds to the currently analyzed por-

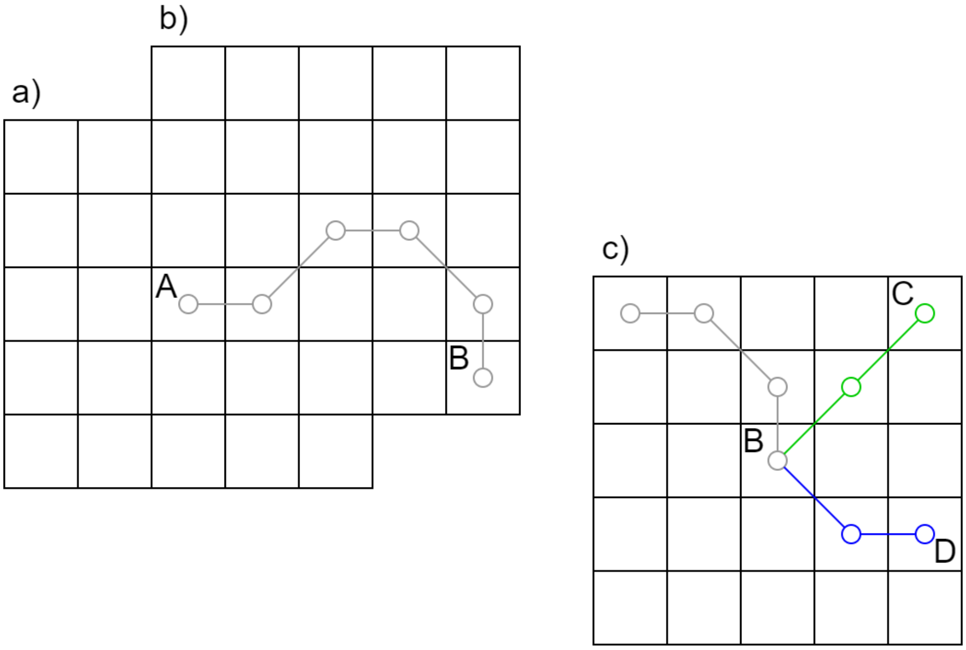


Figure 6.9: Detecting splitting fibers

tion of the fiber with length ζ:

$$\begin{aligned} SMA_z &= \frac{v_{n-z+1} + v_{n-z+2} + \ldots + v_{n+z}}{z} \\ &= \frac{1}{z} \sum_{i=n-z+1}^{n} v_i \end{aligned} \tag{23}$$

SMA is commonly used in financial applications [107]. In this case, for each node, a nodes $v_{n-k+1} + v_{n-k+2} + \ldots + v_{n+k}$ are taken into account to reflect the overall fiber topology better. Figure 6.10 depicts using SMA with window sizes z set to 3, 5, and 7 on a crafted example. It is visible that the larger the window size, the more smooth the fiber becomes. However, the anatomic relevance of such smoothing is in question, like in the U-turn marked with arrows on the lower panes. In this case, the smoothed topology deviates significantly from that determined by the algorithm (red and blue lines). For this reason, smaller window sizes should be used to more faithfully reflect the results of the path search algorithm (green line, upper right pane). Even using such a small window size, SMA can fix artefacts created by favouring diagonal transitions between voxels (marked with arrows, upper right pane in Fig. 6.10).

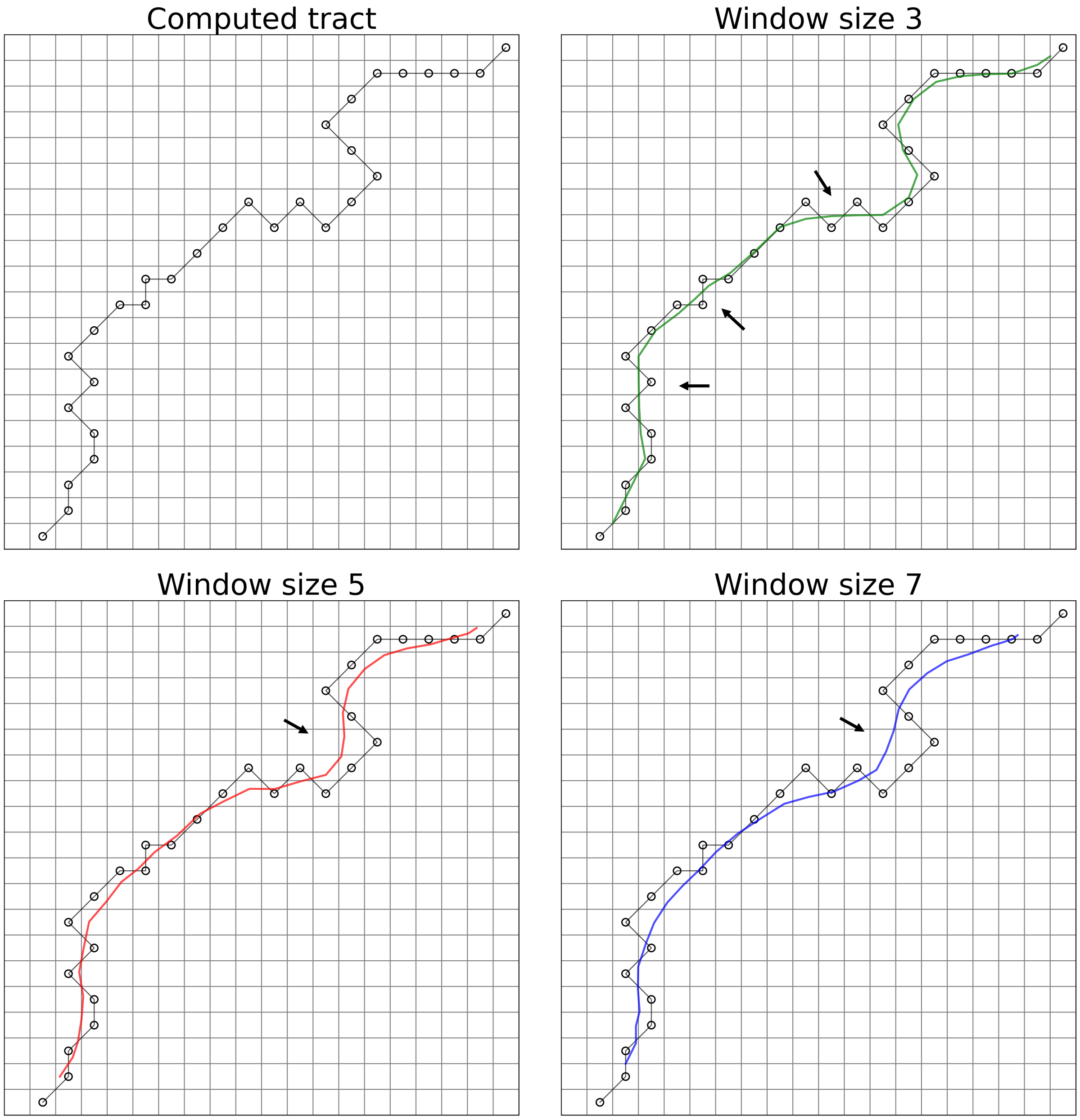


Figure 6.10: Smoothing calculated tracts with SMA; k=3, k=5, k=7.

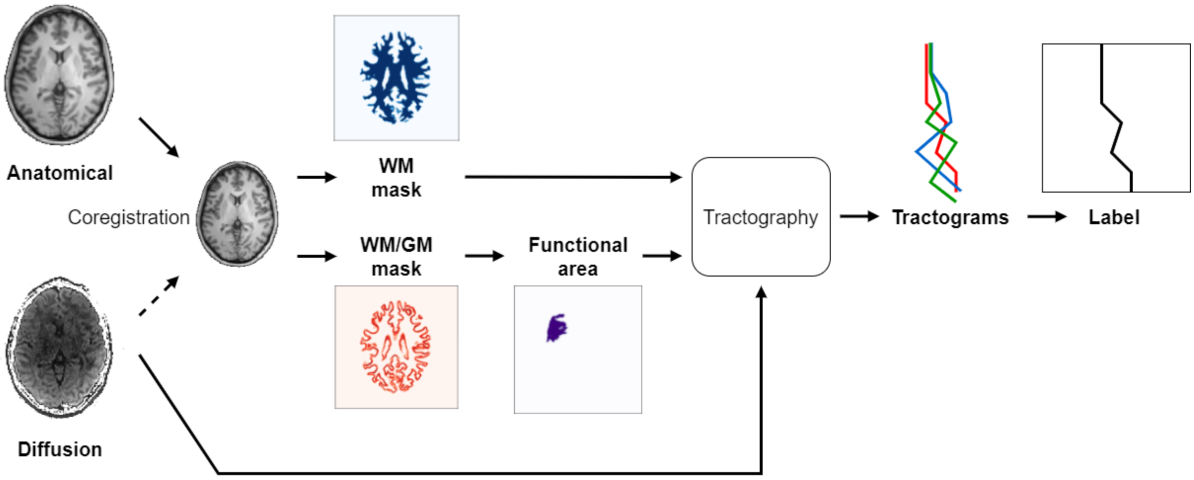


Figure 7.1: Dataset preparation outline

7 Diffusion data analysis and processing

Artificial neural network is one of the key elements of the HyTract method. Training this network requires the preparation of appropriate labels. For this purpose, data from the diffusion tensor imaging (DTI) method and anatomical imaging (T1, see Fig. 2.9) were used. After proper preparation, they were used to create tractograms using the three methods available in the Dipy library [108]. The fibers confirmed by at least two methods were used to create the labels (tractograms), thus substantially decreasing the number of false positives. Available tools, libraries and platforms were reviewed to solve the problems discussed in this chapter. Based on this selection, an efficient computational pipeline was proposed to obtain labels for artificial neural network training.

Creating labels can be divided into four stages presented in Fig. 7.1: coregistration, calculating brain masks, tractography, and label creation. These steps will be discussed in more detail in the following subsections.

7.1 Scan coregistration

The initial step in the label preparation pipeline is the coregistration of different MRI images. Anatomical scans are usually registered in a different resolution than diffusion data. In the case of the HCP subjects, T1 and T2 scans are done with a voxel size of 1mm, while the diffusion is with a voxel size of 1.5 mm. Thus the anatomical scan has to be coregistered to the diffusion scan space to relate spatial information properly.

The first step is to extract the mean B_0 image from the set of diffusion measurements, which can be achieved by dwiextract tool from MRtrix3 package [92]. This image is a rigid reference to which the T1 image will be aligned. Registering T1 to B_0 is done using FMRIB's Linear Image Registration Tool (FLIRT) [109, 110, 111] from FSL software package [112, 113, 114] using six degrees of freedom for transformation. The resulting files are the T1 image in the same space as the diffusion image and a transformation matrix. The resulting T1 image is then used to calculate all the necessary masks used in the further steps.

7.2 Brain masks computation

To prepare tractograms based on which the labels were created, several brain masks were required. These include a white matter mask, a white-matter-gray-matter interface mask, and a functional area mask.

White matter mask can be computed from T1 scan data. Its main role is to target tractography algorithms to generate streamlines in regions containing white matter tissue, where the neural fibers are expected. Firstly a five-tissue-type (5TT) segmented tissue image has to be computed from the T1 image. The 5ttgen tool [115] from MRtrix3 package [92] has been employed. At this stage, to limit the amount of data processed, a non-brain tissue is removed from the image by Brain Extraction Tool (BET) [116] from FSL. White matter mask (WM mask) can be then easily extracted from 5TT image cropped to the brain area with mrconvert tool from MRtrix3 package (Fig. 7.2).

Due to the underlying anatomy of the neural connections, neural tracking usually starts on the interface between gray and white matter. Mask identifying this interface, gray matter-white matter interface mask (GMWMI mask), can be obtained by using a dedicated tool 5tt2gmwmi [115] from MRtrix3 package (Fig. 7.2).

Tractography of the whole brain is an interesting field of study. However, in neurosurgical applications, it is more important to track the fibers coming out of a specific eloquent region of the cerebral cortex. In an ideal scenario, this region should be determined by fMRI. For the presented studies, Juelich histological atlas was used to obtain the location for Broca's region (see Fig. 2.3) [117] and area V1 from the visual cortex [118]. Functional regions in the atlas have been annotated on MNI152 1mm brain, a template T1 image with 1mm voxel resolution [119, 120]. This image was created by linear co-registration of 152 normal T1 images to the common space. The main purpose of this venture carried out by the International

