

LONDON'S GLOBAL UNIVERSITY



# Estimation of tissue microstructure using Deep Learning Techniques

Candidate number: LRSN0<sup>1</sup>

MEng Computer Science

Dr Zhang Gary

Submission date: 4 06 2021

<sup>1</sup>**Disclaimer:** This report is submitted as part requirement for the MEng Computer Science Degree at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report will be distributed to the internal and external examiners, but thereafter may not be copied or distributed except with permission from the author.

## **Abstract**

Over the last 3 decades, diffusion MRI (dMRI) has shown to be a powerful non-invasive method of assessing the microstructural characteristics and organizations within the brain tissue. With the use of advanced diffusion models, specific microstructural properties can be computed and used for the study of numerous neurological disorders, including multiple sclerosis and the Alzheimer's disease. Despite the fact that studies indicate that calculated microstructure measures correlate very well with histological examinations, many of the diffusion models require large acquisition times, making them prohibitive for use in clinical settings. To address this, deep learning and specifically the class of feed-forward neural networks is employed exploiting the main vulnerability of advanced diffusion models, which lies the data processing pipeline used. This project demonstrates that with the use of deep learning, scalar measures characterizing tissue microstructure, can be computed directly from the dMRI signals, thus reducing dMRI data processing to a single optimized step. Additionally, it is illustrated that the most important information is contained in a small number of data points, therefore enabling the computation of scalar measures at a twelve-fold reduced scan time. This, in combination with a significant training time reduction compared to classical diffusion models demonstrate the feasibility of deep learning for inclusion in clinical settings.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Problem Outline . . . . .	2
1.2	Aims . . . . .	2
1.3	Goals . . . . .	2
1.4	Programming Languages, Libraries and Frameworks used . . . . .	3
1.4.1	PyTorch . . . . .	3
1.4.2	Sklearn . . . . .	3
1.4.3	Nibabel and Nilearn . . . . .	3
1.4.4	FSLeyes . . . . .	3
1.5	Report Structure . . . . .	3
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Diffusion . . . . .	5
2.2	Model Fitting Techniques . . . . .	5
2.2.1	Diffusion Tensor Imaging (DTI) . . . . .	5
2.2.2	Neurite orientation dispersion and density imaging (NODDI) . . . . .	6
2.2.3	Accelerated Microstructure Imaging via Convex Optimization (AMICO) . . . . .	7
2.3	Analytical Solutions . . . . .	8
2.4	Deep Learning Techniques . . . . .	8
2.4.1	Comparison between model fitting and deep learning . . . . .	9
<b>3</b>	<b>Methods</b>	<b>10</b>
3.1	Problem statement and aims . . . . .	10
3.1.1	Novelty of Deep Learning . . . . .	10
3.2	Data . . . . .	10
3.2.1	Details of Diffusion-Weighted Images . . . . .	11
3.2.2	Data Exploration . . . . .	11
3.2.3	Data Pre-Processing . . . . .	11
3.3	Feed-Forward Neural Network . . . . .	12
3.4	Implementation Details of the Feed Forward Neural Network . . . . .	13
3.4.1	Hyperparameter tuning . . . . .	13
3.5	Deep Learning Framework . . . . .	14
<b>4</b>	<b>Experiments</b>	<b>15</b>
4.1	Experiment 1: . . . . .	15
4.1.1	Loss Curves . . . . .	15
4.1.2	Difference between ground-truth and predicted values . . . . .	16

4.1.3	Visual comparison between predicted scalar measures and ground-truth . .	16
4.2	Experiment 2: . . . . .	18
4.2.1	Loss Curves . . . . .	18
4.2.2	Difference between ground-truth and predicted values . . . . .	18
4.2.3	Visual comparison between predicted scalar measures and ground-truth . .	18
4.3	Experiment 3: . . . . .	20
4.3.1	Loss Curves . . . . .	20
4.3.2	Difference between ground-truth and predicted values . . . . .	20
4.3.3	Visual Comparison of predicted scalar measures with ground-truth . . . . .	21
4.4	Experiment 4: . . . . .	22
4.4.1	Visual Comparison of predicted scalar measures with ground-truth . . . . .	22
4.5	Experiment 5 . . . . .	24
4.5.1	Visual Comparison of predicted scalar measures with ground-truth . . . . .	24
<b>5</b>	<b>Results</b>	<b>26</b>
5.1	Evaluation Metrics . . . . .	26
5.1.1	Root Mean Square Error (RMSE) . . . . .	26
5.1.2	Mean absolute error (MAE) . . . . .	26
5.1.3	Mean Bias Error (MBE) . . . . .	27
5.2	First 3 experiments Results . . . . .	27
5.3	Fourth Experiment Results . . . . .	29
5.4	Fifth Experiment Results . . . . .	30
5.5	Computation time comparison . . . . .	31
<b>6</b>	<b>Discussion</b>	<b>33</b>
6.1	Achievements . . . . .	33
6.2	Critical Evaluation and Future Work . . . . .	33
6.3	Final Thoughts . . . . .	34

# Chapter 1

## Introduction

### 1.1 Problem Outline

Advanced diffusion models can give detailed characterizations of the brain, as well as important biomarkers which can be used for the diagnosis of brain disorders. To accomplish this, a number of diffusion-weighted images (DWIs) are acquired with varying diffusion weightings and directions. The goal of quantitative dMRI is then to find a mapping from a limited number of dMRI signals to rotationally invariant scalar measures. However, many of the advanced diffusion models used require long acquisition times, making them inapplicable for use in real-world applications, especially when patients are uncooperative. This project demonstrates a deep learning framework which overcomes these limitations.

### 1.2 Aims

The aim of this project is to employ deep learning methods for the estimation of microstructural tissue properties, with focus on reducing the acquisition time needed for traditional model fitting techniques, while also providing a computationally efficient framework for the calculation of these measures.

### 1.3 Goals

In order to achieve the aims set above the following goals have been set:

- Perform a literature review of existing techniques and understand both the advantages and limitations of these methods.
- Gain a thorough understanding of deep learning techniques, specifically the class of feed-forward neural networks that are used in this project. This also involves understanding the differences between traditional model fitting techniques and deep learning models.
- Learn to conduct careful examination of diffusion-weighted imaging (DWI) data, specifically using the software FSLeyes.
- Learn to use the applicable libraries that are used for processing DWI data using the Python programming language. This includes learning the appropriate techniques for both importing, exploring and pre-processing the data used.

- Implement AMICO, which is a model fitting technique used for the estimation of tissue microstructure. AMICO, as it will be described briefly in later sections, provides the ground-truth data that are used for the training of the neural network model.
- Learn to use the PyTorch framework for the implementation of deep learning models. After developing a working model, learn how to perform hyperparameter tuning on deep learning models, so as to achieve the optimization of the model.
- Evaluate the performance of the neural network on multiple datasets containing data of different subjects and data acquired using various MRI scanners.

## 1.4 Programming Languages, Libraries and Frameworks used

### 1.4.1 PyTorch

PyTorch [1] is an open source machine learning library, primarily developed by Facebook’s AI Research lab (FAIR). PyTorch was used in this project for the development of the feed-forward neural network. PyTorch contains important modules that make the development process easier. To name a few, PyTorch contains the optim module which includes various optimization algorithms used for developing neural networks, and the nn module which enables the development of complex neural networks in a few lines of code.

### 1.4.2 Sklearn

Sklearn [2] is a machine learning library for the Python Programming Language. Sklearn offers a wide range of machine learning models such as support vector machines and logistic regression. However, in this project Sklearn was used for data pre-processing tasks, such as normalizing the data.

### 1.4.3 Nibabel and Nilearn

Nibabel [3] provides read/write access to common neuroimaging file formats and it was used in this project to import all the datasets used. Nibabel [4] gives access to data in the form of NumPy arrays, thus it is very straightforward to use. Nilearn was used to perform some pre-processing tasks such as applying a brain mask to the data to capture all the brain-voxels.

### 1.4.4 FSLeaves

FSLeaves [5] is a software used to examine diffusion-weighted imaging data before going to the development process. FSLeaves enabled me to understand the information provided by the data through its various functionalities such as the time-series view which provides the examination of all the measurements made for a given voxel.

## 1.5 Report Structure

Chapter 2 includes background information and literature review on existing diffusion models used for microstructural estimation. Additionally the benefits of using deep learning for this task are discussed.

Chapter 3 describes the techniques and methods used in this project. This section explains step-by-step, the process followed and demonstrates the framework used.

Chapter 4 contains all the experiments conducted in this project. The first 3 experiments aim to test how well the neural network model generalizes on different datasets, while the 4th experiment aims to estimate tissue microstructure using sub-sampled data.

Chapter 5 provides a summary and evaluates the results obtained from the experiments conducted in chapter 4.

Chapter 6 summarises the achievements of the project, includes a critical evaluation of the work done and gives some thoughts on possible future work.

## Chapter 2

# Background

### 2.1 Diffusion

Water makes up around 60-70 percent of the human body. The random Brownian motion of the molecules caused by thermal energy is known as diffusion. Diffusion in perfectly homogeneous media is random and isotropic, meaning that molecules move in all directions with equal probability. However, in the complex case of the human body, water is partitioned between cells and extracellular compartments. Extracellular water molecules show comparatively free diffusion, whereas intracellular water molecules exhibit comparatively "restricted diffusion." Different tissues of the human body have distinct cellular architectures and proportions of intracellular and extracellular compartments, resulting in distinct diffusion characteristics. The pathologic processes have an impact on the relative proportion of water distribution between these compartments. With the use of Diffusion Weighted Imaging (DWI) both quantitative and qualitative information is provided about the diffusion properties. [6]

### 2.2 Model Fitting Techniques

The traditional approach used for the estimation of scalar measures is model fitting, at which scalar measures are estimated using the fitted model parameters. Below you can find description of some of the most widely used diffusion models, that employ model fitting techniques.

#### 2.2.1 Diffusion Tensor Imaging (DTI)

Water molecules in cerebral white matter diffuse more readily along the direction of axonal fascicles than across them. This directional dependence of diffusivity has a special name, and it is referred to as anisotropy. Diffusion Tensor Imaging (DTI) [7] was the first diffusion model that was able to exploit this directional dependence and formally describe the anisotropy of the random motion of water molecules in biological tissues. DTI was introduced, straight after it was demonstrated in vivo, that the organization of the tissue influences the diffusion-weighted signal acquired using magnetic resonance imaging (MRI) and may therefore be assessed non-invasively.

DTI makes a strong assumption that the diffusion process follows a 3D Gaussian Distribution, which can be captured using 6 parameters, describing the variance and covariance of the Gaussian diffusion along the three primary axes. From these parameters, estimations of the principal diffusion direction can be computed. To determine the biophysical characteristics of the underlying



tissue, further uni-variate metrics can be inferred from the tensor model’s parameters. The main metrics calculated from DTI are fractional anisotropy (FA) and mean diffusivity (MD). Fractional anisotropy is a scalar value which lies between zero and one, and describes the degree of anisotropy of a diffusion process. An FA value of zero shows that the diffusion is isotropic, which means that it is unrestricted in all directions. An FA value of one shows that diffusion is fully restricted in all directions except the one axis at which it occurs. [8][cite other thing as well.] MD shows the average molecular displacement magnitude by diffusion. The higher the value of MD, the higher the medium. However, as DTI assumes that the diffusion process follows a Gaussian distribution, it is not possible to resolve more than one fiber population in the same voxel. To overcome the limitation of DTI, a number of methods were introduced including Diffusion Spectrum Imaging [9], methods based on Q-Ball Imaging [10][11][12], and Constrained Spherical Deconvolution [13]. Using these methods, directional information of many fiber populations in a voxel can be computed.

