MODELING BRAIN NETWORKS WITH ARTIFICIAL NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BARAN BARIŞ KIVILCIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2019

Approval of the thesis:

MODELING BRAIN NETWORKS WITH ARTIFICIAL NEURAL NETWORKS

submitted by **BARAN BARIŞ KIVILCIM** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Fatoş T. Yarman Vural
Supervisor, **Computer Engineering, METU** _____

Dr. Itır Önal Ertuğrul
Co-supervisor, **Robotics Institute, Carnegie Mellon University** _____

Examining Committee Members:

Assist. Prof. Dr. Hande Alemdar
Computer Engineering, METU _____

Prof. Dr. Fatoş T. Yarman Vural
Computer Engineering, METU _____

Assist. Prof. Dr. Hamdi Dibeklioğlu
Computer Engineering, Bilkent University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Baran Barış Kıvılcım

Signature :

ABSTRACT

MODELING BRAIN NETWORKS WITH ARTIFICIAL NEURAL NETWORKS

Kıvılcım, Baran Barış
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Fatoş T. Yarman Vural
Co-Supervisor
: Dr. Itır Önal Ertuğrul

January 2019, 100 pages

In this thesis, we propose three novel artificial neural network approaches to capture the functional connectivities among anatomic brain regions. Using the first proposed method, we estimate directed and undirected brain networks for each subject (subject-level) and validate the estimated brain networks on brain decoding performance using Support Vector Machines for classification. We propose our second method in order to estimate brain networks to represent each task (task-level) and we also propose a novel classification method in order to categorize an unseen chunk of fMRI data into cognitive states using reconstruction power of these directed and undirected task-level brain networks. In our last method, we use multilayer perceptrons for classification of brain networks and we propose two novel artificial neural network models. The first model of the last method estimates multiple brain networks simultaneously and the second model is an end to end model that estimates and classifies brain networks in a single architecture. All three proposed methods offer directed and undirected brain networks and we show that both directed and undirected brain networks estimated with the proposed methods offer state of the art classification accuracy on two

functional magnetic resonance imaging datasets, namely HCP (Human Connectome Project) dataset and TOL (Tower of London) dataset.

Keywords: Brain Networks, Brain Decoding, Artificial Neural Networks

ÖZ

BEYİN AĞLARININ YAPAY SİNİR AĞLARIYLA MODELLENMESİ

Kıvılcım, Baran Barış
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Fatoş T. Yarman Vural
Ortak Tez Yöneticisi
: Dr. Itır Önal Ertuğrul

Ocak 2019 , 100 sayfa

Bu tez çalışmasında, yapay sinir ağlarını kullanarak anatomik beyin bölgeri arasındaki işlevsel bağlantıları kestiren üç özgün yöntem öneriyoruz. Önerdiğimiz ilk yöntemi kullanarak, her denek için (denek seviyesinde) yönlü ve yönsüz beyin ağları oluşturup bu beyin ağlarının beyin çözümleme performanslarını destek vektör makineleri aracılığıyla ölçüyoruz. Önerdiğimiz ikinci yöntem ile zihinsel görev seviyesinde beyin ağları oluşturuyor ve bu yönlü ve yönsüz beyin ağlarının geçerliliğini yine kendi önerdiğimiz önceden görülmemiş bir veri yığınını zihinsel görevlere göre kategorileyen ve bu amaçla beyin ağlarının yeniden yapılandırma gücünü kullanan bir sınıflandırma yöntemi ile test ediyoruz. Son yöntemimizde, çok katmanlı algılayıcı mimarisini kullanarak beyin ağlarını sınıflandırıyoruz ve iki özgün yapay sinir ağları modeli öneriyoruz. Bu modellerin ilki, eş zamanlı çoklu beyin ağı oluşturmakta ve ikinci model ise uçtan uca bir yapay sinir ağı yapısı olup, beyin ağlarının oluşturması ve sınıflandırması görevlerini tek bir yapay sinir ağı mimarisinde buluşturmaktadır. Önerdiğimiz üç yöntem de hem yönlü hem de yönsüz beyin ağları sunmaktadır ve iki fonksiyonel manyetik rezonans görüntüleme verisetini, HCP (Human Connec-

tome Project) ve TOL (Tower of London), kullanarak yaptığımız deneyler ile önerilen yönlü ve yönsüz beyin ağlarının literatürdeki beyin ağları oluşturan yöntemlere kıyasla daha yüksek sınıflandırma doğruluğu sağladığını gösteriyoruz.

Anahtar Kelimeler: Beyin Ağları, Beyin Çözümleme, Yapay Sinir Ağları

To my family

ACKNOWLEDGMENTS

I know how hard and sometimes impossible to train a neural network on multiple objectives. However, what is even harder is to do it for a student. Other than and much more important than science and engineering, Prof. Dr. Fatoş Tünay Yarman Vural has been teaching me how to be an essential human with manners, self esteem, curiosity, awareness of his surroundings. Like carving a stone into a flawless statue, I can say she caused my evolution from a basic student with only raw talent to the very person I have become today. I can not thank her, since I know it will never be enough. Instead, I thank my destiny for the great opportunity of studying under her supervision. I will always be proud of this period of my life.

I would also like to thank my co-advisor Dr. Itır Önal Ertuğrul for her guidance, patience, all the contributions she made to this study.

It has a pleasure to work with my colleagues Abdullah, Erbil, and Ömer. The project meetings we had together has been very educational and horizon broadening.

I would also like to thank the other members of the Image Processing and Pattern Recognition Laboratory, Güneş, Hazal, Muhammed, Salih, Barış, Ozan, Nermin, for the cozy and cheerful working environment.

I appreciate the concern and caring of the jury members towards this study, Assist. Prof. Hande Alemdar and Assist. Prof. Hamdi Dibeklioğlu. Their critics reshaped and improved this thesis.

Lastly yet most importantly, I would like to express my deepest gratitude to my parents for supporting me for every decision I took.

The work was supported by TUBITAK (Scientific and Technological Research Council of Turkey) under the grant No: 116E091. We also thank Sharlene Newman, from Indiana University, for providing us the TOL dataset.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF ALGORITHMS xix

LIST OF ABBREVIATIONS xx

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation and Problem Definition 3

1.2 Proposed Methods 5

1.3 Contributions and Novelties 6

1.4 The Outline of the Thesis 7

2 BACKGROUND 9

2.1 Experimental Design of functional Magnetic Resonance Imaging (fMRI) Process 9

2.2 Nature of functional Magnetic Resonance Imaging (fMRI) Data . . . 10

2.3 fMRI Datasets Used to Study the Validity of the Proposed Computational Models . . . 11

2.3.1 Tower of London Experiment . . . 12

2.3.2 Human Connectome Project . . . 13

2.4 Feature Representation . . . 13

2.4.1 Multi-Voxel Pattern Analysis (MVPA) . . . 14

2.4.2 Brain Parcellation . . . 15

2.4.3 Graphical Methods for Brain Decoding . . . 16

2.4.3.1 Pearson Correlation . . . 18

2.4.3.2 Levinson-Durbin Recursion . . . 19

2.4.3.3 Closed Form Solution of Ridge Regression . . . 20

2.4.3.4 Granger Causality . . . 21

2.5 Classification Methods . . . 22

2.5.1 Multilayer Perceptrons (MLP) . . . 22

2.5.2 Convolutional Neural Networks . . . 23

2.5.3 Support Vector Machines . . . 25

3 MODELING SUBJECT- LEVEL BRAIN NETWORK WITH ARTIFICIAL NEURAL NETWORKS . . . 27

3.1 Modeling the Cognitive States of Brain as A Graph . . . 27

3.2 Representation of Brain Networks by the Relationships among the BOLD Responses . . . 29

3.3 Estimation of Edge Weights Using Artificial Neural Networks . . . 29

3.3.1 Directed Artificial Brain Networks (dABN) . . . 30

3.3.2 Undirected Artificial Brain Network (uABN) . . . 32

3.4 Baseline Methods 35
3.4.1 Pearson Correlation 35
3.4.2 Local Mesh Model with Temporal Measurements (LMM-TM) 35
3.5 Experimental Analysis of Artificial Brain Networks 36
3.5.1 Human Connectome Project (HCP) Experiment 36
3.5.2 Tower of London(TOL) Experiment 38
3.6 Chapter Discussion 40
3.7 Chapter Conclusion 41
4 MODELING COGNITIVE TASK- LEVEL BRAIN NETWORKS WITH ARTIFICIAL NEURAL NETWORKS 43
4.1 Estimation of Task-Specific Brain Networks 44
4.2 Representation of Task Specific Brain Networks 46
4.3 Estimation of Task Specific Brain Networks 47
4.4 A Novel Classification Approach Based on Task- Specific Brain Networks 49
4.5 Experimental Analysis of Task Specific Brain Networks 52
4.6 Decoding Performances of Task-Specific Brain Networks 52
4.7 Visualizations 54
4.8 Chapter Discussion 57
4.9 Chapter Conclusion 65
5 AN END TO END MODEL TO ESTIMATE AND DECODE THE ARTIFICIAL BRAIN NETWORKS 67
5.1 Elastic Loss Function 68
5.1.1 Experiments Conducted with Elastic Loss Function 70

5.1.1.1 Classification of Brain Networks using SVM 71
5.2 Multi-Layer Perceptrons for Classification of Brain Networks 74
5.2.1 Training of MLP 75
5.2.2 Classification of Brain Networks using MLP 76
5.3 Parallel Artificial Brain Networks 78
5.4 End to End Model 81
5.4.1 Loss Function of the End to End Model 83
5.4.2 Training the End to End Model 84
5.4.3 Experiments Conducted with the End to End Model 85
5.5 Chapter Discussion 86
5.6 Chapter Conclusion 87
6 CONCLUSION AND FUTURE WORK 89
6.1 Future Work 90
REFERENCES 93

LIST OF TABLES

TABLES

Table 3.1 Within-Subject Performances of Brain Networks on HCP Dataset. . 37

Table 3.2 Across-Subject Performances of Brain Networks on HCP Dataset. . 38

Table 3.3 Within-Session Performances of Mesh Networks on TOL Dataset. . 39

Table 4.1 Decoding Performance of Directed TSN on HCP Dataset. 53

Table 4.2 Decoding Performance of Undirected TSN on HCP Dataset. 54

Table 5.1 SVM classification accuracy on within subject experiments of directed brain networks created using elastic loss on HCP Dataset. 72

Table 5.2 SVM classification accuracy on within subject experiments of undirected brain networks created using elastic loss on HCP Dataset. 72

Table 5.3 SVM classification accuracy on across subject experiments of directed brain networks created using elastic loss on HCP Dataset. 73

Table 5.4 SVM classification accuracy on across subject experiments of undirected brain networks created using elastic loss on HCP Dataset. 73

Table 5.5 MLP classification accuracy on across subject experiments of directed brain networks created using elastic loss on HCP Dataset. 77

Table 5.6 MLP classification accuracy on across subject experiments of undirected brain networks created using elastic loss on HCP Dataset. 77

Table 5.7 End to end performance on HCP, directed, $\lambda = 4$ (Across).. 86

LIST OF FIGURES

FIGURES

Figure 2.1 During the scanning process of fMRI, a 3 dimensional brain volume is captured at each time period (Repetition time). A brain volume includes thousands of atomic units, voxels. A voxel holds the combined BOLD response of the corresponding brain area. Gathering BOLD responses from a voxel through all brain volumes, we form a discrete signal for a voxel. The figure is taken from Wager et. al. [1] 11

Figure 2.2 Sample directed and undirected brain networks formed using anatomic brain regions, R = { Amygdala, Cingulum, Occipital, Parietal }. Similar graphs can be formed for voxel level fMRI data by replacing the anatomic regions with the voxels of interest. 18

Figure 2.3 A perceptron. The incoming input to perceptron the vector $\mathbf{x} = \{x\}_{i=1}^{4}$. The perceptron assign a weight for each incoming connection. Its output the dot product of the weight vector and the input vector. An additional non-linear function, $\sigma(.)$ could be used to scale the output. . 23

Figure 2.4 A Multilayer Perceptron. The left-most nodes are input nodes, and rightmost nodes are the output nodes of the architecture. The middle layer is called a hidden layer. Form left to right output of a layer is fed into the next layer as the input. When used for classification, an output node o_j, returns the probability of the sample belonging to class j. 24

Figure 2.5 Support vectors and a hyperplane drawn by SVM in binary classification problem. The features of the samples, $\{x_1, x_2\}$, are in a two dimensional space. Support vectors pass through the spatially closest samples from opposite classes. 26

Figure 3.1 A sample figure of brain network extraction approach using non-overlapping time windows from fMRI signals of anatomic regions. Starting from the beginning of the signal, we declare time windows with length L until the end of the signal. For each time window, using the proposed methods, we estimate a directed and an undirected brain networks using the representative activations of the anatomic regions within the time interval. 31

Figure 3.2 Directed Artificial Brain Network Architecture. 32

Figure 3.3 Neural Network Structure to Create Undirected Artificial Brain Networks (connections with the same colors are shared). 33

Figure 4.1 Training methodology to estimate task-level brain networks. Assuming we have n cognitive tasks in a task fMRI dataset, we extract a brain network for each cognitive task. For estimation of each brain network, we train an artificial neural network with the architecture we defined in Section 3. The estimated brain networks represent common connectivity patterns shared across all the subjects in the underlying fMRI dataset. 45

Figure 4.2 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Emotion Processing'. 58

Figure 4.3 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Gambling'. 59

Figure 4.4 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Language'. 60

Figure 4.5 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Motor'. 61

Figure 4.6 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Relational Processing'. 62

Figure 4.7 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Social Cognition'. 63

Figure 4.8 Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Working Memory'. 64

Figure 5.1 A simple representation of the multilayer perceptron we employed in our experiments. The input layer has 8100 nodes and each of them takes an entry of the adjacency matrix. The middle layer, hidden layer, consists of 500 nodes. Each of 7 nodes of the output layer return the probability of a sample belonging a cognitive task, for example: o_7 returns the probability of the given input graph belonging to cognitive task y_7, $p(y_k = 7|W_k, V_1, V_2)$. We represent the weight matrix between the input layer and the hidden layer as V_1 and the weight matrix between the hidden layer and the output layer with V_2. 75

Figure 5.2 The proposed end to end architecture is a combination of PABN and MLP architectures. 82

LIST OF ALGORITHMS

ALGORITHMS

Algorithm 3.1 Algorithm to split the fMRI data into time windows. 29

Algorithm 3.2 Algorithm of creating subject level brain networks with the proposed method. In **ABN** process, the adjacency matrices are estimated. **Intialize_Weight_Matrix** process is used for random initialization of adjacency matrices 34

Algorithm 4.1 Algorithm to create Task Specific Brain Networks. Random samples of matrix elements are from uniform distribution (*U(0,1)*) 49

Algorithm 4.2 Algorithm to classify an unseen batch of fMRI data using TSNs. The algorithm search through all task specific brain networks in order to find the one that gives the minimum reconstruction loss. 51

Algorithm 5.1 Algorithm to prepare input tensor for a subject, X. 79

Algorithm 5.2 Process of Initializing the weight tensor for subject s, $\overline{W}$. Elements of random matrices are sampled from a normal distribution ($\mathcal{N}(\mu = 0, \sigma^2 = 1)$). 79

Algorithm 5.3 One epoch of training PABN model. 81

Algorithm 5.4 Initialize input tensor 85

LIST OF ABBREVIATIONS

AAL	Automated Anatomical Labeling
ANOVA	Analysis of Variance
BOLD	Blood Oxygenation Level Dependent
fMRI	functional Magnetic Resonance Imaging
HCP	Human Connectome Project
MVPA	Multi-voxel Pattern Analysis
SVM	Support Vector Machine
TOL	Tower of London
PCA	Principal Component Analysis
ICA	Independent Component Analysis
ABN	Artificial Brain Networks
dABN	Directed Artificial Brain Networks
uABN	Undirected Artificial Brain Networks
TSN	Task Specific Networks
dTSN	Directed Task Specific Networks
uTSN	Undirected Task Specific Networks
SVM	Support Vector Machines
MLP	Multilayer Perceptrons
PABN	Parallel Artificial Brain Networks

CHAPTER 1

INTRODUCTION

Human brain has been a mysterious black box to the humankind since the Homo Sapiens became self-aware. The main power that kept us dominant in the earth has been the one we know the least about. 'How do humans learn a language?', 'How perception of a scene triggers a categorization of an object in the brain?', 'Do all human brains have the same structure or are they all unique?' are some of the questions we keen to find answers to through-out the centuries.

Before the brain imaging techniques, the scientists had focused on the physical properties of the brain. The measures like the ratio of the brain volume to body volume, number of segregated brain lobes, the ratio of brain volume to brain area were shedding lights to determine how intelligent an individual is.

With the advent of brain imaging techniques like functional Magnetic Resonance Imaging (fMRI), Electroencephalography (EEG), and Positron Emission Tomography (PET), we can now observe the brain activities in a deeper and wider perspective. Using these technologies, one can study the blood flow, neurochemistry, dynamics and structure of human brain.

The atomic processing units of the human brain, neurons, need oxygen after they are activated and release chemical signals. As they do not reserve any type of biological energy, the activated neuron requires an immediate oxygen supply more than inactivated neurons. The blood flow in the brain is through these activated brain areas for the delivery of oxygen. This change of oxygen level in the brain cell is called **Hemodynamic Response** of the neuron and **Brain Oxygen Level Dependency (BOLD)** is a mapping of neural activations to a scale. The contrast of BOLD responses differ-

entiate between active and inactive brain areas.

Magnetic properties of an oxygenated body cell is found to be different than deoxygenated body cells. Based on this discovery, fMRI machines scan the human brain in three dimensions and measure the activations of whole brain. This 3-D image is called a **Brain Volume** and its atomic level cubic element is called a **voxel**. A voxel holds the combined BOLD response of the neurons in its range.

During a scanning of the brain of a human subject, fMRI machine captures 3 dimensional brain volumes with a relatively low time frequency compared to dynamics of the human brain. Depending on the spatial resolution of the machine and the interval between two adjacent voxels, the number of voxels in each of the brain volumes varies between 100.000 and 200.000. The temporal resolution of the machine defines the time interval between the capturing of two consecutive brain volumes and it is around 1 second.

fMRI scanning experiments can be designed to measure brain activities while experimentees perform various cognitive tasks, such as solving a puzzle, memorizing a word. This type of fMRI experiments are called **task-based fMRI (tfMRI)**. During a tfMRI scanning, the subjects can perform several cognitive tasks in sequence in a single session, or separate sessions could be reserved for each cognitive tasks. The aim in conducting tfMRI experiments is to examine the human brain under different cognitive states.

It has been assumed that the human brain could be divided into sections, each of which are evolved to specialize in performing a mental task: evaluating visual input, maintaining the physical balance, remembering a scent, etc. Physical pathways, activational responses, and many other neurological findings have been taken as path to determine these distinct brain regions. There have been also various types of mappings that determine which voxel in a brain volume resides in which of these brain regions. These mappings or atlases opened the door for researchers to study the human brain in a more abstract level.

The above-mentioned discoveries and developments have given us an ultimate goal, which is to computationally model the human brain. We believe simulating the dy-

namics of the brain in an artificial environment under our control would take us one step closer to open the magical black box. In this sense, many studies have been conducted to understand what cognitive task a scanned brain is performing, what abnormalities in the brain causes a disorder, or what is the underlying connectivity pattern during a cognitive process. The computational approaches to shed a light to this questions are known to be 'brain decoding' or 'mind reading' in the literature.

To put it more formally, brain decoding is an attempt to deduce the condition of the brain by understanding the underlying dynamics of the organ depending on the observation supplied by the brain imaging techniques. While the condition to be predicted could vary, the raw data used in methodologies in fMRI is the grid-like volumic structure containing the activational information for every time step during scanning. This data could be further processed to obtain time series of anatomical brain regions' representative activations.

In the rest of the study, we use the terms 'brain network', 'brain graph', 'mesh' in exchange to refer the term 'brain graph'.

1.1. Motivation and Problem Definition

We pursue a computational technique to represent the human brain as a collection of networks established among the anatomic regions. One of the power of this computational technique should be the capability to magnify the difference between brain volumes under different cognitive tasks. This property would make it easier for researchers to study transition of brain activation patterns under different cognitive conditions. A second power is interpretability of the outputs of the technique in terms of the relations among anatomic regions. The technique also provides us an interface that we can visualize and examine the brain networks to validate our preliminary knowledge and assumptions of the human brain.

An intuitive approach to represent the brain from fMRI images is to employ raw activity values of voxels or representative activity value of anatomical brain regions. However, employing voxel values as features in our computational model is expensive, since there could be up to 200.000 voxels in a single brain volume. Voxel BOLD

responses or anatomical regions' representative activations are not unique enough to discriminate cognitive tasks. Therefore, they cannot be informative of the underlying cognitive task of a brain volume.

A pioneering approach in the brain decoding field has been to model the brain as a graphical structure and it caused a paradigm shift in the literature. The basic idea is to take advantage of relative neural activations across the brain. This means that a voxel may not carry a valuable information, yet its functional relation with the other voxels in the brain volume could provide a more dynamic and more robust information. This relative relations has been assumed to show dependency, correlation or causality among distinct neural processing units. Many proposals have been made for the process of modeling graphs on raw fMRI signals.

A brain graph consists of three main components: vertices, edges, and edge weights. Every node in the graph represents an anatomical brain region. Existence of edge between any two nodes imply an assumed connection between the two anatomic region these the nodes represent. Weight of an edge shows the power of that connection.

There are two types of weighted graph structures: directed and undirected. We can adapt both of these graph types to model the human brain. In directed brain graphs, the edges have directions and there are at most two edges connecting any two nodes. In undirected brain graphs, the edges have no direction and there are at most one between any two nodes.

Brain graphs are constructed with the aim of accurately modeling the human brain under a cognitive condition. In order to evaluate a graph extraction method, we observe how much a brain graph formed during a cognitive task differs from the brain graphs formed during the other cognitive tasks. We employ classification methods for this purpose. Our classification methods predict the underlying cognitive task of a given brain graph. The accuracy of the classifier's performance is used as an evaluation metric for the graph extraction method. Therefore, in this thesis, we aim to develop graph extraction models that will output brain graphs with higher classification performance than the methods in the literature.

Brain decoding is a multidisciplinary field in which neuroscientific evaluations of

the data and the computational methods occupies a considerable ratio. However, the focus of computer science and neuroscience is very distant. Brain graphs serve as a bridge between these two worlds. While methodologies have been engineered for graph extraction as a focus of one discipline, the other can interpret the connectivity patterns of these graphs in neuroscientific point of view.

The studies concerning brain graphs have been on the track of extracting subject-level graphs. Yet, when it comes to create a generic graph for a cognitive task or a condition, statistical tools have been used as post-processing step. Yet, there has not been any method developed in the literature that produces a task-generic brain graph as an output. Being an open problem, we aim to take the first step in this direction. These characteristic networks would contain a population-level information regarding a task and increase interpretability of the networks.

The basic pipeline of brain decoding using graphs is extraction of the brain graphs as the first step and applying a classification algorithm on these brain graphs as the second step. Yet, there has not been a structure that combines these two steps in a single architecture.

1.2. Proposed Methods

In this thesis, we propose computational models to represent cognitive processes captured by functional magnetic resonance images (fMRI) for extracting brain graphs. We evaluate the validity of our brain graphs by decoding a set of cognitive states recorded in two fMRI data sets, namely HCP (Human Connectome Project) and TOL (Tower of London) data sets.

In our first computational model, we propose two novel neural network architectures, called Artificial Brain Networks (ABN) to estimate brain graphs for each subject and for each cognitive process. The first architecture, directed Artificial Brain Networks (dABN), is used to create directed brain graphs; on the other hand the second architecture, undirected Artificial Brain Networks (uABN), outputs an undirected brain graph of a subject. Both architectures generate subject-level directed and undirected brain networks.

The second model is an extension to ABN. However, instead of creating subject-level brain graphs, we aim to find a characteristic network that represents a cognitive task over the population of subjects. There are also two types of population level brain graphs: directed and undirected. We also propose a novel approach to classify the unseen brain volumes on the consideration of the task-level meshes.