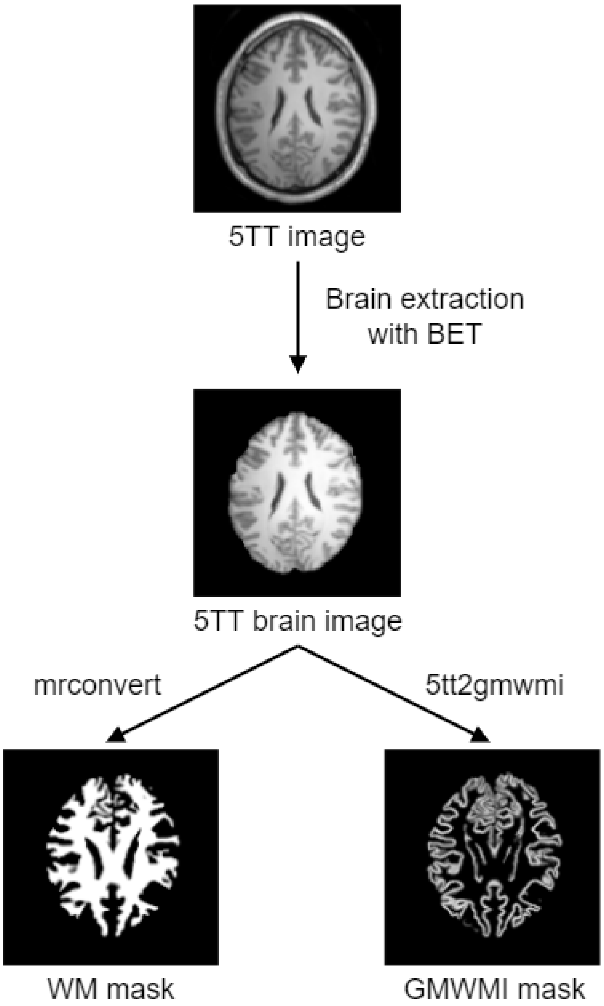


Figure 7.2: Masks extraction

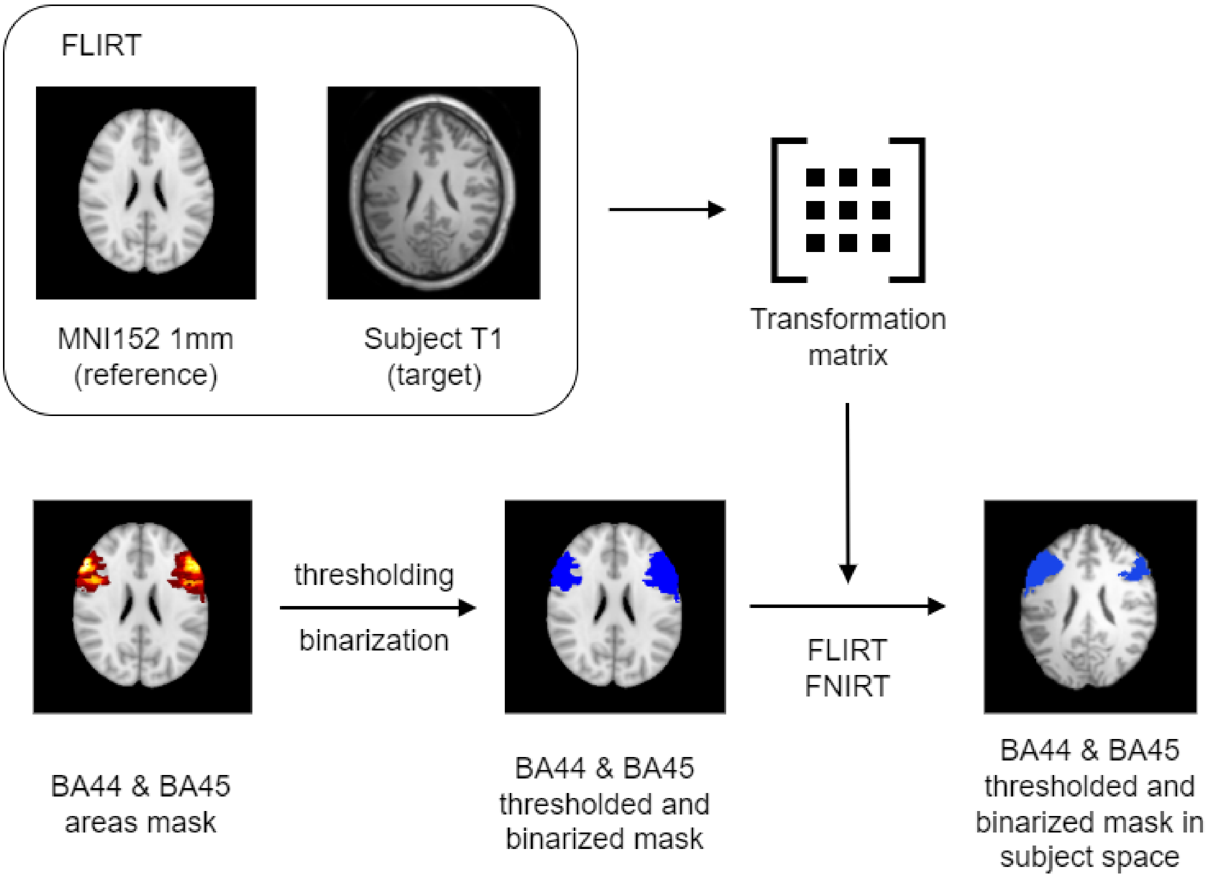


Figure 7.3: Calculating functional mask

Consortium of Brain Mapping (ICBM) was to define the standard anatomy. Herewith, it is possible to create masks for functional regions in MNI152 space, which can then be transformed into the subject space. The values represent the likelihood of belonging to the functional area of interest. FLIRT and FNIRT [121] tools were used to generate a transformation matrix from the subject space to the MNI152 space. This transformation matrix is used to transform functional area masks to the subject space as presented in Fig. 7.3. To use masks efficiently, it is a good practice to binarize them and use only voxels above a given threshold $\Psi \in [0,1]$. For this study, a low threshold of $\Psi = 0.1$ was used not to narrow the starting region too much.

The last computed mask is the gray matter-white matter interface in the functional region, obtained by combining the GMWMI mask with a functional mask.

7.3 Computing tractograms for training labels

To calculate labels to train the models, tractograms were built for nerve fibers originating in the eloquent regions responsible for speech generation. Voxels on the interface between

Figure 7.4: Preparing tracking seeds

white and grey matter within the Broca region in the left and right hemispheres (BA44 and BA45) were used as the seed points. For this, a GMWMI mask was combined with an appropriate functional mask, computed as described in section 7.2. The process is depicted in Fig. 7.4.

In addition to the seed voxels, local tracking requires a method estimating the direction of water diffusion from MRI data and a stopping criterion defining where to stop tracking. Given the high incidence of false positives in tractography results, three algorithms were used: EuDX with Constant Solid Angle (CSA) model [122], Deterministic Maximum Direction Getter (DMDG) [108], and Closest Peak Direction Getter (CPDG) [108]. These methods compute streamlines, designated as a set of adjoining points in a three-dimensional space. Streamlines generated by these methods are combined into a “meta-tractogram” used to compute the labels. Label creation is described in the following section 7.4.

The Constant Solid Angle ODF model fits diffusion data to estimate the Orientation Distribution Function (ODF) at each voxel [123]. The ODF characterizes water diffusion as a function of direction. Peaks of an ODF model can be used to calculate fiber orientations in all voxels of white matter using the EuDX algorithm [122]. In this work, the implementation provided in the Dipy library [108] was used. The CSA model makes use of the spherical harmonic (SH) basis. SH order was set to 6. Peaks were computed directly from the CSA ODF model, with the default sphere providing discrete directions for evaluation. The relative peak threshold was set to 0.8, and the minimum separation angle to 45°. This method also uses a white matter mask to restrict the area where tracking occurs. The stopping criterion was used to hamper tracking to the areas where the ODF shows significantly restricted diffusion. It is achieved by thresholding on the generalized fractional anisotropy (GFA), where a value of 0.1 was used.

Another method used is the Deterministic Maximum Direction Getter (DMDG). It follows the trajectory of the most probable pathway within the tracking constraint. This algorithm can be used with the same local model and parameters as the previously described approach. However, it does not follow the peaks of the local model like EuDX tractography but rather uses the entire orientation distribution. Diffusion data were fitted with the Constrained Spherical Deconvolution (CSD) model to provide the DMDG algorithm with the spherical harmonic representation of the Fiber Distribution Function (FOD). CSD model was fitted only in the voxels belonging to the white matter. Tracking was performed with a maximum separation angle of 30° and a default sphere. The stopping criterion was the same as in the EuDX algorithm, with a threshold of 0.1.

The last method used for computing tractograms in the subjects was the Closest Peak Direction Getter method. It uses the CSD model to fit diffusion data, similar to the case of the DMDG algorithm. A maximum separation angle of 30° was used. The only difference in parameters is in the sphere used. Here, a small sphere is recommended for the best algorithm performance instead of the default sphere used in the previous algorithms. A stopping criterion was used with the same parameters as in the case of the EuDX and DMDG approaches.

Figure 7.5 presents a histogram of streamlines computed using the abovementioned methods. It clearly shows that the EuDX algorithm produces the most streamlines compared to other methods. Most of them are short, depicting connections within the Broca region itself, as depicted by a red arrow in figure 7.6. The blue arrow indicates neural fibers forming towards the posterior part of the brain. These could indicate a connection with a Wernicke's area located in the superior temporal gyrus of the temporal lobe.

Meta-tractograms were created based on the results obtained by all three methods. Since the EuDX algorithm produced the largest number of streamlines, it was used as a reference to which other methods were compared. For each streamline from the reference set, the fiber closest to it in the other two methods was sought to confirm it. One of the parameters considered was the length of the two streamlines. The shorter one should contain at least as many points as 80% of the longer one. Those that do not meet this criterion were skipped automatically. The second criterion is the Mean Euclidian Distance between two streamlines:

$$MED = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(a_{ix}-b_{ix})^2+(a_{iy}-b_{iy})^2+(a_{iz}-b_{iz})^2}, \tag{24}$$

where N is the length of the shorter streamline, a and b, are voxels of two streamlines being compared with coordinates defined by ix, iy, iz for the x, y, and z planes. For each reference

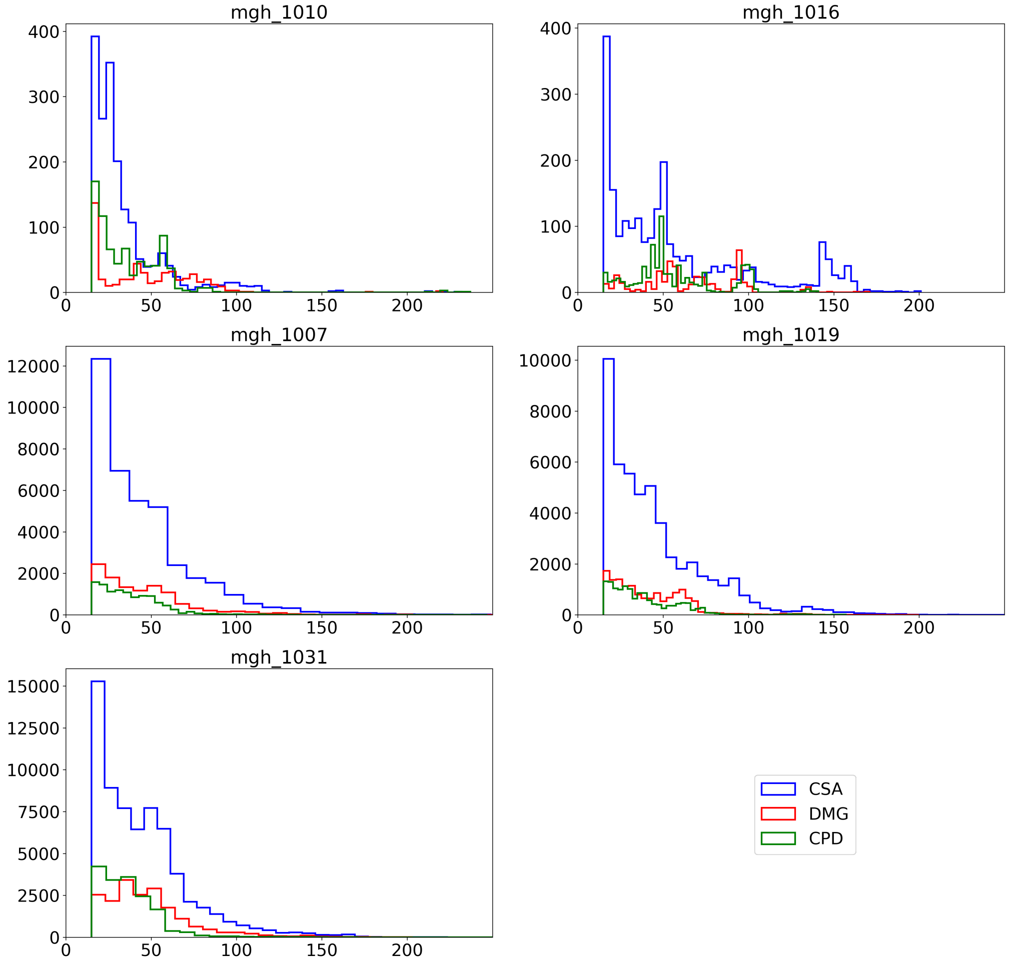


Figure 7.5: Streamline lengths histograms

streamline, a streamline of the lowest possible MED was picked from the two other methods. Meta-tractogram contains those streamlines confirmed by similar streamlines computed by DMDG or CPDG methods with MED smaller or equal to a given threshold.

Figure 7.7 presents meta-tractograms created based on three thresholds of MED value 1, 2, and 3. The reference tractogram, computed with the EuDX method, is shown for comparison. Using a MED value threshold of 1 is too strict, as many essential connections with distant regions are lacking (marked with yellow arrows). They appear when a MED value threshold of 2 is used. However, the amount of streamlines not included is still high, leading to the assumption that this threshold is too strict. A MED value threshold of 3 was chosen to

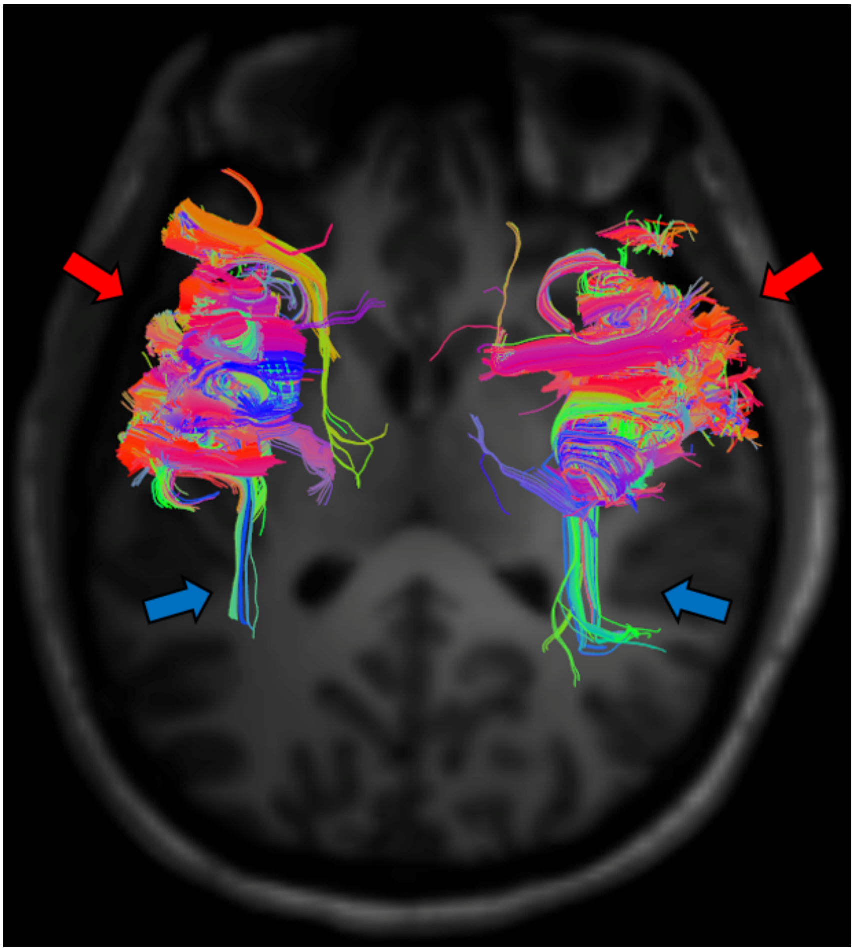


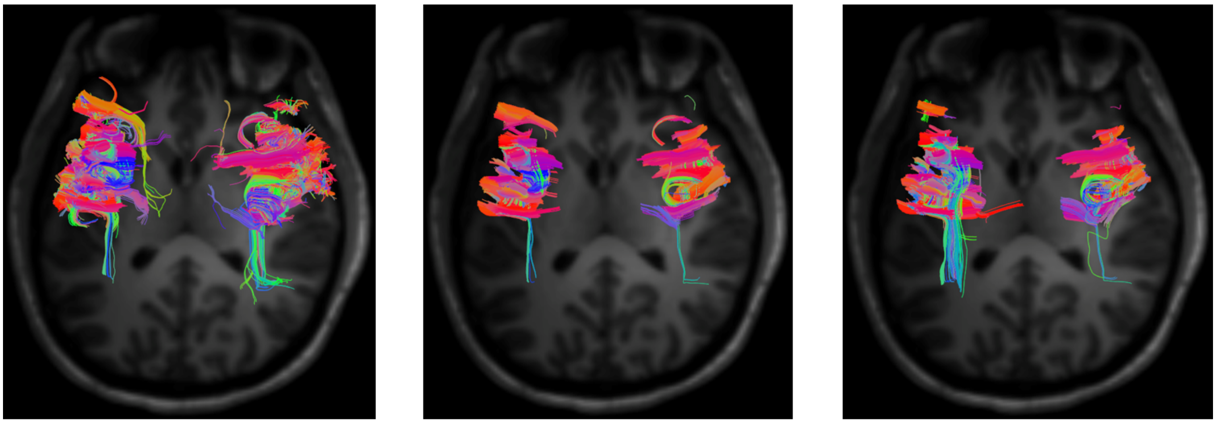
Figure 7.6: Visualisation of EuDX tractogram

allow for more possible pathways, simultaneously leaving out the least probable ones.