### 2.2.2 Neurite orientation dispersion and density imaging (NODDI)

As it has been stated above, the main parameters estimated with the use of Diffusion Tensor Imaging are the mean diffusivity and fractional anisotropy. Despite the fact that these measures have been shown to have high sensitivity, they are lacking specificity for individual tissue microstructure features. [14] It has been demonstrated that a decrease in the value of FA, might be produced by a decrease in neurite density, an increase in the dispersion of neurite orientation distribution, or other changes in the tissue’s microstructure. Therefore, it is very hard to know the specific change in tissue microstructure, just by the fractional anisotropy. To address this issue, a new class of techniques was derived, named Microstructure Imaging. Microstructure Imaging aims to derive tissue microstructural properties by using explicit biophysical modelling of decay patterns in various tissue compartments, such as axons, glial cells, and extra-axonal space.

NODDI, is a Microstructure Imaging technique, which obtains tissue microstructural information by modelling the decay of water molecules in dendrites and axons. Dendrites and axons are projections of neurons in the brain known together as neurites, which are the molecular building blocks of the brain’s electronic circuitry. Neurite morphology, measured in terms of density and orientation distribution, gives an insight into the underlying foundation of brain function in both healthy and unhealthy subjects. Neurite morphology is also an important indication of brain growth and ageing. A rise in the distribution of neurite orientation is related to the growth of the brain, while the decrease of dendritic density is linked to brain ageing. In addition, changes in the anatomy of neurites are linked with various neurological diseases, including the Alzheimer’s disease and amyotrophic lateral sclerosis. Estimated NODDI indices have also been proven to align extremely well with histology observed anatomical patterns. [13][14][15] In comparison to DTI, NODDI requires dMRI data that are scanned with multiple b-values and comparably a larger number of diffusion gradient directions (i.e > 90 directions over two b-shells). NODDI models the dMRI signal using a tissue model which distinguishes intra-cellular, extra-cellular, and CSF microstructural compartments. Each compartment gives a different dMRI signal and the full normalised signal  $A$  can be written as:

$$A = (1 - v_{iso})(v_{ic}A_{ic} + (1 - v_{ic})A_{ec} + v_{iso}A_{iso}), \quad (2.1)$$

where  $A_{ic}$  and  $v_{ic}$  are the normalized signal and volume fraction of the intra-cellular compartment;  $A_{ec}$  is the normalised signal of the extra-cellular compartment; and  $A_{iso}$  and  $v_{iso}$  are the normalized signal and volume fraction of the CSF compartment. [15]

## Model Fitting

The set of model parameters that the NODDI model uses for fitting purposes are:

- *ICVF*: Intra-cellular volume fraction
- *d*: Intrinsic free diffusivity
- *ISOVF*: Isotropic volume fraction
- $\mu$ : Mean Orientation of Watson Distribution
- $v_{iso}$ : Isotropic Volume Fraction
- $d_{iso}$ : Isotropic Diffusivity
- $\kappa$ : Concentration Parameter of Watson Distribution

Diffusivities  $d_{iso}$  and  $d$  are fixed. By then adapting techniques described in [16] for model fitting, the maximum likelihood estimations of the remaining parameters are computed. Orientation Dispersion (OD) calculated using the the estimated value of  $k$ . Below you can find a brief description of some of the scalar measures estimated with the use of NODDI. The measures described are the ones that will be estimated with the deep learning framework that is provided in the Methods section.

**Isotropic volume fraction (ISOVF)** is the volume fraction of the water pool characterized by isotropic diffusion, and can be interpreted as the voxel volume fraction of free water.

**Intra-cellular volume fraction (ICVF)** is an estimate of neurite density.

**Orientation Dispersion Index (ODI)** is NODDI’s summary statistic and characterizes the angular variation of neurite orientation. The value of ODI ranges from 0 (all axon/dendrites parallel) to 1 (all axon/dendrites isotropically randomly oriented) and is calculated by fitting a Watson distribution to the zero-radius cylinder orientation distribution.

### 2.2.3 Accelerated Microstructure Imaging via Convex Optimization (AMICO)

Despite the success of NODDI to compute microstructural information from dMRI data, the non-linear model fitting techniques employed to achieve that are computationally expensive, and therefore the application of NODDI in clinical settings, especially with a large group of people is impractical. To overcome this limitation, AMICO uses the flexibility of convex optimization to reformulate microstructure imaging approaches in equivalent but more practical linear systems that can be solved quickly utilizing rapid algorithms, making it more applicable for use in real-world settings.

#### Spherical Deconvolution Methods

Before, explaining the details of how the AMICO framework works it is important to give a brief explanation of Spherical Deconvolution (SD) methods, as it forms the basis of AMICO. In the SD methods, the dMRI signal  $E(q)$  in each voxel is modelled as a function of the convolution of the fiber orientation distribution (FOD).

$$E(q) = E_0 \int_{S^2} K(q, \hat{u}) f(\hat{u}) d(\hat{u}) \quad (2.2)$$

where  $E_0$  is the signal without diffusion weight and the integral is performed over the unit sphere  $S^2$ . The FOD is usually expressed as a linear combination of  $N_k$  basis functions, also called atoms, as  $f(\hat{u}) = \sum_{i=1}^{N_k} w_i f_i(\hat{u})$ .

Spherical Deconvolution methods have showed great success mainly due to the fact they can easily be expressed as simple linear systems (Eq. (2.3)), which in turn can be solved using a number of algorithms based on convex optimization.

$$y = \phi x + c \quad (2.3)$$

where  $x$  are the coefficients of the FOD to be estimated,  $y$  is the vector containing the  $N_d$  normalised q-space measurements and  $\phi$  is the dictionary that models the convolution operation in Eq.(2.2).

### AMICO Framework

The goal of AMICO is to express microstructure imaging techniques (such as NODDI), as linear systems of the form in Eq. (2.3). To do that, the authors of the paper [16], suggested separating the reconstruction of intra-voxel fiber geometry, i.e. number and orientation of fiber populations, from the assessment of their microstructure qualities, i.e. diameter and density. This enables to deconstruct the original technique's complexity into two easier sub-problems. The first sub-problem i.e the estimation of the orientation of fiber populations can be solved using classical DTI and SD methods. For the second sub-problem, the authors propose methods, including partitioning the dictionary  $\phi$  to blocks, formulating the NODDI problem as a convex optimization problem, and thus accelerating the estimation of the NODDI indices. [17]

## 2.3 Analytical Solutions

Another way of estimating scalar measures is using analytical methods, in the case where closed-form analytical solutions exist. For the DKI diffusion model [18][19], which needs around 150 DWIs, it has been demonstrated that for some DKI scalar measures, substantially fewer DWIs are necessary, and that these measures may be analytically derived from the data in a single step [20]. This gave an indication that a small number of DWIs may contain the most important information and thus additional scalar measures can be computed using a small set of DWIs, reducing scan-time costs.

## 2.4 Deep Learning Techniques

Deep Learning is a sub field of machine learning associated with artificial neural networks that are influenced by the structure and operation of the human brain. A neural network is made up of a set of units called neurons that are organized in layers. There is an input layer where data enters the network, followed by a number of hidden layers that transform the data as the training set is propagated through them, and finally an output layer where the neural network's predictions are made. The network is learned to make useful predictions by detecting patterns in a series of labelled training data points that is fed into the network and the model outputs are

compared to the individual labels using an optimisation function. The network's parameters are adjusted during training until the patterns found by the network result in accurate predictions for the training data. Once the neural network is trained, it can be used for making predictions on new, unseen data.

### 2.4.1 Comparison between model fitting and deep learning

Figure 2.1 shows the data processing steps needed for the calculation of scalar measures using DKI, NODDI, AMICO and deep learning. In essence, the figure also shows the main difference between classical machine learning models (i.e model fitting) and deep learning. Classical machine learning methods are limited in their ability to process natural data in their raw form. Developing a classical machine learning model often demands domain-expertise in order to perform feature engineering on raw-data by constructing features which transform the raw-data in such a way such that when fed to a machine learning model, patterns are able to be detected. On the other hand, deep learning falls in the class of representation learning methods, which enable the model to be fed with raw data and automatically discover patterns, overcoming the need for feature engineering. Additionally, the traditional pipeline does not have feedback from later steps to earlier steps on which parts of the information should be kept and which transformations should be applied. In deep learning, with the use of back-propagation all layers are optimized together. [22]

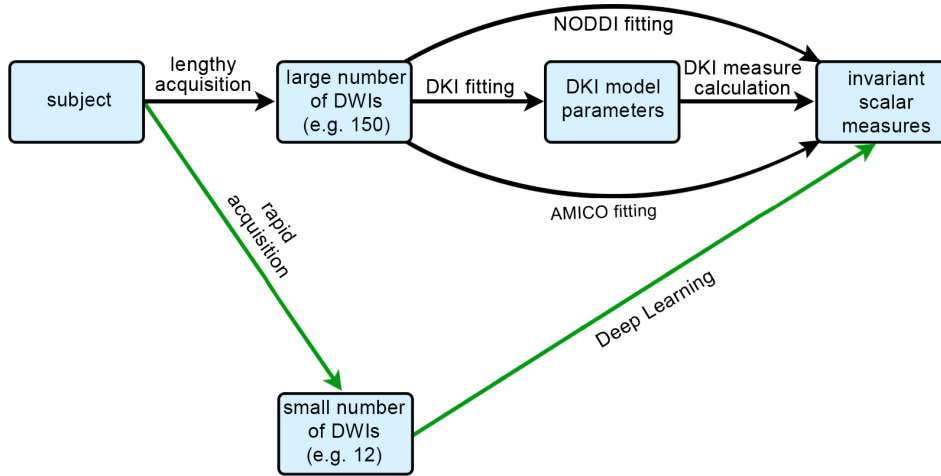


Figure 2.1: Estimation of scalar measures using model fitting techniques in comparison to deep learning techniques.

## Chapter 3

# Methods

### 3.1 Problem statement and aims

As discussed in the background section, a diffusion weighted signal contains information about the structure of the tissue, with the aim of quantitative diffusion MRI to provide a method that computes invariant scalar measures from DWI signals. The method used should provide accurate measurements as well as be clinically feasible, overcoming the limitations that traditional model fitting methods such as NODDI impose. To address the limitations of traditional diffusion modelling methods, deep learning techniques are employed, providing a framework which exploits the classical data processing techniques used in traditional methods by utilizing all the benefits that deep learning provides. Deep Learning models require much shorter acquisition times and thus are able to be used in clinics, reducing scan costs caused by long scan duration, as well as overcome potential issues caused by uncooperative, uncomfortable or unwell patients.

#### 3.1.1 Novelty of Deep Learning

The main novelties of deep learning in the estimation of tissue microstructure are the following:

- DWIs are used as the input directly to the model without the need for feature engineering and feature extraction.
- Much fewer DWIs can be used for the calculation of the scalar measures, resulting in a large scan time reduction.
- Improvement in the training time needed to produce the invariant scalar measures in comparison with both AMICO and NODDI model fitting techniques.