The third model is a novel end-to-end artificial neural network architecture to both generate and classify brain graphs, simultaneously. Having options of directed and undirected graphs, we only show the case where the generated brain graphs are directed.

1.3. Contributions and Novelties

Our method offers both directed and undirected brain graphs by using similar artificial neural network architectures. Hence, we examine the effect of both graph types on brain decoding. We, also, show that both undirected and directed graphs estimated with our method provide more discriminative features to the classifiers, compared to the state of the art methods, such as methods based on Pearson Correlation. We generate directed and undirected brain graphs from a task-fMRI dataset (HCP) and complex problem solving dataset (Tower of London) and show that our classification performance is superior to state-of-the-art techniques in both within-subject classification and across-subject classification.

We studied the effect of different loss functions on uABN and dABN to classification performance. In detail, we show which parameter set on which loss function triggers the best performance. We also made experiments on different types of classifiers to get the best performance with our brain graphs.

To our knowledge, we are the first in the brain decoding literature to generate a generic brain network for a cognitive task. Previously, for the same purpose, basic statistical tools have been used to measure variance of edge weights. Using the generic brain networks we obtain, we define a distance metric to categorize unseen brain graphs.

With our last model, we introduce discriminative features to brain graphs using an

end-to-end model. This end-to-end model also means assembling the two components (graph building and classification) into a single process.

Some of the parts of this thesis has been published in the following papers.

- Kivilcim, B. B., Ertugrul, I. O., & Vural, F. T. Y. (2018). Modeling Brain Networks with Artificial Neural Networks. In Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities (pp. 43-53). Springer, Cham.
- Alchihabi, A., Kivilicim, B. B., Ekmekci, O., Newman, S. D., & Vural, F. T. Y. (2018, May). Decoding cognitive subtasks of complex problem solving using fMRI signals. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
- Alchihabi, A., Ekmekci, O., Kivilcim, B. B., Newman, S. D., & Vural, F. T. Y. (2018). On the Brain Networks of Complex Problem Solving. arXiv preprint arXiv:1810.05077.
- Alchihabi, A., Kivilicim, B. B., Newman, S. D., & Vural, F. T. Y. (2018, April). A dynamic network representation of fMRI for modeling and analyzing the problem solving task. In Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on (pp. 114-117). IEEE.

1.4. The Outline of the Thesis

Chapter 2 provides description of the methods in the literature to build brain graphs and brain decoding methods using fMRI data. We also refer to works which includes the relevant brain graphs in which cases they could be employed. Chapter 2 has also definitions of various classification algorithms that have been utilized to categorize brain graphs in the literature as a separate section.

In chapter 3, we describe how we compute directed and undirected brain graphs from fMRI time series. We share details of the neural network architectures used in the experiments. The chapter contains comparison of our method to the ones in the literature on within-subject and across-subject performances in datasets HCP and TOL.

We also share results of our experiments obtained by using different loss functions and different classification algorithms.

In chapter 4, we give details of the process of building population characteristic brain graphs. The chapter also includes the visual representations of these generic networks. The second section of this chapter provides the distant metric we used to compare two brain graphs and categorize an unseen brain volume. We supply the classification performance we obtained with this metric.

The experiments we describe in chapter 5 are on across-subject classification on HCP dataset. We, first, observe the change of classification performance on HCP brain networks across the epochs using a multi-layer perceptron classifier right after an dABN. We propose our novel end-to-end artificial neural network architecture to classify and create brain networks simultaneously. We compare the performances of our novel architectures to the classical methods available in the literature.

CHAPTER 2

BACKGROUND

From testing the effects of neurochemical drugs to observation of activation patterns of cognitive states, studies to understand the human brain could be on various objectives. However, in this thesis, we focus on only classification of cognitive states in functional Magnetic Resonance Imaging (fMRI) data. We model the problem as a machine learning problem. Therefore, we need to define the basic concepts of machine learning, such as a sample, the label of a sample and what we aim the algorithm to learn and predict.

In order to bring familiarity to the reader, in the beginning of this chapter, we aim to describe scanning procedures of fMRI, the characteristics of the data fMRI supplies, and the fMRI data sets that have been used during our experiments in this thesis. In Section 2.4 and Section 2.5, we elaborate how we model brain decoding process as a machine learning problem.

2.1. Experimental Design of functional Magnetic Resonance Imaging (fMRI) Process

Based on the design of fMRI scanning procedure, we categorize the fMRI experiments into two; resting state fMRI (rs-fMRI) and task fMRI (tfMRI). In resting state fMRI experiments, the aim is to capture the default activation patterns of the human brain. As its name suggests, during resting state fMRI experiments, the subject of interest is required not to perform any explicit mental task. Based on the data obtained from a population of subjects, effect of single variable's change among the population can be studied. This variable could be existence of brain disorder (schizophrenia,

dyslexia etc.) and the difference of brain activation patterns of a healthy subject and a subject suffering from the disorder could be examined [2, 3, 4]. Using (rs-fMRI), the effect of speaking a second language to functional connectivity of the brain is studied in [5], differences between brain networks of female and male subjects is examined [6].

Task-fMRI experiments (tfMRI) are conducted in order to study the effect of cognitive process on the brain activation. During the scanning duration of a tfMRI experiment, the subjects are required to accomplish certain cognitive tasks. The tfMRI experiment could be on audio-visual experiment [7], solving a puzzle [8] or performing a linguistic task [9]. The brain volumes of tfMRI experiments correspond to cognitive task and in brain decoding, one of the aims is to categorize the brain volumes to their underlying cognitive tasks. In this thesis, our objective is to study the brain volumes from various cognitive tasks of two fMRI datasets, and classify them to cognitive tasks using our proposed computational models. We explain the cognitive tasks and the data type of these fMRI datasets in Section 2.3.

2.2. Nature of functional Magnetic Resonance Imaging (fMRI) Data

An fMRI dataset consists of brain volumes taken through a limited time. The time period between any two consecutive brain volume is Repetition Time (TR). TR depends on the characteristic of the fMRI machine and is generally around 1 second. A brain volume surrounds the brain, however there could be empty spaces in the volume due to non-cubic geometry of the human brain.

A brain volume consist of atomic cubic elements, voxels. Depending on the spatial resolution of the machine, there are approximately between 20,000 and 200,000 voxels in a single brain volume. A voxel corresponds to a tiny area of the scanned brain, and its value show the hemodynamic response of that brain area. Apart from representing the fMRI dataset as series of 3 dimensional images using brain volumes, we can also represent the dataset using the time series of individual voxels. Time series of a voxel could be obtained by aligning activations of the voxel through time. Time series representation is useful in studying the responses of the brain to the stimulus

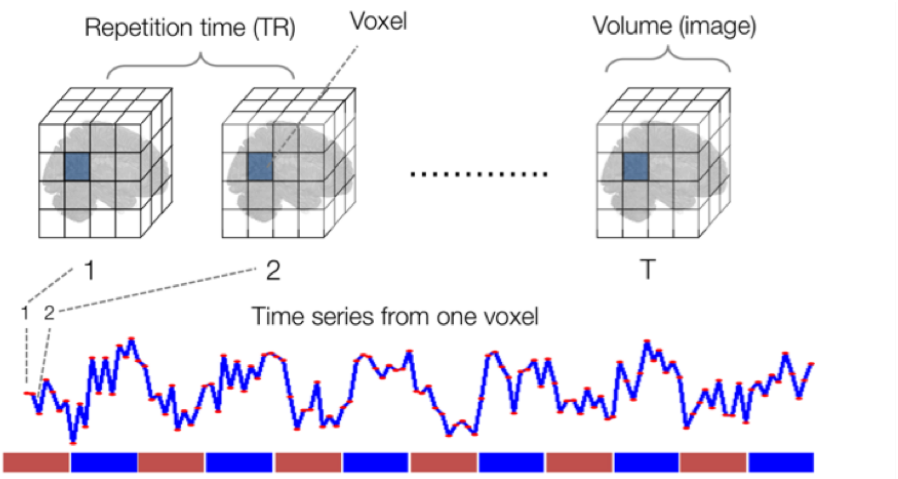


Figure 2.1: During the scanning process of fMRI, a 3 dimensional brain volume is captured at each time period (Repetition time). A brain volume includes thousands of atomic units, voxels. A voxel holds the combined BOLD response of the corresponding brain area. Gathering BOLD responses from a voxel through all brain volumes, we form a discrete signal for a voxel. The figure is taken from Wager et. al. [1]

given during the fMRI experiment.

In the next section, Section 2.3, we describe the characteristics of the fMRI datasets we have used in our brain decoding experiments.

2.3. fMRI Datasets Used to Study the Validity of the Proposed Computational Models

Collecting fMRI data is costly and difficult, due to the restrictions of the fMRI data acquisition equipment. As a result, the size of the data sets is small, compared to the data sets used in the other field of computer science, such as computer vision, bioinformatics, big data analysis etc. These data sets are held private to the experiment conductors or shared online for public use [10].

In our experiments, we use two task fMRI data sets. The first data set, Tower of London, is a private data set supplied to us by Prof. Sharlene Newman, Indiana University, Department of Brain Sciences. The second data set, Human Connectome

Project task-fMRI data set, is a widely studied public fMRI data set.

2.3.1 Tower of London Experiment

Tower of London (TOL) is a puzzle consisting of 3 bins with varying capacities and 3 balls with different colors [11]. Aim in the game is to convert a pre-determined initial configuration of balls to pre-determined final state by moving a ball at a time with the least number of movements.

In the literature of brain decoding, TOL puzzles have been used in order to study complex problem solving as the puzzle solving process requires several cognitive tasks to be performed such as memory, strategy formation [8, 12, 13].

The fMRI experiment was conducted in Indiana University with 18 participants aged between 19 and 38. The subjects played the computerized version of TOL during the scanning. Each subject attended four fMRI sessions and in each of these sessions they solved 18 different TOL puzzles.

There are two sub-phases of TOL puzzle solving process: planning and execution. The time between the presentation of the game on the computer screen and the first click of the subject is considered as the planning phase. Having varying duration limited to five seconds at most, planning phase is the reserved time for the subject for him/her to develop a strategy. The time between the occurrence of first click and the last movement of the subject is considered as the execution phase. Subjects aim to solve the puzzle by rearranging the positions of the balls transforming the initial configuration to the final configuration. The time limit given to the subject for a single puzzle is 15 seconds. If the subject can not solve the puzzle within the given time, the puzzle is considered to be unsolved.

From the beginning of the fMRI session till the start of the first puzzle, there is a fixation period of 32 seconds. Between the end of the execution phase of a puzzle and the beginning of the next puzzle, a resting period is reserved, in which the subjects focus on a $+$ sign appearing on the screen. At the end of eighteenth puzzle of a session, the subject rests until the session takes 590 seconds.

Sample rate of the fMRI machine used in the experiments is $TR = 1$ second meaning that the machine creates a brain volume at each second. Each brain volume in the created dataset has 185,405 voxels. In summary, the dataset consist of brain images of 18 subjects recorded in 4 sessions during 590 seconds, in which they solve 18 different puzzles.

2.3.2 Human Connectome Project

Human Connectome Project aims to investigate functional and structural connectivity patterns of healthy human brains by collecting tfMRI data from vast amount of subjects. As of now, the collected dataset includes 1200 subjects.

In our experiments, we have used an earlier release of the HCP dataset consisting of 807 subjects. Each subject attended 7 sessions of fMRI scanning and in each of they perform a different cognitive task, namely : Emotion Processing [14], Gambling [15], Language [16], Motor [17], Relational Processing [18], Social Cognition [19], and Working Memory. The cognitive tasks covers 176, 253, 316, 284, 232, 274, 405 brain volumes of each subjects in the same order.

2.4. Feature Representation

Intuitive approach to represent the human brain using fMRI technology is to employ raw brain volumes. However, data could be problematic due to a few reasons and a more meaningful representation could be obtained by applying feature representation algorithms. First, dimensionality of the data is too high for any computational model. The number of voxels determines the number of features of a sample. Since, this number could be very high, the memory consumption and running time could be immense. Secondly, all voxels of the brain do not get activated during a cognitive process. That means, there are redundant features in the raw data. Third, one can aim to study activation patterns of the brain in a more abstract level. Lastly, instead of only voxel activations, correlations and dependencies among broader brain areas may be crucial.

In brain decoding literature, there have been various attempts to represent a brain volume with the aim of overcoming the above-mentioned problems. In this section, we describe three main groups of methods that have been used for feature extraction. The first one, Multi-Voxel Pattern analysis, is a feature elimination method designed with purpose of identifying the 'important' voxels for brain decoding analysis. The methods in the second group, Brain Parcellation, groups the voxels in a brain volume. The last method group, Brain Graphs, represent the brain as a graph to denote functional connectivities among brain units.

2.4.1 Multi-Voxel Pattern Analysis (MVPA)

A voxel captures the activations of a tiny brain area and hence represents the activation of very small ratio of all neurons in the brain. Often times, activation of a single unit (a voxel in this case) does not carry sufficient information regarding the underlying cognitive process.

To have a broader view of the underlying cognitive process, one can consider the whole brain image, i.e. all voxels and aim to capture a multivariate pattern. Yet, the fMRI images are known to be very noisy [20] and one would also take account of the noisy voxel values in this scenario. Also, current image techniques are capable of supplying high resolution brain images and therefore, a great number of voxel units. The high number of the features leads to a difficulty in pattern analysis and could cause curse of dimensionality problems.

The aim in MVPA methods is to find a balance between information sufficiency and feature quantity. The way to achieve a sparser representation of a brain volume, a subset of voxels should be discarded and the activation of remaining voxels should be assumed as the representation of the brain volume. Different methods to eliminate voxels have been suggested in the literature [21, 22, 23]. Below, we acknowledge the reader of the two most commonly used procedures.

In the first method, Voxel Selection, the aim is to achieve the maximum separation among the samples from different classes. The approach is to measure the contribution of each voxel to class prediction performance. Various methods have been

proposed in order to measure the discrimination power of a feature on classification and regression [24, 25, 26, 27, 28]. After, estimating the importance of each voxel for learning algorithms, a 'custom' thresholding is applied to select only the top most discriminative voxels. Afrasiyabi et al. analysed the effect of voxel selection on the classification performance in a visual object recognition experiment [21].

In the work of Kriegeskorte et al. [29], the voxels that are informative of the neural conditions are detected by applying 'Multivariate Effect Statistics' to a spherical volume of voxels. This multivariate analysis estimates the information each voxel carries by calculating how much its activation contributes to a linear predictor of the condition. After evaluation, the same steps are applied to another set of voxels which resides in another sphere in the brain. The process is completed after considering each voxel in a brain image as the center of a sphere and contribution of each voxel to the discrimination of cognitive processes evaluated.

Principal Component Analysis (PCA) [30] and Independent Component Analysis (ICA) methods have also been utilised to decrease the number of features in an fMRI dataset. These methods take the voxel intensities as variables and apply a linear transform on them to represent these variables in a new feature space. PCA method, maximizes the variance of the samples, and removing the components with low variance corresponds to reduction of the data dimension [31]. ICA, on the other hand, finds statistically independent components each of which contains distinct information.

The second approach, Region Selection or Region of Interest, could rely on both statistics and neuroscientific findings. In [23], Haxby et al. focused on only the voxels from ventral temporal cortex to classify objects in visual memory experiment. In [8], observation from 8 different ROI which is relevant to strategy-development in the brain have been used to study complex problem solving. Yet, a similar approach to voxel selection could be employed to retain the voxels from discriminative regions.

2.4.2 Brain Parcellation

Neurons are atomic level information processors of the brain and the human brain consists of billions of them. A voxel in a fMRI sample represents the activations of a

group of voxels in its regarding coordination in the brain. Varying with the resolution of the capturing machine, the number of neurons resides in the cubic of the voxel could be up to millions.

Spatially close neurons are known to be in co-operation during mental processes. And taking this observation as a starting point, one would expect several sections in the brain would organize to process cognitive tasks. Similarly, we could assume that, spatially close voxels would correlate functionally as they represent activations of neighboring neurons and a group of spatially adjacent voxels may represent a cluster of organized neurons. From this point on, we will use the term 'brain region' or 'region' to refer to these neuron clusters.

There have been various ways to define brain regions existing in human brain both neuroscientifically and statistically [32]. While in neuroscientific studies certain mappings from voxels to limited number of brain regions are proposed, statistical studies could include various number of regions and dynamic region borders over voxels.

Subject-level parcellation methods find unique segmentation for each considered subject independently. Clustering and segmentation methods have been applied to extract subject-level parcels [33, 34]. [35]. Mogultay et al. [36] proposed to obtain the segments by partitioning the subject's functional connectivity graph into variable number of sub-graphs.

Group-level parcellation methods targets a data-base estimation of a mapping that could be applicable to a wider range of individuals. The estimated template is assumed to depict a common parcellation for a population. A group-level segmentation could be obtained by aggregation of subject-level segmentation [37]. There have also been pre-computed brain segmentation schemes which supplies a generalized template for any individual [38].

2.4.3 Graphical Methods for Brain Decoding

A graph is a structure that visualize objects (nodes, vertices) and connectivities among them (edges) and in some cases the power of these connections (edge weights, arc weights).

In computational neuroscience, a *brain graph* (*brain network*) represents functional or structural connectivity of the brain's neural units. The nodes of the graphs could represent either a subset of brain regions or voxels. An edge implies a connection between two of these units and edge weight denotes the power of connectivity between them.

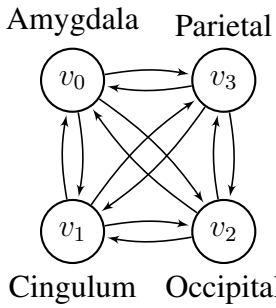
Mathematically, we illustrate a weighted brain graph of anatomic regions as $G = (V, W)$, where $V = \{v_1, v_2, v_3, \ldots, v_M\}$ denotes the vertices of the graph and the vertices corresponds to M anatomic brain regions, $R = \{r_1, r_2, r_3, \ldots, r_M\}$. $W \in \mathbb{R}$ is adjacency matrix of the graph. An entry at coordinate (i, j), $w_{i,j} \in W$, is weight of the edge from vertex i to vertex j.

Based on the existence of a direction of its edges, a weighted brain graph could be of two types: directed and undirected. A directed brain graph is a simple directed graph, meaning that there are at most 2 symmetric edges connecting two nodes. An undirected brain graph, on the other hand, is a simple graph where there are at most 1 undirected edge between any two nodes of the graph. Sample brain graphs of each structures are visualized in Figure 2.2.

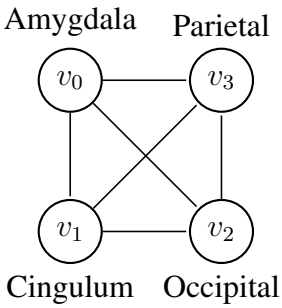
Brain graphs or brain networks have been a tool to understand the dynamics of underlying brain connections of human brains with various disorders. They have been an informative feature representation for brain decoding. [39] have examined the brain networks to study and verify the assumption of missing connections in the brains of schizophrenic patients. In [40], the brain networks of patients with dyslexia have been examined in order to observe difference of brain dynamics in healthy and dyslexic patients.

It has been shown that the brain networks contain more informative and discriminative information compared to raw fMRI data to classify cognitive tasks [41]. Hence, researches have exploit the graph structure of the brain to predict the cognitive tasks [7], gender of the subject [6, 42] or even to categorize subjects with autism [43]

In the rest of this subsection, several approaches in the literature to model the human brain as a graph and methods to estimate adjacency matrices of these graphs, W, are explained. We define the brain network extraction methods in order to denote



(a) A Directed Brain Network.



(b) An Undirected Brain Network.

Figure 2.2: Sample directed and undirected brain networks formed using anatomic brain regions, R = { Amygdala, Cingulum, Occipital, Parietal }. Similar graphs can be formed for voxel level fMRI data by replacing the anatomic regions with the voxels of interest.

connectivity among anatomic brain regions. However, the brain networks could be defined by simply replacing the anatomic regions with voxels.

2.4.3.1 Pearson Correlation

Pearson correlation coefficient (PCC) measures the linear correlation between two variables. The value of PCC varies between -1 and +1, where the value +1 implies a total positive linear association, the value 0 indicates no correlation, and the value -1 means a total negative association.

In their work, Richiardi et. al. defined the functional connectivity [44] between two anatomic region as Pearson correlation coefficient of the representative activations of these anatomic regions. Therefore, the edge weight from node i to node j is calculated as:

$$w_{i,j} = w_{j,i} = \rho_{\overline{\mathbf{b}}_{\mathbf{i,t}}, \overline{\mathbf{b}}_{\mathbf{j,t}}} = \frac{cov(\overline{\mathbf{b}}_{\mathbf{i,t}}, \overline{\mathbf{b}}_{\mathbf{j,t}})}{\sigma_{\overline{\mathbf{b}}_{\mathbf{i,t}}} \sigma_{\overline{\mathbf{b}}_{\mathbf{j,t}}}}, \tag{21}$$

where $\overline{\mathbf{b}}_{\mathbf{i,t}} = [b_{i,t}, b_{i,(t+1)}, ..., b_{i,(t+L)}]$ represents the average time series of BOLD activations of region i between time t and $t + L$, $cov()$ defines the covariance, and σ_s

represents the standard deviation of time series s.

Pearson correlation is symmetric, meaning that the correlation of $\overline{\mathbf{b}}_{\mathbf{i,t}}$ with $\overline{\mathbf{b}}_{\mathbf{j,t}}$ is equal to the correlation of $\overline{\mathbf{b}}_{\mathbf{j,t}}$ with $\overline{\mathbf{b}}_{\mathbf{i,t}}$, $\rho_{\overline{\mathbf{b}}_{\mathbf{i,t}},\overline{\mathbf{b}}_{\mathbf{j,t}}} = \rho_{\overline{\mathbf{b}}_{\mathbf{j,t}},\overline{\mathbf{b}}_{\mathbf{i,t}}}$. Therefore, edges between nodes i and j are also symmetric in the formed graph, $w_{i,j} = w_{j,i}$. Hence, this method estimates an undirected brain network.

With this method, the brain networks are estimated for a time interval, therefore temporal information of the fMRI data is considered. However, finding accurate Pearson correlation coefficients require a long time interval and one of the limitations of the fMRI data is having small number of brain volumes. Also, this method does not concern the locality property of the human brain since the connectivity is estimated by pair-wise calculations.

2.4.3.2 Levinson-Durbin Recursion

The brain graph construction methods that employ Levinson-Durbin recursion [45], represent a linear relationship among the representative activations of anatomic regions. In other words, we estimate representative activation of each anatomic region based on the remaining ones. For region i, we estimate the representative activation at time t, $b_{i,t}$, using :

$$b_{i,t} = \sum_{j=1, j\neq i} w_{i,j} b_{j,t} + \epsilon_{i,t}, \tag{22}$$

where $b_{j,t}$ is the representative activation of anatomic region j, $w_{i,j}$ is the weight of the connectivity from anatomic region i to j, and $\epsilon_{i,t}$ is error of estimation at time t.

Establishing the estimation formula, Equation (22), for each M anatomic brain region, we end up having M linear equations. When solving these equations, the aim is to find the set of weights, $w_{i,j}$, which minimize the error rate, $\epsilon_{i,t}$. In the literature, several algorithms using Levinson-Durbin recursion have been proposed in order to solve the equations to obtain the weights [46, 47, 48].

Using a Levinson-Durbin recursion algorithm, linear regression equations for each

anatomic brain region is solved separately in order to obtain adjacency matrix $W = \{w_{i,j}\}$ at each brain sample. As a result, the obtained brain graph is a directed graph.

Previously, this approach has been adopted to estimate brain networks of a recognition memory experiment for classification of semantic words [7, 41].

This approach takes locality property of the brain into account by avoiding pair-wise comparisons. However, the method is not capable of estimating a brain graph for a time interval. Therefore, temporal information is not considered.