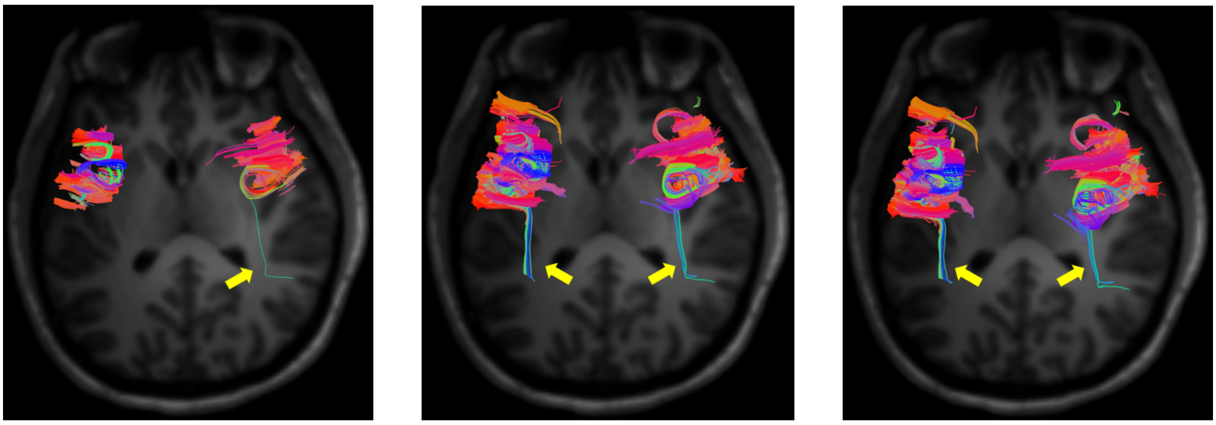
7.4 Creating labels

The labels are created directly from the meta-tractograms. Figure 7.8 depicts (in 2D for simplicity) how a streamline is marked in the three-dimensional space of the diffusion data.

For each streamline in the meta-tractogram (e.g. A and B in Fig. 7.8) labels are computed independently. If a streamline occupies a particular voxel, it is marked with 1. All the other voxels are marked with 0. In this way, a binary mask is obtained, marking the projection of a streamline in the same space as the diffusion data.



EuDX CPDG DMDG



MED = 1 MED = 2 MED = 3

Figure 7.7: Meta-tractograms with different MED thresholds

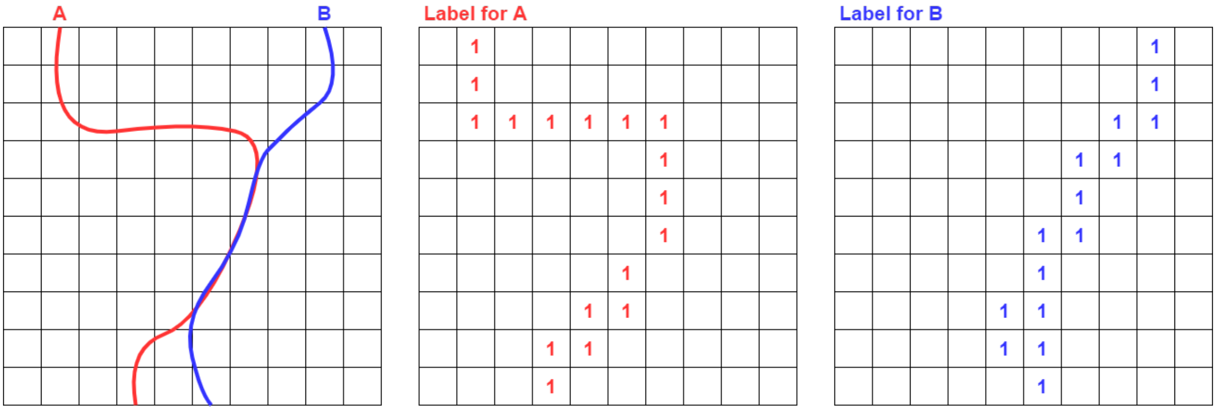


Figure 7.8: Creating labels from meta-tractograms

8 HyTract ANN training and validation

8.1 Experimental setup and datasets

8.1.1 Test scenarios

The research focused on comparing the efficiency and performance of different variants of artificial neural network models and different shortest-path algorithms. In the case of the models, tests were performed with the aim of:

1. selecting the best network architecture for a given task
2. selection of an algorithm for the optimal determination of ANN weights in terms of accuracy and performance
3. comparing the performance of the method for different weight decay parameters and selecting the best value
4. studying the effect of cube size on the quality of the solution.

To evaluate the results of the tests, the area under the ROC curve (AUC) was used [124, 125].

8.1.2 Dataset

Data from the Human Connectome Project (HCP) database (https://ida.loni.usc.edu/login.jsp) was used. HCP is the result of efforts of co-investigators from the University of Southern California, Martinos Center for Biomedical Imaging at Massachusetts General Hospital (MGH), Washington University, and the University of Minnesota [126, 127, 128, 129, 130].

Training and validation subsets were prepared to train the models, based on the data from five subjects randomly selected from the HCP dataset (case ids: mgh_1007, mgh_1010, mgh_1016, and mgh_1019). Meta-tractograms prepared for each subject were split into training and validation in the proportion of 8:2 randomly. Thus streamlines from each subject are represented in both subsets. The dataset was processed as described in chapter 7.

8.2 Results

8.2.1 Various architectures and learning parameters

When using neural networks, choosing the right network architecture is important. This experiment was designed to elucidate the best-performing architecture from those proposed in section 6.3. A set of learning parameters were chosen to compare the results reliably. Each architecture was trained for 100 epochs. The choice of 100 epochs as the training length was based on previous experiments (data not shown). During these experiments, a deceleration in the decline of the loss value and an increase in prediction accuracy was observed. Therefore, using such a constraint to compare different proposed architectures and learning parameters in a reasonable amount of time was deemed reasonable. Experiments were carried out with three learning rates 1e-3, 1e-4, and 1e-5. This approach prevents model rejection due to the lack of generalization caused by a high learning rate. The experiments were run using the Adam optimizer and Binary Cross Entropy loss function [31].

For each training, two plots are drawn (e.g. Fig. 8.1 for Minimal Model), depicting the loss function minimization (a) and network performance measured by calculating the area under the ROC curve (b).

Figure 8.1 shows the training process of Minimal Model architecture (presented in Fig. 6.3). The best results in terms of the loss function minimization and network performance were obtained using a learning rate of 1e-4 with a loss value of 0.1082 and ROC AUC of 0.9531 measured on the validation dataset. Values of the minimal loss and maximal ROC AUC for all experiments are summarized in table 8.1 at the end of this section.

The training process of the Two-Fold Input Model (presented in Fig. 6.4) is presented in Figure 8.2. Similarly to the Minimal Model, the best results were obtained when training with a learning rate of 1e-4. For this experiment, minimal loss on the validation dataset reached a value of 0.1028 and ROC AUC of 0.9587.

Figure 8.3 depicts training of the Two-Fold Input Model with attention weights (pre-

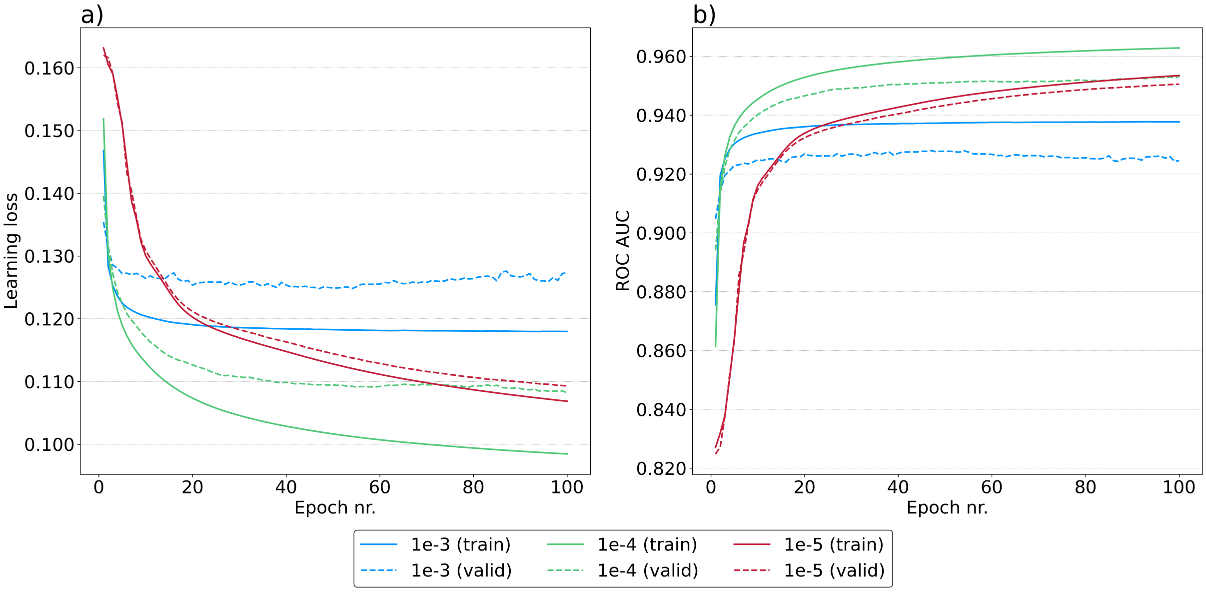


Figure 8.1: Minimal model training

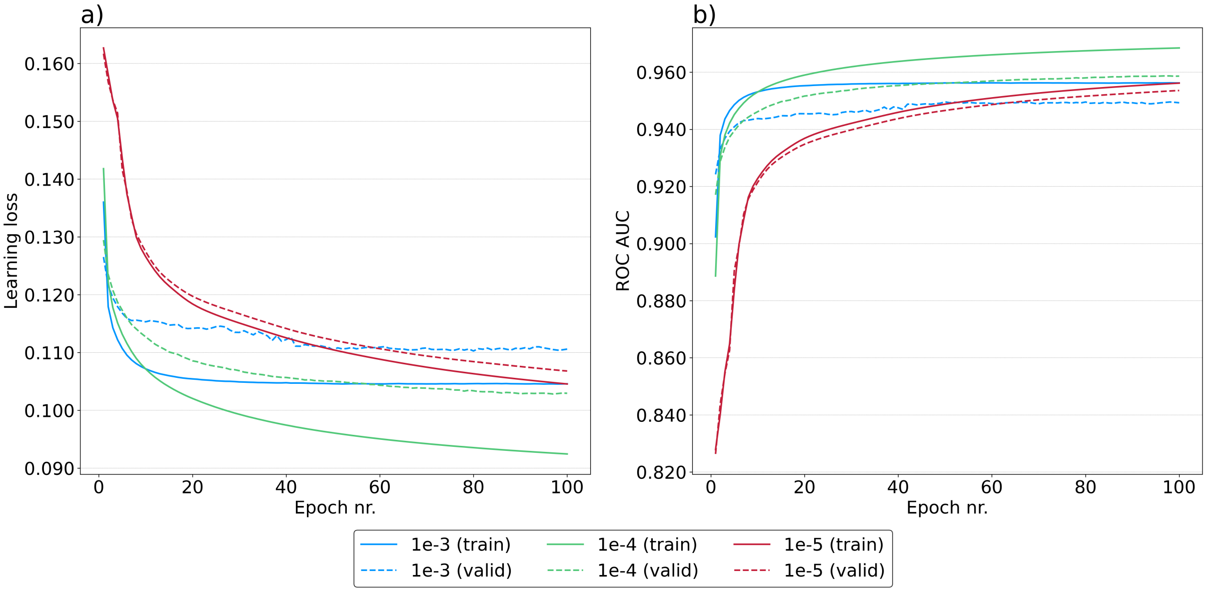


Figure 8.2: Two-fold input model training

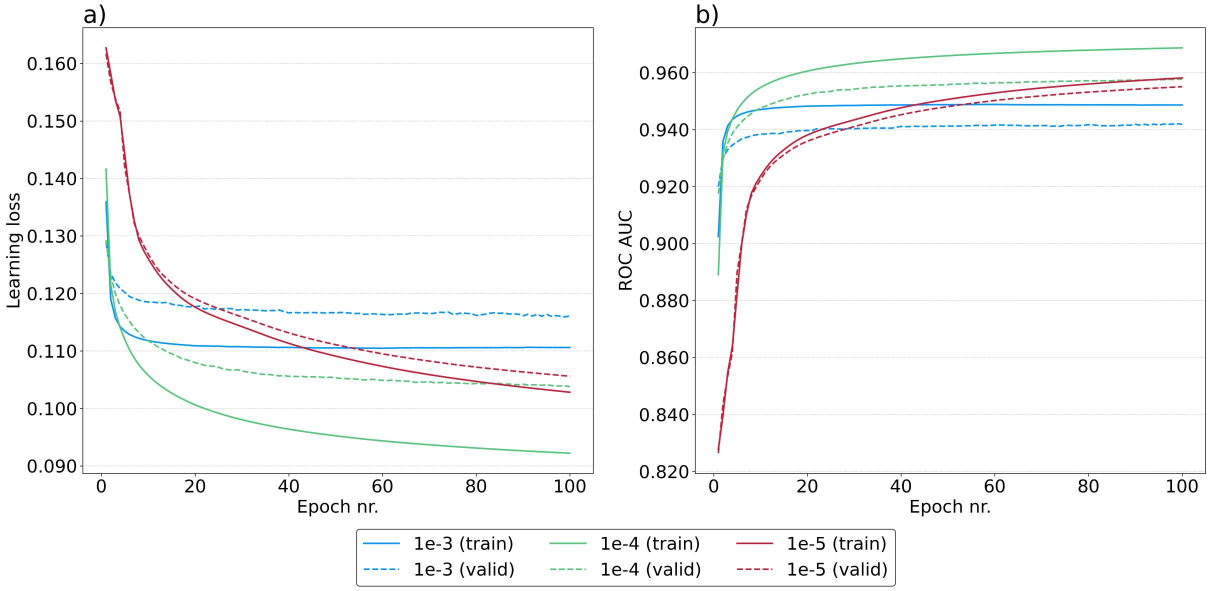


Figure 8.3: Two-fold input model with weights training

sented in Fig. 6.5). As in previous experiments, the best results were achieved with a learning rate of 1e-4. During optimization, the loss function on the validation dataset reached a value of 0.1038 and a ROC AUC value of 0.9578. After training for 100 epochs, the attention weights have values $\gamma = 0.0622$, associated with the gradient table, and $\delta = 4.2595$ associated with the diffusion data.