### 3.2 Data

The dataset used to conduct all experiments consists of three healthy individuals, specifically: one male with 23 years of age and two females with 26 and 23 years of age respectively. Each one of the subject's brains was scanned in 10 different centers, and all scans were processed by the same operator with a fixed operating procedure. For each subject, anatomical images and diffusion images were acquired. For all the experiments implemented, only the diffusion images were used, which details can be seen below.

### 3.2.1 Details of Diffusion-Weighted Images

DWIs were obtained using a simultaneous multi-slice (SMS) spin-echo echo planar imaging (EPI) prototype sequence<sup>28</sup>. The imaging parameters were as follows: TR/TE= 5.4 s/71ms, FOV=220×220 mm<sup>2</sup>, slice number=93, voxel size=1.5×1.5×1.5 mm<sup>3</sup>, bandwidth=1712 Hz/Px, GRAPPA factor=2, and SMS factor=3, with reversed phase-encoding (PE) directions along anteroposterior (AP) and posteroanterior (PA) separately. The diffusion duration and diffusion time were 15.9ms and 34.4ms, respectively, for the monopolar diffusion gradients. The diffusion scheme, containing 30 vectors with uniform angular coverage on each shell (b-values=1000, 2000, and 3000s/mm<sup>2</sup>, non-colinear between any two shells), was generated from a multi-shell vector sampling tool<sup>29</sup>. Six non-diffusion frames were equally distributed in the scheme. The total acquisition time was 19minutes and 04 seconds. [23]

### 3.2.2 Data Exploration

By performing some exploratory data analysis on every dataset used in this project, the table below is filled showing information about the number of DWIs contained in each dataset, the number of brain-voxels, and man/mix intensities found.

Data Exploration				
Dataset	Number of DWIs	Number of brain-voxels	Max Inten-sity	Min Inten-sity
Person 1-Center 01	96	480925	-68	2604
Person 1-Center 02	96	471809	-126	2527
Person 1-Center 03	96	476554	-94.	2586
Person 1-Center 09	96	478220	-76	2588
Person 1-Center 10	96	476628	-76.3	2588
Person 2-Center 10	96	394848	-106.6	2129
Person 3-Center 10	96	451128	-100	2123

Table 3.1: Data Exploration of the datasets used

### 3.2.3 Data Pre-Processing

In order for the data to be used as input to the neural network, some minimal pre-processing had to be done. To start with, the data used contain 4 dimensions. The first dimension contains the number of voxels in the x-direction, the second one contains the number of voxels in the y-direction, the third one contains the number of voxels in z-direction and the last dimension is the volume index i.e it represents all the settings under which a DWI measurement has been made. In all the DWIs the largest proportion of voxels are non-brain voxels. Non-brain voxels contain no information, therefore they had to be removed. To account for this, the brain mask of each one of

the datasets was used to remove all the non-brain voxels. Subsequently, the data were transformed from four-dimensional arrays to two-dimensional arrays, with the first dimension containing all the voxels, and the second one containing the intensity value of each voxel acquired using different settings. Finally, the data were normalised using the min-max normalisation technique. Min-max is one of the most prevalent methods of data normalization. The minimum value of each feature is converted to a 0, the highest value is converted to a 1, and all other values are converted to a number between 0 and 1.

### 3.3 Feed-Forward Neural Network

Following the brief explanation of deep learning in the background section, this section will provide a more detailed explanation of how a feed forward neural network (also known as multi layer perceptron) works. A multi-layer perceptron aims to find a parametric mathematical function  $y = f(x; \theta)$  which maps an input  $x$  to an output  $y$  by finding the values of the parameters  $\theta$  that result in the best function approximation. As it can be seen from Figure 3.1, a multi-layer perceptron contains an input layer (left-most layer), a number of hidden layers (in-between layers) and an output layer (the right-most layer). A feed-forward neural network performs a number of non linear data transformations in each layer. Feed-Forward Neural Networks are fully connected, therefore, each unit is connected with all the units from the previous layer. Each unit has its own bias  $b^{(i)}$ , and a weight  $W^{(i)}$  is assigned to each pair of units in two subsequent layers. Denoting Layer 0 the input layer, layer  $L$  the output layer, the output vector  $h_i^{(l)}$  for data point  $j$  in layer  $i \in \{1, \dots, L\}$  can be expressed as:

$$h_j^{(i)} = \phi_i \left( W^{(i)} h_j^{(i-1)} + b^{(i)} \right) \quad (3.1)$$

To measure how good the output vector  $h_j^{(i)}$  is, we make use of a cost function  $C$  to calculate how close it is to the target vector  $\hat{y}_j$ . The cost function used, which is widely known as the mean square error (MAE), is formally defined as:

$$C = \frac{1}{n} \sum_{j=1}^N (h_j^{(L)} - \hat{y}_j)^2 \quad (3.2)$$

Then during training, the weight and bias vectors are modified in order to optimize the cost function, using the backpropagation algorithm which solves the following optimization problem:

$$\arg \min_{W, b} \sum_j (h_j^{(L)} - \hat{y}_j)^2 \quad (3.3)$$

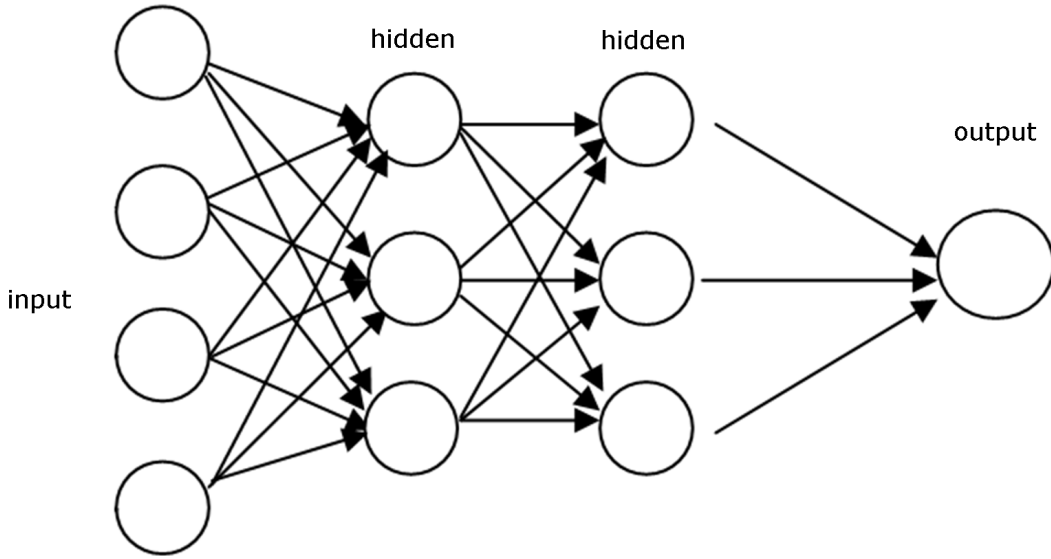


Figure 3.1: Feed Forward Neural Network architecture.

### 3.4 Implementation Details of the Feed Forward Neural Network

This section includes the specific settings used for the multilayer perceptron as well as details on how hyperparameter tuning was performed. To begin with, the neural network was initialised with orthogonal random weights [24], and the rectified linear unit (ReLU) [25] was used for the activation function. The ReLU function is a piecewise linear function which outputs the the input if it is positive or zero otherwise. ReLU is the go to activation function in multilayer perceptrons, as it overcomes the vanishing gradient problem allowing models to run better and faster. It is important to note here, that the model must generalise well on unseen data, and therefore regularization had to be introduced. This was firstly achieved by using dropout. [26] In simple terms, dropout randomly 'drops' units during the training phase to reduce overfitting and improve the generalization error. In addition to dropout, an early stopping class was developed and early stopping was employed when there was no increase in the validation set error after 10 epochs (this is often referred as patience).

#### 3.4.1 Hyperparameter tuning

Hyperparameter tuning was performed on both the neural network architecture parameters (e.g number of hidden layers) as well as on parameters of the optimization function used (e.g weight decay for ADAM optimizer). To start with, experiments were carried out to see which neural network architecture worked best. Specifically, I performed a grid search on both two and three hidden layer structures with the following combinations: [128,128,128], [64,128,64], [64,64,64], [128,128], [64,128], [128,64], [64,64] and I found out that using a three hidden layer structure with 128 hidden units each ([128,128,128]) produced the best results. Afterwards, I compared two optimization techniques: Stochastic Gradient Descent (SGD) and ADAM [27], and it was found out that ADAM was by far superior. By then running a hyperparameter grid search over ADAM parameters, specifically over the learning rate and the weight decay, it was found that a learning rate of 0.001 and



weight decay of 0.1 performed best.

### 3.5 Deep Learning Framework

Given a set of brain-only signal measurements consisting of  $S = nDWIs$ , the task is to compute tissue microstructure-characterizing scalar measures  $M$ , directly from  $S$ . In the framework provided, every voxel is treated as a separate training point. As discussed, training and testing data are pre-processed in order to contain only brain voxels. Moreover, in all experiments performed that are described in detail in the next chapter, training and testing data come from different subjects or from the same subject but acquired using a different scanner. Scalar measures used to construct the training dataset are obtained with the use of the AMICO model fitting framework. It is important to note here, that once the neural network is trained, model fitting (AMICO in this case) is no longer required and the trained neural network can be used directly for predictions on unseen datasets. In other words, the scalar measures computed from AMICO are treated as the ground-truth scalar measures. As one of the aims of this project is to be able to use fewer DWIs as input to the deep learning model in order to speed up acquisition times, sub-sampling techniques are used to generate a sample of the training data. It is important to specify here, that sub-sampling must be consistent i.e the same sub-sampling must be performed on both the training and testing data.

---

**Algorithm 1:** Deep Learning Framework - Estimation of scalar measures directly from DWI signals

---

```
// Training and Testing Datasets
features_train = nDWIs[0];
labels_train = AMICO(features_train);
features_test = nDWIs[1];

// Sub-sampling Method
train_features_sampled = subSample(features_train, a);
labels_train_sampled = subSample(labels_train, a);
test_features_sampled = subSample(features_test, a);

// Neural Network Model Training
model = FeedForwardNeuralNetwork(parameters);
model.train(train_features_sampled, labels_train_sampled);

// Predictions on unseen data
scalar_measures = model.predict(test_features_sampled);
```

---

## Chapter 4

# Experiments

This section contains all the experiments carried out in this project. As discussed, the dataset provided contains data of three subjects acquired using various scanners. The variety of the datasets used provided me with the ability to perform a number of different experiments. All the experiments carried out follow the deep learning framework provided in the Methods section. The first three experiments aim to test how well the neural network model generalizes on unseen data of different types. The fourth and fifth experiment aim to estimate scalar measures using sub-sampled data, with experiment 4 using a reduced set of DWIs, and experiment 5 using a sample of brain data for the training of the neural network.

### 4.1 Experiment 1:

In this experiment, both training and testing data are of the same subject, but acquired using a different scanner. To test how well the model generalizes, the feed-forward neural network model is trained using data of one of the subjects acquired from one scanner and tested on data of the same subject on the remaining four scanners. The aim of this experiment is to demonstrate the use of dMRI signals directly as input to the neural network model and test the performance of the model on this type of test sets.

