DEVELOPMENT OF A NEW COMPUTATIONAL MODEL FOR MAPPING, LEARNING AND MINING OF 3D SPATIO-TEMPORAL fMRI DATA

NORHANIFAH BINTI MURLI

A Thesis submitted to
Auckland University of Technology
in fulfilment of the requirements for the degree of
Doctor of Philosophy (PhD)

2015

Knowledge Engineering and Discovery Research Institute

Faculty of Design and Creative Technologies

Primary Supervisor: Prof. Nikola Kasabov

Secondary Supervisor: Dr. Russel Pears

CONTENTS

DECLARATION

I hereby declare that the submission is my own and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extend has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Auckland, 2015

Norhanifah binti Murli

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3D	three dimensional
AER	Address Event Representation
ANNs	Artificial Neural Networks
BCI	Brain Computer Interface
BD	Bilateral Dorsolateral
BOLD	Blood Oxygen Level Dependent
BSA	Ben's Spike Algorithm
CALC	Calcarine Sulcus
CMU	Carnegie Mellon University
CPU	Central Processing Unit
CT	Computed Tomography
deSNN	Dynamic Evolving Spiking Neural Networks
DICOM	Digital Imaging and Communications in Medicine
DTI	Diffusion Tensor Imaging
ECoS	Evolving Connectionist System
EEG	Electroencephalography
ESNN	Evolving Spiking Neural Networks
FEF	Frontal Eye Field
FIR	Finite Impulse Response
FMRI	functional Magnetic Resonance Imaging
GENESIS	General Neural Simulation System
GRN	Gene Regulatory Networks
GSC	Generalized Sparse Classifier
HHM	Hodgkin-Huxley Model
IFG	Inferior Frontal Gyrus
IM	Izhikevich Model
IPL	Inferior Parietal Lobule
IPS	Intraparietal Sulcus
ITG	Inferior Temporal Gyrus
ITL	Inferior Temporal Lobule
LIFM	Leaky Integrate-and-Fire Model
LTD	Long Term Depression

LTP	Long Term Potential
LSM	Liquid State Machine
MEG	Magneto Encephalography
MFG	Medial Frontal Gyrus
MLP	Multilayer Perceptron
MLR	Multiple Linear Regression
MNI	Montreal Neurological Institute
MRI	Magnetic Resonance Imaging
NEST	Neural Simulation Tool
NIRS	Near Infrared Spectroscopy
OL	Occipital Lobe
OPER	Opercularis
PET	Positron Emission Tomography
PPREC	Posterior Precentral Sulcus
pSNM	Probabilistic Spiking Neuron Model
pSNN	probabilistic Spiking Neural Networks
POC	Population Rank Order Coding
PROC	Population Rank Order Coding
PSO	Particle Swarm Optimization
PSP	Postsynaptic Potential
PSTH	Peri-Stimulus-Time Histogram
RO	Rank Order
ROC	Rank Order Coding
ROI	Region of Interest
SDSP	Spike-Driven Synaptic Plasticity
SFG	Superior Frontal Gyrus
SMA	Supplementary Motor Area
SNN	Spiking Neural Networks
SNR	Signal to Noise
SMG	Supramarginal Gyrus
SPAN	Spike Pattern Association Neuron
SpikeNET	Spike Network
SPL	Superior Parietal Lobe
SPM	Statistical Parametric Mapping
SRM	Spike Response Mode

SSTD	Spatio- Spectro Temporal Data
STBD	spatio- and spectro-temporal brain data
STDP	Spike Timing Dependent Plasticity
STG	Superior Temporal Gyrus
STP	Spatio-temporal Pattern
STPR	Spike Time Pattern Recognition
SWC	Small World Connections
SVM	Support Vector Machine
TL	Temporal Lobe
TM	Thorpe Model
TR	Time of Repetition
TRIA	Triangularis

ACKNOWLEDGEMENTS

I would like to extend my gratitude and to acknowledge some of the many individuals who have educated, inspired, motivated and supported me towards the completion of this PhD study.