2.4.3.3 Closed Form Solution of Ridge Regression

Ridge Regression is a form of loss function which measures the difference between the estimation of a variable and the true value of that variable. The function also utilises a regularization on the weights that will be estimated to penalize large weights.

In this approach, similar to Levinson-Durbin approach, we assume that we can estimate the activation of an anatomical region by applying a linear regression on the representative activations of the remaining anatomic regions. In addition to this assumption, we also construct this linear relation within a time interval as in Pearson correlation method. Formally, we reconstruct the activation of anatomic region i during the time interval $[t, t + L]$ as:

$$b_{i,t'} = \sum_{j=1, j\neq i} w_{i,j} b_{j,t'} + \epsilon_{i,t'} \qquad \forall t' \in [t, t + L] \tag{23}$$

where $\epsilon_{i,t'}$ is the error of estimation at time t'. The aim is to minimize the expectation of error rate through the time. For this purpose, a loss function is defined to measure total error of estimation :

$$Loss_{ridge}(\overline{b}_{i,t}; w) = E((b_{i,t'} - \sum_{j\neq i, j=1}^{M} w_{i,j} b_{j,t'})^2) + \lambda \mathbf{w}_i^T \mathbf{w}_i \tag{24}$$

where $\overline{\mathbf{b}}_{\mathbf{i},\mathbf{t}} = \{b_{i,t'}\}_{t' \in [t,t+L]} \in \mathbb{R}^L$ is the vector consisting of representative activation of anatomic region i, $E(.)$ is expectation operator taken through time. The last term

in the equation is the regularization included in order to increase the generalization capability of the regression and λ is the regularization hyperparameter and $\mathbf{w}_i = \{w_{i,j}\}_{j=1,j\neq i}^{M} \in \mathbb{R}^{M-1}$ is the weights of the regression. The weights, $\mathbf{w}_i$, that minimize the defined loss function is calculated with the following closed form solution :

$$\overline{\mathbf{w}}_\mathbf{i} = (\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})^{-1}\mathbf{B}^T\overline{\mathbf{b}}_{\mathbf{j},\mathbf{t}}, \tag{25}$$

where the matrix $\mathbf{B} \in \mathbb{R}^{L\times(M-1)}$

where $\mathbf{B} \in \mathbb{R}^{L\times(M-1)}$ is a matrix, whose columns consist of the average BOLD activations of anatomic regions except for the region i in the time interval $[t, t+L]$ such that column j of matrix $\mathbf{B}$ is $\overline{\mathbf{b}}_{\mathbf{j},\mathbf{t}}$, $B_{:,j} = \overline{\mathbf{b}}_{\mathbf{i},\mathbf{t}}$. $\lambda \in \mathbb{R}$ represents the regularization hyperparameter. $I \in \mathbb{R}^{(M-1)\times(M-1)}$ in the formula is an identity matrix.

This method has been used in order to estimate brain graphs on an fMRI dataset of an object recognition experiment and on HCP dataset [49, 50, 51].

The brain graphs created using the closed form solution of ridge regression are directed graphs. This approach includes both temporal information of the fMRI data and the locality property of the human brain. However, the solution, Equation (25), involves pseudo inverse of relative a big matrix which could cause an inaccurate estimation.

2.4.3.4 Granger Causality

Granger causality model was first introduced in economics [52]. Being an effective method for time series, the method can be used to estimate the effect of a signal upon another signal. One of the assumptions of the model is that the signals to be modeled have the causality property, meaning that the current value of signal, $x[t]$ depends on the past values of the signal.

Granger causality method, estimates how much the current value of the signal, x, depends on the past values of another signal, y, and its own past values. Hence, these two signals, x and y are modeled as:

$$x[n] = \sum_{m=1}^{M} a_x[m]x[n-m] + \sum_{m=1}^{M} b_y[m]y[n-m] + \epsilon_x[n]$$

$$x[n] = \sum_{m=1}^{M} b_y[m]y[n-m] + \sum_{m=1}^{M} a_x[m]x[n-m] + \epsilon_y[n]$$

where a_x and b_y are coefficients of how much the past values improve the prediction of the current of value of the signal x and y respectively; a_y, b_y represent the effect of one's past values to the other's current value; $\epsilon_x[n], \epsilon_y[n]$ are the error margins for the time $t = n$.

Geweke et. al. proposed a solution to measure the dependence of two auto-regressive models onto each other [53]. This method has been utilised to study the influence of brain regions to each other in visual cortex of various animals [54, 55] and to create effective brain graphs in resting state fMRI [56].

Calculating pairwise Granger causality of anatomic brain regions, a directed brain graph could be obtained for a time interval. This approach would consider the temporal information of the data, however pairwise estimations prevents the use of brain's locality property.

2.5. Classification Methods

In this section, we describe the classification methods used in the literature to categorize the brain graphs with respect to cognitive states. While most of these methods are for general use in machine learning, some of the architectures are specifically designed to classify brain graphs.

2.5.1 Multilayer Perceptrons (MLP)

Multilayer perceptrons (MLP) is a type of neural network architecture that has been used for supervised classification. As the name suggests the architecture consists of layers of perceptron stacked on top of each other.

A perceptron is the basic, atomic processing unit of an MLP structure, shown in 2.3. Inspired by the actual neural cells, a perceptron weights every incoming input, propagates the summation of these weighted inputs. $o = \sum_i^N w_i x_i$. An activation

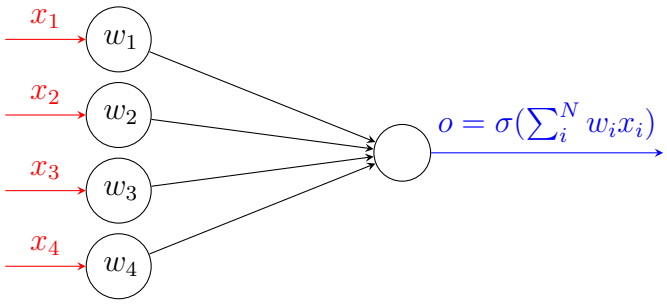


Figure 2.3: A perceptron. The incoming input to perceptron the vector $\mathbf{x} = \{x\}_{i=1}^{4}$. The perceptron assign a weight for each incoming connection. Its output the dot product of the weight vector and the input vector. An additional non-linear function, $\sigma(.)$ could be used to scale the output.

function can be applied for non-linearity after the summation.

An MLP could be devised in very different configurations since number of layers, number of nodes in each layer, activation function of each layer are design parameters. The first level in the architecture is named as input layer and the very last one is called output layer. Hidden layers are the ones between them and if there are more than one of them, the model considered a deep one, otherwise it is a shallow architecture.

There are two flows in an MLP: forward propagation and backward propagation. If the flow is from bottom to top, it is called forward propagation and networks make a prediction in this flow. If it is top to bottom, the network adjust the weights in a way to correct its mistakes in the prediction phase. This optimization phase is completed by employing back-propagating algorithm for a number of epochs [57].

MLPs have been employed to perform classification in big fMRI datasets. A deep neural network has been used classify cognitive tasks across all subjects in Human Connectome Project dataset in [58]. Plis et. al. employed an MLP in order to predict schizophrenia in a given patient's functional brain images [59].

2.5.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are special type of artificial neural networks that are generally specified to solve image related problems: such as image classifi-

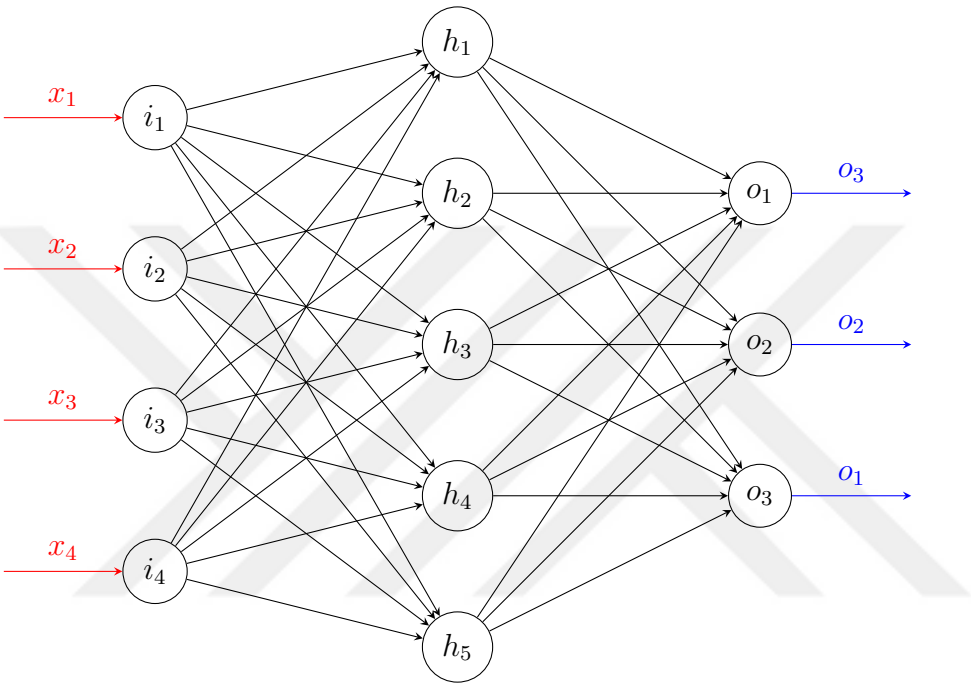


Figure 2.4: A Multilayer Perceptron. The left-most nodes are input nodes, and right-most nodes are the output nodes of the architecture. The middle layer is called a hidden layer. Form left to right output of a layer is fed into the next layer as the input. When used for classification, an output node o_j, returns the probability of the sample belonging to class j.

cation, object detection. CNNs aim to replace hand-crafted filters that have been used in the older image processing algorithms.

At a convolution layer, an incoming image is further processed by convolution with the filters in that layer and this processed image is propagated to an upper layer. In a vanilla CNN architecture classification or regression task is performed by the fully connected layers at the end of the architecture.

Requiring very specific input structures, CNNs can only work on grid-like inputs. This could be a 1-D time signal, 2-D image, 3-D brain volume.

Variations of CNNs have been applied in brain decoding. Kawahara et. al. proposed using 1 dimensional CNNs on rows and columns of adjacency matrices of brain networks to estimate neurodevelopment of premature infants [60]. In [61], Spectral CNNs are employed to predict Alzheimer's disease.

2.5.3 Support Vector Machines

Support Vector Machines are one of the most popular methods in Machine Learning literature which could be used for both classification and regression problems. It is also a memory efficient and a robust method that can be used in high dimensional feature spaces.

At the core of SVMs lies the support vectors. In a binary classification problem, support vectors are the coordinates of two spatially closest samples from opposite classes. These support vectors are considered when drawing the decision boundary.

After the support vectors are selected among the samples, decision function draws decision boundary in the space. Decision function is a design issue and could be selected as a linear or a non-linear function. Linear decision functions result on hyperplane in the space that is equidistant from support vectors. The distance between the hyperplane to a support vector is called the margin and it is the main objective of the algorithm to maximize this distance.

The SVM method could be generalized for multi-class problem by drawing 'one versus the rest' decision boundaries for each of the classes. That means we would end

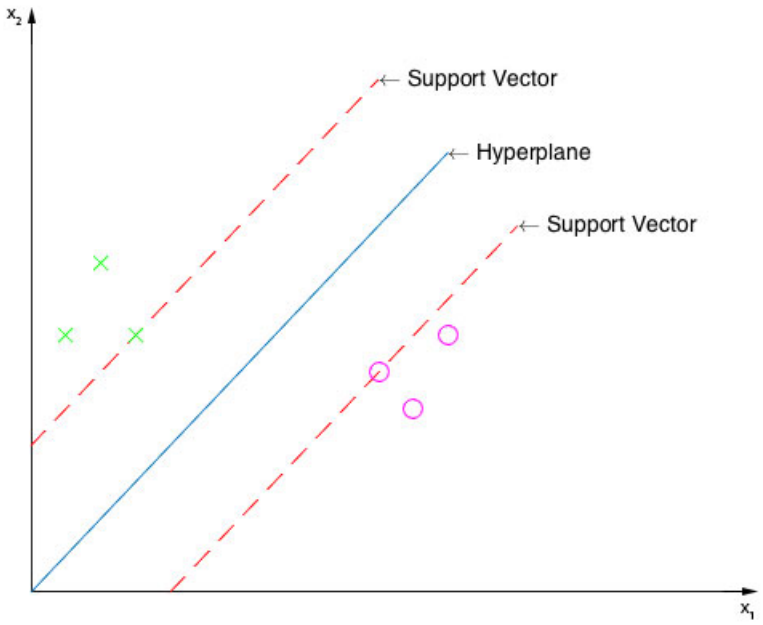


Figure 2.5: Support vectors and a hyperplane drawn by SVM in binary classification problem. The features of the samples, $\{x_1, x_2\}$, are in a two dimensional space. Support vectors pass through the spatially closest samples from opposite classes.

up having $n * (n-1)/2$ decision boundaries in multi-class example.

CHAPTER 3

MODELING SUBJECT- LEVEL BRAIN NETWORK WITH ARTIFICIAL NEURAL NETWORKS

In this chapter, we propose a novel method to model each cognitive state, when a subject is exposed to a pre-defined stimulus. The proposed model generates subject-level directed and undirected brain networks using artificial neural networks. A representative activation time series are extracted from the fMRI data recorded during a subject performs a cognitive task. These representative time series are then employed to estimate the connectivity of anatomical regions.

We generate brain networks using fMRI data from two datasets: HCP (Human Connectome Project) and TOL(Tower of London). We train an Support Vector Machine on these brain graphs to predict the underlying cognitive process. We show that our method achieves the state of the art classification performance in both of the fMRI datasets.

3.1. Modeling the Cognitive States of Brain as A Graph

In this section, we explain how we estimate the edge weights of directed and undirected brain networks using artificial neural networks. Throughout this chapter, we represent a brain network by $G = (V, W)$, where $V = \{v_1, v_2, v_3, \ldots, v_M\}$ denotes the vertices of the network, which represent $M = 90$ anatomical brain regions. Each node has an attribute which consists of average voxel time series which resides in the anatomical region.

$W = (w_{i,j})$ is the adjacency matrix of the graph G. $w_{i,j}$ denotes the weight of the

edge from v_i to v_j. If G is undirected, the adjacency matrix is symmetrical $W = W^T$ and therefore $w_{i,j} = w_{j,i}$. Sample representations of directed and undirected brain networks are shown in Figure 2.2 in Chapter 2.

The anatomical brain regions we use are defined by Anatomical Atlas Labeling (AAL) [38]. We discard the regions residing in Cerebellum and Vermis, since they do not contribute to cognitive processes we are interested in. We use the remaining $M = 90$ anatomic regions to denote the vertices of the graph and later in the next section.

As mentioned in Chapter 2, each anatomic region is represented by a time series obtained by averaging all the voxel time series in in that region, defined over a time interval.

Let us now, describe how the representative activations of these anatomic regions are used to estimate adjacency matrix of the graph.

We start by partitioning the time series of the fMRI signal recorded at each voxel into non-overlapping time windows with length L. Therefore, the kth time window, win_k denotes the time interval $[k * (L - 1), k * L]$ of the data. Overall, we obtain a list of time windows $win_{all} = \{win_1, win_2 \dots win_n\}$ from the entire fMRI time series, where $n = Z/L$ denotes the number of time windows and Z is the length of the fMRI signal. Using representative activations of anatomical regions within each time window, we estimate a different brain graph. Therefore, we create a list of brain networks $G_{all} = \{G_1, G_2 \dots G_n\}$. A graph G_k will represent the spatio-temporal relationship among anatomic regions during time window win_k. Assuming that, a single cognitive process takes place during a time window win_k, we represent the underlying cognitive process as y_k, and for the whole signal we collect $y_{all} = \{y_1, y_2 \dots y_n\}$. We define the process of partitioning the time series of an fMRI signal

in Algorithm 3.1.

Algorithm 3.1: Algorithm to split the fMRI data into time windows.

Process `Extract_time_windows(`$data$`)`**:**

```
time_windows = { } ;
for t=0; t + L < data.length ; t ← t+L do
    time_windows.add( [t, t + L] ) ;
end
return time_windows ;
```

3.2. Representation of Brain Networks by the Relationships among the BOLD Responses

For a given time window win_k, weights of incoming edges to vertex v_i is defined by an M dimensional vector, $\overline{\mathbf{w}}_\mathbf{i} = [w_{i,1}, w_{i,2} \dots w_{i,M}]$. Note that the ith entry $w_{i,i} = 0$, which implies that a node does not have an edge value into itself. These edge weights can be represented as a linear combination of activation, $b_{i,t}$, of region i at time t to the activations of the remaining regions, $b_{j,t}$:

$$b_{i,t'} = \sum_{j \neq i, j=1}^{M} w_{i,j} b_{j,t'} + \epsilon_{i,t'} = \hat{b}_{i,t'} + \epsilon_{i,t'} \qquad \forall t' \in win_k \tag{31}$$

where $\hat{b}_{i,t'}$ is the estimated value of $b_{i,t'}$ at time t' with error rate $\epsilon_{i,t'}$, which is the difference between the real and estimated activation.

3.3. Estimation of Edge Weights Using Artificial Neural Networks

We design an artificial neural network model with two layers to estimate the edge weights of brain networks. This artificial neural network will estimate the representative activations of each anatomic region, $b_{i,t}$, using the representative activations of the remaining anatomic regions, $b_{j,t}$. Therefore, there are $M = 90$ input takes taking regional representative activations and there are $M = 90$ output nodes supplying estimation of these activation in the artificial neural network.

The weights of the artificial neural network we design corresponds to the edge weights of the directed brain network, such that the weight from $input_i$ to $output_j$ is $w_{i,j} \in W$ where W is the adjacency matrix of the brain network.

We use Ridge Regression as the loss function of artificial neural network. Using the loss function we penalize the error rate in the estimation of regional representative activations. We also append a regularization term to the loss function in order to increase the generalization capability of the artificial neural network. We define the loss of output node i as:

$$Loss_i(\overline{b}_i; \overline{w}_i) = E((b_{i,t'} - \sum_{j \neq i, j=1}^{M} w_{i,j} b_{j,t'})^2) + \lambda \overline{w}_i^T \overline{w}_i \tag{32}$$

where vector $\overline{w}_i = [w_{i,1}, w_{i,2} \dots w_{i,M}]$ is the weights connected to node i, the vector $\overline{b}_i = [b_{i,t}, b_{i,t+1} \dots b_{i,t+L}]$ is representative activations of region i during win_k, and λ is the regularization hyperparameter, and $E(.)$ is the expectation operator taken over time interval win_k.

Using Equation (32), we define the proposed artificial neural network architectures for estimation of adjacency matrices of directed and undirected brain networks in the following subsections. For each time window we define on the fMRI signal, we extract brain networks of both structures using the artificial neural networks. The overall process can be seen in Figure 3.1.

3.3.1 Directed Artificial Brain Networks (dABN)

The artificial neural network we design for estimating directed brain network, shown in 3.2 is straightforward. We do not have any restrictions on the weight matrix of directed brain networks, such as the symmetry property of the adjacency matrix, $W = W^T$.

We use loss function define in Equation (32) for the artificial neural network. We

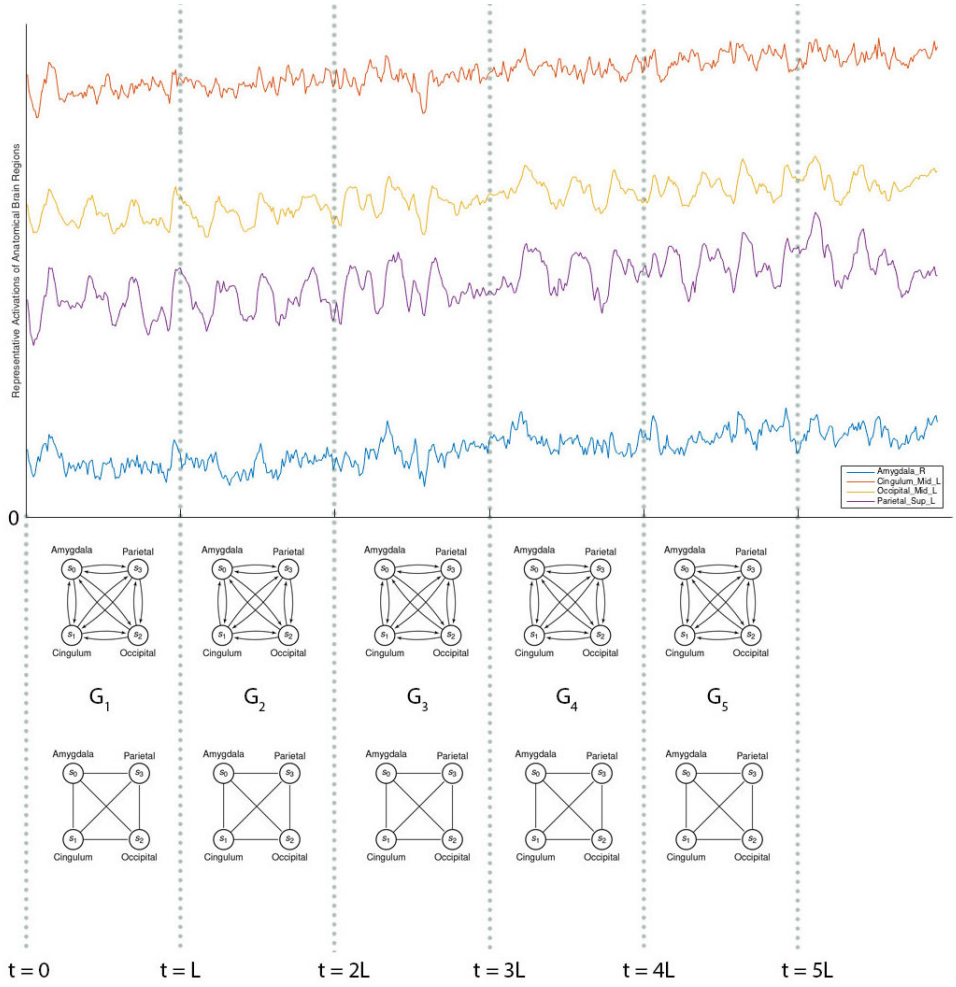


Figure 3.1: A sample figure of brain network extraction approach using non-overlapping time windows from fMRI signals of anatomic regions. Starting from the beginning of the signal, we declare time windows with length L until the end of the signal. For each time window, using the proposed methods, we estimate a directed and an undirected brain networks using the representative activations of the anatomic regions within the time interval.

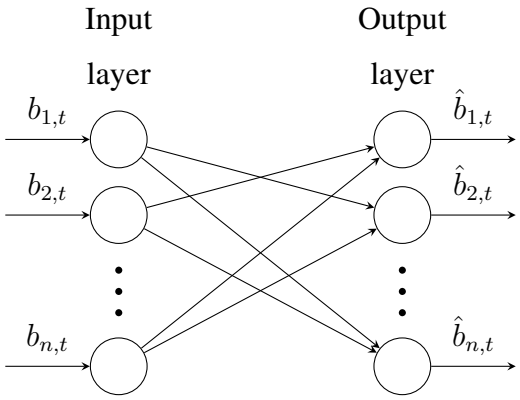


Figure 3.2: Directed Artificial Brain Network Architecture.

apply gradient descent on the loss function to update the weights, $w_{i,j}$:

$$w_{i,j} \leftarrow w_{ij} - \alpha \frac{\partial Loss_i(\overline{b}_i; \overline{w}_i)}{\partial w_{i,j}}. \tag{33}$$

We train the artificial neural network for an empirically chosen number of epochs, $\#epochs$. At the end of the training, we obtain the optimal weights for minimizing the loss function.

3.3.2 Undirected Artificial Brain Network (uABN)

The adjacency matrix of undirected brain network should be symmetrical, $W = W^T$, $w_{i,j} = w_{j,i}$. To create the symmetry, we apply a weight sharing mechanism in the artificial neural network by sharing the weight of every $(w_{i,j}, w_{j,i})$ pair. The artificial neural network we use for creating undirected brain networks is shown in Fig. 3.3.