Training set used: Person 1 - Center 09

Testing sets used: Person 1 - Center 10, Person 1 - Center 01, Person 1 - Center 02, Person 1 - Center 03

#### 4.1.1 Loss Curves

The loss curves obtained for ISOVF, ICVF and OD, when the model is trained using data of Person 1 and Center 9 are provided. The loss curves show the training set error, validation set error and the early stopping check-point, which is the epoch at which the validation set error was the lowest using a patience value of 10. At the early stopping checkpoint all the parameters of the neural network are saved, and are subsequently used for predictions on the test data.

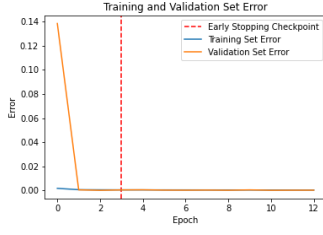


Figure 4.1: ISOVF

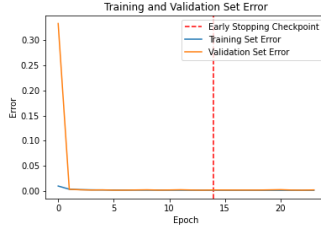


Figure 4.2: ICVF

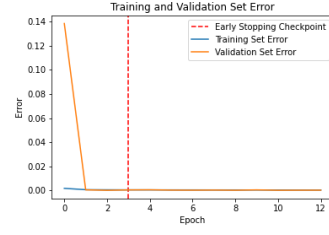


Figure 4.3: OD

#### 4.1.2 Difference between ground-truth and predicted values

A histogram of the voxel-wise intensity difference between ground-truth and predicted values for the Person 1 - Center 10 test set is displayed here, where it is visible, that the largest majority of predicted values are extremely close to ground truth. The differences are also symmetrical around zero and not skewed in any direction.

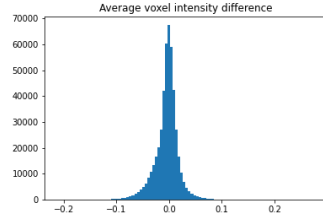


Figure 4.4: ISOVF

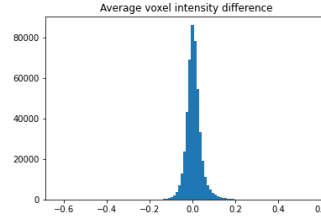


Figure 4.5: ICVF

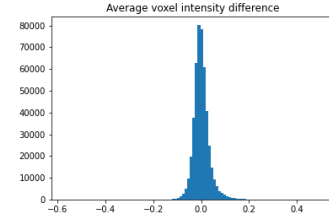


Figure 4.6: OD

#### 4.1.3 Visual comparison between predicted scalar measures and ground-truth

Here, a visual comparison between predicted scalar measure acquired with the use of the neural network (labeled MLP on figures) and ground-truth scalar measures acquired using AMICO (labeled AMICO on figures) is shown. Additionally, error maps are provided, which show the absolute difference between predicted and ground-truth data. Except from the visual part, the difference between the error maps and the histograms produced above, is that the error maps show the absolute difference whereas the histograms show the actual difference.

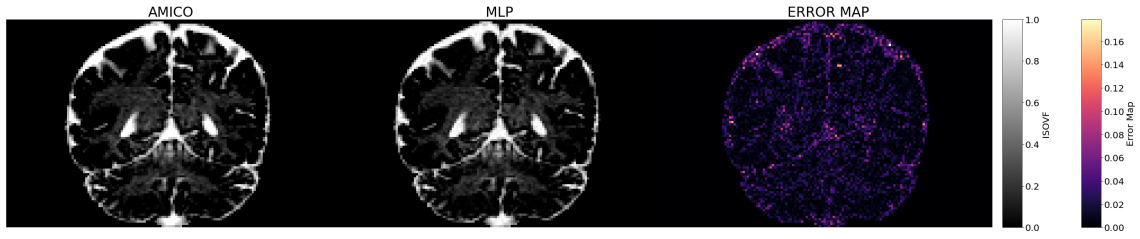


Figure 4.7: Visual comparison of ISOVF and associated error maps

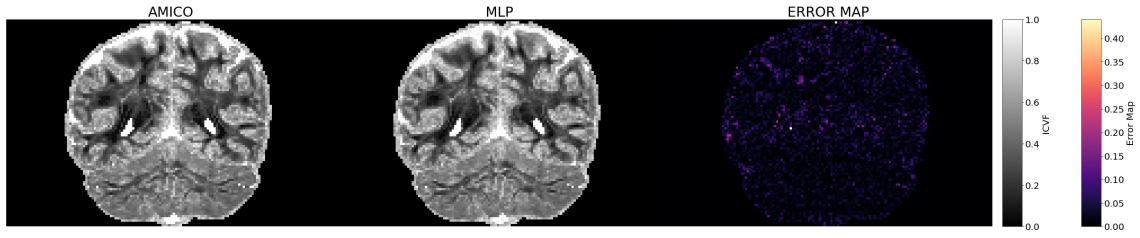


Figure 4.8: Visual comparison of ICVF and associated error maps

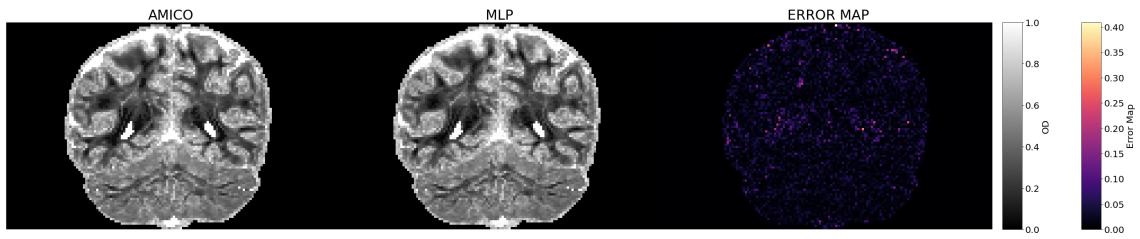


Figure 4.9: Visual comparison of OD and associated error maps

## 4.2 Experiment 2:

In experiment 2, both training and testing data come from different subjects acquired using the same scanner. The aim of this experiment is to see if the model is able to produce accurate predictions on test data of a different subjects, which have different brain structure.

Training set used: Person 1 - Center 10

Testing sets used: Person 2 - Center 10, Person 3 - Center 10

### 4.2.1 Loss Curves

The loss curves found when model was trained using data of Person 1 - Center 10 as training data are visible below. As it can be seen the loss curves obtained, are very similar (almost exactly the same) as the ones obtained from experiment 1. This gives an indication, which is then confirmed by both the evaluation metrics and the images of the scalar measurements produced, that the model is able to predict scalar measures on test sets of different subjects.

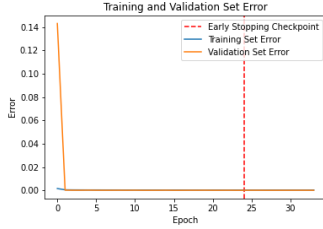


Figure 4.10: ISOVF

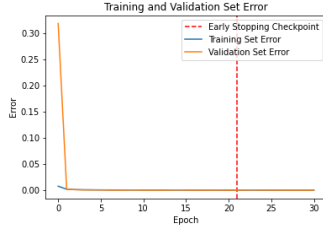


Figure 4.11: ICVF

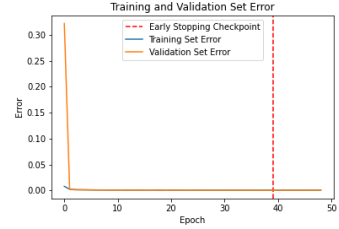


Figure 4.12: OD

### 4.2.2 Difference between ground-truth and predicted values

All the figures shown below are when the model is evaluated using the test set of Person 2 - Center 10. The histograms below, as in experiment 1 show the voxel-wise difference between ground-truth and predicted values. From the images, it easy to see that most voxels predicted are very close to ground-truth.

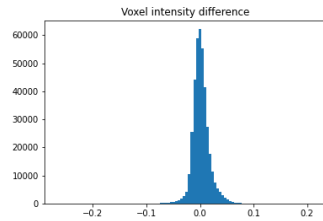


Figure 4.13: ISOVF

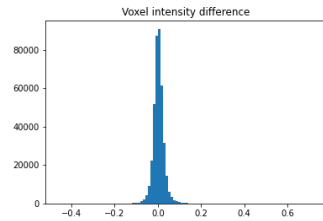


Figure 4.14: ICVF

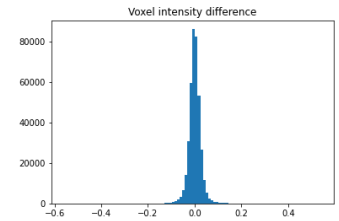


Figure 4.15: OD

### 4.2.3 Visual comparison between predicted scalar measures and ground-truth

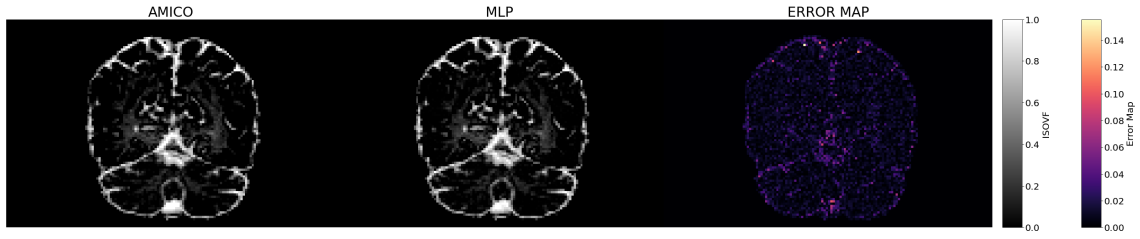


Figure 4.16: Visual comparison of ISOVF and associated error maps

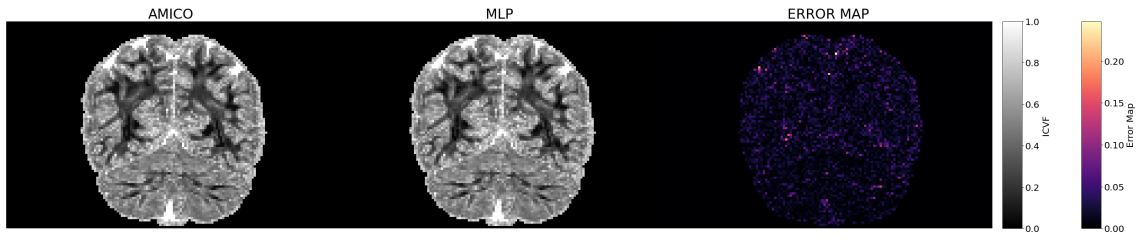


Figure 4.17: Visual comparison of ICVF and associated error maps

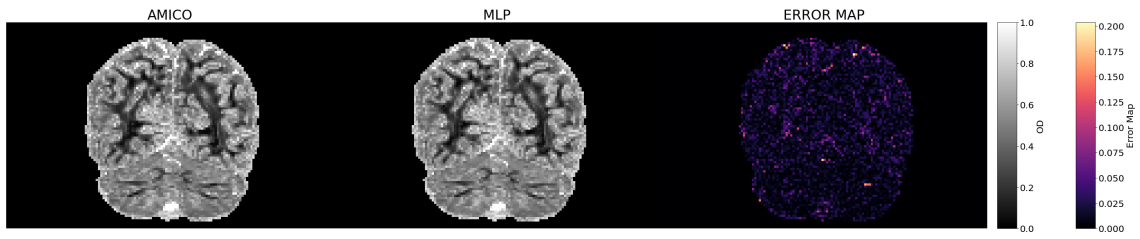


Figure 4.18: Visual comparison of OD and associated error maps

## 4.3 Experiment 3:

Experiments 1 and 2 are combined to form the 3rd experiment at which both training and testing data come from different subjects and are acquired using different scanners. The aim of this experiment is again to test how well the model generalizes on this type of test sets.