Architecture containing a convolutional layer processing diffusion data input (presented in Fig. 6.5) did not show an improvement in terms of a lower training loss or increased ROC AUC value. The training process is depicted in Figure 8.4. As in all other experiments, the best results were obtained using a learning rate of 1e-4, with training loss achieving a value of 0.1062 and ROC AUC 0.9552, measured on the validation dataset.

Using a weight decay is a regularization method allowing for better generalization [131]. In this experiment, various values of the weight decay were tested on selected architecture to investigate whether it would improve model performance further.

Figure 8.5 shows the results of three experiments with weight decay values of 1e-2, 1e-3 and 1e-4. Plots for a learning loss and ROC AUC values clearly indicate that adding weight decay did not improve the model performance. Learning loss did not decrease for any of the testing values. Even though ROC AUC increased for each experiment in the training subset, it remained constant for the validation dataset, indicating a lack of generalization. Hence, a weight decay shall be set to 0 when training the final model.

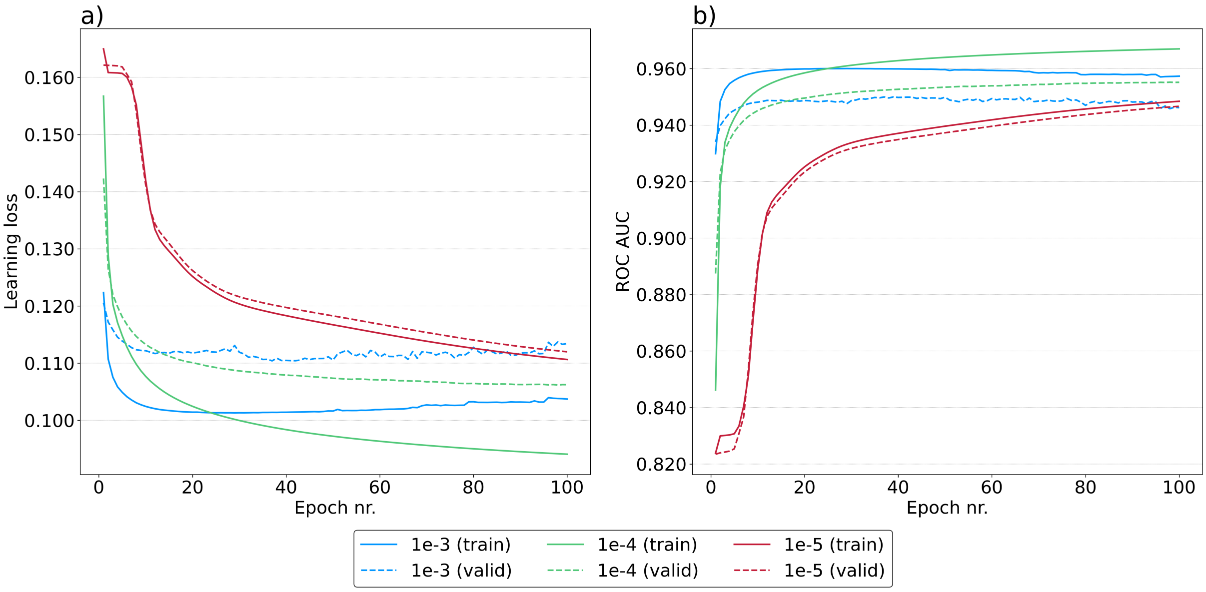


Figure 8.4: Two-fold input model with convolution training

Architecture	Learning rate	Min loss		Max ROC AUC	
		Train	Val	Train	Val
Minimal model	1e-3	0.118	0.1248	0.9378	0.928
	1e-4	0.0984	0.1082	0.9628	0.9531
	1e-5	0.1069	0.1093	0.9534	0.9505
Two-fold input model	1e-3	0.1046	0.1103	0.9563	0.9496
	1e-4	0.0925	0.1028	0.9685	0.9587
	1e-5	0.1046	0.1068	0.9562	0.9536
Two-fold input model with weights	1e-3	0.1105	0.1159	0.9488	0.942
	1e-4	0.0922	0.1038	0.9687	0.9578
	1e-5	0.1028	0.1056	0.9581	0.9551
Two-fold input model with convolution	1e-3	0.1013	0.1104	0.960	0.950
	1e-4	0.0941	0.1062	0.967	0.9552
	1e-5	0.1106	0.112	0.9484	0.9467

Table 8.1: ANN models performance with different architectures (training and validation phases)

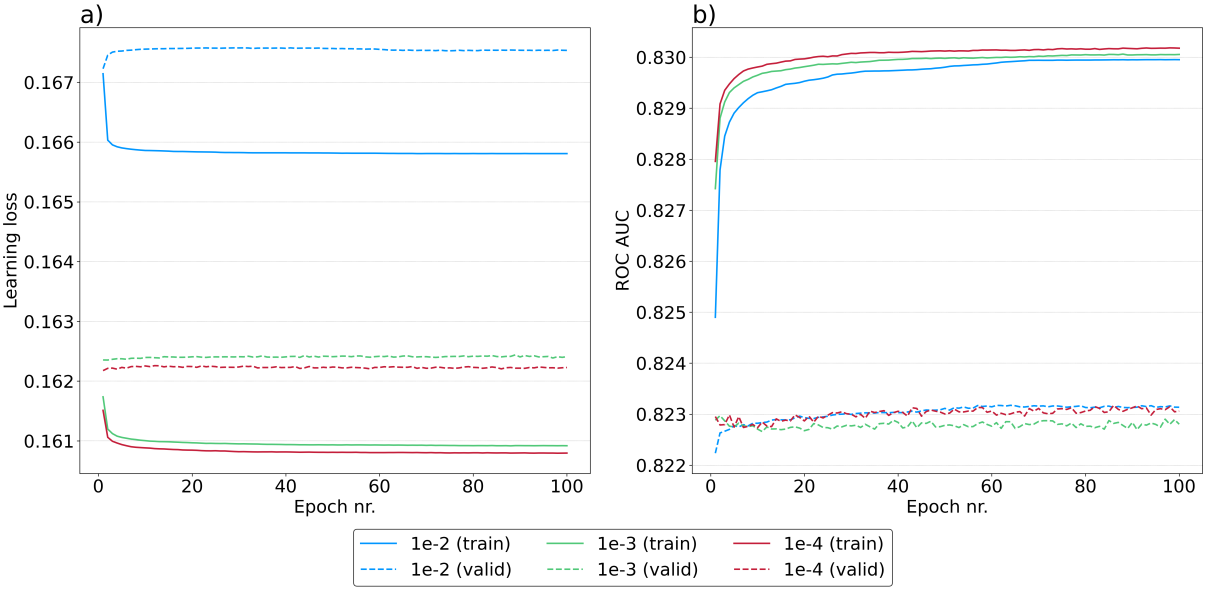


Figure 8.5: Influence of the weight decay on training

8.2.2 Different optimization algorithms

Despite Adam Optimizer being the most widely used, other algorithms have been proven to perform better in some cases. Hence, this experiment has tested different optimisation algorithms on the Two-Fold Input Model with attention weights. Besides Adam, classical SGD and SAM optimizers were tested. Given a SAM requires a basal optimizer to work on, two experiments were run, with SAM using SGD and Adam. Figure 8.6 depicts these experiments. For the SGD optimizer and SAM with SGD as the backbone, a learning loss decreased in the first few epochs to remain constant throughout the rest of the training. ROC AUC for these two optimizers reached levels below 0.85, which is much less than for other optimizers tested. The plots are basically the same for the training and validation phases. In the case of Adam and SAM with Adam backbone, a steady drop in learning loss can be observed throughout the whole training. Values of ROC AUC are similar and reach levels above 0.95 which are expected for Adam optimizer as seen in the experiments elucidating the best architecture. The use of SAM optimizer with Adam backbone does not lay better results. This indicates that the pure Adam optimizer shall be used, especially given the longer time required for training with SAM + Adam. This is caused by the need for two backward passes to minimize the Adam and the SAM optimizer.

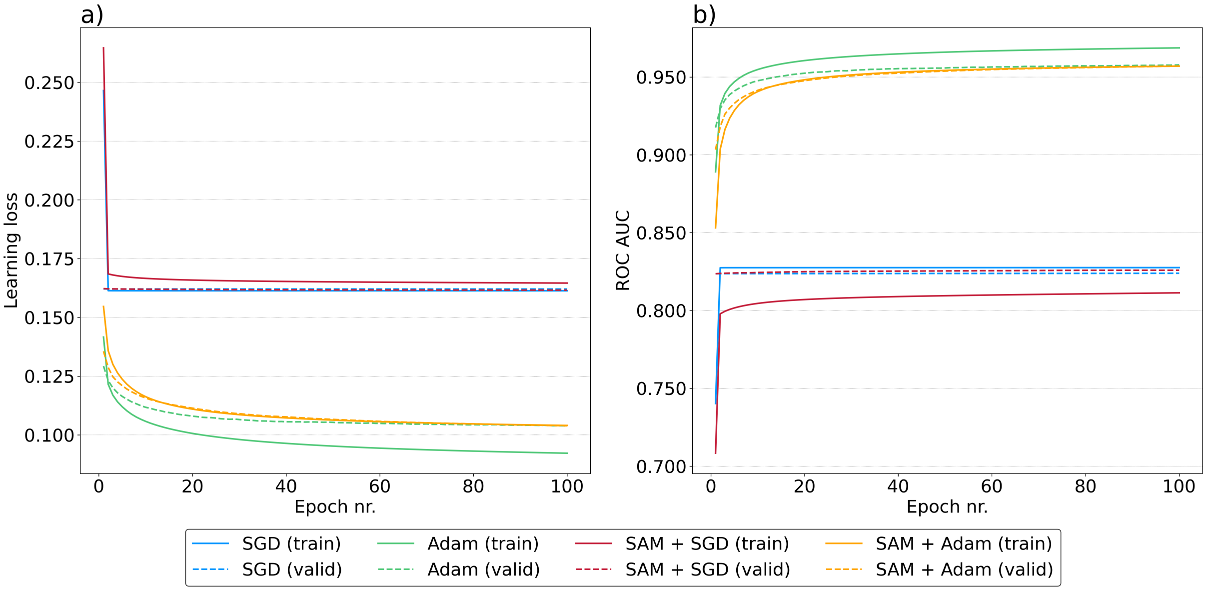


Figure 8.6: Two-fold input model with weights training using various optimization algorithms

8.2.3 Effect of a cube size on results

Previous experiments have examined the effect of specific parameters on the model performance without considering different input sizes. One of the parameters that can affect the path search stage profoundly is the size of the cube used. With increasing cube size the amount of the data processed by the network at once is bigger and the granularity of the search changes. A first step in this experiment was to train neural network for three cube sizes: 5 × 5 × 5 used in the previous experiments, 7 × 7 × 7 and 9 × 9 × 9. A matrix 3 × 3 × 3 was not considered since, in such an approach, the voxels on the border directly surround a central voxel. Hence, there is very little room for the path search algorithm to work. Increasing the cube size significantly increases the number of model parameters.

Figure 8.7 depicts the training process by visualizing the loss function and ROC AUC metric, as in the previous experiments. All three networks were trained on the same dataset, with the same parameters of learning rate 1e-4, and weight decay set to 0. As the plots clearly show the models for cube size 7 and 9 outperforms significantly those for cube size 5. The model for cube size 7 reaches a ROC AUC of 0.9714 on a validation dataset after 100 training epochs. The model for cube size 9 reaches an even higher ROC AUC value on the same dataset - 0.9787. All values are summarized in a table 8.2.