We use Equation (32) loss function of the artificial neural networks. The weight matrix of uABN is initialized symmetrically, $w_{i,j} = w_{j,i}$ and in order to satisfy the symmetry constraint through the training process, we define the following update rule for the weights, $w_{i,j}$ and $w_{j,i}$:

$$w_{i,j} = w_{j,i} \leftarrow w_{i,j} - \frac{1}{2}\alpha \left[\frac{\partial Loss_i(\overline{b}_i; \overline{w}_i)}{\partial w_{i,j}} + \frac{\partial Loss_j(\overline{b}_j; \overline{w}_j)}{\partial w_{j,i}} \right]. \tag{34}$$

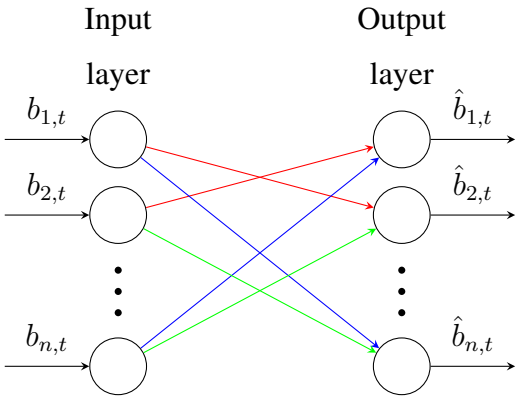


Figure 3.3: Neural Network Structure to Create Undirected Artificial Brain Networks (connections with the same colors are shared).

where vectors $\overline{w}_i = [w_{i,1}, w_{i,2} \dots w_{i,M}]$, $\overline{w}_j = [w_{j,1}, w_{j,2} \dots w_{j,M}]$ are incoming weights to output node i and j of the artificial neural network respectively. The vectors $\overline{b}_i$ and $\overline{b}_j$ denote the representative activation series of region i and region j during time interval win_k respectively.

We optimize the weight of the artificial neural network by updating the weights for an empirically chosen number of epochs, $\#epochs$.

We explain the overall process of extracting directed and undirected brain networks

with a pseudocode in Algorithm 3.2.

Algorithm 3.2: Algorithm of creating subject level brain networks with the proposed method. In **ABN** process, the adjacency matrices are estimated. **Intialize_Weight_Matrix** process is used for random initialization of adjacency matrices

Process `Intialize_Weight_Matrix(` $graphType$`)`**:**
- $W \leftarrow$ randomly initialized (uniform distribution, unif(0,1)) $M \times M$ matrix;
- **if** *undirected* **then**
  - $W \leftarrow \frac{1}{2}(W + W^T)$;
- **end**
- **return** W ;

Process `ABN(`$subject_data$, λ, $\#epochs$`)`**:**
- time_windows = Extract_time_windows($subject_data$)
- Adjacency_matrices = { } ;
- labels = { } ;
- **for** *win in time_windows* **do**
  - win_data = $subject_data$(*win*) ;
  - $W \leftarrow$ Initialize_Adjacency_Matrix($graphType$) ;
  - $epoch = 0$;
  - **while** $epoch$ < $\#epochs$ **do**
    - Calculate loss of the artificial neural network using Equation (32) ;
    - Calculate derivative of loss with respect to W ;
    - Update W with Equation (33) for directed, Equation (34) for undirected case;
    - $epoch \leftarrow epoch + 1$;
  - **end**
  - label $\leftarrow$ cognitive task of win.
  - labels.add(label) ;
  - Adjacency_matrices.add(W) ;
- **end**
- **return** Adjacency_matrices, labels ;

3.4. Baseline Methods

In this subsection, we briefly describe the state of the art methods that have been used to build functional connectivity graphs, in order to provide some comparison for the suggested Artificial Brain Network.

3.4.1 Pearson Correlation

In their work, Richiardi et al. [44] defined the functional connectivity between two anatomic regions as pairwise Pearson correlation coefficients computed between the average activations of these regions in a time interval. The edge weights $w_{i,j}$ and $w_{j,i}$ are computed within time interval win_k as :

$$w_{i,j} = w_{j,i} = \rho_{\overline{b}_i,\overline{b}_j} = \frac{cov(\overline{b}_i, \overline{b}_j)}{\sigma_{\overline{b}_i}\sigma_{\overline{b}_j}}, \tag{35}$$

where $\overline{b}_i = [b_{i,t}, b_{i,(t+1)}, ..., b_{i,(t+L)}]$ representative activation series of region i within time interval win_k, $cov()$ defines the covariance, and σ_s represents the standard deviation of time series s.

3.4.2 Local Mesh Model with Temporal Measurements (LMM-TM)

In order to estimate adjacency matrix of a brain graph within time interval win_k with the proposed method in [49], we estimate the representative activation of a region based on the activations of its neighboring regions. We minimize the loss function in Equation 32 using closed form solution for ridge regression. In estimation of edge weight incoming to region i, $\overline{w}_i = [w_{i,1}, w_{i,2} \dots w_{i,M}]$, we minimize the loss function with respect to $\overline{w}_i$. The following equation is the closed form solution of ridge regression function to find $\overline{w}_i$:

$$\overline{\mathbf{w}}_\mathbf{i} = (\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})^{-1}\mathbf{B}^T\overline{b}_i, \tag{36}$$

where $\mathbf{B}$ is an $L \times (M-1)$ matrix, whose columns consist of the average BOLD

activations of anatomic regions except for the region i in the time interval win_k, such that column j of matrix $\mathbf{B}$ is $\bar{b}_j$. $\lambda \in \mathbb{R}$ represents the regularization hyperparameter.

3.5. Experimental Analysis of Artificial Brain Networks

The aim in creating a brain network is to capture the underlying connectivity patterns of the human brain during a cognitive process. In the previous section of this chapter, we defined how we can estimate a brain network within a time interval of a cognitive task. In this section, we explore how a brain network, G_k, we created with the proposed method describes the underlying cognitive task, y_k. For this purpose, we employ a classification algorithm to categorize a given task into one of the cognitive tasks in the dataset. Based on the performance of the classifier, we evaluate the quality of the brain networks' representation.

In order to compare our method with other brain network extraction methods in the literature, we implement two methods mentioned above: LMM-TM and Pearson Correlation method. Similar to our proposed model, they estimate a brain network, G_k, for every time interval, win_k. Therefore, the methods are evaluated on how informative a brain network they create, G_k is to a classifier about the underlying cognitive task, y_k.

We work on two fMRI datasets: TOL and HCP. On both of these datasets, we apply our brain network extraction methods and the baseline methods to create brain networks.

3.5.1 Human Connectome Project (HCP) Experiment

In Human Connectome Project dataset, 807 subjects attended 7 sessions of fMRI scanning in each of which the subjects were required to complete a different cognitive task with various durations, namely, Emotion Processing, Gambling, Language, Motor, Relational Processing, Social Cognition, and Working Memory. We aim to discriminate these 7 tasks using the edge weights of the formed brain graphs.

In the experiments, the learning rate α was empirically chosen as $\alpha = 10^{-5}$ for both

Table 3.1: Within-Subject Performances of Brain Networks on HCP Dataset.

	Pearson Corr.		Ridge Reg.		dABN		uABN	
λ	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	0.7194	0.16	-	-	**0.7435**	0.13	0.5918	0.13
1	0.7194	0.16	0.5293	0.13	**0.9236**	0.07	**0.9226**	0.07
2	0.7194	0.16	0.5663	0.13	**0.9453**	0.06	**0.9449**	0.06
4	0.7194	0.16	0.6188	0.13	**0.9440**	0.07	**0.9439**	0.07
8	0.7194	0.16	0.6782	0.13	**0.9203**	0.08	**0.9201**	0.08
16	0.7194	0.16	0.7408	0.12	**0.8597**	0.11	**0.8595**	0.11

dABN and uABN and window size is chosen as $L = 40$. We compared the directed and undirected Artificial Brain Networks and Ridge Regression method using various λ values. Since computation of Pearson correlation does not require any hyper parameter estimation, a single performance result is obtained for the Pearson correlation method.

After estimating the arc weights of Artificial Brain Networks and forming the feature vectors from the graph adjacency matrix, we performed within-subject and across-subject experiments using Support Vector Machines with linear kernel. During the within-subjects experiments, we performed 3-fold cross validation using only the samples of a single subject. Table 3.1 shows the average within-subject experiment results over 807 subjects, when the classification is performed using a single subject brain network of 7 tasks. During the across-subject experiments, we performed 3-fold cross validation using the samples obtained from 807 subjects. For each fold we employed the samples from 538 subjects to train and 269 subject to test the classifier. Table 3.2 shows the across-subject experiment results.

Table 3.1 shows that in within subject experiments our methods, dABN and uABN, give the best performances in classifying the cognitive task under different λ values. Furthermore performances of directed networks are slightly better than undirected ones. It can be observed that as λ increases, generalization of our models also increase up to $\lambda = 2$.

Table 3.2: Across-Subject Performances of Brain Networks on HCP Dataset.

	Pearson Corr.		Ridge Reg.		dABN		uABN	
λ	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	**0.7524**	0.01	-	-	0.6654	0.01	0.5681	0.01
1	0.7524	0.01	0.6737	0.00	**0.8211**	0.00	0.8179	0.00
2	0.7524	0.01	0.6931	0.00	**0.8347**	0.01	**0.8334**	0.01
4	0.7524	0.01	0.7189	0.00	**0.8410**	0.01	**0.8406**	0.01
8	0.7524	0.01	0.7481	0.01	**0.8400**	0.01	**0.8402**	0.01
16	0.7524	0.01	0.7760	0.00	**0.8327**	0.01	**0.8328**	0.01

Table 3.2 shows that our methods outperform the others within a range of λ values, $\lambda = \{1, 2, 4, 8, 16\}$. Pearson Correlation results in the best accuracy when no regularization is applied to Artificial Brain Networks.

3.5.2 Tower of London(TOL) Experiment

We also test the validity of the suggested Artificial Brain Network on a relatively more difficult fMRI dataset, recorded when the subjects solve Tower of London (TOL) problem. TOL is a puzzle game which has been used to study complex problem solving tasks in human brain. TOL dataset used in our experiments contains fMRI measurements of 18 subjects attending 4 sessions of problem solving experiment. In the fMRI experiments, subjects were asked to solve 18 different puzzles on computerized version of TOL problem [8]. There are two labeled subtasks of problem solving with varying time periods namely, planning and execution phases.

As the nature of the data is not suitable for a sliding window approach and the dimensionality is too high for a computational model, in our previous study [62], a series of preprocessing steps were suggested for the TOL dataset. In this study, we employ the first two steps of the pipeline. In the first step, *voxel selection and regrouping*, a feature selection method is applied on time series of voxels to select the "important" ones. Then, the activations of the selected voxels in the same region are averaged

Table 3.3: Within-Session Performances of Mesh Networks on TOL Dataset.

	Pearson Corr.		Ridge Reg.		dABN		uABN	
λ	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	0.6119	0.09	-	-	**0.8914**	0.11	0.8499	0.12
1	0.6119	0.09	0.6688	0.10	**0.8913**	0.11	0.8499	0.12
2	0.6119	0.09	0.6651	0.10	**0.8914**	0.11	0.8499	0.12
4	0.6119	0.09	0.6679	0.10	**0.8906**	0.11	0.8499	0.12
8	0.6119	0.09	0.6685	0.10	**0.8905**	0.11	0.8500	0.12
16	0.6119	0.09	0.6705	0.10	**0.8912**	0.11	0.8498	0.12

to obtain the activity of corresponding region. As a result, a more informative and lower dimensional representation is achieved. In the second step, bicubic spline interpolation is applied to every consecutive brain volumes and a number of new brain volumes are inserted between two brain volumes to increase temporal resolution. For the details of interpolation, refer to [62]. In this study, the optimal number of volumes inserted between two consecutive brain volumes are found empirically and it is set to 4. Therefore, the time resolution of the data is increased four times.

We applied the above-mentioned preprocessing steps to all of the 72 sessions in the dataset. After the voxel selection phase, number of regions containing selected voxels is much less than 116 regions. Note that, we discard regions located in Cerebellum and Vermis. Window size for this dataset was set to $L = 5$, since there are at least 5 measurements for every sub-phase after the interpolation. The neural network parameters used in our experiments are $\alpha = 10^{-6}$ and $\#epochs = 10$. Table 3.3 shows the mean and standard deviation of classification accuracy obtained with our method and the base-line methods. Similar to HCP experiments, we slid non-overlapping windows on the measurements and we performed 3-fold cross validation during TOL experiments.

Table 3.3 shows that the suggested Artificial Brain Networks model gives better performances than that of Pearson Correlation and Closed Form Ridge Regression methods in classifying sub-phases of complex problem solving under various regulariza-

tion parameters. We observe that decoding performances of directed brain networks outperforms those of undirected brain networks.

3.6. Chapter Discussion

In this chapter, we applied classification on brain networks and took advantage of their discrimination power. We showed that, representation power of the proposed methods is greater than the power of existing brain network extraction methods. Main focus of this chapter was the representative power of our networks.

We considered studying brain networks to reveal connectivity patterns of the brain. For this purpose, use of network properties [63, 64] and brain graph visualization tools [65, 66, 67] could be convenient options. However, we estimated almost 100,000 brain networks for TOL dataset, more than 200,000 brain networks for HCP dataset. Quantity of the brain networks makes any analysis to be applied unfeasible and incorrect due to big differences among brain networks of different subjects.

In the next chapter, Chapter 4, in order to solve the problems of brain networks mentioned above, we propose an algorithm that estimates task level brain networks. It means that, we will describe a cognitive task of a tfMRI dataset with a single brain network. This brain network will present common connectivity patterns of all the subjects in the dataset. Reducing the number of the brain networks and settling individual differences among the subjects will help us to examine the functional connections of the human brain in a deeper sense.

In this chapter, we considered complete brain graphs where all regions are assumed to have connections with each other. A sparser brain representation can be computationally more efficient and neuroscientifically more accurate. For sparsification of brain networks, we will apply thresholding to the weight matrices in Chapter 4.

Although the number of publicly available fMRI datasets increases [10], there is not a dataset commonly used by the majority of computational neuroscience researchers. Hence, we can not ensure the classification accuracy we obtain with our brain networks is the highest up to now. For a more detailed analysis, we leave observing the

performance of the proposed methods in other tfMRI dataset as a future work.

It has been shown that the segmentation of the brain plays an essential role in brain decoding and discrimination power of brain networks [32, 68]. In this study, we used a precomputed brain atlas for the segmentation of the brain. However, a computational group-level parcellation method could be employed in order to group the voxels of a brain volume into clusters where the voxels with similar functional behaviours will be gathered into the same cluster. We believe, a custom brain parcellation would improve the classification performance of the brain networks.

3.7. Chapter Conclusion

In this chapter, we proposed two novel methods to estimate directed and undirected brain networks from fMRI data. We designed classification experiments in two fMRI datasets, namely HCP and TOL, to show the power of feature representation of our methods. With HCP dataset, we conducted within and across subject classification experiments. We show that the classification accuracy is higher when we employ our methods compared to the two state of the art brain network extraction methods in both within and across subject experiments. In HCP dataset, we also show the classification performances obtained with dABN and uABN are very close to each other.

We conducted within-session classification using TOL dataset. Since the fMRI data of TOL experiment is not suitable for brain network extraction, we applied a preprocessing pipeline that served us an output that we can build brain networks on. Although it is a more challenging dataset than HCP, we achieved the highest classification performance with our methods.

CHAPTER 4

MODELING COGNITIVE TASK- LEVEL BRAIN NETWORKS WITH ARTIFICIAL NEURAL NETWORKS

Classical brain network extraction methods in the literature creates subject level brain networks from fMRI data. Similar to our approach in Chapter 3, for a single subject, a set of brain networks are created for each cognitive process the subject goes through. Using brain networks of all subjects in a dataset, a classification algorithm is run in order to evaluate the discrimination power of these brain networks.

In this chapter, we suggest a different approach compared to the classical approaches for classification of brain networks. We propose a method to create a single brain network for each cognitive task in a fMRI dataset. A task-specific brain network of a cognitive task enables us to study the similar patterns shared among all the subjects and shared connectivity patterns occurring in the brains of all subjects in the dataset. Therefore, brain networks will enable us to study population-wise functional brain connectivities.

We estimate the task specific brain networks using artificial neural network architectures, defined in Chapter 3. We adapt the methods for extracting both directed and undirected Artificial Brain Networks. Therefore, we estimate both directed Task Specific Brain Networks (dTSN) and undirected Task Specific Brain Networks(uTSN). We visualize the estimated artificial brain networks using Brainnet Viewer toolbox [65] to study them in a deeper sense. Observing these network visualizations, we draw some surprising conclusions regarding the structure of brain networks. For example, we observe that human brain tends to form undirected connectivity patterns among the anatomic regions during a cognitive process.

We propose a classification method to categorize an unseen chunk of fMRI data into a cognitive task. Using this classification method, we perform across subject classification on HCP dataset to evaluate the validity of the estimated directed and undirected brain networks.

4.1. Estimation of Task-Specific Brain Networks

In this section, we introduce a new method to estimate generic directed and undirected Task-Specific Networks (TSN) as an alternative to estimate an Artificial Brain Network for each subject and for each cognitive task.

We denote the task specific brain network of cognitive task c as $G_c = (V, W_c)$. As in Chapter 3, $V = \{v_1, v_2, v_3, \ldots, v_R\}$ is for vertices representing $M = 90$ anatomical brain regions. We denote the adjacency matrix of the graph by $W_c = (w_{i,j}^{(c)})$ where $w_{j,k}^{(c)}$ is the weight to edge from v_i to v_j in G_c.

We aim to construct one Task Specific Brain Network for each cognitive task by estimating the relationships among the node attributes of $G_c = (V, W_c)$. Recall that each node attribute is a vector with entries corresponding to the samples of representative time series, obtained for each anatomic region. We train a single neural network by employing the representative time series of anatomical brain regions of all subjects in the datasets to create task specific brain networks.This training methodology enables us to capture the shared properties of a cognitive task across all the subjects. We denote the representative time series (action) of anatomical region i at time t as $b_{i,t}$.

When we have n different cognitive tasks, at total, we have n directed or undirected TSNs, $G_{all} = \{G_1, G_2 \ldots G_n\}$. The overall process can be seen in Figure 4.1.

We group temporally consecutive representative time series from all of the R brain regions into a chunk. A chunk with index k, $B_k \in R^{L \times M}$, consist of fMRI activations within time window k. Since, we use time intervals with fixed length L, the time window k, win_k, represents the time interval $[k * (L-1), k * L]$.

We define a matrix for each chunk at the input of our suggested neural network archi-

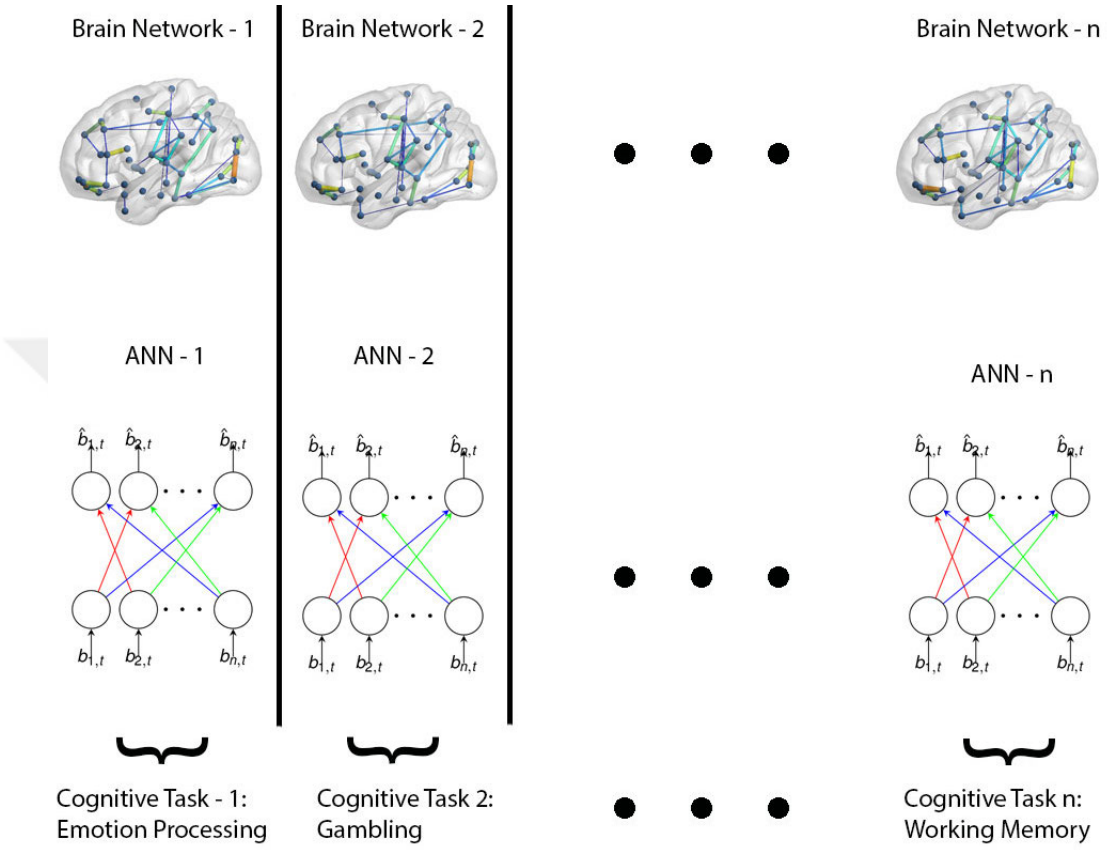


Figure 4.1: Training methodology to estimate task-level brain networks. Assuming we have n cognitive tasks in a task fMRI dataset, we extract a brain network for each cognitive task. For estimation of each brain network, we train an artificial neural network with the architecture we defined in Section 3. The estimated brain networks represent common connectivity patterns shared across all the subjects in the underlying fMRI dataset.

tecture. The $1th$ chunk we create is represented by B_1, in the following way:

$$B_1 = \begin{bmatrix} b_{1,1} & b_{2,1} & b_{3,1} & \dots & b_{M,1} \\ b_{1,2} & b_{2,2} & b_{3,2} & \dots & b_{M,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{1,L} & b_{2,L} & b_{3,L} & \dots & b_{M,L} \end{bmatrix}$$

Recall that L represents the window length of the time series and M is the number of nodes corresponding to each anatomic region.

4.2. Representation of Task Specific Brain Networks

Suppose that, during a given time interval a chunk B_k is recorded corresponding to a single cognitive process. Suppose, also, that the index of this cognitive process is c. We expect the task specific brain network corresponding to this cognitive process, G_c, represents the connectivity of this chunk.

As suggested in Chapter 3, we define the connectivity of anatomic regions as the reconstruction capability of a node attribute from the linear combination of the node attributes of rest of the node attributes. In other words, a node attribute vector is reconstructed from the linear combination of the rest of the node attributes. This approach enables us to estimate the entries of the adjacency matrix of G_c from the column vectors of B_k . Formally speaking, a representative activation at time instance t', for an anatomic region i, is assumed to satisfy the following equation:

$$b_{i,t'} = \sum_{j=1, j\neq i}^{M} w_{i,j}^{(c)} b_{j,t'} + \epsilon_{i,t'} = \hat{b}_{i,t'} + \epsilon_{i,t'}, \qquad \forall t' \in win_k \tag{41}$$

where $w_{i,j}^{(c)} \in W_c$ denotes the edge weights of graph G_c, $\hat{b}_{i,t'}$ denotes the estimated representative activation of region i at time t', $\epsilon_{i,t'}$ denotes the estimation error for region i at time t and win_k denotes the time interval $[k*(L-1), k*L]$.

4.3. Estimation of Task Specific Brain Networks

In order to estimate the weight matrix of a generic task-specific brain network, we adopt the artificial neural networks, defined in Chapter 3. Since we aim to create n brain networks for n cognitive tasks, we create n distinct artificial neural networks for each cognitive task.