Training set used: Person 3 - Center 10

Testing sets used: Person 1 - Center 09, Person 1 - Center 02, Person 1 - Center 03, Person 1 - Center 01, Person 2 - Center 01, Person 2 - Center 02

### 4.3.1 Loss Curves

The loss curves found when the neural network model was trained using data of Person 3 - Center 10 are shown. It is noticeable again that the loss curves acquired are very similar to the ones from the first two experiments.

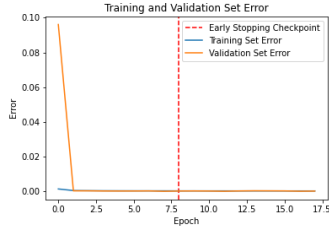


Figure 4.19: ISOVF

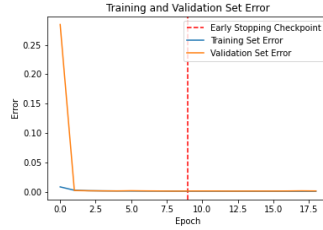


Figure 4.20: ICVF

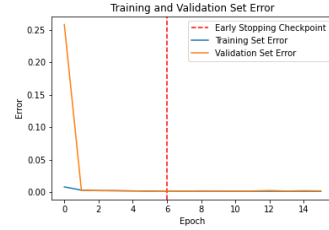


Figure 4.21: OD

### 4.3.2 Difference between ground-truth and predicted values

Histograms showing the difference between ground-truth data and predicted data when the model is evaluated on the Person 1 - Center 09 test set.

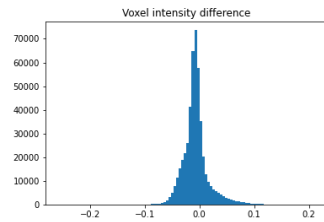


Figure 4.22: ISOVF

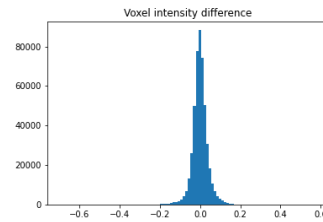


Figure 4.23: ICVF

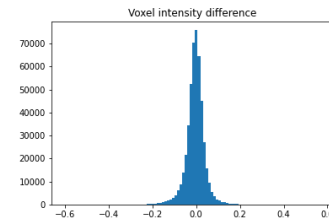


Figure 4.24: OD

### 4.3.3 Visual Comparison of predicted scalar measures with ground-truth

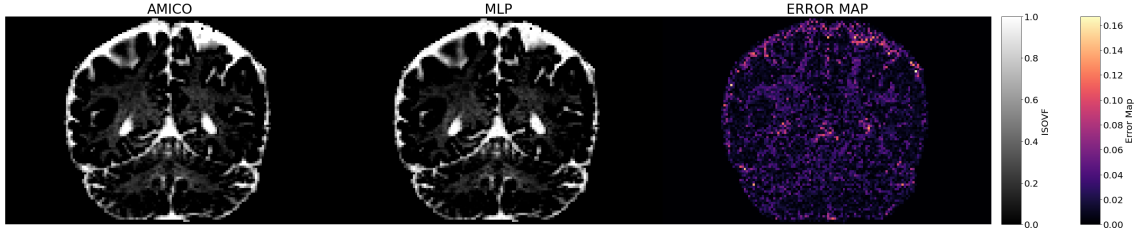


Figure 4.25: Visual comparison of ISOVF and associated error maps

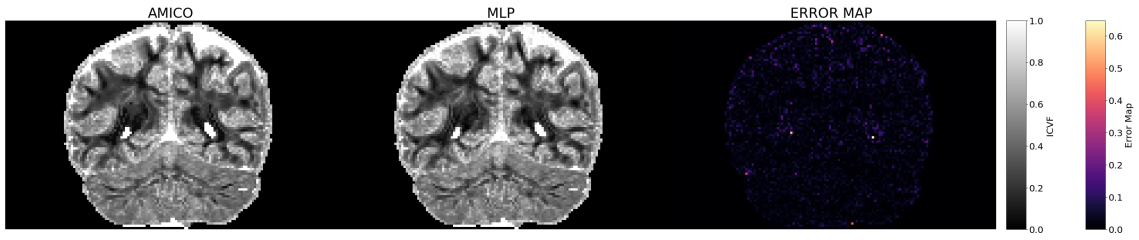


Figure 4.26: Visual comparison of ICVF and associated error maps

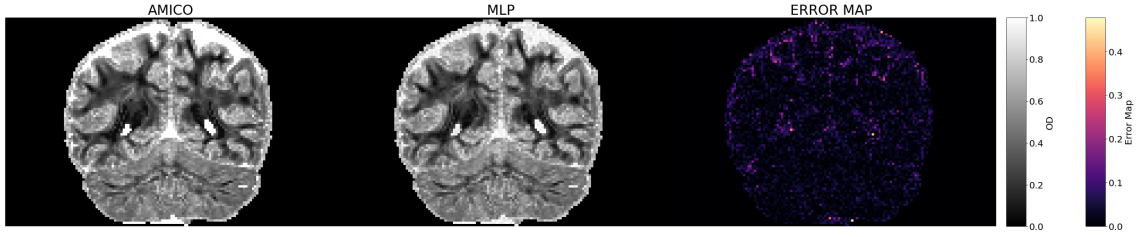


Figure 4.27: Visual comparison of OD and associated error maps



## 4.4 Experiment 4:

As it can be seen from the first three experiments, the neural network model does a very good job in generalizing and predicting on unseen data of different subjects acquired using different scanners. It is evident, that by using the fully sampled training dataset, the model is able to produce predictions extremely close to the ground-truth scalar measures. The promising results of the first experiments, led to fourth experiment, which aims to predict scalar measures using a sub-sampled set of DWIs. The original training set consists of 96 DWIs, and in this experiment sub-sampling factors of 2, 3, 4, 6, 12 and 24 are tested i.e using 48, 32, 24, 16, 8 and 4 DWIs. It is worthwhile to note, that in this experiment training and testing data come from a different subject, acquired from different scanner. In this experiment only the visual comparison is shown.

Training set used: Person 3 - Center 10

Testing set used: Person 1 - Center 09

### 4.4.1 Visual Comparison of predicted scalar measures with ground-truth

Below, you can see maps of the orientation dispersion index (OD) in the human brain using the sub-sampling factors discussed above. It is important to note that the AMICO map is constant in all figures and shows the ground-truth data computed using the fully sampled training set (i.e using 96 DWIs). The mappings of ICVF and ISOVF can be found in the appendix (ADD SECTION)