My sincerest and highest appreciation to my primary supervisor, Professor Nikola Kasabov, who offered me the opportunity to be in his dedicated PhD supervision and gave me the chance to be one of the family members' to the prominent Knowledge Engineering and Discovery Research Institute (KEDRI). Your intellectual supports, great ideas, comments, and encouragement were very significant for me in ensuring this study persisted in the right path. It was you who have motivated and inspired me to continue with the study and not giving up during the fuzzy and blurry duration of my study years. Thank you so much for your patience, tolerance and confidence.

Special thanks to my secondary supervisor, Dr. Russel Pears who introduced me to the machine learning and computational intelligence field, and for allowing me to attend his class on Data Mining and Knowledge Discovery. My acknowledgment is also to Dr. Bana Handaga for his continuous help and support. He has constantly spent countless hours to teach me programming and to help in my paper writings.

I had a great time learning and working in KEDRI which provided a multiracial and exciting research environment where we could exchange knowledge and ideas, and once in a while to have party just to release the pressure and tension. To Joyce D'Mello, the manager of KEDRI who has very beautiful eyes, you will always be remembered for your kindness, encouragement and thoughtfulness especially during my hard times. My life would be difficult and miserable if it wasn't with your help. Thank you heaps!

My thanks also go to Muhaini Othman for being a good friend, for being there through my ups and down, providing me with the updated versions of the NeuCube model, helping me with the coding, and of course for her delicious cakes! I would also like to extend my gratitude to the past and current KEDRI team members: Haza Nuzly, Kshitij Dhoble, Nuntapod Nuntalid, Linda Liang, Maryam Doborjeh, Lissa Capecci, Neelava Sengupta, Enmei Tu, Nathan Scott, and Reggio Hartono for many discussions and conversations, ranging from technical to non-technical issues. The group has been a great team that continuously gives assistance especially in technical and professional

advice. My gratitude also to my UTHM-PGR4 girlfriends: Hannani Aman, Norlida Hassan, Isredza Rahmi A. Hamid, Nurul Aswa, Mahfuzoh Wasikon, Yana Mazwin Hassim and Nor Yusliza Abdullah for your continuous supports and motivation.

I would like also to thank the Malaysian Ministry of Higher Education and Universiti Tun Hussein Onn Malaysia for providing the scholarship during the entire phase of my study, and Auckland University of Technology, specifically KEDRI, for providing good research environment and state of the art facilities.

Last but not least, special thanks also go to my beloved family: Abang – Azlani Sanip, Mak – Misem Limin, Mak – Bizliah Kamar, Bapak – Sanip Nor, Afiq, Qaisara and Qaseh for your endless love and supports throughout my study journey. Also to my dearest brothers and sisters: Kak Long & Abg Mansor, Kak Imah & Abg Akir, Abg Jaini & Kak Nina, Abg Mat & Kak Nora, Kak Idah & Abg Yaim, and Abg Man & Kak Rina for your care and supports. You are all the pillars to my journey and this journey will not be successful if it wasn't with your constant prayers and encouragement.

ABSTRACT

The application of data mining techniques, particularly classification of spatio-temporal 3D functional magnetic resonance images has received growing attention in the literature. Spatio or spatial component as well as temporal component are factors of high importance in determining and recognizing brain state in response to external stimuli. Structural and functional brain data have been hugely collected, in an attempt to improve brain cognitive abilities and processing capabilities, as well as advancement in medicine, health, education, Brain Computer Interface and games. A particular spatio- and spectro-temporal brain data (STBD), functional Magnetic Resonance Imaging (fMRI), provides a comprehensive detail of brain activation when a certain stimulus is presented to the subject resulting from the changes of oxygen level in the blood vessel of the brain. This oxygen difference between active neurons and inactive neurons is captured in sequence, and the images generated from this (the fMRI) are composed of tens of thousands of individual voxels. These massive voxels are the features to this thesis, which became one of the challenges that had to be faced, in addition to the complex format of the data itself.

To some extent, conventional machine learning techniques has successfully process and classify fMRI data. However, these techniques are only best at dealing spatial data, which completely neglect the temporal information that this data has.

Thus, this study proposes and presents a novel computational model that specifically process spatial and temporal information of fMRI data, which make use of the newly proposed NeuCube model as its foundation. The derived model, denoted as NeuCube^{B} utilized the 3D evolving SNN architecture of NeuCube in mapping and learning the data. The model learns from the data; then creates and updates connections between the neurons based on their weights. These connections represent chains of neuronal activities which could be reproduced even when only part of the stimuli data is presented, therefore making the NeuCube connections as an associative memory. The model can be used not only to classify brain activation patterns, but also to determine functional trail from the data i.e. to identify brain areas that receive the most activation from the stimulus.