The suggested task specific neural network is trained by initializing the adjacency matrices of each brain networks with random rational numbers sampled from a uniform distribution ($unif(0,1)$). In addition to random initialization, we satisfy the symmetry property of undirected brain networks with the assignment $W_c \leftarrow \frac{1}{2}(W_c + W_c^T) \qquad \forall c \in \{1, 2 \dots n\}$.

We use ridge regression function as the loss function of the artificial neural networks. The following calculates the loss of the neural network corresponding to cognitive task c when the input is chunk B_k:

$$\begin{aligned} Loss(B_k; W_c) &= \frac{1}{L} \sum_{i=1}^{M} \sum_{t' \in win_k} (b_{i,t'} - \hat{b}_{i,t'})^2 + \lambda \sum_{i}^{M} \sum_{j}^{M} (w_{i,j}^{(c)})^2 \\ &= \frac{1}{L} \sum_{i=1}^{M} \sum_{t' \in win_k} (b_{i,t'} - \sum_{j=1, j \neq i}^{M} w_{i,j} b_{j,t'})^2 + \lambda \sum_{i}^{M} \sum_{j}^{M} (w_{i,j}^{(c)})^2, \end{aligned} \tag{42}$$

where the second term is a regularization term used to increase the generalization capability of the neural networks and λ is regularization hyperparameter. At each epoch, gradient descent is applied to update the weights. We use the following equation to update the weights of a neural network for directed Task Specific Brain Network estimation:

$$w_{i,j}^{(c)} \leftarrow w_{ij}^{(c)} - \alpha \frac{\partial Loss(B_k; W_c)}{\partial w_{i,j}^{(c)}}, \tag{43}$$

where α is learning rate hyperparameter and the gradient is calculated with the fol-

lowing formula:

$$\frac{\partial Loss(B_k; W_c)}{\partial w_{i,j}^{(c)}} = \frac{2}{L} \sum_{t'=t}^{t+L} -(b_{i,t'} - \sum_{j=1, j\neq i}^{M} w_{i,j}^{(c)} b_{j,t'}) b_{j,t'} + 2\lambda w_{i,j}^{(c)}. \tag{44}$$

The above equation can be generalized to all of the chunks B_k by gradient calculation for all output nodes with matrix operations:

$$\nabla W_c = -\frac{2}{L}(B_k - B_k W_c^T)B + 2\lambda W_c. \tag{45}$$

For the gradient descent in undirected brain graph extraction however, we cannot use equation 44 or 45, since these equations disturb the symmetry of the weight matrix. The symmetry property of the weight matrix can be satisfied by employing the following equation to ensure the equality between each $(w_{i,j}, w_{j,i})$ pair.

$$w_{i,j}^{(c)} = w_{j,i}^{(c)} \leftarrow w_{i,j}^{(c)} - \frac{1}{2}\alpha \left[\frac{\partial Loss(output_i)}{\partial w_{i,j}^{(c)}} + \frac{\partial Loss(output_j)}{\partial w_{i,j}^{(c)}} \right]. \tag{46}$$

We define gradient calculation for undirected brain graph extraction by the following trick after the calculating ∇W_c with equation 45:

$$\nabla W_c \leftarrow \frac{1}{2}(\nabla W_c + (\nabla W_c)^T) \tag{47}$$

A generic task-specific brain network is estimated by training a neural network for each specific task, with chunks from all subjects in fMRI dataset, in each iteration. For each chunk, we update the weights of the neural network that corresponds to the cognitive process performed during that chunk. We do the iteration for an empirically chosen number of repetitions, $\#epochs$. The overall procedure to create task-generic

brain networks is explained as a pseudocode in Algorithm 4.1.

Algorithm 4.1: Algorithm to create Task Specific Brain Networks. Random samples of matrix elements are from uniform distribution (*U(0,1)*)

Result: n directed or undirected Task Specific Brain Networks

for *task c in cognitive tasks* **do**
  if *directed* **then**
    $W_c \leftarrow$ randomly initialized $M \times M$ matrix with 0 diagonal;
  end
  if *undirected* **then**
    $W_c \leftarrow$ randomly initialized $M \times M$ **symmetric** matrix with 0 diagonal;
  end
end
epoch = 0 ;
while *epoch* < *#epochs* **do**
  for *subj in subjects* **do**
    for *chunk in subj's chunks* **do**
      label $\leftarrow$ cognitive task of chunk ;
      calculate loss with equation 42 ;
      calculate ∇W_{label} with Equation (45) for directed, (47) for undirected case ;
      update weight matrix $W_{label} \leftarrow W_{label} + \alpha \nabla W_{label}$;
    end
  end
  epoch $\leftarrow$ epoch$+1$
end

4.4. A Novel Classification Approach Based on Task- Specific Brain Networks

In the previous section, we introduce a new method for estimating Task-Specific Brain Networks. Our goal was to find a brain network, which provides common information about the connectivity patterns of a specific cognitive task for a subject population. We partition the fMRI data set into chunks of time series at each anatomic region

for all of the subjects who performed the same set of cognitive tasks. These chunks are used to train an artificial neural network for each cognitive task in order to learn the arc-weights of the brain networks. The major assumption of this approach is that the functional relations among the anatomic brain regions can be estimated from the linear relationships of representative time series among the anatomic regions. We, also, assume that the estimated brain networks hold compressed information about a cognitive process captured by the fMRI data, recorded during an experimental setup by modeling the spatio-temporal activation patterns in human brain for a pre-defined cognitive task.

In this section, we aim to develop a supervised classification algorithm to test the validity of our assumptions, about the generic Task-specific Brain Networks. For this purpose, we train a classifier using the adjacency matrices of brain networks and measure the test performance. If the representation power of the estimated network is high, then we expect to obtain a relatively high performances compared to the state of the art network representation models. Since the algorithm is a supervised learning algorithm, we define the training process and testing process of the algorithm.

In training phase of the algorithm, our aim is to train the model on the features so that it forms a unique representation for each cognitive task. In our case, the features are representative activations of anatomical regions at each time step and class of the features are cognitive task performed in this time steps.

We consider the Task Specific Brain Network generation process as the training phase of classification algorithm. For each class, we created a neural network that learns the unique properties of data chunks of its corresponding class.

In testing phase of the algorithm, we aim to predict an unseen chunk of data using the knowledge we obtained during the classification phase. Our main objective during the development of Task Specific Brain Networks algorithms was to train an artificial neural network that can reconstruct a given chunk of fMRI data. Since, each neural network we constructed are specialized in a certain cognitive task, we expect that they can reconstruct a chunk from their cognitive task with minimum error. We formulate

this error as a reconstruction loss of the network in following way:

$$\begin{aligned} Loss_{TSN}(B_k; W_c) &= \frac{1}{L}\sum_{i=1}^{M}\sum_{t'\in win_k}(b_{i,t'} - \hat{b}_{i,t'})^2 \\ &= \frac{1}{L}\sum_{i=1}^{M}\sum_{t'\in win_k}(b_{i,t'} - \sum_{j=1,j\neq i}^{M} w_{i,j}^{(c)} b_{j,t'})^2 \end{aligned} \tag{48}$$

where $W_c = (w_{i,j}^{(c)})$ is the neural network weights for cognitive task c and B_k is an unseen chunk consisting of $b_{i,t'}$, win_k the time interval B_k represents. When assigning a class for batch B_k, we measure the loss of each n neural network to see whose estimation causes the smallest loss. We estimate the label of B_k, $\hat{y}_k$, as:

$$\hat{y}_k = \arg\min_c Loss_{TSN}(B_k; W_c) \tag{49}$$

For a given batch of fMRI data with unknown underlying cognitive task, we give the pseudocode of the classification algorithm that assigns a label for the batch of data in Algorithm 4.2.

Algorithm 4.2: Algorithm to classify an unseen batch of fMRI data using TSNs. The algorithm search through all task specific brain networks in order to find the one that gives the minimum reconstruction loss.

```
Process Classifiy_batch(batch : B_k):
    min_loss = ∞ ;
    assigned_class = -1 ;
    for task c in cognitive tasks do
        c_loss = Loss_TSN(B_k; W_c) ;
        if c_loss < min_loss then
            min_loss = c_loss ;
            assigned_class = c ;
        end
    end
```

4.5. Experimental Analysis of Task Specific Brain Networks

In this section, we describe the experiments, conducted on the Task- Specific Brain Networks. In our experiments, we used Human Connectome Project (HCP) fMRI dataset. In this data set there are total of 807 subjects, which attended to the experiment,which records the fMRI signals under 7 cognitive tasks. The number of sessions are also 7, when the subjects perform these seven tasks. In each of these sessions, the experimentees were required to perform a different cognitive task, namely, Emotion Processing, Gambling, Language, Motor, Relational Processing, Social Cognition, and Working Memory.

We represent $n = 7$ cognitive tasks as both directed and undirected Task Specific Brain Networks. We define $M = 90$ nodes for the graphs and edges between every pair of nodes. This nodes represents the anatomical brain regions defined by AAL atlas [38].

We use representative activations of anatomical regions. We use time windows with length $L = 40$ in order to temporally partition from the data and create data chunks.

We use 807 subject of HCP dataset in our experiments. We perform 3 fold cross validation. We reserve 538 subject for training directed and undirected Task Specific Brain Networks. We apply our classification method on the remaining 269 subject using estimated brain networks. We repeat the training and testing procedure for 3 times and each time we select different subset of subject for testing.

Our methods include 3 hyperparameters, regularization hyperparameter (λ), learning rate of neural networks (α) and number of iterations ($\#epochs$). We fix the iteration number for our all experiments to $\#epochs = 200$. We select λ from the set $\lambda = \{1, 2, 4, 8, 16\}$ and α from the set $\alpha = \{5*10^{-5}, 5*10^{-6}, 5*10^{-7}, 5*10^{-8}, 5*10^{-9}\}$.

4.6. Decoding Performances of Task-Specific Brain Networks

We share the classification results of directed and undirected Task Specific Brain Networks created with the hyperparameter set defined in the previous section. Table 4.1

shows the classification performance of directed Task Specific Networks on HCP dataset and 4.2 shows the performance of undirected networks.

A brief analysis of 4.1 and 4.2 reveals two important facts: First of all, when optimized with the appropriate hyper parameters, both directed and undirected networks provide very similar decoding performances. This fact can be interpreted as the connectivity among the anatomic regions is rather undirected. Because extra weights in the opposite direction do not introduce additional information to the brain networks in terms of brain decoding performances. Secondly, the decoding performance of the task-specific networks for HCP dataset is observed around 75%. When we compare this result to the across subject performances of Table 3.2, where each subject has a different network for the same task, we observe that the decoding performance is around 84%. Although there is almost 10% performance loss in the task specific brain networks, this result is very interesting showing that task specific networks still have a substantial representation power. Therefore, one may conclude that although there is a great between - subject variability, a lot of common connectivity properties among the anatomic regions across all the subjects are shared.

Table 4.1: Decoding Performance of Directed TSN on HCP Dataset.

	$\alpha = 5 \times 10^{-8}$		$\alpha = 5 \times 10^{-7}$		$\alpha = 5 \times 10^{-6}$		$\alpha = 5 \times 10^{-5}$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
$\lambda = 1$	0.2238	0.00	0.2222	0.00	**0.7522**	0.01	0.7474	0.01
$\lambda = 2$	0.2235	0.00	0.2331	0.00	0.7325	0.01	0.7289	0.01
$\lambda = 4$	0.2223	0.00	0.5486	0.01	0.6961	0.01	0.6906	0.01
$\lambda = 8$	0.2222	0.00	0.6289	0.01	0.6354	0.01	0.6223	0.01
$\lambda = 16$	0.2222	0.00	0.5544	0.01	0.5542	0.01	0.5258	0.00

Table 4.2: Decoding Performance of Undirected TSN on HCP Dataset.

	$\alpha = 5 \times 10^{-8}$		$\alpha = 5 \times 10^{-7}$		$\alpha = 5 \times 10^{-6}$		$\alpha = 5 \times 10^{-5}$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
$\lambda = 1$	0.2222	0.00	0.2249	0.00	**0.7513**	0.01	0.7493	0.01
$\lambda = 2$	0.2221	0.00	0.3138	0.01	0.7325	0.01	0.7302	0.01
$\lambda = 4$	0.2222	0.00	0.5925	0.01	0.6961	0.01	0.6916	0.01
$\lambda = 8$	0.2222	0.00	0.6325	0.01	0.6351	0.01	0.6241	0.01
$\lambda = 16$	0.2222	0.00	0.5548	0.01	0.5543	0.01	0.5261	0.00

Table 4.1 and 4.2 clearly shows that the prediction accuracy improves with the increase of α hyperparameter. The reason behind this could be the slow convergence rates of the artificial neural networks caused by small α values or insufficient number of epochs for these slow learners. Up to the point $\alpha = 5 \times 10^{-7}$, the common results for all various λ selections results in accuracy of 22%, this could be due to the prediction being stuck with the dominant class in the dataset which also compose a similar ratio of the overall dataset. Because of these reasons, we can focus on the models with greater learning rates, as they ensure a convergence.

We can observe that the classification performance of dTSN and uTSN are very close to each other under the same hyperparameter selections, although undirected networks have only half of the feature number that directed networks have. As mentioned above, the reason behind this redundancy could be the human brain's tendency to operate with undirected connectivity patterns. In order to have a better understanding about the differences and similarities between the directed and undirected brain networks let us try to visualize and examine the brain graphs, in the next section.

4.7. Visualizations

As we described the structure of the estimated brain networks in Section 4.1; the nodes of the network represents the average time series of anatomical brain regions

and the edges between these nodes are the assumed to be functional connectivities of the corresponding anatomical regions. We visualize the brain networks that we estimated on a brain template using the physical coordinates of the brain regions.

We use BrainNet Viewer Toolbox [65] to visualize directed and undirected Task Specific Brain Networks, estimated by our artificial neural network approach. As we have $n = 7$ cognitive tasks (Emotion, Gambling, Language, Motor, Relational, Social, and Working Memory), in HCP dataset, we have 7 figures for directed brain networks and 7 figures for undirected brain networks. Since the human brain is a matter in a 3 dimensional space that cannot be sketched on a paper, we share the views from different angles for better understanding. The edge weights in the figures are coded with colors and thickness. Thick red lines describe the powerful connections whereas thin blue lines describe weak relations.

Drawing a complete graph with 90 nodes results in a very complex and cluttered figures. Therefore, we sparsify the edges by retaining the most powerful top 3% of the edges. With this sparsification, we end up with ~ 240 edges in directed graphs and ~ 120 edges in undirected graphs.

The brain graphs are visualized for each specific cognitive task for directed and undirected brain networks. The first two figures, Figure 4.2a and Figure 4.2b, are visualizations of task 1's (Emotion) dTGN and uTGN, Figure 4.3a and Figure 4.3b are for task 2 and so on.

Examining the brain graphs, we draw the following conclusions:

- In general, the edges of sparsified directed graphs are in pairs. If $w_{i,j}$ occurs to be in the sparsified version, $w_{j,i}$ is also included. This observation indicates that the degree of the connectivity is somehow symmetric.

- Directed brain graphs have the tendency to resemble undirected graphs. The weight of two edges between any two nodes, $w_{i,j}$ and $w_{j,i}$, are almost symmetrical.

- The connections are parallel between the right and left lobe of the brain. This observation can be interpreted as a sort of symmetry between the left and right

lobes of the brain. In other words, if there is an edge between these two lobes, it connects the left side and the right side of the same region.For example, right and left amygdala is connected by a strong edge value.

- There is an uncanny resemblance between directed and undirected graphs in every cognitive task. The edge pattern and connectivity power of the edges are very similar to each other.

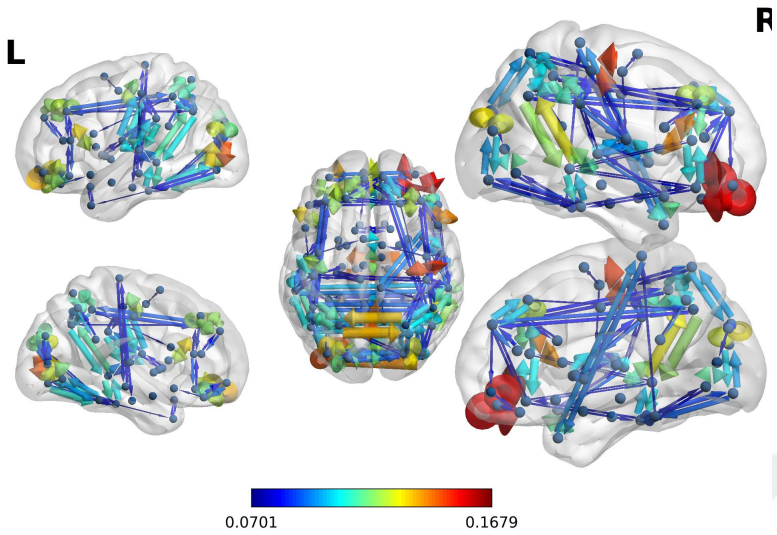
4.8. Chapter Discussion

In this chapter, we modified the brain network estimation algorithm that we proposed in the previous chapter in order to generate task- level brain networks. We could have extended the other brain network estimation algorithms in the literature (such as LMM-TM, Pearson Correlation) for the same purpose by simply concatenating fMRI signals of all subjects in a dataset and applying the brain network estimation method of interest on the entire dataset. However, we would not be able to define a metric for comparing the resulting task specific brain networks in terms of their representation power. We were able to define a metric for our algorithm due to our reconstruction assumption of the algorithm.

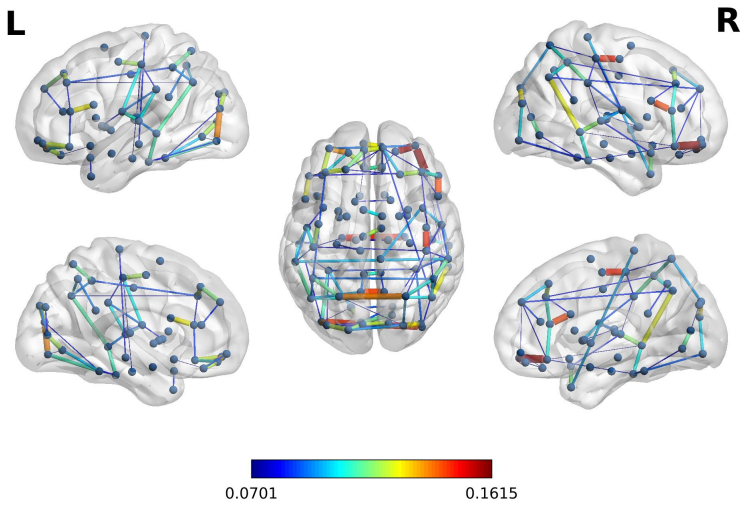
In this chapter, in order to make the brain networks sparse, we applied a thresholding method on the adjacency matrices of the brain networks. In the next chapter, we will also observe the effect of $L1$ regularizer, which is known to be capable of creating sparse networks on the classification performance of brain networks.

While designing the artificial neural networks in this chapter and in Chapter 3, our assumption was that ridge regression would be the best loss function for reconstruction of input signals. Therefore, we have not studied the effect of any other loss function on brain network estimation. In Chapter 5, we will perform experiments with brain networks using two other loss functions, namely, lasso and elastic loss.

Our main strategy towards brain decoding in this chapter and the previous chapter, could be summarized in two steps: Brain network extraction and classification of brain networks. We kept these two phases separate from each other since our methods does not allow any information flow between the phases. This restriction prevents the obtained brain networks from having more discrimination power against the cognitive tasks. In order to establish an information flow between network extraction and classification methods, we propose an end to end artificial neural network architecture in Chapter 5. We observe that retrieving a feedback from the classifier lead to a better classification performance.

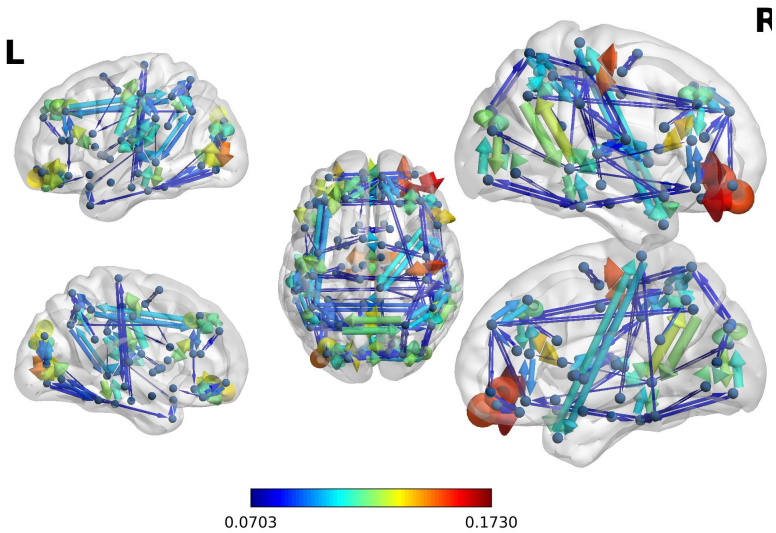


(a) Directed Task Specific Brain Network of Cognitive Task 'Emotion Processing'

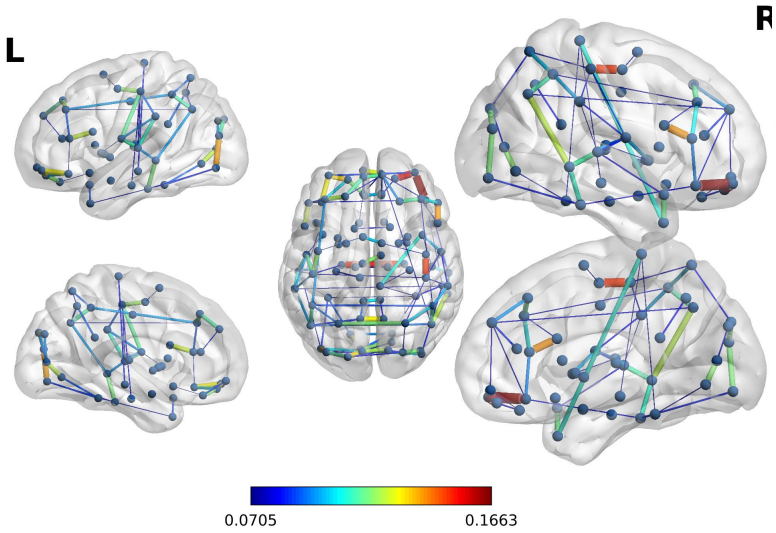


(b) Undirected Task Specific Brain Network of Cognitive Task 'Emotion Processing'

Figure 4.2: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Emotion Processing'.

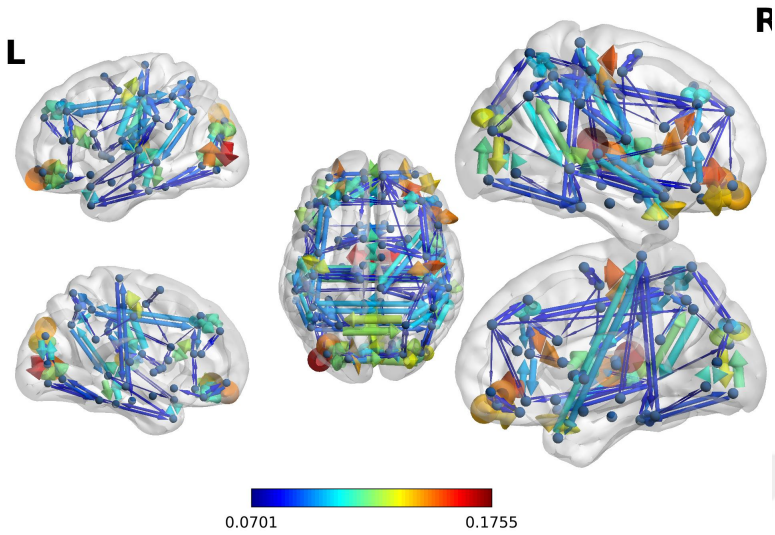


(a) Directed Task Specific Brain Network of Cognitive Task 'Gambling'.