#### Sub-sampling factor 2: 48 DWIs

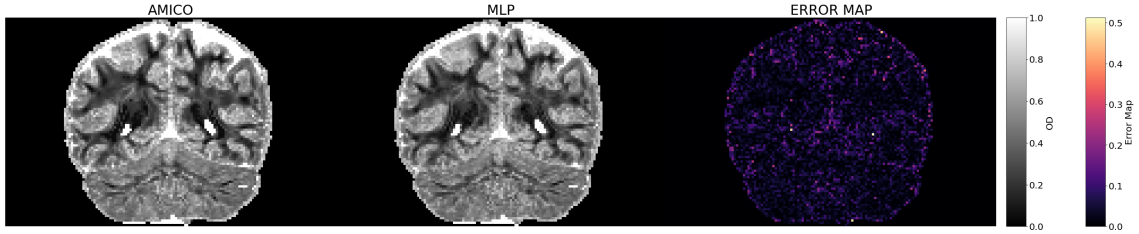


Figure 4.28: Visual comparison of OD and associated error maps using 48 DWIs

#### Sub-sampling factor 3: 36 DWIs

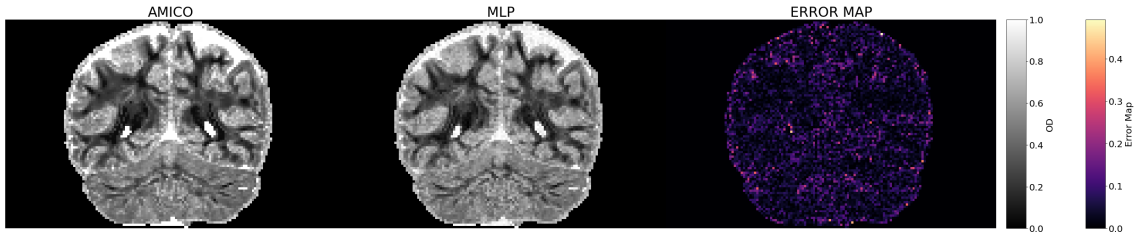


Figure 4.29: Visual comparison of OD and associated error maps using 36 DWIs

**Sub-sampling factor 4: 24 DWIs**

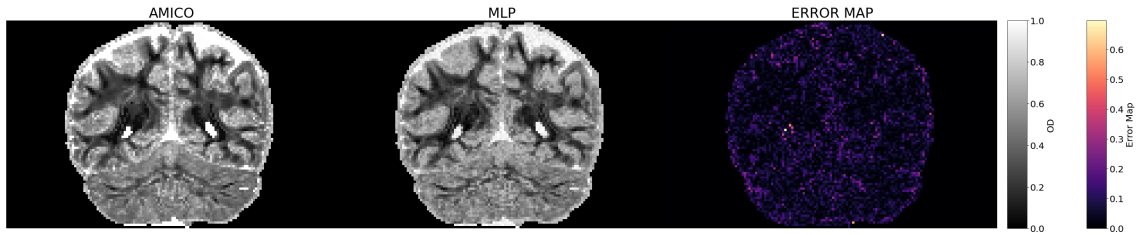


Figure 4.30: Visual comparison of OD and associated error maps using 24 DWIs

**Sub-sampling factor 6: 16 DWIs**

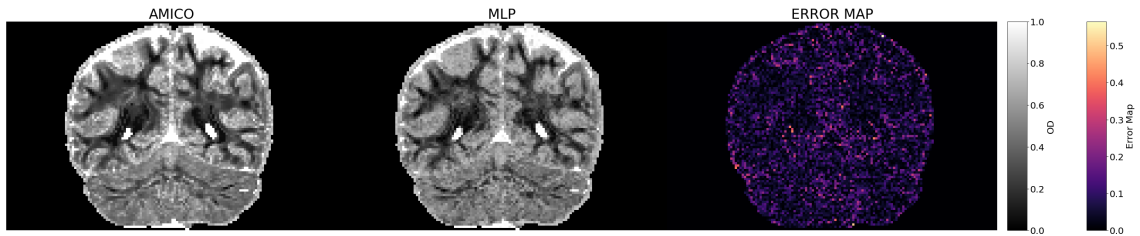


Figure 4.31: Visual comparison of OD and associated error maps using 16 DWIs

**Sub-sampling factor 12: 8 DWIs**

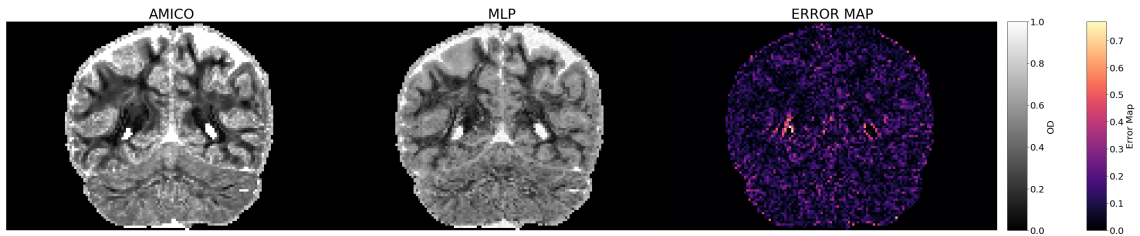


Figure 4.32: Visual comparison of OD and associated error maps using 8 DWIs

**Sub-sampling factor 24: 4 DWIs**

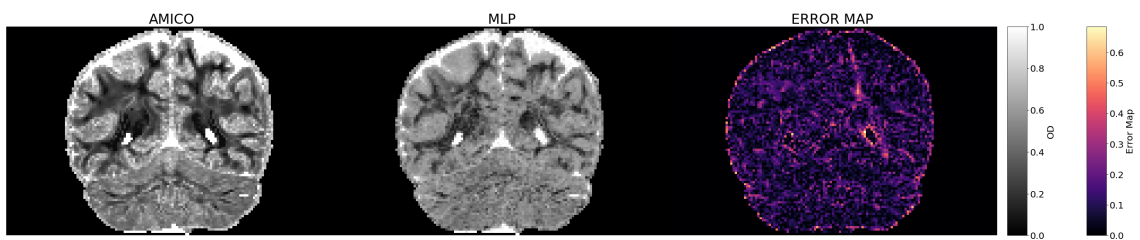


Figure 4.33: Visual comparison of OD and associated error maps using 4 DWIs

## 4.5 Experiment 5

Experiment 5 aims again to estimate scalar measures using sub-sampled data, but instead of using a sub-sample of DWI's as in the 4th experiment, this time a sample of the brain data is used for the training of the neural network. Results from experiment 4 shows that a lot of information is contained in a small number of DWI's, so it is of interest to check if the same holds when using a sample of the brain data. In this experiment sub-sampling factors of 2, 4, 6, 10, and 20 were tested i.e using 1/2 of the brain data, 1/4 of the brain data, 1/6 of the brain data, 1/10 of the brain data and 1/20 of the brain data.

Training set used: Person 3 - Center 10

Testing set used: Person 1 - Center 09

### 4.5.1 Visual Comparison of predicted scalar measures with ground-truth

**Sub-sampling factor 2: 1/2 of the brain**

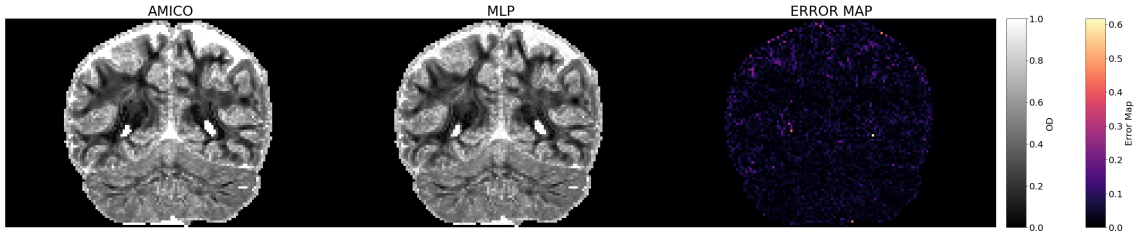


Figure 4.34: Visual comparison of OD and associated error maps using 1/2 of the brain voxels for training

**Sub-sampling factor 4: 1/4 of the brain**

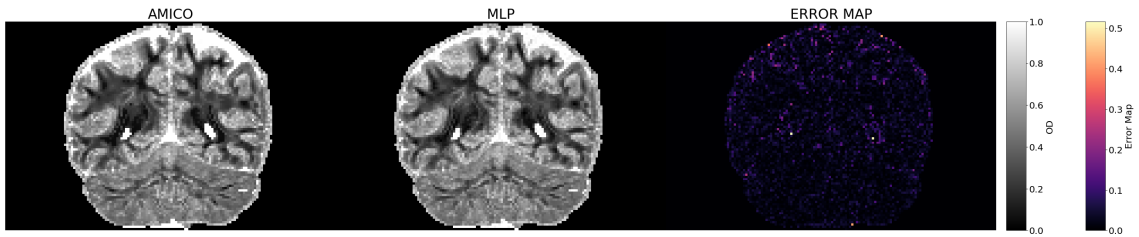


Figure 4.35: Visual comparison of OD and associated error maps using 1/4 of the brain voxels for training

**Sub-sampling factor 6: 1/6 of the brain**

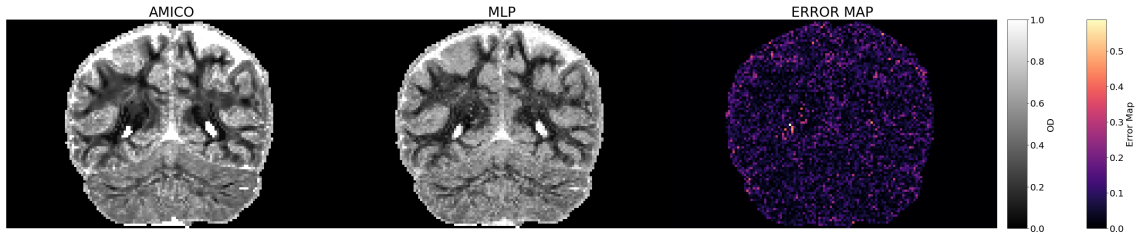


Figure 4.36: Visual comparison of OD and associated error maps using 1/6 of the brain voxels for training

**Sub-sampling factor 10: 1/10 of the brain**

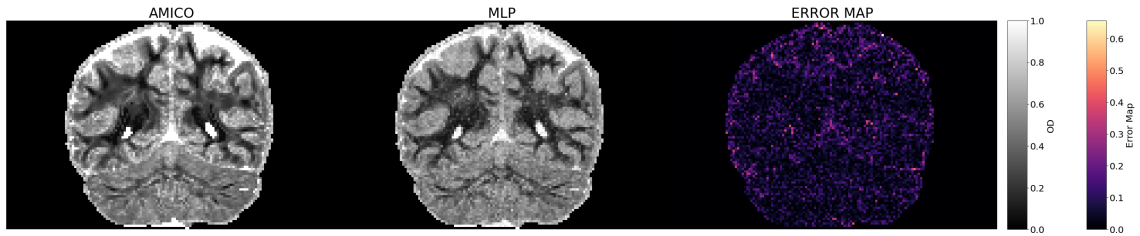


Figure 4.37: Visual comparison of OD and associated error maps using 1/10 of the brain voxels for training

**Sub-sampling factor 20: 1/20 of the brain**

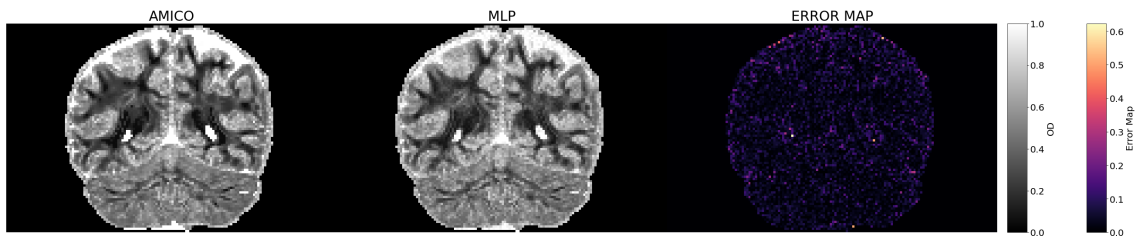


Figure 4.38: Visual comparison of OD and associated error maps using 1/20 of the brain voxels for training

# Chapter 5

## Results

The aim of this section is to introduce the evaluation metrics used to evaluate the performance of the feed-forward neural network and provide a concise summary of the results obtained from the experiments conducted in the Experiments section.

### 5.1 Evaluation Metrics

The evaluation metrics described below were used to evaluate the performance of the neural network.

#### 5.1.1 Root Mean Square Error (RMSE)

Root Mean Square Error is defined as the square root of mean squared error (MSE). RMSE computes the differences between values predicted by the neural network model and the ground-truth values. RMSE measures the quality of the fit between the ground-truth data and the predicted model, and is one of the most frequently used measures of the goodness of fit of generalized regression models. As the errors are squared before the average is taken, RMSE gives a relatively high weight to large errors, therefore making it a useful metric when large errors are particularly undesirable. RMSE, is defined formally as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^I (\hat{y}_i - y_{igt})^2}{J}} \quad (5.1)$$

where, the sum is taken over all  $I$  voxels,  $\hat{y}_i$  is the predicted intensity at voxel  $i$  and  $y_{igt}$  is the considered true intensity at voxel  $i$ .

#### 5.1.2 Mean absolute error (MAE)

Mean absolute error is the average of the absolute values of the differences between model predictions and ground-truth values. In MAE is a linear measure, therefore, all the individual voxel intensity differences are equally weighted in the average. MAE is defined formally as:

$$MAE = \frac{1}{n} \sum_{i=1}^I | \hat{y}_i - y_{igt} | \quad (5.2)$$

where, the sum is taken over all  $I$  voxels,  $\hat{y}_i$  is the predicted intensity at voxel  $i$  and  $y_{igt}$  is the considered true intensity at voxel  $i$ .

### 5.1.3 Mean Bias Error (MBE)

Mean bias error is the average of differences between model predictions and ground-truth values and captures the average bias in the prediction. It is important to note here that the mean bias error wasn't used as the main evaluation metric (RMSE and MAE were) for all the experiments conducted, but rather a measure used to decide if any further steps should be taken to address any abnormal/unexpected results. MBE is defined formally as:

$$MBE = \frac{1}{n} \sum_{i=1}^I (\hat{y}_i - y_{igt}) \quad (5.3)$$

where, the sum is taken over all  $I$  voxels,  $\hat{y}_i$  is the predicted intensity at voxel  $i$  and  $y_{igt}$  is the considered true intensity at voxel  $i$ .

## 5.2 First 3 experiments Results

The aim of the first three experiments was to test the ability of the model to generalize on unseen data. This was approached by evaluating the model on test sets of the same subject acquired from a different scanner, evaluating the model on test sets of different subjects acquired from the same scanner, and lastly evaluating the model on test sets of different subjects acquired from different scanners. In addition, another implied goal of the first three experiments was to demonstrate the use of dMRI signal values directly as input to the deep learning model, unlike traditional model fitting methods. Tables 5.1, 5.2 and 5.3 show the evaluation metrics results got from the first three experiments. It is important to note here, that separate RMSE, MAE, MBE were computed for each one of the scalar measures ISOVF, ICVF, OD, hence the values shown in the table are averaged over the three measures.