One of the explanations for this might be the number of parameters (Tab. 8.3). The model for a cube size 7 is twice as big as the one for a cube size 5. The model for cube size 9 is even

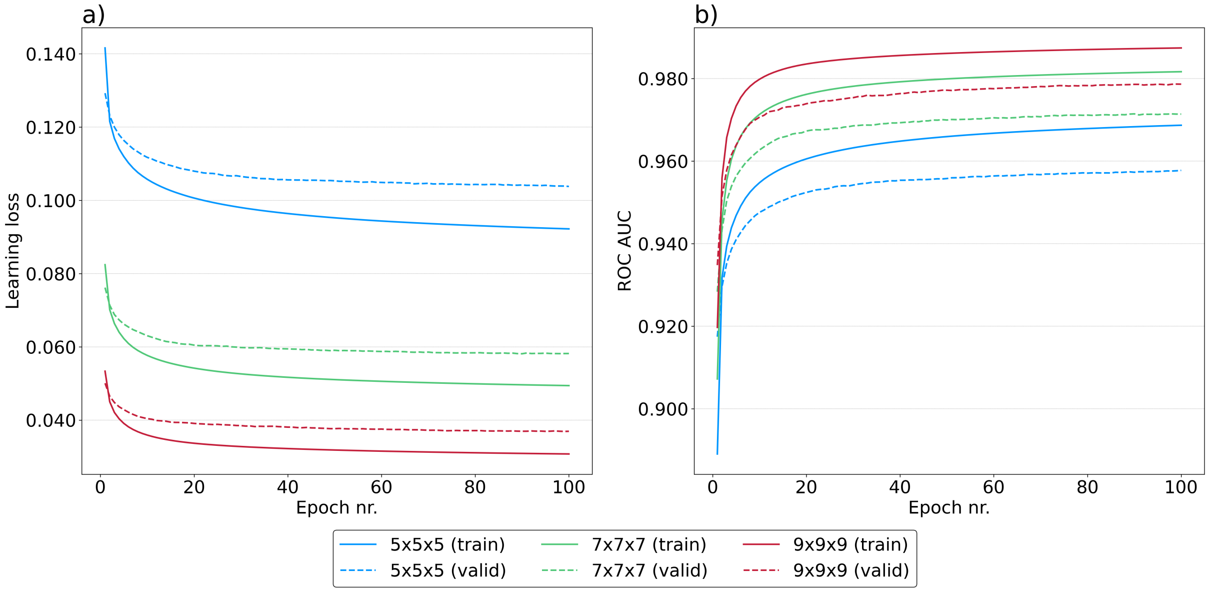


Figure 8.7: Two-fold input model with weights training with various cube sizes

Cube size	**Min loss**		**Max ROC AUC**	
	Train	**Val**	**Train**	**Val**
5x5x5	0.0865	0.1012	0.9734	0.961
7x7x7	0.0487	0.0579	0.9824	0.9717
9x9x9	0.0307	0.0369	0.9875	0.9788

Table 8.2: Loss and ROC AUC values for training Two-fold input model with weights with various cube sizes

Cube size	**Number of model parameters**
5x5x5	14 938 127
7x7x7	30 416 345
9x9x9	57 822 731

Table 8.3: Effect of cube size on the number of trainable model parameters in Two-fold input model architecture

bigger, and the number of parameters totals almost 58 mln. In bigger cubes, the view is also much broader. It allows to check if a branching fiber has a continuation and is a true fiber, or if the branching is just a data error.

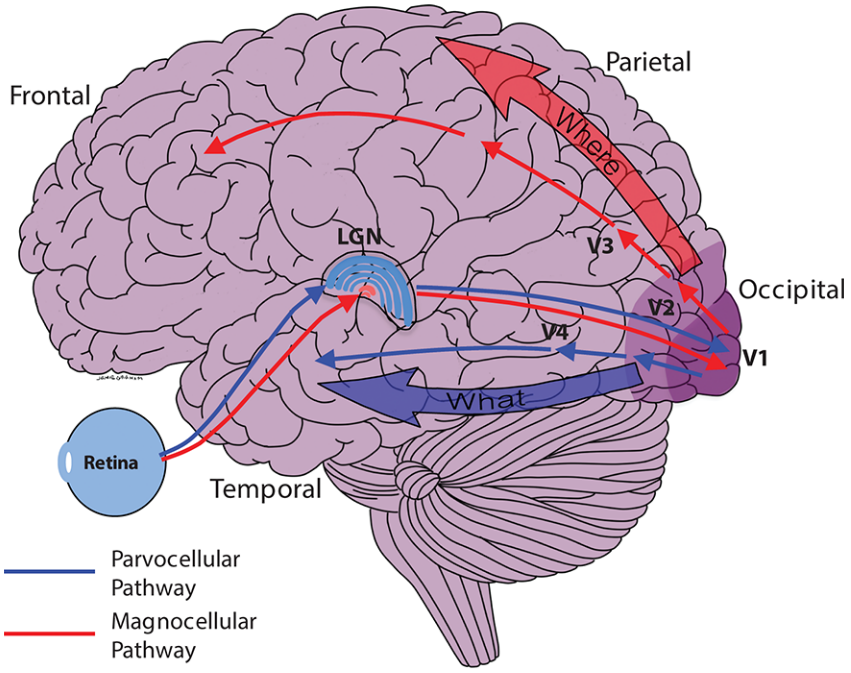
8.3 Training summary and discussion

The network training experiments focused on selecting the appropriate architecture and examining the impact of the various learning parameters on its effect. Multiple architectures with a controlled set of parameters showed very similar performance measured on the validation dataset. However, even a small increase of 0.01 in ROC AUC is desirable in medical applications. During these experiments, the architecture of the two-fold input model with attention weights was selected for further testing, where it was combined with a path search algorithm to build the desired tractograms. The attention weights values for a trained model are $\delta = 4.2595$ (associated with the diffusion data) and $\gamma = 0.0622$ (associated with the gradient table). It indicates that for obtaining a desirable network output scaling on the diffusion data has to be much higher than for the gradient table. It follows the intuition that the information is carried mainly in the intensity of voxels. However, information from the gradient table is also important as it describes the MRI magnetic field when collecting data, hence the value of $\gamma \neq 0$.

The optimization of the weights of the network proceeds best with the Adam optimizer, with a learning rate of 1e-4, leading to the model with a ROC AUC of 0.96, measured on the validation dataset.

Due to the simple architecture of the network and the relatively small number of parameters to be learned, the training process did not require the use of regularization in the form of the application of the weight-decay parameter, and its use even worsened the results. The performance of the selected network admittedly was very similar to the network without attention weights. However, their inclusion in the architecture makes it possible to ensure explainability regarding the relevance of individual input data.

Experiments with different cube sizes indicate that the cube sizes of 7 and 9 shall be used, despite the bigger model sizes. However, such judgment shall not be made solely on the results of the model training but also on the results of the path search stage of the hybrid model.



Source: Tony Mosconi, Victoria Graham:
Neuroscience for Rehabilitation
Copyright © McGraw-Hill Education. All rights reserved.

Figure 9.1: Primary visual cortex with main pathways [132]

9 HyTract testing

9.1 Test scenarios and datasets

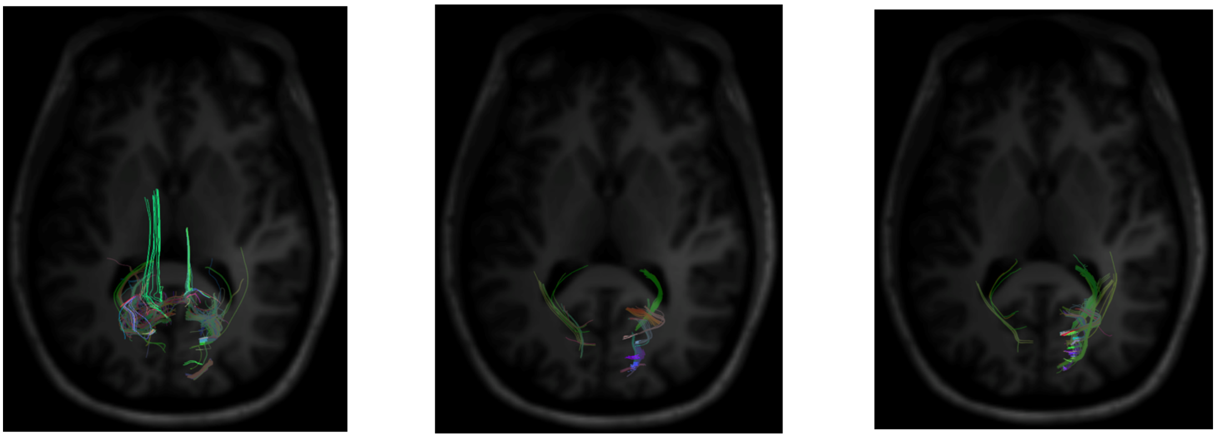
9.1.1 Test scenarios

In the experiments described in the previous chapter, the best artificial neural network architecture was selected and then combined with the shortest path algorithm. In the case of the hybrid model, it was examined:

- the effect of cube size and the value of the threshold variable
- the effect of window size on smoothing

9.1.2 Dataset

A validation dataset is disjoint from the training and evaluation. It has been created using data from a single randomly selected study, not included in the dataset used while training, with an id of mgh_1027. The dataset was prepared in the same way as for training, as described in section 8.1.2.



EuDX CPDG DMDG

Figure 9.2: Validation dataset

To check for bias toward detecting fibers originating in the Broca region, tractograms were computed for the neural pathways originating in the V1 area of the visual cortex in the occipital lobe (Fig. 9.1). A functional mask of the V1 area was obtained from Juelich histological atlas. It was binarized at the threshold value of 80 and combined with the GMWMI mask [118]. The methodology behind creating masks, seeds, and labels is analogous to the methodology used to create training and evaluation datasets, including algorithms and their parameters. Figure 9.2 presents the EuDX, DMDG and CPDG tractograms used for reference validation.

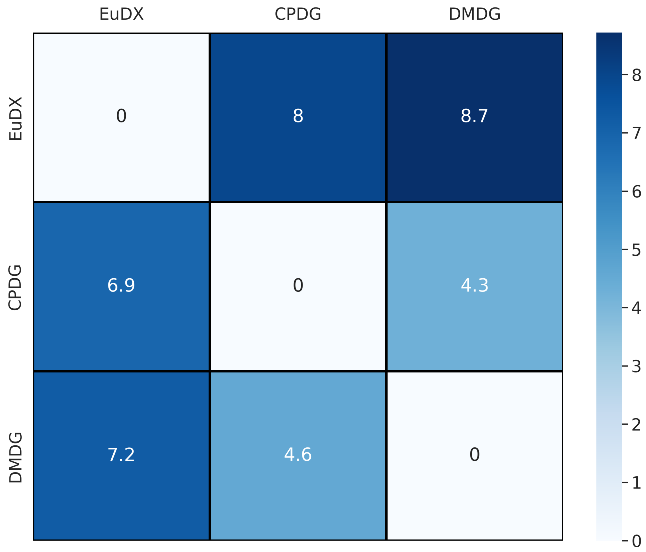


Figure 9.3: Differences between reference tractograms

For validation, a subset of seeds was used, selected as follows. From each of the full reference tractograms (computed with EuDX, DMDG, and CPDG), the 100 longest fibers were selected. For each of them, the position of the initial voxel was determined. All the starting voxels were then collected into a single unique array. These starting voxels were used as seeds for computing tractograms with different variants of the method presented in this thesis. For comparison, reference tractograms were limited to the streamlines originating in the selected seeds.

As the computed tractograms will be compared with these methods independently, it is important to evaluate how much the reference methods differ from each other. Figure 9.3 is a heatmap showing pairwise differences between these methods, expressed as a mean MED computed for the closest streamlines in both.

9.2 Results

9.2.1 Fiber reconstruction with a path search algorithm

Model checkpoints representing a two-fold input with weights architecture were selected to compare fibre tracking using different cube sizes. These checkpoints were obtained by training a model to process data cubes of sizes $5 \times 5 \times 5$, $7 \times 7 \times 7$ and $9 \times 9 \times 9$ as indicated in the Table 8.2.

Tractogram computation was run for each cube size independently on diffusion data

Cube size	Threshold	# streamlines	Streamline length	
			Mean	Max
5x5x5	0.4	4	17	18
7x7x7	0.4	4848	24.80	73
	0.5	1190	22.53	46
	0.6	403	19.86	33
	0.7	67	18.46	26
	0.8	18	16.94	21
9x9x9	0.4	12899	32.79	105
	0.5	6459	24.94	51
	0.6	3373	24.21	92
	0.7	889	20.69	48
	0.8	449	18.73	40
	0.9	296	18.90	36

Table 9.1: Basic statistics of streamlines obtained by computing tractograms with different window sizes and thresholds

available for subject *mgh_1027*. As a threshold for the cube border targets, the following values were used: 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. Other parameters were set constant. Euclidian distance for the stopping criterion was set to 70 voxels, and the smoothing window to 5, following the intuition explained in section 6.4.2. For clarity, only streamlines longer or equal to 15 voxels were considered.

Table 9.1 shows the basic statistics for computed tractograms. Surprisingly using a method variant with a cube size of 5 did not lay any streamlines longer or equal to 15 voxels unless a low threshold of 0.4 was used. Even with such a low threshold, only four streamlines were found with mean length of 17. Such a result indicates that this cube size should not be used, as it does not lay longer fibers, large numbers of which are found in the brain.

Both variants for cube sizes 7 and 9 yield fibers much longer than the assumed threshold of 15 voxels, some exceeding a 100. With increasing threshold values, the number of streamlines and their length decrease. Such results are expected, given the more restrictive selection rules for cube border targets. At the same time, increasing the cube size increases the number of computed streamlines and their length. It is understandable, as with bigger cubes, the network can interpret the data in a broader context. Moreover, the path between the cube center and the border is longer, allowing to pass through the voxels of lower values, that might not have been included otherwise.

Each of the computed tractograms was compared to the reference methods (EuDX, CPDG, and DMDG using mean euclidian distance (MED), as described in section 7.3. During validation, there was no length coverage restriction when searching for the most similar streamline in the reference tractogram. This restriction was dropped, as the meta-tractogram was not computed, and the presented method uses a different stopping criterion than the reference ones.

Tables 9.2, 9.3, 9.4 depict the comparison between reference tractograms (EuDX, CPDG, and DMDG), and tractograms computed with different cube sizes and thresholds. The tractograms calculated by various variants of the presented method were closest to the EuDX reference tractogram. The mean MED value is below 10 for five variants, highlighted green in Table 9.2. These tractograms are presented in Figure 9.4 and 9.5. MED values below 10 suggest that tractograms differ with EuDX on a similar level as the reference methods, as presented in Figure 9.3. Such inconsistency is not surprising. Using different algorithms and thresholds in tractography can land different results.

The comparison to CPDG and DMDG methods was not as good, with a MED value between 10 and 20 voxels. The three best variants, compared to CPDG, are cube size 7 with thresholds 0.4 and 0.8 and cube size 9 with threshold value 0.5 (marked green in Table 9.3). The two best variants with a cube size of 7 are consistent with comparison to the EuDX method. Very similar is the comparison of the different variants with the DMDG method. The three lowest mean MED values were obtained for variants with window size 7 and the threshold values of 0.4, 0.6 and 0.8 (marked green in Table 9.4. These results are consistent with other comparisons made to EuDX and CPDG methods. Surprisingly, all mean MED values for the variant with cube size 9 are much higher than those with cube size 7.

Comparisons made to the reference methods show that computed streamlines are highly significant. The best variant assumes a cube size of 7 and a threshold value of 0.4, resulting in the lowest mean MED value compared to all three reference tractograms. Anatomical significance is also relevant. Tractograms resulting in a low number of short streamlines do not reflect well anatomical fibers of the brain. Hence, the tractograms computed with a cube size of 7 and threshold values of 0.6 and 0.8 shall not be considered anatomically significant, similar to a cube size of 9 with threshold values of 0.8 and 0.9.