There are two case studies presented in the thesis involving different set of fMRI data which are in different format. The dataset is used and experimented by many re-

searchers, which utilized different types of conventional machine learning techniques. In NeuCube^{B}, the fMRI features (voxels) are modelled and studied as both spatial and temporal information involving phases of data reading, mapping, and encoding, before they are transferred to initialization and unsupervised learning stage. Connectivity of neurons in the network could be visualized and studied. The visualization can reveal crucial spatio-temporal relationship unseen from the data that are completely ignored by the standard classifiers. For both experiments involving two different sets of fMRI data, NeuCube^{B} model results in better classification accuracy as compared to the standard classifiers. From this result, it can be concluded that NeuCube^{B} model is not vulnerable to noise, that normally reside in fMRI data. In addition the result can be further interpreted to better understand the brain activation under which the brain data is collected. However, these results and interpretations could still be improved, and further exploration on the subject matter is indeed a huge research prospect.

Chapter 1

INTRODUCTION

1.1 BACKGROUND

The human brain is the command centre for the complex network of nerves and cells that carry signals to and from the brain to various part of the human body. For instance, when a person looks at a picture, a particular region in the brain will be activated. In order to capture the spatio temporal activation, brain imaging technique, in particular functional magnetic resonance imaging (fMRI) technique, is often used. The technique which detects metabolic activity of brain areas (not neuronal firing itself) is widely available and it allows subjects to be examined noninvasively. More precisely, fMRI measures the blood oxygen level (oxygenated haemoglobin to deoxygenated haemoglobin) at many individual locations within the brain. This neural activity is widely believed to influence the level of oxygen in the blood and is known as blood oxygen level dependent (BOLD) response (Mitchell, et. al. 2004).

Images generated from this technique are constructed from two components – spatial/spectral (or spatio) and temporal components. Spatial component is identified as a volume of a brain that can be further sub-divided into smaller three-dimensional (3D) cuboids, known as voxels (volume element). In addition, temporal component is the time acquired scanning the whole volume of a brain. The combination of these spatial and temporal data of the brain images will be the main characteristics to be investigated in this study. Apart from fMRI, electroencephalogram (EEG), diffusion tensor imaging (DTI), magnetoencephalography (MEG) and near-infrared spectroscopy (NIRS) are also known as spatio- and spectro-temporal brain data (STBD). These are largely collected and available for researchers in the field of medicine, health, cognitive sciences, education and Brain Computer Interface (BCI) to take advantage of.

STBD in particular fMRI, is not only large, consisting millions of data points, but also has complex structure both in space and time. To deal with such complex and massive data points, a new computational framework is needed in order to process and understand this data, which was only partly solved previously by using standard machine learning techniques. In recent years, spiking neural networks (SNN) that have the

same working principle as STBD in terms of its spiking activity is anticipated to be suitable for the creation of a computational framework for learning and understanding of fMRI as well as other brain data. A newly proposed architecture called NeuCube (Kasabov, 2014) that is based on a 3D evolving SNN (eSNN) is developed to map and learn STBD, thus to better understand the data.

eSNN architecture (Wysoski, Benuskova, & Kasabov, 2006) was inspired by the Evolving Connectionist System (ECoS) (Kasabov, 1998) principles that evolves its structures incrementally in time, i.e. in its neurons and neurons connections. Besides the fact that ECoS could perform training very fast, it is also capable of processing large amount of spatio-temporal data, and it has structures that can accommodate new features to be added in at a later stage of the learning process. The advantages that ECoS and eSNN have to offer make NeuCube the best architecture to map, train and learn STBD. Previous experiments involving the use of NeuCube in modeling and learning spatio-temporal stroke data (Kasabov et al., 2014) has produced high accuracy in predicting stroke occurrences, thus inspiring more studies to experiment with other variants of STBD.

Undeniably many efforts have been performed to model and process fMRI or other spatio-temporal data but classical machine learning often neglect the temporal component that this data has. We believe that beside the spatial component of fMRI that holds so much information about the brain, the other equally important component i.e. temporal component is also very crucial and very much desired in making accurate decisions.