The results of the first three experiments show that predicted scalar measures are very close to ground-truth scalar measures obtained from traditional model fitting. Evaluation metrics results demonstrate that the model is able to predict accurate scalar measures when evaluated on both test data of different scanners and test data of different subjects. Comparing the results of experiment 1 and experiment 2, it is noticeable that the evaluation metrics of experiment 1 are higher than the ones in the 2nd experiment. Even though the differences are minimal, this gives an indication that data acquired from different scanners are generally harder to predict than data of different subjects acquired with the same scanner. Results of the 3rd experiment are higher than the rest, and this was expected as the test data come from different subjects and scanners.

To conclude, the first 3 experiments demonstrate that deep learning is able to predict scalar measures accurately when the fully-sampled training sets are used.

Experiment 1: Evaluation Metrics Results			
Mean Train Error	Mean Test Error		
Person 1 - Center 10	0.032	0.022	0.00091
Person 1 - Center 01	0.031	0.022	-0.00058
Person 1 - Center 02	0.031	0.022	-0.00013
Person 1 - Center 03	0.031	0.022	0.00088

Table 5.1: Evaluation metrics results from experiment 1 showing the average RMSE, average MAE and average MBE for all three scalar measures. Testing data used were of the same subject acquired with a different scanner.

Experiment 2: Evaluation Metrics Results			
Testing Dataset	RMSE	MAE	MBE
Person 2 - Center 10	0.025	0.017	-0.00079
Person 3 - Center 10	0.023	0.016	-0.00081

Table 5.2: Evaluation metrics results from experiment 2 showing the average RMSE, average MAE and average MBE for all three scalar measures. Testing data used were of different subjects acquired with the same scanner.

Experiment 3: Evaluation Metrics Results			
Testing Dataset	RMSE	MAE	MBE
Person 1 - Center 09	0.037	0.025	-0.00041
Person 1 - Center 02	0.037	0.028	-0.00058
Person 1 - Center 03	0.041	0.031	-0.00059
Person 1 - Center 01	0.038	0.028	0.00057
Person 2 - Center 01	0.042	0.033	-0.00064
Person 2 - Center 02	0.044	0.034	0.00065

Table 5.3: Evaluation metrics results from experiment 3 showing the average RMSE, average MAE and average MBE for all three scalar measures. Testing data used were of different subjects acquired with different scanners.

### 5.3 Fourth Experiment Results

The aim of the fourth experiment was to compute microstructural indices from a reduced set of DWIs. Figure 5.1 shows the RMSE values calculated for all the three scalar measures (ISOVF, ICVF, OD) separately using 48, 36, 16, 8 and 4 DWIs. Table 5.1 also shows the RMSE, MAE and MBE values for all the experiments. It should be noted that in the Table 5.1 the average values are shown across the three scalar measures.

In experiment 3, where the fully sampled training set was used, the values found were: RMSE = 0.025, MAE = 0.017 and MBE = -0.00079. Comparing that with the results of the figures below, it is noticeable that the neural network is able to compute scalar measures extremely close to ground-truth with only 16 DWIs, with the difference in RMSE being only 0.054 (in comparison with the fully sampled training set). A large increase is seen when 8 and 4 DWIs are used with the RMSE being around 10 times more than in the previous settings, but as it was illustrated in the experiments section through the visual maps, using 4 and 8 DWIs acceptable results are achieved. It is also important to note, that the ISOVF scalar measure produced the lowest RMSE values compared to others across all the sub-sampling factors tested in this experiment. With just 8 DWIs the RMSE value of ISOVF was around 10 times less than the others, hence the large increase seen when using 8 DWIs is largely caused by the increase in OD and ICVF scalar measures. This is something that could be examined in future work, in order to understand the reason why some measures are predicted with more accuracy. Possible explanations for this are that some scalar measures are easier to be computed due to some inherent characteristic they have, or maybe further fine-tuning of the model is required.

To conclude, this experiment demonstrated that scalar measures can be estimated with high accuracy with a reduced set of 16 DWIs (for ICVF and OD) and 8 DWIs (for ISOVF), offering a possible reduction in scan-time up to a factor of 12, significantly improving clinical feasibility.

Experiment 4: Evaluation Metrics Results			
Number of DWIs	RMSE	MAE	MBE
48	0.052	0.036	-0.0055
36	0.059	0.043	-0.0024
24	0.069	0.049	-0.0053
16	0.079	0.057	-0.0084
8	0.10	0.076	-0.0098
4	0.14	0.10	-0.0039

Table 5.4: Evaluation metrics results from experiment 4 showing the average RMSE, average MAE and average MBE for all three scalar measures.



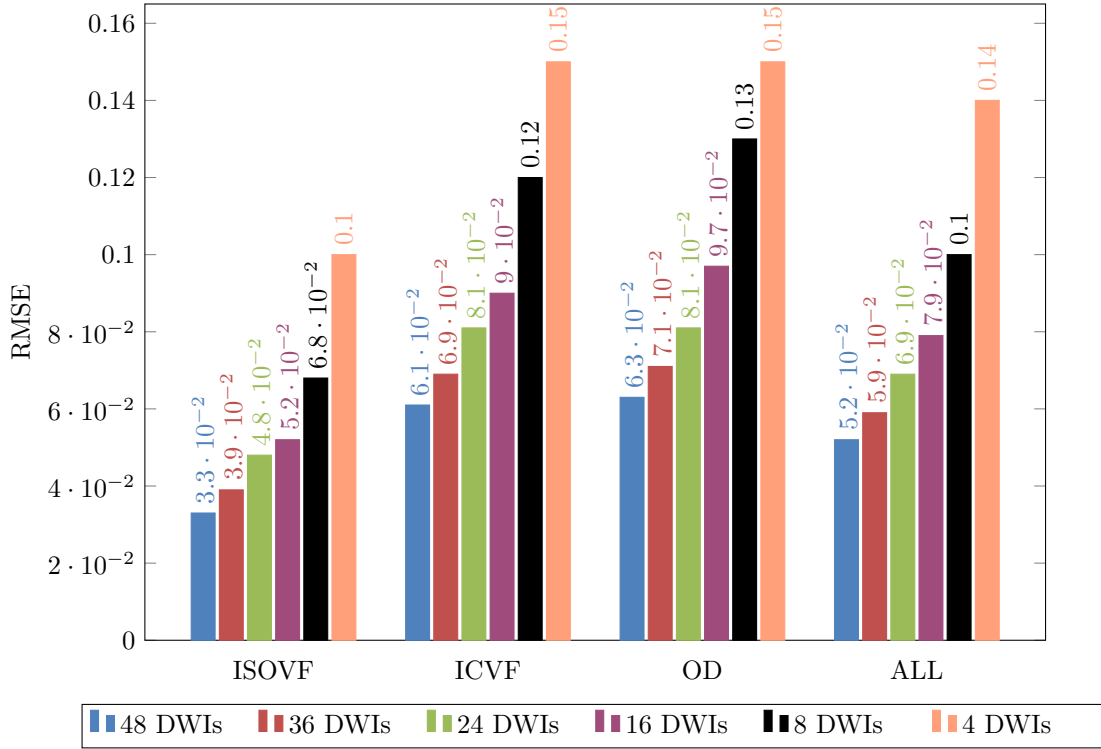


Figure 5.1: RMSE values computed for ISOVF, ICVF and OD using 48 DWIs, 36 DWIs, 24 DWIs, 16 DWIs, 8 DWIs and 4 DWIs.

## 5.4 Fifth Experiment Results

The aim of the 5th experiment was again to compute microstructural indices using sub-sampled data, but instead of using a reduced set of DWIs, the goal here was to use a sub-sample of the total brain voxels that are subsequently used for the training of the neural network. Figure 5.2 shows the RMSE values got when using 1/2, 1/4, 1/6, 1/10 and 1/20 of the brain data for all the scalar measures separately. Table 5.2 displays the RMSE, MAE, MBE averaged across all three scalar measures.

In experiment 3, where the fully sampled training set was used, the values found were: RMSE = 0.025, MAE = 0.017 and MBE = -0.00079. As it can be seen from the figures below, in all the sub-sampling factors tested, the evaluation metrics are very close to ground-truth. Specifically, an increase in RMSE of just 0.044 is seen when using 1/20 of the brain instead of the full brain for training. The minimal difference shows that it is not necessary to train the neural network using the full brain data and that using a sample of as low as the 1/20 of the total data still produces accurate results. It is worthwhile to say here, that a particular region of the brain wasn't selected in this experiment (i.e the first half of the brain). Instead, brain voxels were sampled uniformly. The effect and possibly a limitation of random-sampling can be seen from the difference in the results produced when 10 and 20 sub-sampling factors were applied, respectively. RMSE and MAE of the former are larger, and the logical explanation for this is that the random data selected were more helpful for the neural network to learn, even though twice as more voxels were used for training in the latter. In the future, it would be interesting to examine the differences between selective sampling and random sampling.

To conclude, this experiment showed that a sub-sample of the brain data can be used for the training of the neural network instead of using all the brain voxels. This reduces the computation

time required for training the neural network, and illustrates the fact that it is not necessary to scan the whole brain to compute microstructural indices, achieving a further reduction in scan time.

Experiment 5: Evaluation Metrics Results				
Sub-sampling factor	Number of brain voxels	RMSE	MAE	MBE
2	225564	0.037	0.025	-0.0052
4	112782	0.051	0.035	-0.0025
6	75188	0.072	0.053	-0.0056
10	45112	0.079	0.058	-0.0020
20	22556	0.069	0.050	0.0017

Table 5.5: Evaluation metrics results from experiment 5 showing the average RMSE, average MAE and average MBE for all three scalar measures.