A visual evaluation of the tractograms supports these conclusions. Figure 9.4 and 9.5) shows a comparison between EuDX reference tractogram (Fig. 9.4, a) on three sections (sagittal, axial and coronal). All variants of the proposed method show a difference from EuDX below 10 voxels of MED. The highest number of streamlines (4848) was computed us-

Cube size	**Threshold**	**EuDX**		
		Min MED	**Max MED**	**Mean MED**
5x5x5	0.4	13.9	17.31	16.13
7x7x7	0.4	3.41	19.82	9.34
	0.5	3.45	18.63	10.12
	0.6	3.63	17.45	8.12
	0.7	4.73	16.00	10.34
	0.8	6.64	11.96	8.52
9x9x9	0.4	3.83	20.47	11.51
	0.5	3.38	20.15	10.40
	0.6	3.73	19.79	10.27
	0.7	3.18	20.31	10.20
	0.8	4.36	15.88	9.54
	0.9	5.04	14.86	9.40

Table 9.2: Comparison between tractograms computed with different cube sizes and thresholds with a reference tractogram from EuDX method

Cube size	**Threshold**	**CPDG**		
		Min MED	**Max MED**	**Mean MED**
5x5x5	0.4	16.38	57.76	37.56
7x7x7	0.4	4.09	26.78	15.28
	0.5	4.09	26.78	16.51
	0.6	8.65	25.10	16.66
	0.7	9.48	23.06	18.45
	0.8	4.31	23.06	14.42
9x9x9	0.4	3.63	26.29	16.74
	0.5	4.37	26.03	16.08
	0.6	5.92	27.74	16.90
	0.7	6.24	25.64	17.51
	0.8	8.11	25.64	18.32
	0.9	8.90	25.42	19.42

Table 9.3: Comparison between tractograms computed with different cube sizes and thresholds with a reference tractogram from CPDG method

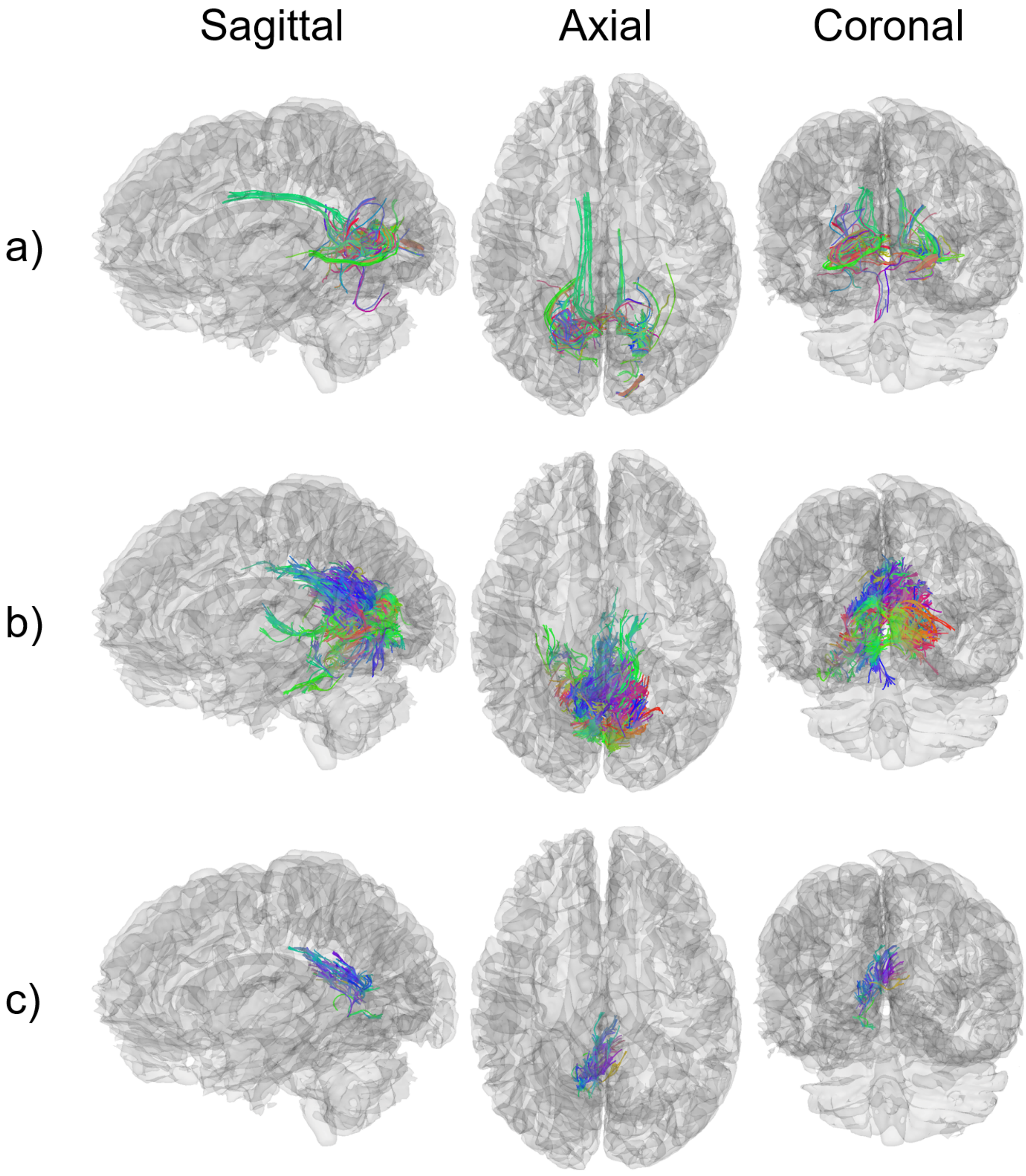


Figure 9.4: Comparison of selected tractograms with EuDX algorithm; a) EuDX; b) HyTract with cube size 7, threshold 0.4, no. of streamlines 4848, mean MED 9.34; c) HyTract with cube size 7, threshold 0.6, no. of streamlines 403, mean MED 8.12

Cube size	Threshold	DMDG		
		Min MED	Max MED	Mean MED
5x5x5	0.4	15.4	60.34	38.58
7x7x7	0.4	4.74	35.55	15.60
	0.5	4.78	28.94	16.25
	0.6	7.23	27.07	15.63
	0.7	10.90	23.40	17.97
	0.8	4.77	23.40	14.62
9x9x9	0.4	4.36	34.54	19.27
	0.5	5.48	31.78	17.21
	0.6	5.27	34.24	17.30
	0.7	6.34	31.37	18.02
	0.8	6.34	29.07	17.62
	0.9	11.38	27.09	19.25

Table 9.4: Comparison between tractograms computed with different cube sizes and thresholds with a reference tractogram from DMDG method

Smoothing window size	EuDX	CPDG	DMDG
0	9.61	15.45	15.78
3	9.49	15.38	15.70
5	9.34	15.28	15.60
7	9.17	15.16	15.49
9	8.98	15.03	15.36

Table 9.5: Different smoothing window sizes compared with reference tractograms

ing a cube size of 7 and a threshold of 0.4 (Fig. 9.4, b). They span both hemispheres and reach the anterior parts, as expected from anatomy. One major track arranges superior to the *Corpus callosum*. Another one, in the middle of the sagittal section, reaches towards a *thalamus*. A part of this structure, the lateral geniculate nucleus (LGN), is a gateway through which visual information reaches the cerebral cortex [133]. The third major track reaches towards the temporal lobe, creating a connection important for visual perception and memory [134].

Other presented variants lack most of the expected fibers. For instance, the variant with a cube size of 7 and a threshold of 0.6 (Fig. 9.4) reconstructs a track superior to *Corpus callosum* but does not reconstruct others. Furthermore, most variants compute only a small number of short streamlines originating mainly in the left hemisphere, as shown in Fig. 9.5.

9.2.2 Different window size in path smoothing

For experiments with different smoothing window sizes, a tractogram computed with a cube size of 7 and a threshold of 0.4 was used. The smoothing method was applied to all streamlines, with a window size of 3, 5, 7, and 9. Results are gathered in Table 9.5. With increasing

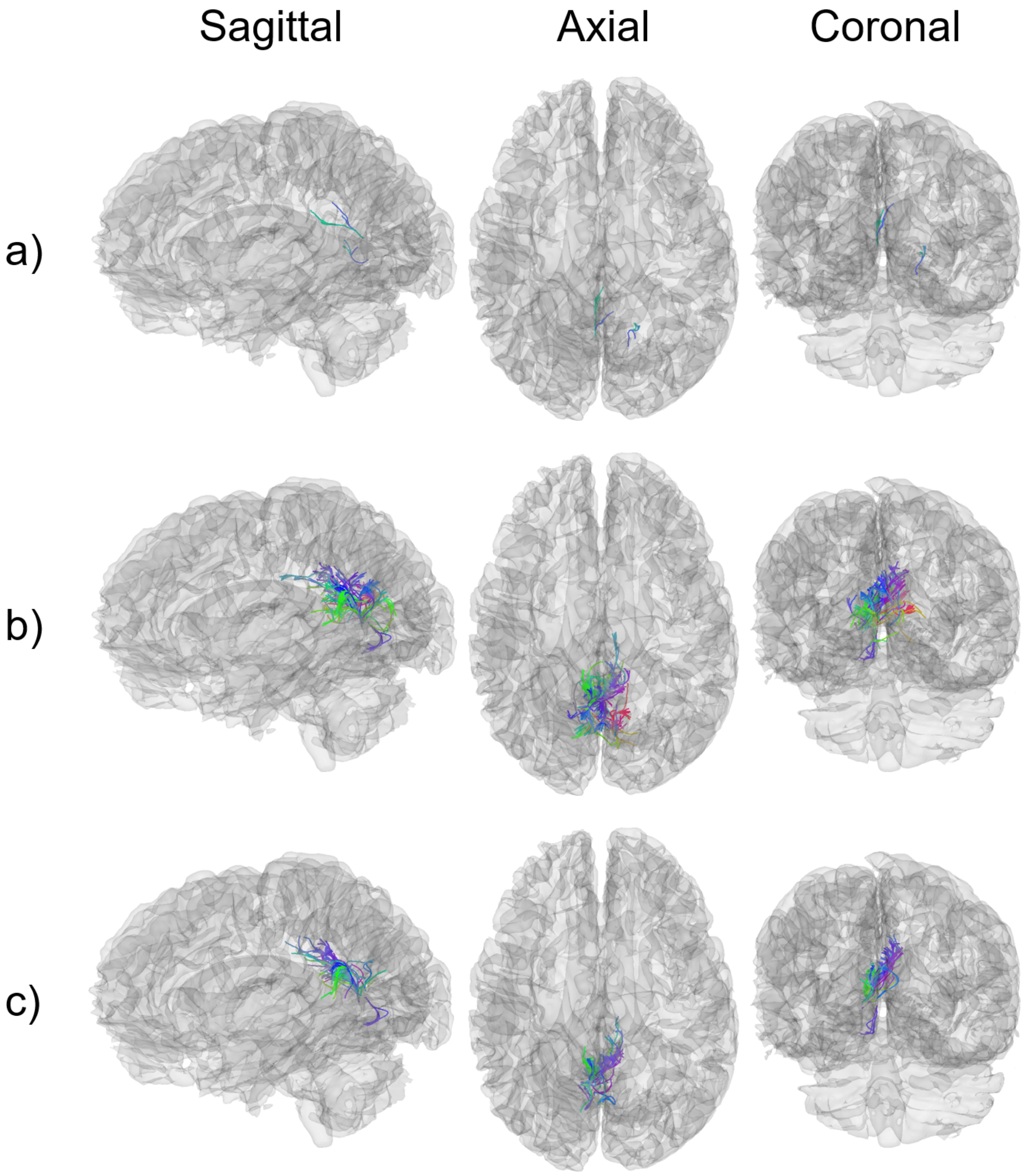


Figure 9.5: Selected tractograms computed with the hybrid model; a) cube size 7, threshold 0.8, no. of streamlines 18, mean MED 8.52; b) cube size 9, threshold 0.8, no. of streamlines 449, mean MED 9.54; c) cube size 9, threshold 0.8, no. of streamlines 296, mean MED 9.40

window size, the mean MED value tend to decrease. However, the change is quite small, especially considering the large number of streamlines in the set.

9.3 Tests summary and discussion

Final results were obtained by performing tractography for selected starting points of the gray-white matter interface of cortical region V1. These computations were performed on data from a patient who was not a part of the training and validation sets. The resulting tractograms were compared with reference ones obtained for the same starting points (using methods EuDX, CPDG and DMDG). Both visual analysis of the results and metrics using the mean Euclidean distance (MED) between fibers indicate significant potential for the clinical application of this method. Compared to one of the methods mentioned above, the average MED values were less than 10 voxels, comparable to the differences observed between the reference methods. The best variant assumes using a cube size of 7 and a threshold value of 0.4. It lands many long streamlines, closely reflecting streamlines computed by reference methods. Even more critical, computed streamlines follow anatomical tracks expected to originate from the V1 cortex.

The use of smoothing shall not be neglected at all, given the anatomical organization of the fiber tracks in the brain. As shown in Figure 6.10, the presented method can result in paths with sharp turns. These do not occur naturally. The smoothing shall be used, preferably with a relatively small window size, e.g. 5. A larger window size could significantly change the fiber topology determined by the path search algorithm, which is not desirable.

10 Conclusions

Tools that support the work of neurosurgeons in planning and performing operations are invaluable. Thanks to their use, the operations' time is significantly reduced. Most importantly, fed with the right data, such systems allow neurosurgeons to determine the entry point and scope of the intervention precisely. Good planning, in effect, significantly reduces the risk of complications and unwanted side effects, such as aphasia or paralysis.

This thesis introduces a new tractography technique that can be successfully used in neurosurgical planning - HyTract. The method uses a hybrid model consisting of an artificial neural network and an A* path search algorithm. The neural network is responsible for analyzing diffusion data (DTI), and the result of its analysis is the input to the A* algorithm.

The diffusion data is processed in small portions, cubic chunks of the entire scan with a side length of 5, 7, 9, or more voxels. With this approach, it is unnecessary to process all the scan data simultaneously. The locations from which the slices are picked are determined during the tracking itself using the A* algorithm. As a result, it is possible to calculate the topology of nerve fibers starting at the specified locations (seed points) of the 3D MRI scan.

The deep network architectures developed in this work were trained on five scans obtained from a publicized database of the international Human Connectome Project (HCP) initiative. The resulting models achieved excellent performance with a ROC AUC measure above 0.95.