1.2 MOTIVATION

As have been pointed out in recent publications (Gerstner, Sprekeler, & Deco, 2012; Poline & Poldrack, 2012; Van Essen et al., 2012), an integrated model that could model and analyze brain data is very much required, and machine learning is one of the approaches mentioned. Methods such as Support Vector Machine (SVM), Multilayer Perceptron neural networks (MLP), Gaussian Naive Bayes and many more have been used successfully in processing static brain data (Mitchell et al., 2003; Rustandi, 2007a, 2007b; Wang, Hutchinson, & Mitchell, 2003), but unfortunately they are not efficient in capturing and processing complex spatio-temporal relationship of STBD. These methods also could not hold prior knowledge about the brain in the models. In addition to these efforts, researchers have also developed modeling paradigm that could provide

accurate structural and functional brain models (e.g. Izhikevich & Edelman, 2008; Markram, 2006; Toga, Thompson, Mori, Amunts, & Zilles, 2006). However they still could not be used for machine learning and pattern recognition of STBD, which specifically could handle fMRI data modeling and processing that would facilitate researchers to find efficient solutions to important problems such as detecting fMRI cognitive functions accurately and interpreting the data model.

Thus there is a significant gap to develop a computational framework that have a generic spatial structure of an approximate map of the human brain in certain coordinate system (e.g. MNI or Talairach); that could also map the spatial location(i.e. voxel coordinates) of the incoming brain data; that later on could represent and process the voxels as train of spikes and STBD; and finally that have brain-like learning rules to learn the STBD and evolve when new STBD patterns are presented into the framework. Following the principle of brain cognitive development, evolving is in terms of learning and recognizing new STBD patterns and adding those patterns into the model. In addition, the framework preserves a spatio-temporal associative memory (i.e. in its neuron connections) that can be inferred for new knowledge discovery.

We also believed that the temporal order of fMRI data acquire significant meanings to the interpretation and understanding of the brain data. Thus, a framework that can handle this temporal information is very much required i.e. to be able to train the data according to their temporal order, in which the output of the post-synaptic neurons will be greatly influenced by the characteristics of the neurons before them. At the end of the neurons computations, it is hoped that the framework is able to produce more accurate results as compared to the temporal-ignorant classifiers that disregard the sequence of the data.

On top of this, conventional classifiers developed so far are incapable of handling large and noisy data, unable to deal with stochastic and dynamic processes while ignoring several equally important parameters required in the computations of spiking neurons (Kasabov, 2010). These variables include the probability of synapses to accept spikes and the emission of neurotransmitters (Lauger, 1995), synaptic scaling, STDP and synaptic redistribution (Abbott & Nelson, 2000), physical properties of neurons' connections (Huguenard, 2000), new information about neuronal information processing in biological neural networks (Gerstner & Kistler, 2002) and gene and protein expression (Kojima & Katsumata, 2008).

This research addresses this challenge with the development of such framework that is based on recently proposed SNN-based architecture – NeuCube (Kasabov, 2014).

Because of the complexity of the fMRI data itself (having spatio and temporal information) together with the fact that it is big in nature (thousands of features but with low number of samples), there are chances that NeuCube^{B} could provide an environment that could better process and analyze the fMRI data. The analysis is critical in such a way that it could map the brain data, convert the voxel intensity values into spikes and STBD patterns, learn the STBD patterns, classify and visualize the connections between neurons after the learning process.

1.3 RESEARCH OBJECTIVES

This research has the following objectives:

a. To develop a new methodology for spatio-temporal fMRI data processing

b. To develop a novel fMRI mapping strategy

c. To analyze the neurons connectivity

d. To conduct practical experiments on benchmark datasets

e. To classify the dataset into predefined classes and to compare the findings with performance of other existing well known methods

f. To share research outcomes through presentations and publication of related research community

1.4 SPECIFIC RESEARCH QUESTIONS

Aligned with the research objectives, the specific research questions of this study are as follows:

1. How to map the spatial location of the benchmark fMRI dataset into the NeuCube^{B} architecture? NeuCube^{B} (Kasabov, 2014) was predefined as having 1,471 neurons constructed in 3-dimensional Talairach coordinate system. It is crucial to bear in mind that brain data from different subject will definitely have differ-

ent brain sizes and structures, i.e. mapping of these brains will impose different spatial estimation.