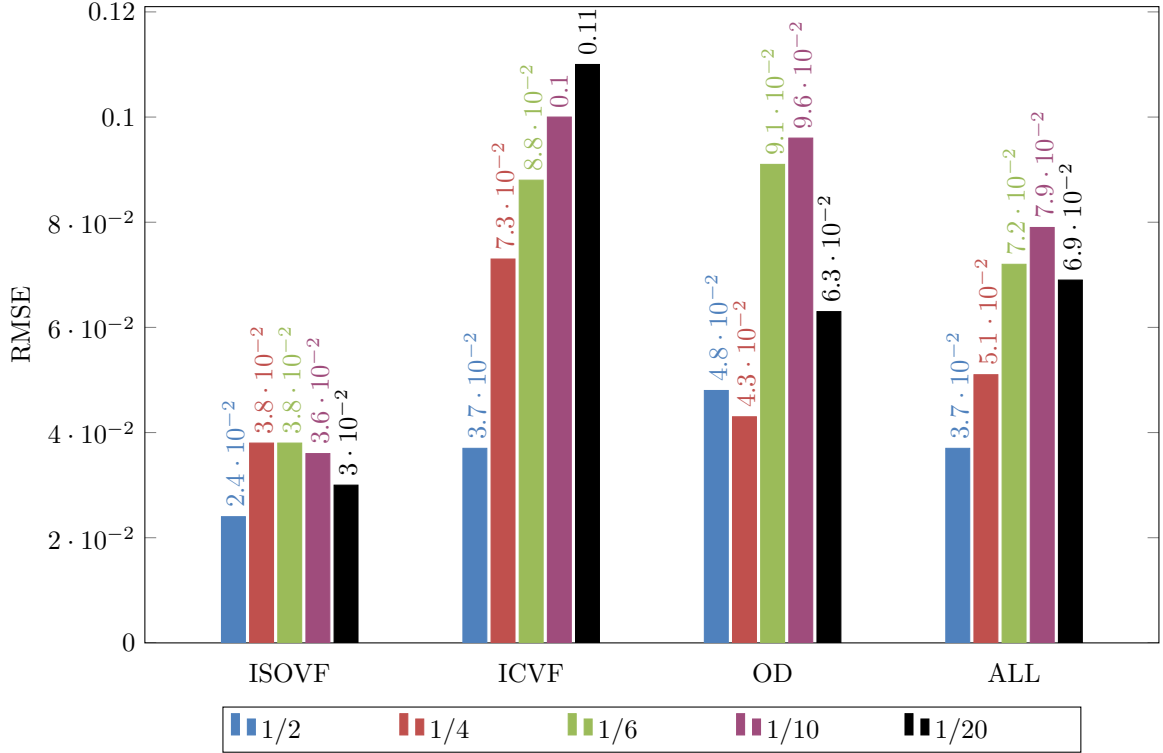


Figure 5.2: RMSE values computed for ISOVF, ICVF and OD using 1/2, 1/4, 1/6, 1/10, 1/20 of the total brain voxels for training

## 5.5 Computation time comparison

Finally, a comparison between the computation time required for training the AMICO model and the neural network model is shown. All the experiments conducted were executed on 2.3GHz Intel Core i5 CPU with 16GB of DDR3 RAM. Computation times were compared by training both models on the fully-sampled training datasets illustrated in table 5.6. Results indicate that the neural network is on average 32 times faster than AMICO.

Computation time comparison (in seconds)			
Training Dataset	AMICO	MLP	Ratio
Person 1 - Center 01	990	33	30
Person 1 - Center 02	959	27	35
Person 1 - Center 03	972	27	35
Person 1 - Center 10	1054	27	38
Person 2 - Center 10	741	29	25
Person 3 - Center 03	1016	33	30

Table 5.6: Computation time comparison between AMICO and the multi-layer perceptron on multiple training datasets.

# Chapter 6

## Discussion

### 6.1 Achievements

- The use of DWI signals directly as input to the deep learning model, which require minimal pre-processing in comparison with traditional model-fitting techniques.
- In the experiments section, it has been demonstrated that the neural network model generalizes well on unseen data and is able to produce accurate characterizations of tissue microstructure on data of different subjects acquired with the use of a different scanners.
- A large amount of information is included in a small number of DWIs, and as such a dataset using a reduced set of DWIs (up to 8), instead of a fully sampled can be used for the training of the neural network. Using a sub-sampled set, scan-time reduction is reduced by a factor of 12, demonstrating the potential that deep learning has for use in clinical environments.
- It has been demonstrated that the neural network can be trained with a sample (up to 1/20) of the total brain voxels and still compute accurate scalar measures, resulting in training time reduction. This experiment has also demonstrated that it is not necessary to scan the whole brain to estimate tissue microstructure, and thus there is opportunity for reducing further the acquisition time.
- A significant training time reduction compared to the traditional model fitting technique, AMICO. More precisely, the deep learning method is on average 32 times faster than AMICO, enabling large-scale studies to be conducted for the estimation of tissue microstructure.

### 6.2 Critical Evaluation and Future Work

In this project, the use of a deep learning framework for the estimation of invariant scalar measures that quantify tissue microstructure was demonstrated. The main parts of the framework are the feed-forward neural network and the random down-sampling technique used which enables the neural network to learn with a sub-sampled set instead of the fully-sampled one. Although it has been demonstrated that the use of a multi-layer perceptron in combination with random-down sampling produces accurate characterizations of tissue microstructure, both parts of the framework could be further improved. Below I outline my suggestions for future work for both of the parts stated, that would in turn lead to a new research project.

- Selective sampling techniques can be tested instead of random sampling in order to see if there is any improvement in the accuracy of the estimated scalar measures. Using selective sampling could also lead in computing scalar measures with less DWIs, thus further reducing the acquisition times. Heuristic experiments can be conducted first to gauge the most relevant DWIs and then the neural network can be trained with the sets produced from the heuristic experiments.
- Different neural network architectures can also be employed. Specifically, convolutional neural networks can be used which have been proven to be very effective in image processing related tasks, as they take into account spacial information of the surrounding voxels. Using a convolution neural network, could lead in improving estimation accuracy, as well as computing scalar measures with less DWIs.
- Although the computation time of the feed-forward neural network used in this project is very low (on average 28s), further training time reduction can be obtained by training the neural network with importance sampling. The authors of [28] indicate that neural network training spends most of the computation on training points that are handled properly and thus could be ignored. They propose a way of training neural networks with importance sampling, where the neural network spends more time learning "informative" examples.

## 6.3 Final Thoughts

Overall, I can say that this project has been successful, although there is a lot of room for improvement. Given the fact that I have never taken deep learning or image processing related modules before, this project has been a great challenge. I believe this project could have delivered more, if more emphasis was given in first understanding some methods/techniques before going straight to development. To conclude, this project offered me the opportunity to gain an understanding of deep learning techniques and appreciate a lot their application in the healthcare field.

# Bibliography

- [1] Pytorch.org. 2021. PyTorch. [online] Available at: <https://pytorch.org/>
- [2] Scikit-learn.org. 2021. scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation. [online] Available at: <https://scikit-learn.org/stable/>
- [3] Nipy.org. 2021. Neuroimaging in Python — NiBabel 3.2.1+100.g9fdf5e3e documentation. [online] Available at: <https://nipy.org/nibabel/>
- [4] <https://nilearn.github.io/>
- [5] Fsl.fmrib.ox.ac.uk. 2021. FSLeyes - FslWiki. [online] Available at: <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes>
- [6] Baliyan, V., Das, C., Sharma, R. and Gupta, A., 2016. Diffusion weighted imaging: Technique and applications. pp. 1-3, 2016
- [7] Basser, P., Mattiello, J. and LeBihan, D., MR diffusion tensor spectroscopy and imaging *Biophysical Journal*, 66(1), pp. 259-267,1994
- [8] Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., van der Walt, S., Descoteaux, M. and Nimmo-Smith, I. Dipy, a library for the analysis of diffusion MRI data. pp. 6-14,2014
- [9] Wedeen, V., Wang, R., Schmahmann, J., Benner, T., Tseng, W., Dai, G., Pandya, D., Hagmann, P., D’Arceuil, H. and de Crespigny, A., Diffusion spectrum magnetic resonance imaging (DSI) tractography of crossing fibers, *NeuroImage*, 41(4), pp.1267-1277,2008
- [10] Tuch, D., Q-ball imaging, *Magnetic Resonance in Medicine*, 52(6), pp.1358-1372. pp.1267-1277,2004
- [11] Canales-Rodríguez, E., Melie-García, L. and Iturria-Medina, Y., Mathematical description of q-space in spherical coordinates: Exact q-ball imaging, *Magnetic Resonance in Medicine*, 61(6), pp.1350-1367,2009
- [12] Aganj, I., Lenglet, C., Sapiro, G., Yacoub, E., Ugurbil, K. and Harel, N., Reconstruction of the orientation distribution function in single- and multiple-shell q-ball imaging within constant solid angle, *Magnetic Resonance in Medicine*, 64(2), pp.554-566,2010
- [13] Descoteaux, M., Deriche, R., Knosche, T. and Anwander, A., Deterministic and Probabilistic Tractography Based on Complex Fibre Orientation Distributions, *IEEE Transactions on Medical Imaging*, 28(2), pp.269-286,2009

- [14] Pierpaoli, C., Jezzard, P., Basser, P., Barnett, A. and Di Chiro, G., Diffusion tensor MR imaging of the human brain, *Radiology*, 201(3), pp.637-648, 1996
- [15] Zhang, H., Schneider, T., Wheeler-Kingshott, C. and Alexander, D., 2012. NODDI: Practical in vivo neurite orientation dispersion and density imaging of the human brain, *NeuroImage*, 61(4) pp.1000-1016, 2012
- [16] Alexander, D., Hubbard, P., Hall, M., Moore, E., Ptito, M., Parker, G. and Dyrby, T., Orientationally invariant indices of axon diameter and density from diffusion MRI, *NeuroImage*, 52(4), pp.1374-1389, 2010
- [17] Dyrby, T., Sgaard, L., Hall, M., Ptito, M. and Alexander, D., Contrast and stability of the axon diameter index from microstructure imaging with diffusion MRI, *Magnetic Resonance in Medicine*, 70(3), pp.711-721, 2012
- [18] Daducci, A., Canales-Rodríguez, E., Zhang, H., Dyrby, T., Alexander, D. and Thiran, J., Accelerated Microstructure Imaging via Convex Optimization (AMICO) from diffusion MRI data, *NeuroImage*, 105, pp.32-44, 2015
- [19] Jensen, J., Helpert, J., Ramani, A., Lu, H. and Kaczynski, K., Diffusional kurtosis imaging: The quantification of non-gaussian water diffusion by means of magnetic resonance imaging, *Magnetic Resonance in Medicine*, 53(6), pp.1432-1440, 2005
- [20] Lu, H., Jensen, J., Ramani, A. and Helpert, J., Three-dimensional characterization of non-gaussian water diffusion in humans using diffusion kurtosis imaging, *NMR in Biomedicine*, 19(2), pp.236-247, 2006
- [21] Hansen, B., Lund, T., Sangill, R. and Jespersen, S., Experimentally and computationally fast method for estimation of a mean kurtosis, *Magnetic Resonance in Medicine*, 69(6), pp.1754-1760, 2013
- [22] LeCun, Y., Bengio, Y. and Hinton, G., Deep learning, *Deep learning. Nature*, 521(7553) pp.436-444, 2015
- [23] Tong, Q., He, H., Gong, T., Li, C., Liang, P., Qian, T., Sun, Y., Ding, Q., Li, K. and Zhong, J., Multicenter dataset of multi-shell diffusion MRI in healthy traveling adults with identical settings *MRI in healthy traveling adults with identical settings. Scientific Data*, 7(1), 2020
- [24] Saxe, Andrew McClelland, James Ganguli, Surya, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2013  
*The Annals of Statistics*, 48(4), 2020
- [25] Kohler, M. and Langer, S., Discussion of: "Nonparametric regression using deep neural networks with ReLU activation function", *The Annals of Statistics*, 48(4), 2020
- [26] Srivastava, Nitish Hinton, Geoffrey Krizhevsky, Alex Sutskever, Ilya Salakhutdinov, Ruslan. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*. 15, pp 1929-1958, 2014
- [27] Kingma, Diederik Ba, Jimmy, Adam: A Method for Stochastic Optimization, *International Conference on Learning Representations*, 2014

- [28] Katharopoulos, Angelos Fleuret, Francois, Not All Samples Are Created Equal: Deep Learning with Importance Sampling, 2018
- [29] Weiss, Tomer Vedula, Sanketh Senouf, Ortal Michailovich, Oleg AlexBronstein, Towards learned optimal q-space sampling in diffusion MRI, 2020



## Appendix A

### Other appendices, e.g., code listing

All the code used in this dissertation can be accessed using this link <https://github.com/AlexFrangos/Dissertation> which gives access to the GitHub repository.

.