Tests performed on a single subject, which was not a part of the training and validation subsets, show that ANN combined with a path search algorithm allows for determining neural tracts with satisfying accuracy. Moreover, these experiments were carried out in another brain area (V1 cortex) as training (Broca region), showing the method is not biased towards the brain area it was trained with.

The HyTract method presented in this thesis is an original contribution to the technical sciences in the field of information technology and telecommunication. It provides a novel technique to compute tractograms, which can be successfully used in the preoperative planning pipeline, as it only requires basic knowledge of tractography from the user and assumes no prior programming expertise. As the neural network processes the DTI diffusion data, it does not require tedious preprocessing steps as in the case of already available methods. When trained on raw material, the network tends to cope very well with noise and artefacts, naturally occurring in imaging studies when a strong magnetic field is used.

Furthermore, the proposed ANN architecture can be trained for other tasks like detecting crossing fiber points. Implementation of the path search algorithm can be replaced by others, allowing to obtain additional features of the white matter using the same underlying data from the artificial neural network. An example is connectomic studies, concerned with elucidating neural fibers connecting different functional areas of the cortex, passing the vicinity of the surgery site. Such knowledge is incredibly valuable, allowing neurosurgeons to assess the risks of impairments associated with specific motor or cognitive tasks. The modularity of the technique itself makes it very versatile, providing a wide range of applications.

The versatility of the method comes with straightforward architecture. ANN has a rather simple architecture, with a small number of learnable parameters, nevertheless allowing it to achieve the goal. Splitting the DTI data into two separate inputs, processed by an individual set of weights, boosts the network's performance, allowing the use of just two additional fully-connected layers with a limited number of parameters. Given the small size of the network, such architecture can be trained with ease, even on a personal computer with a graphical processing unit (GPU) not designed for professional use.

The artificial neural network used in the HyTract processes the input data in small samples, a cube with a side length of 5, 7 or 9 voxels. The sampling is based on labels prepared from the tractograms during the training. Hence, thousands of cubes can be sampled from a single study, which allows for training the network on a small number of patient scans. As the amount of data available in clinical settings is often limited, such an approach is favourable.

The hybrid nature of this technique ensures the explainability of the obtained results. The output of the artificial neural network is used to build a graph, which is then used to determine the topology of the neural fibers. This makes it possible to interpret the reason for which individual fibers were computed. This is incredibly important, especially in medical applications when such a system is intended to support the work of clinicians. The correlation of the method's reasoning with medical knowledge increases physicians' confidence while using it in preoperative planning.

The results presented in this dissertation prove the thesis that an artificial neural network combined with a path search algorithm is an efficient method for determining the topology of nerve fibers.

References

[1] David N Louis, Hiroko Ohgaki, Otmar D Wiestler, Webster K Cavenee, Peter C Burger, Anne Jouvet, Bernd W Scheithauer, and Paul Kleihues. The 2007 WHO Classification of Tumours of the Central Nervous System. *Acta Neuropathologica*, 114(2):97–109, 8 2007.

[2] Antonio Omuro and Lisa M DeAngelis. Glioblastoma and Other Malignant Gliomas: A Clinical Review. *JAMA*, 310(17):1842–1850, 11 2013.

[3] Nader Sanai, Mei Yin Polley, Michael W. McDermott, Andrew T. Parsa, and Mitchel S. Berger. An extent of resection threshold for newly diagnosed glioblastomas: Clinical article. *Journal of Neurosurgery*, 115(1):3–8, 7 2011.

[4] Kaisorn L. Chaichana, Ignacio Jusue-Torres, Rodrigo Navarro-Ramirez, Shaan M. Raza, Maria Pascual-Gallego, Aly Ibrahim, Marta Hernandez-Hermann, Luis Gomez, Xiaobu Ye, Jon D. Weingart, Alessandro Olivi, Jaishri Blakeley, Gary L. Gallia, Michael Lim, Henry Brem, and Alfredo Quinones-Hinojosa. Establishing percent resection and residual volume thresholds affecting survival and recurrence for patients with newly diagnosed intracranial glioblastoma. *Neuro-Oncology*, 16(1):113–122, 1 2014.

[5] H. Duffau. Brain plasticity and tumors. *Advances and technical standards in neurosurgery*, 33:3–33, 2008.

[6] Robert A. Pooley. Fundamental Physics of MR Imaging. *https://doi.org/10.1148/rg.254055027*, 25(4):1087–1099, 7 2005.

[7] Nikos K Logothetis, Jon Pauls, Mark Augath, Torsten Trinath, and Axel Oeltermann. Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412(68436843):150–157, 7 2001.

[8] Matthew J. Singleton. Functional Magnetic Resonance Imaging. *The Yale Journal of Biology and Medicine*, 82(4):233, 2009.

[9] Patric Hagmann, Lisa Jonasson, Philippe Maeder, Jean Philippe Thiran, J. Van Wedeen, and Reto Meuli. Understanding Diffusion MR Imaging Techniques:

From Scalar Diffusion-weighted Imaging to Diffusion Tensor Imaging and Beyond1. *https://doi.org/10.1148/rg.26si065510*, 26(SPEC. ISS.), 10 2006.

[10] P J Basser, J Mattiello, and D LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66(1):259–267, 1 1994.

[11] J D Martin. *Neuroanatomy text and atlas, fifth edition.* McGraw Hill LLC, 2019.

[12] Kristjan R Jessen and Rhona Mirsky. Glial cells in the enteric nervous system contain glial fibrillary acidic protein. *Nature*, 286(57745774):736–737, 8 1980.

[13] Blausen.com staff. Medical gallery of Blausen Medical 2014. *WikiJournal of Medicine*, (1 (2)), 2014.

[14] J G Betts, K A Young, J A Wise, E Johnson, B Poe, D H Kruse, O Korol, J E Johnson, M Womble, and P DeSaix. *Anatomy and Physiology.* OpenStax, 2013.

[15] N F Dronkers, O Plaisant, M T Iba-Zizen, and E A Cabanis. Paul Broca's historic cases: high resolution MR imaging of the brains of Leborgne and Lelong. *Brain*, 130(5):1432–1441, 4 2007.

[16] Kenneth S Saladin. *Anatomy & physiology : the unity of form and function / Kenneth S. Saladin.* McGraw-Hill, New York, NY, 6th ed. edition, 2012.

[17] Xuhui Wang, Sudhir Pathak, Lucia Stefaneanu, Fang-Cheng Yeh, Shiting Li, and Juan C Fernandez-Miranda. Subcomponents and connectivity of the superior longitudinal fasciculus in the human brain. *Brain Structure & Function*, 221(4):2075–2092, 5 2016.

[18] M de Schotten, F Dell'Acqua, P Ratiu, A Leslie, H Howells, E Cabanis, M T Iba-Zizen, O Plaisant, A Simmons, N F Dronkers, S Corkin, and M Catani. From Phineas Gage and Monsieur Leborgne to H.M.: Revisiting Disconnection Syndromes. *Cerebral Cortex*, 25(12):4812–4827, 12 2015.

[19] Asgeir S Jakola, Kristin S Myrmel, Roar Kloster, Sverre H Torp, Sigurd Lindal, Geirmund Unsgård, and Ole Solheim. Comparison of a strategy favoring early surgical resection vs a strategy favoring watchful waiting in low-grade gliomas. *JAMA*, 308(18):1881–1888, 11 2012.

[20] Seiji Ogawa, Tso-Ming Lee, Asha S Nayak, and Paul Glynn. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magnetic Resonance in Medicine*, 14(1):68–78, 1990.

[21] Peter A Kim Seong-Gi
}and Bandettini. Principles of Functional MRI. In Feroze B Faro Scott H.
}and Mohamed, editor, *BOLD fMRI: A Guide to Functional Imaging for Neuroscientists*, pages 3–22. Springer New York, New York, NY, 2010.

[22] Nikos K Logothetis and Brian A Wandell. Interpreting the BOLD signal. *Annual Review of Physiology*, 66:735–769, 2004.

[23] Serge A. R. B. Rombouts, Frederik Barkhof, and Philip. Scheltens. *Clinical applications of functional brain MRI.* Oxford University Press, 2007.

[24] Klaus-Dietmar Merboldt, Wolfgang Hanicke, and Jens Frahm. Self-diffusion NMR imaging using stimulated echoes. *Journal of Magnetic Resonance (1969)*, 64(3):479–486, 10 1985.

[25] D G Taylor and M C Bushell. The spatial mapping of translational diffusion coefficients by the NMR imaging technique. *Physics in Medicine & Biology*, 30(4):345, 4 1985.

[26] Jahn Andrew, Levitas Dan, Holscher Eric, Johnson John T., Sayal Alexandre, Wiesner Johannes, Clucas Jon, and Tapera Tinashe Michael. *Andy's Brain Book.* 2022.

[27] Yoav Freund and Robert E Schapire. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37:277–296, 1998.

[28] M Minsky and S Papert. *Perceptrons; an introduction to computational geometry.* MIT Press, 1969.

[29] Kenji Suzuki. *Artificial Neural Networks.* IntechOpen, Rijeka, 4 2011.

[30] Juergen Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, 1 2015.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[32] Dan C Cireşan, Ueli Meier, Jonathan Masci, Luca M Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1237–1242, Barcelona, Catalonia, Spain, 7 2011. AAAI Press.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, Red Hook, NY, USA, 12 2012. Curran Associates Inc.

[34] Kouichi Yamaguchi, Kenji Sakamoto, Toshio Akabane, and Yoshiji Fujimoto. A neural network for speaker-independent isolated word recognition. *First International Conference on Spoken Language Processing (ICSLP 1990)*, pages 1077–1080, 11 1990.

[35] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 6 2012.

[36] Kamila Lis, Mateusz Koryciński, and Konrad A. Ciecierski. Classification of masked image data. *PLOS ONE*, 16(7):e0254181, 7 2021.

[37] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 11 2018.

[38] Xiangang Li and Xihong Wu. Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition. (arXiv:1410.4281), 5 2015.

[39] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. (arXiv:1402.1128), 2 2014.

[40] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A Novel Connectionist System for Unconstrained Hand-

writing Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 5 2009.

[41] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[42] Jürgen Schmidhuber. Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks. *Neural Computation*, 4(1):131–139, 1 1992.

[43] J. Schmidhuber. Reducing the Ratio Between Learning Complexity and Number of Time Varying Variables in Fully Recurrent Nets. *ICANN '93*, pages 460–463, 1993.

[44] Imanol Schlag, Kazuki Irie, and J ¨ Urgen Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers. 2 2021.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 2017-December:5999–6009, 6 2017.

[46] Herbert E Robbins. A Stochastic Approximation Method. *Annals of Mathematical Statistics*, 22:400–407, 2007.

[47] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. (arXiv:1412.6980), 1 2017.

[48] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*, 2021.

[49] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks, 12 2021.

[50] Sali Abubaker Bagabir, Nahla Khamis Ibrahim, Hala Abubaker Bagabir, and Raghdah Hashem Ateeq. Covid-19 and Artificial Intelligence: Genome sequencing, drug development and vaccine discovery. *Journal of Infection and Public Health*, 15(2):289–296, 2 2022.

[51] Jiabao Sun, Jagdish C. Patra, and Yongjin Li. Functional Link Artificial Neural Network-based disease gene prediction. *Proceedings of the International Joint Conference on Neural Networks*, pages 3003–3010, 2009.

[52] Corinna Cortes, Vladimir Vapnik, and Lorenza Saitta. Support-vector networks. *Machine Learning 1995 20:3*, 20(3):273–297, 9 1995.

[53] Mateusz Korycinski, Reinhard Albrecht, Astrid Ursinus, Marcus D. Hartmann, Murray Coles, Jörg Martin, Stanislaw Dunin-Horkawicz, and Andrei N. Lupas. STAC - A New Domain Associated with Transmembrane Solute Transport and Two-Component Signal Transduction Systems. *Journal of Molecular Biology*, 427(20):3327–3339, 10 2015.

[54] Andrei N. Lupas, Hongbo Zhu, and Mateusz Korycinski. The Thalidomide-Binding Domain of Cereblon Defines the CULT Domain Family and Is a New Member of the β-Tent Fold. *PLoS Computational Biology*, 11(1), 1 2015.

[55] Andriy Kryshtafovych, John Moult, Arnaud Baslé, Alex Burgin, Timothy K. Craig, Robert A. Edwards, Deborah Fass, Marcus D. Hartmann, Mateusz Korycinski, Richard J. Lewis, Donald Lorimer, Andrei N. Lupas, Janet Newman, Thomas S. Peat, Kurt H. Piepenbrink, Janani Prahlad, Mark J. van Raaij, Forest Rohwer, Anca M. Segall, Victor Seguritan, Eric J. Sundberg, Abhimanyu K. Singh, Mark A. Wilson, and Torsten Schwede. Some of the most interesting CASP11 targets through the eyes of their authors. *Proteins: Structure, Function and Bioinformatics*, 84(S1):34–50, 9 2016.

[56] Lothar Terfloth and Johann Gasteiger. Neural networks and genetic algorithms in drug design. *Drug Discovery Today*, 6(SUPPL.2):102–108, 8 2001.

[57] Nathan Brown, Peter Ertl, Richard Lewis, Torsten Luksch, Daniel Reker, and Nadine Schneider. Artificial intelligence in chemistry and drug design. *Journal of Computer-Aided Molecular Design*, 34(7):709–715, 7 2020.

[58] Reed T. Sutton, David Pincock, Daniel C. Baumgart, Daniel C. Sadowski, Richard N. Fedorak, and Karen I. Kroeker. An overview of clinical decision support systems: benefits, risks, and strategies for success. *npj Digital Medicine 2020 3:1*, 3(1):1–10, 2 2020.

[59] I. Sim, P. Gorman, R. A. Greenes, R. B. Haynes, B. Kaplan, H. Lehmann, and P. C. Tang. Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, 8(6):527–534, 11 2001.

[60] L. A. Riesenberg and D. Riesenberg. Diagnostic decision support systems. *The Journal of medical practice management : MPM*, 17(3):163–165, 2001.

[61] Eta S. Berner. Diagnostic Decision Support Systems: Why Aren't They Used More And What Can We Do About It? *AMIA Annual Symposium Proceedings*, 2006:1167, 2006.

[62] Reeda Kunhimangalam, Sujith Ovallath, and Paul K. Joseph. A clinical decision support system with an integrated EMR for diagnosis of peripheral neuropathy. *Journal of Medical Systems*, 38(4):1–14, 4 2014.

[63] Konrad A. Ciecierski. Mathematical methods of signal analysis applied in medical diagnostic. *International Journal of Applied Mathematics and Computer Science*, 30(3):449–462, 9 2020.