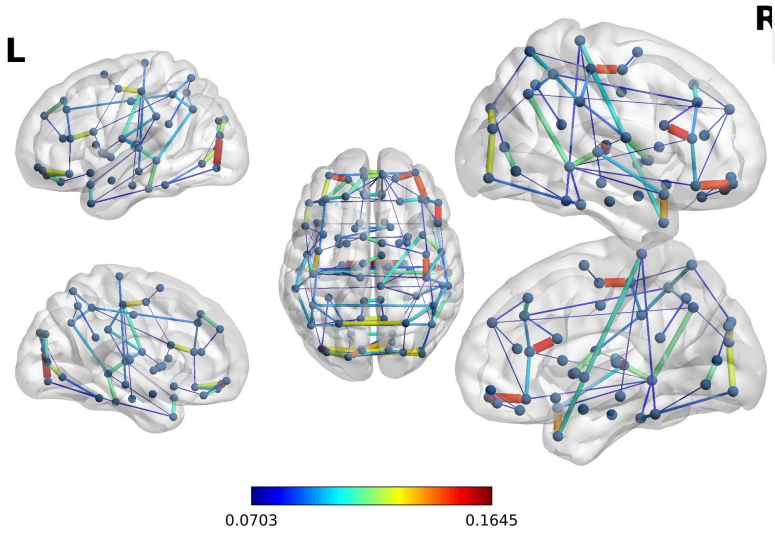


(b) Undirected Task Specific Brain Network of Cognitive Task 'Gambling'.

Figure 4.3: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Gambling'.

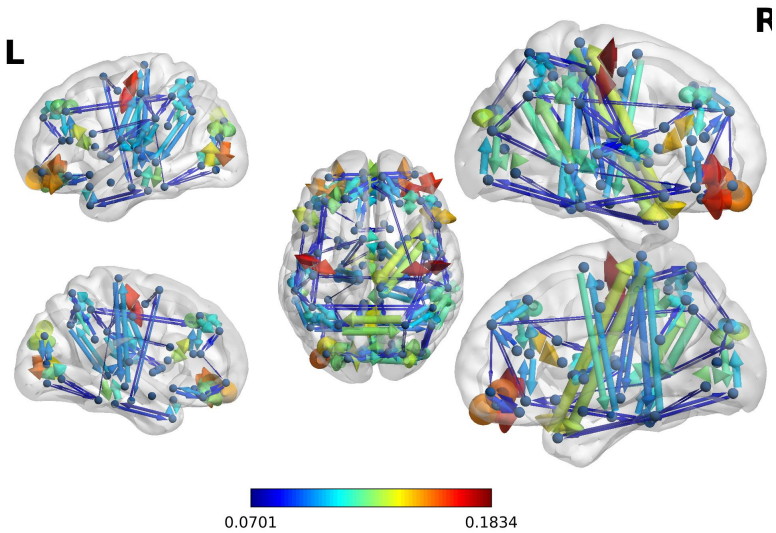


(a) Directed Task Specific Brain Network of Cognitive Task 'Language'.

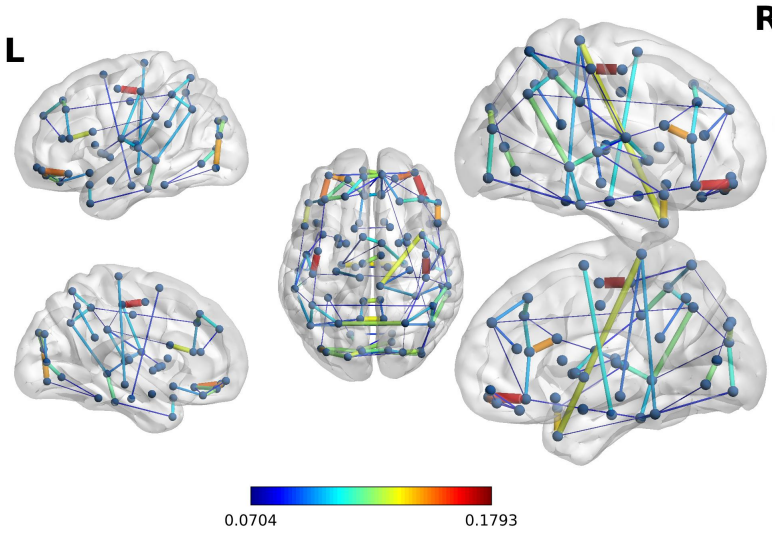


(b) Undirected Task Specific Brain Network of Cognitive Task 'Language'.

Figure 4.4: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Language'.

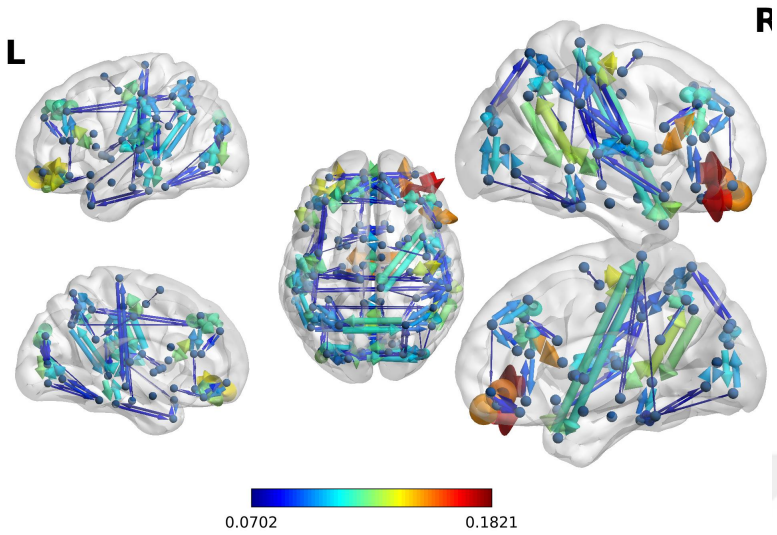


(a) Directed Task Specific Brain Network of Cognitive Task 'Motor'.

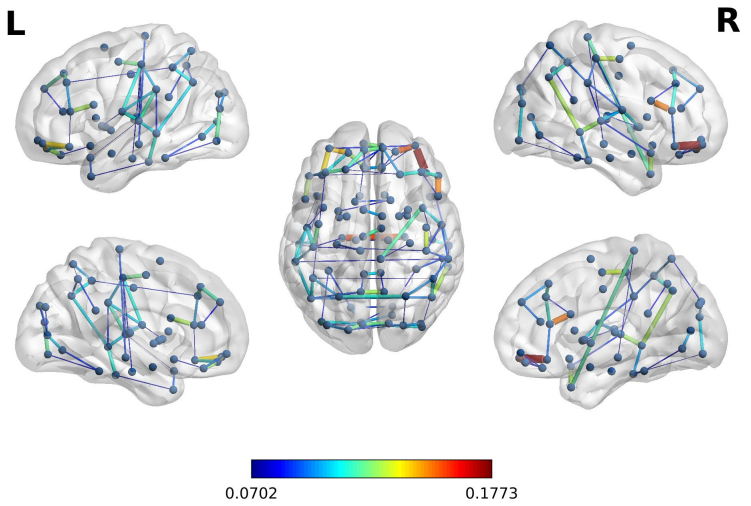


(b) Undirected Task Specific Brain Network of Cognitive Task 'Motor'.

Figure 4.5: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Motor'.

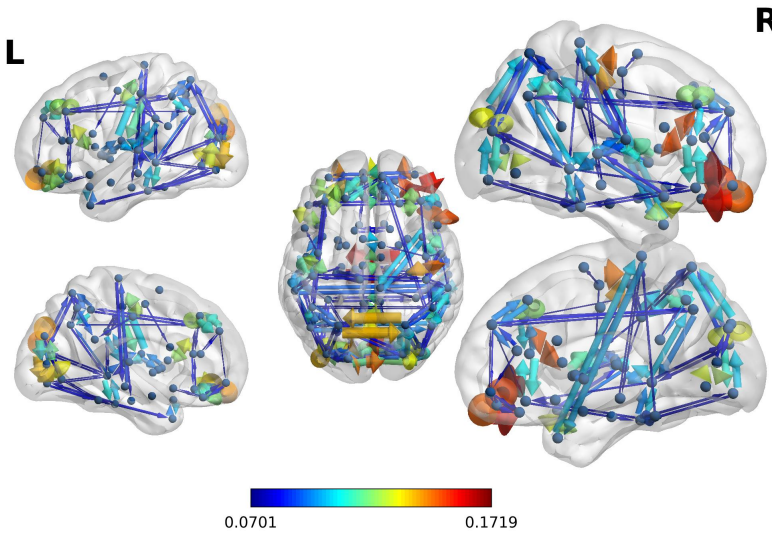


(a) Directed Task Specific Brain Network of Cognitive Task 'Relational Processing'.

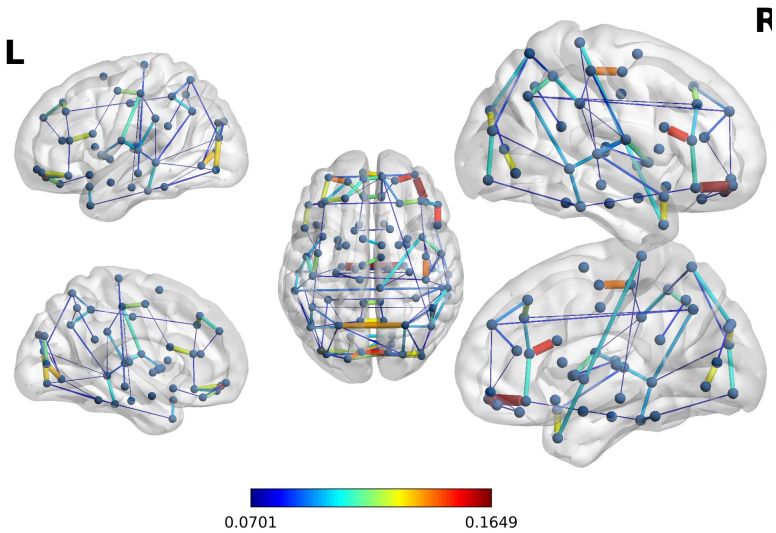


(b) Undirected Task Specific Brain Network of Cognitive Task 'Relational Processing'.

Figure 4.6: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Relational Processing'.

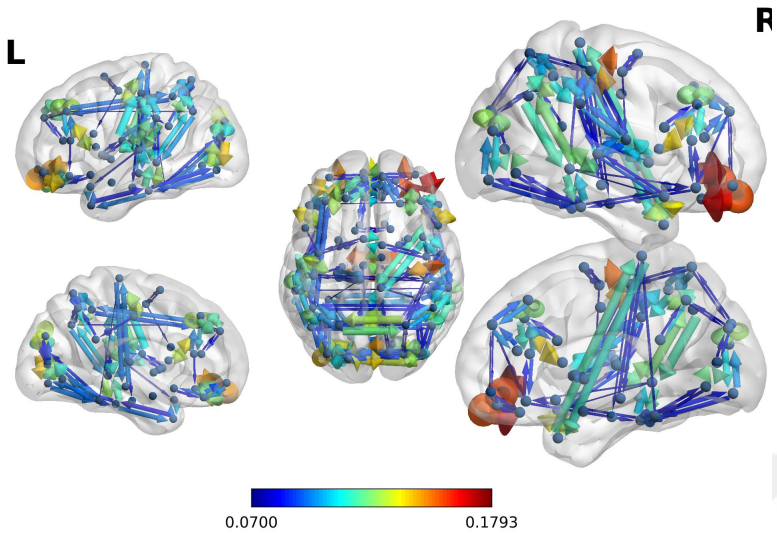


(a) Directed Task Specific Brain Network of Cognitive Task 'Social Cognition'.

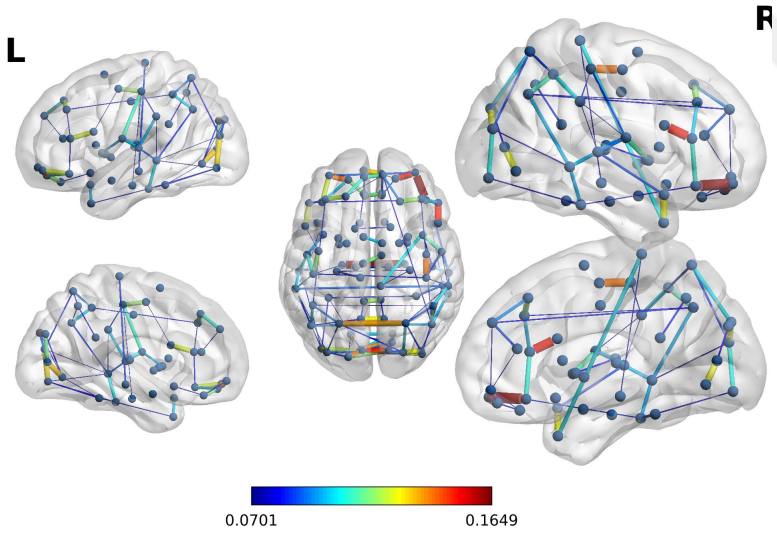


(b) Undirected Task Specific Brain Network of Cognitive Task 'Social Cognition'.

Figure 4.7: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Social Cognition'.



(a) Directed Task Specific Brain Network of Cognitive Task 'Working Memory'.



(b) Undirected Task Specific Brain Network of Cognitive Task 'Working Memory'.

Figure 4.8: Directed (a) and undirected (b) Task Specific Brain Networks of Cognitive Task 'Working Memory'.

4.9. Chapter Conclusion

In this chapter, we took the first step toward estimating task level brain networks. Using our method, we can study the common connectivity patterns in the brain among a subject population. Since our method offers directed and undirected Task Specific Brain Networks, we can study the effect of directionality in the brain network creation.

We validated our Task Specific Brain Networks by a classification algorithm we proposed. We have visualized the task specific brain networks with the highest classification performance.

CHAPTER 5

AN END TO END MODEL TO ESTIMATE AND DECODE THE ARTIFICIAL BRAIN NETWORKS

In Chapter 3, we suggest a computational model to estimate a brain network and using the estimated brain networks, we suggest a brain decoding schema. This is a common approach used in the literature for brain decoding which uses two consecutive steps for model estimation and decoding. In the first step, a computational model is employed in order to estimate the brain graphs. In the second step, a brain decoding algorithm measures the discrimination power of the estimated brain graphs. The estimation and classification models are independent from each other. Furthermore, they optimize different objective functions to achieve the same goal of brain decoding. and they are aiming different objectives, the information flow between them is prohibited and therefore, they cannot run for a common goal.

In this chapter, we define an end to end model that combines construction and classification of brain graphs in a single neural network architecture. In the following sections, we define the evolution of the proposed model step by step. In Section 5.1, we explore the classification performance of Artificial Brain Networks when we replace ridge regression loss function with elastic loss function. In Section 5.2, we use multilayer perceptrons to classify brain graphs into cognitive tasks. In Section 5.3, we redefine the artificial brain network architecture to compute multiple brain networks simultaneously. Finally, combining our brain networks extraction architecture with a multilayer perceptron architecture, we build an end-to-end model in section 5.4.

5.1. Elastic Loss Function

In Chapter 3, we defined non overlapping time windows, $[win_1, win_2 \dots win_n]$ with length L where $n = Z/L$ defines the number of time windows and Z is the length of the fMRI signal. Since we estimate a brain network for every time interval, we had n brain networks, $[G_1, G_2 \dots G_n]$, formed during cognitive tasks $[y_1, y_2 \dots y_n]$, for the entire fMRI signal belonging to a subject.

We represented a brain network formed for a time window win_k as $G_k = (V, W_k)$. For each one of M anatomic brain regions, we defined a vertex and denoted the vertices with V and represented the functional connectivity among the anatomic brain regions, $w_{i,j}^{(k)} \in W_k$, as the elements of the adjacency matrix W_k. We built the relation between the adjacency matrix and the regional activations of time interval win_k as:

$$b_{i,t'} = \sum_{j=1, j\neq i}^{M} w_{i,j}^{(k)} b_{j,t'} + \epsilon_{i,t'} = \hat{b}_{i,t'} + \epsilon_{i,t'} \qquad \forall t' \in win_k, \tag{51}$$

where $w_{i,k}^{(k)} \in W_k$ is an element of the adjacency matrix W_k, $\epsilon_{i,t'}$ is error rate of the estimation of $b_{i,t'}$. In order to minimize the error rate, we proposed two artificial neural architectures, Figure 3.3 and 3.2, with the aim of estimating directed and undirected brain networks. We employed the following loss function, ridge regression loss function, for our artificial neural network architectures:

$$Loss_{ridge}(B_k, W_k) = \frac{1}{L} \sum_{i=1}^{M} \sum_{t' \in win_k} (b_{i,t'} - \hat{b}_{i,t'})^2 + \lambda_2 \sum_i \sum_j w_{i,j}^2, \tag{52}$$

where $win_k = [t, t+1 \dots t+L]$ defines the time interval with length L, B_k defines the activations of $M = 90$ anatomic regions within time interval win_k, such that ith column of B_k defines series of activations of region i during win_k, $\{b_{i,t'}\}_{t'=t}^{t+L}$, $b_{i,t'}$ is the representative activation of region i at time t' and $\hat{b}_{i,t'}$ is the estimation of this activation done by the artificial neural network. Second term in the equation, $\lambda \sum_i \sum_j w_{i,j}^2$ is known as $L2$ regularization. $L2$ regularization is applied for increasing the generalization capability of an artificial neural network. We assumed that the estimated brain network G_k describes the regional connectivities of cognitive task y_k

which takes places during time window win_k.

In this section, we aim to study the effect of $L1$ regularization to the representation power of brain networks. $L1$ regularizer is another regularizer for artificial neural networks and it is used in order to create sparser weight matrices, since it drives some of the weights of the neural network to 0. When we replace $L2$ regularization part of ridge regression function with $L1$ regularization, Equation (52), the loss function is called lasso regression:

$$Loss_{lasso}(B_k, W_k) = \frac{1}{L}\sum_{i=1}^{M}\sum_{t' \in win_k}(b_{i,t'} - \hat{b}_{i,t'})^2 + \lambda_1 \sum_i \sum_j |w_{i,j}|, \tag{53}$$

where λ_1 is the hyperparameter for the regularizer and the neural network is penalized with absolute value of its weights, $|w_{i,j}|$. We aim to have a neural network with generalization capability and sparse weight matrix. The regularization method for this purpose is called elastic loss and it is a combination of $L1$ and $L2$ regularizers. We adapt the elastic loss the following way:

$$Loss_{elastic}(B_k, W_k) = \frac{1}{L}\sum_{i=1}^{M}\sum_{t' \in win_k}(b_{i,t'} - \hat{b}_{i,t'})^2 + \lambda_1 \sum_i \sum_j |w_{i,j}| + \lambda_2 \sum_i \sum_j w_{i,j}^2 \tag{54}$$

where λ_1 and λ_2 are the hyperparameters of $L1$ and $L2$ regularizers respectively. In order to train the weight of the artificial neural networks and optimize the weight matrix W_k with respect to the elastic loss function, $Loss_{elastic}$, we use gradient descent algorithm as we applied in Chapter 3 and 4. At each epoch of training of the neural network, we update the weight matrix as:

$$W_k \leftarrow W_k - \alpha_1 \nabla W_k \tag{55}$$

where α_1 is the learning rate of the optimization process and the $\nabla W_k \in \mathbb{R}^2$ is the derivative of elastic loss function with respect to elastic loss function. We calculate

∇W_k as:

$$\nabla W_k = -\frac{2}{L}(B_k - B_k W_k^T)B + 2\lambda_1 sign(W_k) + 2\lambda_2 W_k \tag{56}$$

where the matrix $sign(W_k) \in \mathbb{R}^{M \times M}$ contains the signs of the elements of the matrix W_k. For $w_{i,j}$, the $sign(.)$ function is defined as:

$$sign(w_{i,j}) = \begin{cases} 1, & \text{if } w_{i,j} > 0 \\ 0, & \text{if } w_{i,j} = 0 \\ -1, & \text{if } w_{i,j} < 0 \end{cases} \tag{57}$$

In order to sustain the symmetry of adjacency matrices of undirected brain networks, we apply the following trick on the gradient matrix, before updating the weights:

$$\nabla W_k \leftarrow \frac{1}{2}(\nabla W_k + (\nabla W_k)^T) \tag{58}$$

The artificial neural networks are trained for an empirically chosen number of epochs with the update update rule defined in Equation (55) in order to estimate the adjacency matrices of dirceted and undirected brain networks.

5.1.1 Experiments Conducted with Elastic Loss Function

We aim to study the effect of elastic loss function to the classification performance of the brain networks. For this purpose, we use HCP tfMRI dataset, since there are high number of subjects attended to the experiment. The dataset consist of scans of 807 subjects performing 7 different cognitive tasks. Using brain networks created for each subject, we aim to predict the cognitive task of an unseen brain network.

We create time windows with length $L = 40$. For each time window we estimate a directed and an undirected brain network. We cross validate λ_1 hyperparameter from the list $\lambda_1 = \{0.2, 0.4, 0.6, 0.8, 1, 2, 4, 8\}$ and λ_2 from the list $\lambda_2 = \{1, 2, 4, 8\}$. We choose the learning rate α as $\alpha = 5 * 10^{-5}$.

We want to observe the representation power of G_k of the cognitive task, y_k. For this purpose, we train a classifier on the adjacency matrices of the estimated brain networks and have a prediction of the underlying cognitive task. We conduct this classification on within subject and across subject experiments. In within subject experiments, we apply the classification using only the brain networks and cognitive task labels from a single subject, and repeating the experiment for all subjects in the dataset. In across subject experiments, on the other hand, we consider the brain networks of all subjects in classification phase.

For classification of brain networks, we use support vector machine (SVM) and multi-layer perceptron (MLP) methods. Using SVM we conduct within subject and across subject experiments. We conduct only across subject experiment using MLP. The reason we cannot conduct within subject experiments using MLP is that the method requires a larger quantity of instances than the number of brain network we have for a subject, $n = 45$.

For both of the classification methods, we apply $k = 3$ cross fold validation. In within subject experiments, we train the classification algorithm of interest with $n = 30$ brain networks and their cognitive tasks, use the remaining data for testing and measuring the accuracy of the method. In across subjects experiments, the brain networks of 538 subjects are reserved for training and the remaining 269 subjects are used for testing.

5.1.1.1 Classification of Brain Networks using SVM

We use an SVM with linear kernel and $L2$ penalization for conducting within subject and across subject experiments. We measure the accuracy of the model as the accuracy of correctly predicted cognitive tasks. Tables 5.1 and 5.2 show the within subject performances. Tables 5.3 and 5.4 show the across subject performances.

Table 5.1: SVM classification accuracy on within subject experiments of directed brain networks created using elastic loss on HCP Dataset.

	$\lambda_2 = 1$		$\lambda_2 = 2$		$\lambda_2 = 4$		$\lambda_2 = 8$	
λ_1	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
0.0	**92.36**	7.39	**94.53**	6.41	**94.40**	6.65	**92.03**	8.19
0.2	86.22	10.32	86.39	10.55	84.90	11.22	80.83	12.49
0.4	68.16	14.98	69.08	14.86	68.10	14.97	64.32	15.33
0.6	49.87	15.55	51.39	15.57	51.37	15.47	48.78	15.10
0.8	35.78	14.06	37.33	14.19	38.12	13.90	36.83	13.30
1.0	26.16	11.89	27.58	11.87	28.64	11.80	28.30	10.99
2.0	15.89	7.18	17.16	6.51	18.80	5.62	20.23	4.45
4.0	21.73	2.01	21.80	1.85	21.86	1.73	21.93	1.59
8.0	22.21	0.80	22.21	0.81	22.22	6.78	22.23	0.75

Table 5.2: SVM classification accuracy on within subject experiments of undirected brain networks created using elastic loss on HCP Dataset.