[64] Konrad Ciecierski, Tomasz Mandat, Rafał Rola, Zbigniew W. Raś, and Andrzej W. Przybyszewski. Computer aided subthalamic nucleus (STN) localization during deep brain stimulation (DBS) surgery in Parkinson's patients. *Annales Academiae Medicae Silesiensis*, 5(68):275–283, 2014.

[65] Arne Jensen and Anders la Cour-Harbo. Ripples in Mathematics. *Ripples in Mathematics*, 2001.

[66] Hui Li, Miguel Castro, Pascal Haigron, Jean Philippe Verhoye, and Vito Giovanni Ruggieri. Decision support system for the planning of minimally invasive aortic valve replacement surgery. *International journal of computer assisted radiology and surgery*, 13(8):1245–1255, 8 2018.

[67] Rolf Adams and Leanne Bischof. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[68] Ahmed Hosny, Chintan Parmar, John Quackenbush, Lawrence H Schwartz, and Hugo J W L Aerts. Artificial intelligence in radiology. *Nature Reviews Cancer*, 18(8):500–510, 2018.

[69] Aras M. Ismael and Abdulkadir Şengür. Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, 164:114054, 2 2021.

[70] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014.

[71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, 12 2015.

[72] Ecem Sogancioglu, Keelin Murphy, Erdi Calli, Ernst T. Scholten, Steven Schalekamp, and Bram Van Ginneken. Cardiomegaly Detection on Chest Radiographs: Segmentation Versus Classification. *IEEE Access*, 8:94631–94642, 2020.

[73] Hossam H. Sultan, Nancy M. Salem, and Walid Al-Atabany. Multi-Classification of Brain Tumor Images Using Deep Neural Network. *IEEE Access*, 7:69215–69225, 2019.

[74] Anita Bakrania, Narottam Joshi, Xun Zhao, Gang Zheng, and Mamatha Bhat. fetArtificial Intelligence in Liver Cancers: Decoding the Impact of Machine Learning Models in Clinical Diagnosis of Primary Liver Cancers and Liver Cancer Metastases. *Pharmacological research*, page 106706, 2 2023.

[75] Le Hang Guo, Dan Wang, Yi Yi Qian, Xiao Zheng, Chong Ke Zhao, Xiao Long Li, Xiao Wan Bo, Wen Wen Yue, Qi Zhang, Jun Shi, and Hui Xiong Xu. A two-stage multi-view learning framework based computer-aided diagnosis of liver tumors with contrast enhanced ultrasound images. *Clinical Hemorheology and Microcirculation*, 69(3):343–354, 1 2018.

[76] Casey N. Ta, Yuko Kono, Mohammad Eghtedari, Young Taik Oh, Michelle L. Robbin, Richard G. Barr, Andrew C. Kummel, and Robert F. Mattrey. Focal Liver Lesions: Computer-aided Diagnosis by Using Contrast-enhanced US Cine Recordings. *https://doi.org/10.1148/radiol.2017170365*, 286(3):1062–1071, 10 2017.

[77] Manan Binth Taj Noor, Nusrat Zerin Zenia, M. Shamim Kaiser, Mufti Mahmud, and Shamim Al Mamun. Detecting Neurodegenerative Disease from MRI: A Brief Review

on a Deep Learning Perspective. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11976 LNAI:115–125, 2019.

[78] Adrien Payan and Giovanni Montana. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. *ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings*, 2:355–362, 2 2015.

[79] S. Sivaranjini and C. M. Sujatha. Deep learning based diagnosis of Parkinson's disease using convolutional neural network. *Multimedia Tools and Applications*, 79(21-22):15467–15479, 6 2020.

[80] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 12 1959.

[81] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[82] Stamatios N. Sotiropoulos, Li Bai, Paul S. Morgan, Cris S. Constantinescu, and Christopher R. Tench. Brain tractography using Q-ball imaging and graph theory: Improved connectivities through fibre crossings via a model-based approach. *NeuroImage*, 49(3):2444–2456, 2 2010.

[83] Ben Jeurissen, Maxime Descoteaux, Susumu Mori, and Alexander Leemans. Diffusion MRI fiber tractography of the brain. *NMR in Biomedicine*, 32(4):1–22, 2019.

[84] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*, 6 2021.

[85] Rohan Gupta, Devesh Srivastava, Mehar Sahu, Swati Tiwari, Rashmi K Ambasta, and Pravir Kumar. *Artificial intelligence to deep learning: machine intelligence approach for drug discovery.* Number 0123456789. Springer International Publishing, 2021.

[86] Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015.

[87] Peter J Basser. Fiber-tractography via diffusion tensor MRI (DT-MRI). *Proceedings of the 6th Annual Meeting ISMRM*, 1(January 1998):3, 1998.

[88] Mariana Lazar, David M Weinstein, Jay S Tsuruda, Khader M Hasan, Konstantinos Arfanakis, M Elizabeth Meyerand, Benham Badie, Howard A Rowley, Victor Haughton, Aaron Field, and Andrew L Alexander. White matter tractography using diffusion tensor deflection. *Human Brain Mapping*, 18(4):306–321, 2003.

[89] Andrew Zalesky. DT-MRI fiber tracking: A shortest paths approach. *IEEE Transactions on Medical Imaging*, 27(10):1458–1471, 10 2008.

[90] David S. Tuch. Q-ball imaging. *Magnetic Resonance in Medicine*, 52(6):1358–1372, 12 2004.

[91] C. Aronis, K. Delibasis, M. Fanariotis, and Ilias Maglogiannis. A tractography algorithm for MR diffusion tensor imaging based on minimum-cost path. *IFMBE Proceedings*, 57:302–307, 2016.

[92] J-Donald Tournier, Robert Smith, David Raffelt, Rami Tabbara, Thijs Dhollander, Maximilian Pietsch, Daan Christiaens, Ben Jeurissen, Chun-Hung Yeh, and Alan Connelly. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202:116137, 11 2019.

[93] Daan Christiaens, Marco Reisert, Thijs Dhollander, Stefan Sunaert, Paul Suetens, and Frederik Maes. Global tractography of multi-shell diffusion-weighted imaging data using a multi-tissue model. *NeuroImage*, 123:89–101, 12 2015.

[94] Marco Reisert, Irina Mader, Constantin Anastasopoulos, Matthias Weigel, Susanne Schnell, and Valerij Kiselev. Global fiber reconstruction becomes practical. *NeuroImage*, 54(2):955–962, 1 2011.

[95] Ben Jeurissen, Jacques Donald Tournier, Thijs Dhollander, Alan Connelly, and Jan Sijbers. Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data. *NeuroImage*, 103:411–426, 12 2014.

[96] J. Donald Tournier, Fernando Calamante, and Alan Connelly. Robust determination of the fibre orientation distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 35(4):1459–1472, 5 2007.

[97] J. Donald Tournier, Fernando Calamante, and Alan Connelly. MRtrix: Diffusion tractography in crossing fiber regions. *International Journal of Imaging Systems and Technology*, 22(1):53–66, 3 2012.

[98] Maxime Descoteaux, Rachid Deriche, Thomas R. Knösche, and Alfred Anwander. Deterministic and probabilistic tractography based on complex fibre orientation distributions. *IEEE Transactions on Medical Imaging*, 28(2):269–286, 2 2009.

[99] A. Anwander, M. Tittgemeyer, D. Y. Von Cramon, A. D. Friederici, and T. R. Knösche. Connectivity-Based Parcellation of Broca's Area. *Cerebral Cortex*, 17(4):816–825, 4 2007.

[100] Martin A. Koch, David G. Norris, and Margret Hund-Georgiadis. An Investigation of Functional and Anatomical Connectivity Using Magnetic Resonance Imaging. *NeuroImage*, 16(1):241–250, 5 2002.

[101] Peter F. Neher, Michael Götz, Tobias Norajitra, Christian Weber, and Klaus H. Maier-Hein. A machine learning based approach to fiber tractography using classifier voting. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9349:45–52, 2015.

[102] Peter F. Neher, Marc Alexandre Côté, Jean Christophe Houde, Maxime Descoteaux, and Klaus H. Maier-Hein. Fiber tractography using machine learning. *NeuroImage*, 158:417–429, 9 2017.

[103] Philippe Poulin, Marc-Alexandre Côté, Jean-Christophe Houde, Laurent Petit, Peter F. Neher, Klaus H. Maier-Hein, Hugo Larochelle, and Maxime Descoteaux. Learn to Track: Deep Learning for Tractography. *bioRxiv*, page 146688, 6 2017.

[104] Philippe Poulin, Francois Rheault, Etienne St-Onge, Pierre-Marc Jodoin, and Maxime Descoteaux. Bundle-Wise Deep Tracker: Learning to track bundle-specific streamline paths. *Proceedings of the International Society for Magnetic Resonance in medicine ISMRM-ESMRMB*, 2018.

[105] Itay Benou and Tammy Riklin-Raviv. DeepTract: A Probabilistic Deep Learning Framework for White Matter Fiber Tractography. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11766 LNCS:626–635, 12 2018.

[106] Eric G. Booth, Jeffrey F. Mount, and Joshua H. Viers. Hydrologic Variability of the Cosumnes River Floodplain. *San Francisco Estuary and Watershed Science*, 4(2), 9 2006.

[107] Hazim El Baz, Ibrahim Al Awadhi, and Assia Lasfer. SMA and MACD combinations for stock investment decisions in frontier markets: evidence from Dubai financial market. *International Journal of Financial Engineering and Risk Management*, 1(2):113, 2013.

[108] Eleftherios Garyfallidis, Matthew Brett, Bagrat Amirbekian, Ariel Rokem, Stefan Van Der Walt, Maxime Descoteaux, and Ian Nimmo-Smith. Dipy, a library for the analysis of diffusion MRI data. *Frontiers in Neuroinformatics*, 8, 2014.

[109] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images. *NeuroImage*, 17(2):825–841, 10 2002.

[110] Mark Jenkinson and Stephen Smith. A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, 5(2):143–156, 6 2001.

[111] Douglas N Greve and Bruce Fischl. Accurate and robust brain image alignment using boundary-based registration. *NeuroImage*, 48(1):63–72, 10 2009.

[112] Mark W Woolrich, Saad Jbabdi, Brian Patenaude, Michael Chappell, Salima Makni, Timothy Behrens, Christian Beckmann, Mark Jenkinson, and Stephen M Smith. Bayesian analysis of neuroimaging data in FSL. *NeuroImage*, 45(1 Suppl):173–186, 3 2009.

[113] Stephen M Smith, Mark Jenkinson, Mark W Woolrich, Christian F Beckmann, Timothy E J Behrens, Heidi Johansen-Berg, Peter R Bannister, Marilena De Luca, Ivana Drobnjak, David E Flitney, Rami K Niazy, James Saunders, John Vickers, Yongyue Zhang, Nicola De Stefano, J Michael Brady, and Paul M Matthews. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23:S208–S219, 1 2004.

[114] Mark Jenkinson, Christian F Beckmann, Timothy E J Behrens, Mark W Woolrich, and Stephen M Smith. FSL. *NeuroImage*, 62(2):782–790, 8 2012.

[115] Robert E Smith, Jacques-Donald Tournier, Fernando Calamante, and Alan Connelly. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 62(3):1924–1938, 9 2012.

[116] Stephen M Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, 2002.

[117] Katrin Amunts, Axel Schleicher, Uli Bürgel, Hartmut Mohlberg, Harry B M Uylings, and Karl Zilles. Broca's region revisited: Cytoarchitecture and intersubject variability. *Journal of Comparative Neurology*, 412(2):319–341, 1999.

[118] Katrin Amunts, Aleksandar Malikovic, Hartmut Mohlberg, Thorsten Schormann, and Karl Zilles. Brodmann's Areas 17 and 18 Brought into Stereotaxic Space—Where and How Variable? *NeuroImage*, 11(1):66–84, 1 2000.

[119] Vladimir Fonov, Alan C Evans, Kelly Botteron, C Robert Almli, Robert C McKinstry, and D Louis Collins. Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage*, 54(1):313–327, 1 2011.

[120] V S Fonov, A C Evans, R C McKinstry, C R Almli, and D L Collins. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47:S102, 7 2009.

[121] J L R Andersson, M Jenkinson, and S Smith. Non-linear registration aka spatial normalisation FMRIB. *FMRIB technical report TR07JA2*, 2010.

[122] Eleftherios Garyfallidis. *Towards an accurate brain tractography*. PhD thesis, University of Cambridge, 2012.

[123] Iman Aganj, Christophe Lenglet, and Guillermo Sapiro. ODF reconstruction in q-ball imaging with solid angle consideration. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1398–1401, 6 2009.

[124] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 6 2006.

[125] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *https://doi.org/10.1148/radiology.143.1.7063747*, 143(1):29–36, 4 1982.

[126] D C Van Essen, K Ugurbil, E Auerbach, D Barch, T E J Behrens, and R Bucholz et al. The Human Connectome Project: A data acquisition perspective. *NeuroImage*, 62(4):2222–2231, 2012.

[127] S Moeller, E Yacoub, C A Olman, E Auerbach, J Strupp, N Harel, and K Uğurbil. Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI. *Magnetic Resonance in Medicine*, 63(5):1144–1153, 2010.

[128] D A Feinberg, S Moeller, S M Smith, E Auerbach, S Ramanna, and M F Glasser et al. Multiplexed Echo Planar Imaging for Sub-Second Whole Brain FMRI and Fast Diffusion Imaging. *PLOS ONE*, 5(12):1–11, 12 2010.

[129] K Setsompop, B A Gagoski, J R Polimeni, T Witzel, V J Wedeen, and L L Wald. Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty. *Magnetic Resonance in Medicine*, 67(5):1210–1224, 2012.

[130] J Xu, S Moeller, J Strupp, E J Auerbach, L Chen, and D A Feinberg et al. Highly Accelerated Whole Brain Imaging Using Aligned-Blipped-Controlled-aliasing Multiband EPI. page 1, 2012.

[131] Anders Krogh and John A Hertz. A Simple Weight Decay Can Improve Generalization. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, pages 950–957, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[132] Tony Mosconi and Victoria Graham. *Neuroscience for rehabilitation.* McGraw-Hill Education, 2017.

[133] W Martin Usrey and Henry J Alitto. Visual Functions of the Thalamus. *Annual Review of Vision Science*, 1(1):351–371, 2015.

[134] Zafar U Khan, Elisa Mart\'\in-Montañez, and Mark G Baxter. Visual perception and memory systems: from cortex to medial temporal lobe. *Cellular and Molecular Life Sciences*, 68:1737–1754, 2011.