	$\lambda_2 = 1$		$\lambda_2 = 2$		$\lambda_2 = 4$		$\lambda_2 = 8$	
λ_1	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	**92.26**	7.45	**94.49**	6.46	**94.38**	6.65	**92.01**	8.19
0.2	90.41	8.54	89.72	9.10	87.34	10.30	82.39	12.02
0.4	81.07	11.73	79.22	12.53	75.19	13.54	68.46	14.69
0.6	68.16	14.25	65.66	14.47	61.12	14.74	54.42	14.95
0.8	54.48	14.96	52.31	14.81	48.13	14.42	42.27	13.72
1	41.71	13.86	40.03	13.44	37.01	12.92	32.97	11.69
2	18.70	6.83	19.48	5.95	20.37	4.93	20.85	4.12
4	21.81	1.82	21.88	1.68	21.95	1.56	22.00	1.46
8	22.22	0.76	22.22	0.78	22.22	0.77	22.22	0.17

Table 5.3: SVM classification accuracy on across subject experiments of directed brain networks created using elastic loss on HCP Dataset.

	$\lambda_2 = 1$		$\lambda_2 = 2$		$\lambda_2 = 4$		$\lambda_2 = 8$	
λ_1	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	**82.11**	0.28	**83.47**	0.60	**84.10**	0.54	**84.00**	0.61
0.2	80.04	0.47	81.59	0.49	82.74	0.46	83.01	0.40
0.4	77.15	0.58	79.17	0.56	80.57	0.44	81.10	0.39
0.6	74.03	0.53	76.26	0.67	77.65	0.54	78.38	0.56
0.8	70.72	0.76	73.00	0.68	74.53	0.59	75.28	0.54
1.0	66.71	0.72	69.05	0.64	70.60	0.52	71.40	0.55
2.0	49.13	0.68	50.78	0.87	52.36	0.77	53.45	0.67
4.0	44.46	0.84	45.32	0.73	45.67	0.58	46.15	0.51
8.0	37.36	0.24	37.60	0.45	37.52	0.45	37.74	0.33

Table 5.4: SVM classification accuracy on across subject experiments of undirected brain networks created using elastic loss on HCP Dataset.

	$\lambda_2 = 1$		$\lambda_2 = 2$		$\lambda_2 = 4$		$\lambda_2 = 8$	
λ_1	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	**81.79**	0.28	**83.34**	0.56	**84.06**	0.56	**84.02**	0.57
0.2	79.53	0.76	81.30	0.57	82.54	0.38	82.95	0.46
0.4	76.76	0.56	78.72	0.39	80.26	0.36	81.00	0.51
0.6	73.73	0.75	75.95	0.79	77.39	0.63	78.23	0.62
0.8	70.56	0.50	72.61	0.66	74.19	0.67	75.15	0.57
1.0	67.12	0.62	68.96	0.65	70.53	0.68	71.15	0.63
2.0	49.27	0.77	50.66	0.81	52.35	0.92	53.07	0.90
4.0	43.23	0.59	42.91	0.49	43.63	0.59	44.09	0.69
8.0	33.14	0.41	33.52	0.32	33.31	0.72	33.38	0.25

The Tables 5.1 and 5.2 show that as we increase the impact of $L1$ regularizer, the classification accuracy decreases and we have the best performance when $\lambda_1 = 0$. For the λ_1 values above $\lambda_1 = 1$, we can see that performance is around or even below the chance level, 22%. We observe that $L1$ also hurts the across subject performance, Tables 5.3 and 5.4. In summary, we observe that the performance of brain networks created using ridge regression function, $\lambda_1 = 0$, outperforms all brain network created using elastic loss function in SVM classification.

5.2. Multi-Layer Perceptrons for Classification of Brain Networks

Multilayer Perceptrons (MLP) are a type of feedforward neural networks that has been replacing Support Vector Machines for classification since they could be devised as a more powerful classification tool. For this purpose, we replace SVMs that we used in Chapter 3 with MLPs for classification of brain networks.

We employ an MLP architecture with three layers. The first layer, the input layer, takes the elements of the adjacency matrix of the brain networks. The adjacency matrices we estimate are $M \times M$ matrices and we use $M = 90$ anatomic regions in our experiments. Therefore, we model the input layer with 8100 nodes for the size of a flattened adjacency matrix.

The number of nodes in the middle layer of the constructed MLP is empirically chosen as 500. The layer is called a hidden layer and it further process the input data.

The final layer, output layer, of our architecture returns the probability of a sample belonging to a class. For each class in the dataset, we add an output node that will estimate the posterior probability of its corresponding class. We employ cross entropy loss in order to calculate the loss of prediction:

$$Loss_{MLP}(\mathbf{y}^{(k)}, \hat{\mathbf{y}}^{(k)}, W_k; V_1, V_2) = -\sum_{c=1}^{C} \mathbf{y}_c^{(k)} log(\hat{y}_{k,c}) \tag{59}$$

where $\mathbf{y}$ is an one hot binary vector that holds the true class of the input W_k, such that $\mathbf{y}_t^{(k)} = 1$ means that the sample belongs to class t. The vector $\hat{\mathbf{y}}^{(k)}$ in the formula,

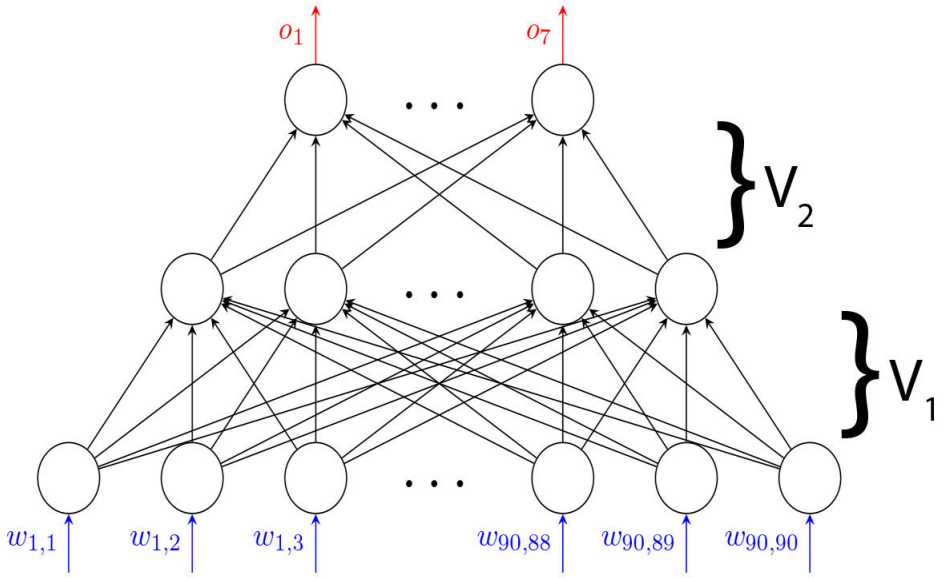


Figure 5.1: A simple representation of the multilayer perceptron we employed in our experiments. The input layer has 8100 nodes and each of them takes an entry of the adjacency matrix. The middle layer, hidden layer, consists of 500 nodes. Each of 7 nodes of the output layer return the probability of a sample belonging a cognitive task, for example: o_7 returns the probability of the given input graph belonging to cognitive task y_7, $p(y_k = 7|W_k, V_1, V_2)$. We represent the weight matrix between the input layer and the hidden layer as V_1 and the weight matrix between the hidden layer and the output layer with V_2.

holds the posterior probabilities of C classes. V_1 and V_2 are the weight matrices of the architecture. The architecture is shown in Figure 5.1.

5.2.1 Training of MLP

The training process of the employed MLP is similar to the training process of the two layered artificial neural networks that we defined in Chapter 3 and Chapter 4. However, there is an extra layer in the MLP architecture and therefore there are two weight matrices, V_1 and V_2, to be optimized. At an epoch of training, we update an

weight matrix, V_l, as:

$$V_l \leftarrow V_l - \alpha_2 \frac{\partial Loss_{MLP}(\mathbf{y}^{(k)}, \hat{\mathbf{y}}^{(k)}, W_k; V_1, V_2)}{\partial V_l} \tag{510}$$

where α_2 is the learning rate of the gradient descent algorithm. During an epoch, we update the weight matrix of every layer using Equation (510).

We use a batch of brain networks for training MLP for a quicker convergence of the model. Therefore, the update rule is applied with the adjacency matrices and cognitive tasks of a batch of brain networks and the weights of the neural networks are penalized for each misclassified sample of the batch

5.2.2 Classification of Brain Networks using MLP

We study the learning capability of MLPs in brain decoding. Therefore, we train the MLP structure that we defined in the previous section on the brain networks that we extracted using elastic loss. We aim to have an MLP with the optimal hyperparameters that will surpass the classification accuracy obtained with SVM. Classification accuracy of the MLP with brain networks created with various regularization hyperparameters, λ_1 and λ_2, will also guide us to the optimal hyperparameters of brain network creation process.

We repeat the across subjects experiments that we conducted using SVM on HCP dataset in Section 5.1.1.1. We apply $k = 3$ fold cross validation on the brain networks of the all 807 subjects. In each fold, we retain the brain networks 538 subject for classification of the MLP and the brain networks of the remaining 269 subjects as the test split in order to observe the classification accuracy of the model. Table 5.5 and Table 5.6, show the across subject performances of directed and undirected brain networks, respectively.

Table 5.5: MLP classification accuracy on across subject experiments of directed brain networks created using elastic loss on HCP Dataset.

	$\lambda_2 = 1$		$\lambda_2 = 2$		$\lambda_2 = 4$		$\lambda_2 = 8$	
λ_1	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	**87.95**	0.38	**88.02**	0.50	**87.91**	0.38	**87.17**	0.48
0.2	84.52	0.57	85.31	0.59	85.78	0.55	85.53	0.44
0.4	81.23	0.66	82.41	0.73	82.66	0.52	82.79	0.37
0.6	77.92	0.78	78.92	0.50	79.49	0.58	79.66	0.49
0.8	74.81	0.78	75.92	0.48	76.83	0.63	76.81	0.42
1.0	71.75	0.82	72.39	0.57	73.29	0.80	73.69	0.85
2.0	62.26	1.25	62.96	1.11	63.34	1.33	63.77	0.98
4.0	51.92	0.92	51.87	0.64	51.77	0.64	52.40	1.03
8.0	36.00	0.13	36.12	0.48	36.13	0.63	36.78	0.36

Table 5.6: MLP classification accuracy on across subject experiments of undirected brain networks created using elastic loss on HCP Dataset.

	$\lambda_2 = 1$		$\lambda_2 = 2$		$\lambda_2 = 4$		$\lambda_2 = 8$	
λ_1	Mean	Std	Mean	Std	Mean	Std	Mean	Std
0	**87.66**	0.36	**88.15**	0.43	**87.93**	0.40	**87.24**	0.49
0.2	84.08	0.69	85.23	0.66	85.51	0.58	85.59	0.60
0.4	80.62	0.74	81.90	0.51	82.66	0.56	82.43	0.50
0.6	77.62	0.58	78.67	0.68	79.36	0.55	79.59	0.68
0.8	74.74	0.45	75.97	0.48	76.66	0.54	76.87	0.43
1.0	71.57	0.41	72.79	0.58	73.22	0.91	73.43	0.75
2.0	61.43	0.99	61.58	0.60	61.89	0.81	62.51	1.11
4.0	47.54	0.42	47.50	0.52	48.17	0.90	47.75	0.44
8.0	32.44	0.49	32.71	0.12	32.55	0.92	32.62	0.52

Table 5.5 and 5.6 show the across subject performance of directed and undirected brain networks created using Elastic Loss. The best classification accuracy using MLP is 88.02% with directed brain networks and 88.15% with undirected brain networks. The accuracy obtained with SVM are 84.10% and 84.06 in the same order. Using MLP, we obtain 4% higher performance compared to SVM.

Observing the results of within subject experiments conducted with SVM (Table 5.1 and Table 5.2), across subject experiments conducted with SVM (Table 5.3 and Table 5.4) and across subject experiments conducted with MLP (Table 5.5 and Table 5.6), we see that the $L1$ regularization hurts the classification accuracy even with small λ_1 for every λ_2 values and in each directed and undirected cases. So, in the rest of this study, we use ridge regression function as the default loss function in the process of brain network estimation.

5.3. Parallel Artificial Brain Networks

In Chapter 3, we estimated brain graphs for each time window of each subject. For each graph estimation, we constructed a neural from scratch and trained it for a certain number of epochs. After creating brain graphs for a subject, we repeated the procedure for all the other subject, hyperparameters and both of the weighted graph types. This approach is very time consuming and we cannot integrate such a model to an end to end architecture because of its asynchronous outputs.

We can, however, redefine the Artificial Brain Network architecture, its input and output in a way that it will work with 3 dimensional matrix operations instead of 2 dimensional ones. This will give us the power to create brain graphs in parallel. Since the neural networks we use are not deep nor wide, we can fit certain number of them to Graphical Processing Units and expect the whole process to run faster.

In Chapter 3, in order to create a single brain network within a time interval win_k with length L we defined a chunk $B_k \in R^{L \times M}$ consisting of the representative activations of $M = 90$ brain regions. In this section, we aim to estimate brain networks for all N time intervals, $win_{all} = [win_1, win_2 \dots win_N]$, simultaneously. In order to achieve this goal, we concatenate the defined chunks into a 3 dimensional tensor

$X = [B_1, B_2 \dots B_N]$ where $X_{k,:,:} = B_k$. Algorithm 5.1 defines the process of input preparation.

Algorithm 5.1: Algorithm to prepare input tensor for a subject, X.

Process `PrepareX(`*subject_data*`)`**:**

windows $\leftarrow$ divideToWindows (subject_data) ;
for *win in windows* **do**
  $B_{win} \leftarrow$ subject data within time interval *win* ;
  $X_{win,:,:} \leftarrow B_{win}$;
end
return X ;

We follow the same approach for the adjacency matrices of the brain networks to be estimated. We define the weight tensor $\overline{W} \in R^{N \times M \times M}$ as the connection of adjacency matrices estimated for each window, $\overline{W} = [W_1, W_2 \dots W_N]$ where $\overline{W}_{k,:,:} = W_k$. We describe the initialization of the tensor $\overline{W}$ with a pseudocode in Algorithm 5.2.

Algorithm 5.2: Process of Initializing the weight tensor for subject s, $\overline{W}$. Elements of random matrices are sampled from a normal distribution ($\mathcal{N}(\mu = 0, \sigma^2 = 1)$).

Process `InitalizeW(`*subject_data*`)`**:**

windows $\leftarrow$ divideToWindows(subject_data) ;
for *win in windows* **do**
  $\overline{W}_{win,:,:} \leftarrow$ randomly assign an $M \times M$ matrix with 0 diagonal. ;
  if *undirected* **then**
    $\overline{W}_{win,:,:} \leftarrow \frac{1}{2}(\overline{W}^T_{win,:,:} + \overline{W}_{win,:,:})$;
  end
end
return $\overline{W}$;

The output of the proposed model $O \in R^{N \times L \times M}$ consists of estimation of anatomic brain region's representative activations during all of the time interval, such that, $O_{k,:,i}$ is the estimations for region i during time window k, $O_{k,:,i} = [\hat{b}_{i,t}, \hat{b}_{i,t+1} \dots \hat{b}_{i,t+L}]$.

We define the forward pass of the model as the following:

$$O = X \odot \overline{W}^T \tag{511}$$

where the operator $\odot$ defines matrix multiplication of sub-matrices of left-hand and right-hand tensors, such that the sub-matrix $O_{i,:,:}$ is the result of matrix multiplication of $X_{i,:,:}$ and $\overline{W}^T_{i,:,:}$. We define transpose of $\overline{W}$, $\overline{W}^T$ as the concatenation of transpose of every sub-matrix of $\overline{W}$, $\overline{W}^T = [\overline{W}^T_{1,:,:}, \overline{W}^T_{2,:,:} \dots \overline{W}^T_{N,:,:}]$.

We measure the error of the estimation O with ridge regression loss function we defined in Equation (52). In order to update the weight tensor, we use gradient descent algorithm. We calculate the gradient of the loss function with respect weight tensor as:

$$\nabla \overline{W} = (-2(X - O) \odot X)/L + 2\lambda_2 W \tag{512}$$

where λ_2 is $L2$ regularization hyperparameter. In order to estimate directed brain networks, we use the following update rule at each epoch of the training process:

$$\overline{W} \leftarrow \overline{W} - \alpha_1 \nabla \overline{W} \tag{513}$$

where α is learning rate of the method. For estimation of undirected brian networks, we use the following trick to sustain the symmetry of each adjacency matrix :

$$\overline{W} \leftarrow \overline{W} - \frac{1}{2}\alpha_1(\nabla \overline{W} + \nabla \overline{W}^T) \tag{514}$$

After training the model for a number of epoch, we obtain a set of adjacency matrices.

The process of training the model for an epoch is summarized in Algorithm 5.3.

Algorithm 5.3: One epoch of training PABN model.

Process `PABN(` X, $\overline{W}$ `)` **:**

$O = X \odot \overline{W}^T$;

$\nabla\overline{W} = (-2(X - O) \odot X)/L + 2\lambda_2\overline{W}$;

if *undirected* **then**

$\nabla\overline{W} \leftarrow \frac{1}{2}(\nabla\overline{W} + \nabla\overline{W}^T)$;

end

$\overline{W} \leftarrow \overline{W} - \alpha_1\nabla\overline{W}$;

return $\overline{W}$;

5.4. End to End Model

In this thesis, our approach toward brain decoding has been constructing brain networks by training an artificial neural network and separately run a classification algorithm on the constructed brain networks to predict their underlying cognitive task.

In Chapter 3, we have shown that our proposed brain networks, ABN, outperform the brain networks estimated with the state of the art methods. We also observed the effect of various regularization parameters on the classification performance of ABN in Section 5.1.

In Section 5.2, we used multilayer perceptrons for classification of brain networks and showed that the classification accuracy we have with MLP is greater than the accuracy we have with SVM for any λ_1 and λ_2 selection. The MLP architecture we employed takes a batch of brain networks for training, it optimizes its weights using the loss caused by all of the brain network in the batch.

In Section 5.3, we defined an approach to estimate multiple brain networks simultaneously. Beside having a shorter running time, we could define the output of the proposed method as a batch of input to MLP.

In this section, we unite the best classification model and best brain network estimation model to our experience in a single end to end artificial neural network archi-

tecture. For this purpose, we concatenate parallel brain network architecture and the MLP architecture we employed for classification in Section 5.2. Using this architecture, we aim to obtain brain networks that are both reconstructive and informative of their cognitive tasks. We show a sample representation of the proposed architecture in Figure 5.2.

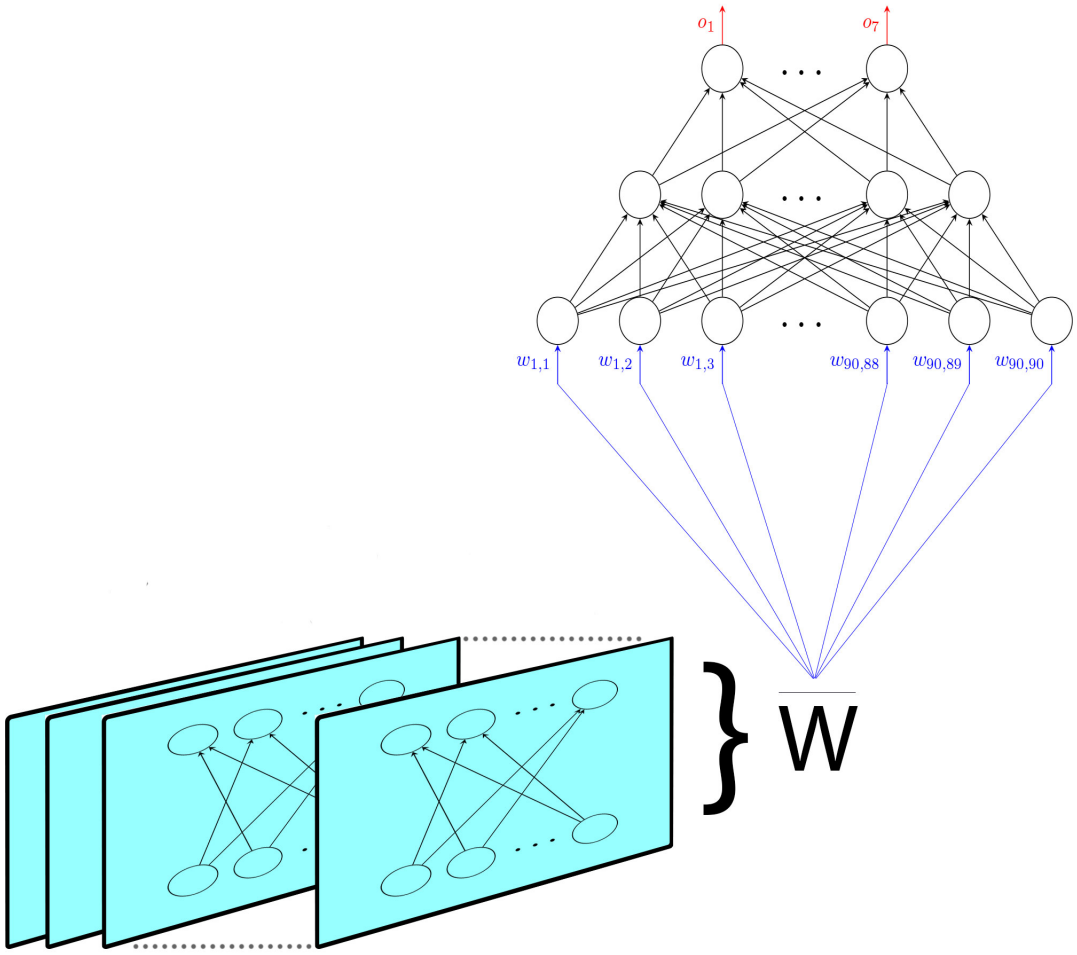


Figure 5.2: The proposed end to end architecture is a combination of PABN and MLP architectures.

Similar to PABN architecture, the end to end architecture also takes a set of fMRI data chunks in the form a three dimensional tensor, $X = [B_1, B_2 \dots B_N]$. In return, the output of the architecture will be class posterior probabilities for each data chunk and

also the reconstruction loss of each brain network estimated from the data chunks. In the following subsection, we describe the principals and the training process of the architecture.

5.4.1 Loss Function of the End to End Model

Both parallel artificial brain network architecture and MLP are neural network architectures and they can be trained using gradient descent method. However, we employed these architectures for different objectives, our two layered brain network extraction model aims to reconstruct its input on the output and MLP applies a classification on its input. Mathematically, we define the following loss function for the parallel artificial brain networks model:

$$Loss_{PABN}(X;\overline{W}) = \frac{1}{L}\sum_{win_k}\sum_{i=1}^{M}\sum_{t=1}^{L}(X_{k,t,i} - O_{k,t,i})^2 + \lambda\sum_{win_k}\sum_{i}^{M}\sum_{j}^{M}\overline{W}_{k,i,j}^2 \quad (515)$$

where X is input tensor, $\overline{W}$ is weight tensor and O is output of the model. The loss function of an MLP with weight matrices V_1, V_2 that does C class classification on a batch of N brain networks $\overline{W}$, the loss function is defined as:

$$Loss_{MLP}(Y,\hat{Y},\overline{W};V_1,V_2) = -\frac{1}{N}\sum_{k=1}^{N}\sum_{c=1}^{C}Y_{k,c}log(\hat{Y}_{k,c}) \quad (516)$$

where matrix $Y \in R^{N,C}$ holds one hot binary vectors of N instances, i.e. if $Y_{k,c}$ is 1 the class of batch k is c. Similarly, $\hat{Y}$ is a one hot binary representation of MLP's estimated classes. The matrices V_1 and v_2 are the inner weight matrices of MLP. Note that the weights of PABN, W, is now an input to MLP.

We combine the architectures PABN and MLP into an end to end model as shown in Fig. 5.2. We define a common loss function for PABN and MLP:

$$Loss_{E2E}(X;\overline{W},V_1,V_2) = \theta Loss_{PABN}(X;\overline{W}) + (1-\theta)Loss_{MLP}(Y,\hat{Y},\overline{W}) \quad (517)$$

where θ is the hyperparameter to balance the effect of the loss functions of PABN and MLP on the total loss.

5.4.2 Training the End to End Model

In the end to end architecture, there are two weight matrices belonging to MLP part, V_1 and V_2, and also a weight tensor, $\overline{W}$, for each subject to be optimized. We apply gradient descent algorithm on these weights in order to minimize end to end loss function, Equation (517). We update the weights of the model for a number of epochs until the loss function reaches its lowest value.

With our experience with ABNs in Section 3, we know that estimation of brain networks could takes several a big number of epochs of training. Therefore, we expect the brain networks that PABN part of the architecture estimates during the training epochs would be incomplete and during the training process, the nature of the data supplied to the MLP part would change drastically. Therefore, at the beginning of each training epoch, we create a new MLP by simply re-initializing its weights.

In order to prepare the input tensors to the end to end model, we take the same approach that we took in PABN architecture, Algorithm 5.1. For each subject of the concerning fMRI dataset, we form an input batch, $X^{(s)}$, and we use every batch of data from all the subjects, $X_{all} = [X^{(1)}, X^{(2)} \dots X^{(S)}]$. Similarly, we also create a set of weight tensors for all subjects, $\overline{W}_{all} = [\overline{W}^{(1)}, \overline{W}^{(1)} \dots \overline{W}^{(S)}]$.

During an epoch of training the model, we use input tensors from each subject in turn to update the weights of the model. For a subject s, the input tensor $X^{(s)}$ is used as input to the model for updating both the tensor $\overline{W}^{(s)}$, V_1, and V_2. The training process

is described as a pseudocode in Algorithm 5.4.

Algorithm 5.4: Initialize input tensor

Process `End2End(` *subjects*`)` **:**

$X_{all} = \{\}, \overline{W}_{all} = \{\}$;

for *subject* s *in subjects* **do**

subject_data $\leftarrow$ load_subject_data(s) ;

$X^{(s)} \leftarrow$ prepareX(subject_data) ;

$\overline{W}^{(s)} \leftarrow$ InitalizeW(subject_data) ;

X_all.add($X^{(s)}$), W_all.add($\overline{W}^{(s)}$)

end

for *epoch in* $[1, 2, \#epochs]$ **do**

Initialize MLP weights V_1 and V_2 ;

for *subject s in subjects* **do**

Update V_1 ;

Update $\overline{W}^{(s)}$, $\overline{W}^{(s)} \leftarrow \overline{W}^{(s)} - \alpha_1 \frac{\partial Loss_{E2E}(X;\overline{W}^{(s)},V_1,V_2)}{\partial \overline{W}^{(s)}}$;

Update v_1, $V_1 \leftarrow V_1 - \alpha_2 \frac{\partial Loss_{E2E}(X;\overline{W}^{(s)},V_1,V_2)}{\partial \overline{V_2}}$;

Update v_2, $V_2 \leftarrow V_2 - \alpha_2 \frac{\partial Loss_{E2E}(X;\overline{W}^{(s)},V_1,V_2)}{\partial \overline{V_2}}$;

end

end

5.4.3 Experiments Conducted with the End to End Model

In this section, we describe the process of the experiments we conduct with the end-to-end model. In our experiments, we use HCP fMRI dataset. The dataset consist of 807 subjects and 7 cognitive tasks.

We created a batch for each of the subjects. Therefore, we have $S = 807$ batches, $X_{all} = [X_1, X_2 \dots X_{807}]$, 807 label matrices, $Y_{all} = [Y_1, Y_2 \dots Y_{807}]$, and $S = 807$ weight tensors to estimate $W_{all} = [\overline{W}_1, \overline{W}_2 \dots \overline{W}_{807}]$.

We reserve the first 269 batches for testing of the model and remaining 538 for training of the model. The PABN weights of a test fold, $\overline{W}_k$, is trained using only loss function of PABN in order to avoid bias. For the training fold, we use end-to-end loss

for training.

We use $\alpha_1 = 5 * 10^{-3}$ for the learning rate of PABN part and $\alpha_2 = 10^{-4}$ for the learning rate of MLP part. We apply $L2$ regularization on PABN loss where the hyperparameter is $\lambda_2 = 4$. We train the model for $\#epochs = 3000$ epochs.

We can create both directed and undirected brain networks using the proposed end to end model, however during our experiments we concentrated on creation of only directed brain networks. The model can be modified to keep the symmetry of adjacency matrices during training in order to create undirected brain networks.

Our first experiment is to observe the classification performance of the model when $\theta = 1.0$ in Equation (517). For $\theta = 1.0$, the brain networks are created using only the loss of PABN. The performance we obtain with these settings will be a baseline for our remaining experiments.

We have tested the effect of various θ values on the classification performance of the model. In Table 5.7, we show the performance of the model under various θ values.

Table 5.7: End to end performance on HCP, directed, $\lambda = 4$ (Across)..

	$\theta = 0.33$	$\theta = 0.83$	$\theta = 0.90$	$\theta = 1.0$
Accuracy (%)	86.32	87.10	86.90	86.10

Table 5.7 shows that the baseline performance we aim to surpass is 86.10 when $\theta = 1.0$. When we choose $\theta = 0.83$, the performance is the greatest among all results. The table implies more discriminative brain networks can be created if the feedback of MLP is considered during the brain network estimation process.

5.5. Chapter Discussion

With our experiments, we have shown that the classification accuracy of brain networks as a result of using ridge regression loss function is greater than the classification accuracy caused by elastic loss function . We assumed that this is a result of

$L1$ regularization part of elastic loss. $L1$ regularization, however, is used for weight sparsification for artificial neural networks. Therefore, the brain networks we obtained employing $L1$ regularization could be sparser and it may help us discarding unnecessary connections in the brain networks.

In this chapter, we employed an MLP architecture for classification of brain networks. Having superior classification accuracy, we also integrate the architecture to the end to end model. However, in recent years, Graph Convolutional Networks (GCN) becoming very popular for the purpose categorizing graphical structure [6, 69]. Compared to MLP, training a GCN is more challenging. However, the architecture could be more suitable for the task in hand since it is specifically designed for processing graphical inputs. We leave employing a GCN for classification of brain networks as a future work.

We designed PABN architecture for two purposes: compatibility and speed. We have shown that the architecture is compatible with the MLP architecture, since it produces a batch of input and it could be fed to the MLP. However, we have not done any analysis on the elapsing time of the brain network estimation process using the architecture. We leave comparison of the speed of ABN and PABN architectures as a future work.

The proposed end to end architecture has been tested for a very specific and limited subset of hyperparameters since training the model is time consuming and very expensive due to big number of matrices that need to be optimized. It is possible to have a greater performance with a different $L2$ regularization hyperparameter, λ_2, than we obtained in Subsection 5.4.3.

5.6. Chapter Conclusion

In this chapter, we studied the effect of various loss function on the classification performances of brain networks. We have shown that the ridge regression function results in better classification performance than elastic loss function. We have replaced SVM with MLP for classification and shown that they can outperform SVM on brain decoding. We have proposed a new approach for creating multiple brain networks

faster and simultaneously. Finally, we have proposed an end-to-end model in order to conduct brain network creation and classification with a single architecture and to replace the classical methods which do these two tasks separately. We have shown that the end to end model has potential to provide the state-of-the-art classification accuracy in brain decoding using brain networks.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we introduce novel methods to estimate brain networks to represent the cognitive states of human brain, recorded by fMRI techniques. The edge weights of the suggested brain networks are estimated using various architectures of artificial neural networks. Each of the brain network models serve to different purposes, such as, analysis of human brain connectome with respect to subjects and tasks, decoding the cognitive states, etc.

The first model, called Artificial Brain Network, estimates the edge weights between the anatomical regions by training a simple 2-layer neural network where the input is the vector of time series recorded from a subject, during the time course of the same cognitive process and the loss function is the sum of squared error of Ridge regression. The suggested network successfully estimates the edge weights of the network and decodes the underlying cognitive task with a performance of $85 - 95\%$. Compared to the state of the art brain decoding methods, the suggested model has a higher representation power for subject specific brain decoding tasks. Furthermore, the suggested model is capable to estimate both directed and undirected brain networks. To our knowledge this method is the first model which computes the the weights of the brain network as the weights of artificial the neural networks. We experimentally show that the suggested artificial neural networks estimates better weights compared to the Pearson correlation and the closed form solutions of the Ridge regression.

The second model, called Task-Specific Networks, estimated a brain network for a predefined task across all the subjects. This approach is very crucial for investigating the shared properties of a cognitive task, under consideration. On the contrary of the previous approach the task-specific brain network enables us to represent a cognitive

task by a network which shows the functional connectivities among the anatomic regions. To our knowledge, this is the first model which learns a generic brain network across the subjects, for a particular cognitive task. All of the previous approaches created subject-level brain networks as there was no suitable computational model to estimate task specific brain networks.

The third model combines the brain network estimation and decoding steps under the same artificial neural network architecture with an end to end model. As a result, optimization of the loss functions for brain network estimation and brain decoding is accomplished, simultaneously, under the same artificial neural network architecture. Although the idea of combining a two step brain decoding method under an end-to-end model was quite good, we could not catch a better performances compared to the sequential methods, developed in this thesis. This approach needs to be further studied for hyperparameter optimization and selecting a better oveall cost function.

The representation power of the suggested brain networks are examined by measuring the brain decoding performances in two fMRI data sets, namely HCP and/or TOL data sets. The performances of the brain decoding is higher in all of the three models compared to the state of the art brain decoding methods.

Furthermore, we experimentally show that directed brain networks resemble each other, indicating that the relationships among the anatomic regions are symmetrical. In other words, if there is a significant connectivity among two anatomic regions this connectivity is significant in both directions. Therefore, one can conclude that an undirected brain network representation is sufficient to study the network properties of human brain.

6.1. Future Work

In this thesis, the validity of the estimated brain networks are analyzed through the performances of brain decoding, measured in TOL and HCP datasets. However, the suggested brain network estimation methods are not limited to solve the brain decoding problem.It can be easily be applicable to study various neurological diseases, sleep disorders and properties of resting state fMRI data.

We tested our first model, ABNs, on two datasets, yet our two other methods have not been tested on any datasets other than HCP. We left task specific brain networks creation for TOL as a future work. We also plan to apply MLP classification and end to end model to TOL dataset.

We took the first step towards the end to end model classification model. We have only built directed brain networks with the model, yet we can extend it for undirected graphs. We have shown that we obtain better classification performance when we employ derivatives from both generative and discriminative networks. However, the classifier we used was trained for only one epoch for timing efficiency and therefore the test accuracy does not surpass the raw MLP performance. We plan to conduct a wider experiment on the end to end model to overcome this problem.

REFERENCES

[1] T. D. Wager and M. A. Lindquist, "Principles of fmri," *Leanpub*, 2015.

[2] M. J. Hoptman, X.-N. Zuo, P. D. Butler, D. C. Javitt, D. D'Angelo, C. J. Mauro, and M. P. Milham, "Amplitude of low-frequency oscillations in schizophrenia: a resting state fmri study," *Schizophrenia research*, vol. 117, no. 1, pp. 13–20, 2010.

[3] W. Zhou, Z. Xia, Y. Bi, and H. Shu, "Altered connectivity of the dorsal and ventral visual regions in dyslexic children: a resting-state fmri study," *Frontiers in human neuroscience*, vol. 9, p. 495, 2015.

[4] M. Schurz, H. Wimmer, F. Richlan, P. Ludersdorfer, J. Klackl, and M. Kronbichler, "Resting-state and task-based functional brain connectivity in developmental dyslexia," *Cerebral Cortex*, vol. 25, no. 10, pp. 3502–3514, 2014.

[5] J. A. Berken, X. Chai, J.-K. Chen, V. L. Gracco, and D. Klein, "Effects of early and late bilingualism on resting-state functional connectivity," *Journal of Neuroscience*, vol. 36, no. 4, pp. 1165–1172, 2016.

[6] S. Arslan, S. I. Ktena, B. Glocker, and D. Rueckert, "Graph saliency maps through spectral convolutional networks: Application to sex classification with brain connectivity," *arXiv preprint arXiv:1806.01764*, 2018.

[7] M. Ozay, I. Öztekin, U. Öztekin, and F. T. Y. Vural, "Mesh learning for classifying cognitive processes," *arXiv preprint arXiv:1205.2382*, 2012.

[8] S. D. Newman, J. A. Greco, and D. Lee, "An fmri study of the tower of london: a look at problem structure differences," *Brain research*, vol. 1286, pp. 123–132, 2009.

[9] I. Kovelman, S. A. Baker, and L.-A. Petitto, "Bilingual and monolingual brains compared: a functional magnetic resonance imaging investigation of syntactic

processing and a possible “neural signature” of bilingualism,” *Journal of cognitive neuroscience*, vol. 20, no. 1, pp. 153–169, 2008.

[10] R. A. Poldrack and K. J. Gorgolewski, “Openfmri: Open sharing of task fmri data,” *Neuroimage*, vol. 144, pp. 259–261, 2017.

[11] T. Shallice, “Specific impairments of planning,” *Phil. Trans. R. Soc. Lond. B*, vol. 298, no. 1089, pp. 199–209, 1982.

[12] Y.-K. Chang, C.-L. Tsai, T.-M. Hung, E. C. So, F.-T. Chen, and J. L. Etnier, “Effects of acute exercise on executive function: a study with a tower of london task,” *Journal of Sport and Exercise Psychology*, vol. 33, no. 6, pp. 847–865, 2011.

[13] W. K. Berg, D. L. Byrd, J. P. McNamara, and K. Case, “Deconstructing the tower: Parameters and predictors of problem difficulty on the tower of london task,” *Brain and Cognition*, vol. 72, no. 3, pp. 472–482, 2010.

[14] A. R. Hariri, A. Tessitore, V. S. Mattay, F. Fera, D. R. Weinberger, *et al.*, “The amygdala response to emotional stimuli: a comparison of faces and scenes,” *Neuroimage*, vol. 17, no. 1, pp. 317–323, 2002.

[15] M. R. Delgado, L. E. Nystrom, C. Fissell, D. Noll, and J. A. Fiez, “Tracking the hemodynamic responses to reward and punishment in the striatum,” *Journal of neurophysiology*, vol. 84, no. 6, pp. 3072–3077, 2000.

[16] J. R. Binder, W. L. Gross, J. B. Allendorfer, L. Bonilha, J. Chapin, J. C. Edwards, T. J. Grabowski, J. T. Langfitt, D. W. Loring, M. J. Lowe, *et al.*, “Mapping anterior temporal lobe language areas with fmri: a multicenter normative study,” *Neuroimage*, vol. 54, no. 2, pp. 1465–1475, 2011.

[17] R. L. Buckner, F. M. Krienen, A. Castellanos, J. C. Diaz, and B. T. Yeo, “The organization of the human cerebellum estimated by intrinsic functional connectivity,” *American Journal of Physiology-Heart and Circulatory Physiology*, 2011.

[18] R. Smith, K. Keramatian, and K. Christoff, “Localizing the rostrolateral prefrontal cortex at the individual level,” *Neuroimage*, vol. 36, no. 4, pp. 1387–1396, 2007.

[19] F. Castelli, F. Happé, U. Frith, and C. Frith, "Movement and mind: a functional imaging study of perception and interpretation of complex intentional movement patterns," *Neuroimage*, vol. 12, no. 3, pp. 314–325, 2000.

[20] T. E. Lund, K. H. Madsen, K. Sidaros, W.-L. Luo, and T. E. Nichols, "Non-white noise in fmri: does modelling have an impact?," *Neuroimage*, vol. 29, no. 1, pp. 54–66, 2006.

[21] A. Afrasiyabi, I. Onal, and F. T. Y. Vural, "Effect of voxel selection on temporal mesh model for brain decoding," in *Signal Processing and Communication Application Conference (SIU), 2016 24th*, pp. 2249–2252, IEEE, 2016.

[22] T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman, "Learning to decode cognitive states from brain images," *Machine learning*, vol. 57, no. 1-2, pp. 145–175, 2004.

[23] J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini, "Distributed and overlapping representations of faces and objects in ventral temporal cortex," *Science*, vol. 293, no. 5539, pp. 2425–2430, 2001.

[24] G. R. Iversen, A. R. Wildt, H. Norpoth, and H. P. Norpoth, *Analysis of variance*. No. 1, Sage, 1987.

[25] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Aaai*, vol. 2, pp. 129–134, 1992.

[26] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with relieff," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997.

[27] M. Robnik-Šikonja and I. Kononenko, "An adaptation of relief for attribute estimation in regression," in *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, vol. 5, pp. 296–304, 1997.

[28] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[29] N. Kriegeskorte, R. Goebel, and P. Bandettini, "Information-based functional brain mapping," *Proceedings of the National Academy of Sciences*, vol. 103, no. 10, pp. 3863–3868, 2006.

[30] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*, pp. 1094–1096, Springer, 2011.

[31] S. M. Smith, A. Hyvärinen, G. Varoquaux, K. L. Miller, and C. F. Beckmann, "Group-pca for very large fmri datasets," *Neuroimage*, vol. 101, pp. 738–749, 2014.

[32] S. Arslan, S. I. Ktena, A. Makropoulos, E. C. Robinson, D. Rueckert, and S. Parisot, "Human brain mapping: A systematic comparison of parcellation methods for the human cerebral cortex," *Neuroimage*, 2017.

[33] G. Flandin, F. Kherif, X. Pennec, G. Malandain, N. Ayache, and J.-B. Poline, "Improved detection sensitivity in functional mri data using a brain parcelling technique," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 467–474, Springer, 2002.

[34] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, "Predicting human brain activity associated with the meanings of nouns," *science*, vol. 320, no. 5880, pp. 1191–1195, 2008.

[35] R. C. Craddock, G. A. James, P. E. Holtzheimer III, X. P. Hu, and H. S. Mayberg, "A whole brain fmri atlas generated via spatially constrained spectral clustering," *Human brain mapping*, vol. 33, no. 8, pp. 1914–1928, 2012.

[36] H. Mogultay and F. T. Y. Vural, "Brainparcel: A brain parcellation algorithm for cognitive state classification," in *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities*, pp. 32–42, Springer, 2018.

[37] M. Van Den Heuvel, R. Mandl, and H. H. Pol, "Normalized cut group clustering of resting-state fmri data," *PloS one*, vol. 3, no. 4, p. e2001, 2008.

[38] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, "Automated anatomical labeling of

activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain," *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002.

[39] Y. Liu, M. Liang, Y. Zhou, Y. He, Y. Hao, M. Song, C. Yu, H. Liu, Z. Liu, and T. Jiang, "Disrupted small-world networks in schizophrenia," *Brain*, vol. 131, no. 4, pp. 945–961, 2008.

[40] E. S. Finn, X. Shen, J. M. Holahan, D. Scheinost, C. Lacadie, X. Papademetris, S. E. Shaywitz, B. A. Shaywitz, and R. T. Constable, "Disruption of functional networks in dyslexia: a whole-brain, data-driven analysis of connectivity," *Biological psychiatry*, vol. 76, no. 5, pp. 397–404, 2014.

[41] O. Fırat, M. Özay, I. Önal, İ. Öztekiny, and F. T. Y. Vural, "Functional mesh learning for pattern analysis of cognitive processes," in *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2013 12th IEEE International Conference on*, pp. 161–167, IEEE, 2013.

[42] I. Onal Ertugrul, M. Ozay, and F. T. Y. Vural, "Gender classification using mesh networks on multiresolution multitask fmri data," *Brain imaging and behavior*, pp. 1–17, 2019.

[43] C. J. Brown, J. Kawahara, and G. Hamarneh, "Connectome priors in deep neural networks to predict autism," in *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, pp. 110–113, IEEE, 2018.

[44] J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville, "Decoding brain states from fmri connectivity graphs," *Neuroimage*, vol. 56, no. 2, pp. 616–626, 2011.

[45] P. Vaidyanathan, "The theory of linear prediction," *Synthesis lectures on signal processing*, vol. 2, no. 1, pp. 1–184, 2007.

[46] N. Levinson, "The wiener rms error criterion in filter design and prediction, appendix b of wiener, n.(1949)," *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, 1949.

[47] J. Durbin, "The fitting of time-series models," *Revue de l'Institut International de Statistique*, pp. 233–244, 1960.

[48] P. Delsarte and Y. Genin, "The split levinson algorithm," *IEEE transactions on acoustics, speech, and signal processing*, vol. 34, no. 3, pp. 470–478, 1986.

[49] I. Onal, M. Ozay, and F. T. Y. Vural, "Modeling voxel connectivity for brain decoding," in *Pattern Recognition in NeuroImaging (PRNI), 2015 International Workshop on*, pp. 5–8, IEEE, 2015.

[50] I. Onal, M. Ozay, and F. T. Y. Vural, "Functional mesh model with temporal measurements for brain decoding," in *Engineering in medicine and biology society (EMBC), 2015 37th annual international conference of the IEEE*, pp. 2624–2628, IEEE, 2015.

[51] I. Onal, M. Ozay, E. Mizrak, I. Oztekin, and F. T. Y. Vural, "A new representation of fmri signal by a set of local meshes for brain decoding," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 4, pp. 683–694, 2017.

[52] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[53] J. Geweke, "Measurement of linear dependence and feedback between multiple time series," *Journal of the American statistical association*, vol. 77, no. 378, pp. 304–313, 1982.

[54] C. Bernasconi and P. KoÈnig, "On the directionality of cortical interactions studied by structural analysis of electrophysiological recordings," *Biological cybernetics*, vol. 81, no. 3, pp. 199–210, 1999.

[55] H. Liang, M. Ding, R. Nakamura, and S. L. Bressler, "Causal influences in primate cerebral cortex during visual pattern discrimination," *Neuroreport*, vol. 11, no. 13, pp. 2875–2880, 2000.

[56] A. Roebroeck, E. Formisano, and R. Goebel, "Mapping directed influence over the brain using granger causality and fmri," *Neuroimage*, vol. 25, no. 1, pp. 230–242, 2005.

[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[58] S. Koyamada, Y. Shikauchi, K. Nakae, M. Koyama, and S. Ishii, "Deep learning of fmri big data: a novel approach to subject-transfer decoding," *arXiv preprint arXiv:1502.00093*, 2015.

[59] S. M. Plis, D. R. Hjelm, R. Salakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. J. Johnson, J. S. Paulsen, J. A. Turner, and V. D. Calhoun, "Deep learning for neuroimaging: a validation study," *Frontiers in neuroscience*, vol. 8, p. 229, 2014.

[60] J. Kawahara, C. J. Brown, S. P. Miller, B. G. Booth, V. Chau, R. E. Grunau, J. G. Zwicker, and G. Hamarneh, "Brainnetcnn: convolutional neural networks for brain networks; towards predicting neurodevelopment," *NeuroImage*, vol. 146, pp. 1038–1049, 2017.

[61] X. Li, Y. Li, and X. Li, "Predicting clinical outcomes of alzheimer's disease from complex brain networks," in *International Conference on Advanced Data Mining and Applications*, pp. 519–525, Springer, 2017.

[62] A. Alchihabi, B. B. Kivilicim, O. Ekmekci, S. D. Newman, and F. T. Y. Vural, "Decoding cognitive subtasks of complex problem solving using fmri signals," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.

[63] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[64] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, no. 3, p. 186, 2009.

[65] M. Xia, J. Wang, and Y. He, "Brainnet viewer: a network visualization tool for human brain connectomics," *PloS one*, vol. 8, no. 7, p. e68910, 2013.

[66] B. Nasir and F. T. Y. Vural, "Cerebra: A 3-d visualization tool for brain network extracted from fmri data," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 1147–1150, IEEE, 2016.

[67] S. Ryali, T. Chen, K. Supekar, and V. Menon, "Estimation of functional connectivity in fmri data using stability selection-based sparse partial correlation with elastic net penalty," *NeuroImage*, vol. 59, no. 4, pp. 3852–3861, 2012.

[68] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich, "Network modelling methods for fmri," *Neuroimage*, vol. 54, no. 2, pp. 875–891, 2011.

[69] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.