2. Can the learning of large and spatio-temporal fMRI data in the NeuCube^{B} be better and improved compared to other standard machine learning techniques?

3. What can be learned from the connectivity of neurons in the NeuCube^{B} before and after the training? Which part of the brain region will be more activated when a certain stimulus is presented to the subject? Different task will stimulate neurons in different locations to spike, thus more neurons connections will be created.

In brief, this research intends to develop a computational model that could map, learn and mine spatio-temporal brain data specifically fMRI, in which the model is developed, based on the new proposed NeuCube^{B} architecture. This research also aims to achieve better classification result and better understanding of the fMRI data.

1.5 THESIS STRUCTURE

The thesis is organized in eight chapters and will be explained briefly as the following:

CHAPTER 2 This chapter explains literatures on the backbone and the inspiration of the NeuCube^{B}, which is SNN. Components of SNN that will also be discussed include: models of neurons, available encoding methods that can be used to transform voxels intensity values into spikes and, models of learning. Other related principles that will be introduced in the chapter include eSNN, liquid state machine (LSM) and available SNN tools and applications.

CHAPTER 3 This section starts with an introduction to imaging techniques, followed by reviews in detail of fMRI as one of the imaging techniques that provide images of brain structure in great resolutions, continued with existing methods for fMRI data analysis and finally briefly explains of feature selection.

CHAPTER 4 A review on the existing techniques that have been applied to the benchmark StarPlus dataset is presented in this chapter. A preliminary result involving

standard classifiers, which are Support Vector Machine (SVM) and Multiple Linear Regression (MLR), tested on the dataset is also discussed later in the chapter.

CHAPTER 5 This chapter explains the NeuCube based methodology used in the research. It includes the architecture and principles of the generic NeuCube which include the unsupervised and supervised mode of learning. There will also be reviews on the three levels that define the architecture: the input module (lowest level), the NeuCube module (the middle level) and the output module (the highest level).

CHAPTER 6 The first case study is presented in this chapter which involves the use of StarPlus data as the input to the NeuCube^{B}. The steps in conducting the experiment i.e. data mapping, data encoding, data learning, data classification and result analysis are explained.

CHAPTER 7 Another experiment is carried out using different set of fMRI data which is in a different format from the StarPlus. The same steps are applied to the data, but with different mappings, which represents a new challenge to the research. The reason is because of the NII format, which in principal is associated with NIfTI-1 data format. This file type is defined in a 4-dimensional data structure to save the volumetric spatio-temporal fMRI data. The results of the experiment are discussed at the end of the chapter.

CHAPTER 8 This chapter explains the conclusions and future direction of the study.

1.6 THESIS CONTRIBUTION

The following diagram shown in Figure 1-1 summarizes the contributions of this research, in terms of the proposed model, datasets used and the problems to be solved. The newly proposed NeuCube^{B} framework is experimented using two different sets of data namely StarPlus and Haxby. From the figure, the first crucial step in solving the problem i.e. to understand the data is to map the selected voxels (both voxel coordinates and values) into the 3D SNN cube to ensure that the spatial locations are maintained. This will be the first contribution of the study. Selected voxels are transformed into spike trains using AER algorithm and are then propagated continuously (i.e. according

to their temporal order) into the evolving cube via the input neurons. These spike sequences are trained using the unsupervised Spike Timing Dependent Plasticity (STDP) learning rule and the temporal relationship between the data are encoded to the connection weights. The spike sequences are then fed into a supervised dynamic evolving Spiking Neural Networks (deSNN) learning rule. An output neuron is created and its connection weights are calculated for the incoming new data (i.e. test data). These generated weights are compared with the weights generated during the unsupervised learning. In short, the same neurons are spiking to evoke the same temporal connectivity generated during the unsupervised learning (Izhikevich, 2006). Accordingly, in this study, the output neurons learned to classify the spike patterns of fMRI data for both mentioned datasets and this would be the second contribution. And finally, for the third contribution, this study presented spatio-temporal neurons connections based on the fMRI data spiking activity which are displayed using blue line and red line to denote positive spikes and negative spikes respectively.

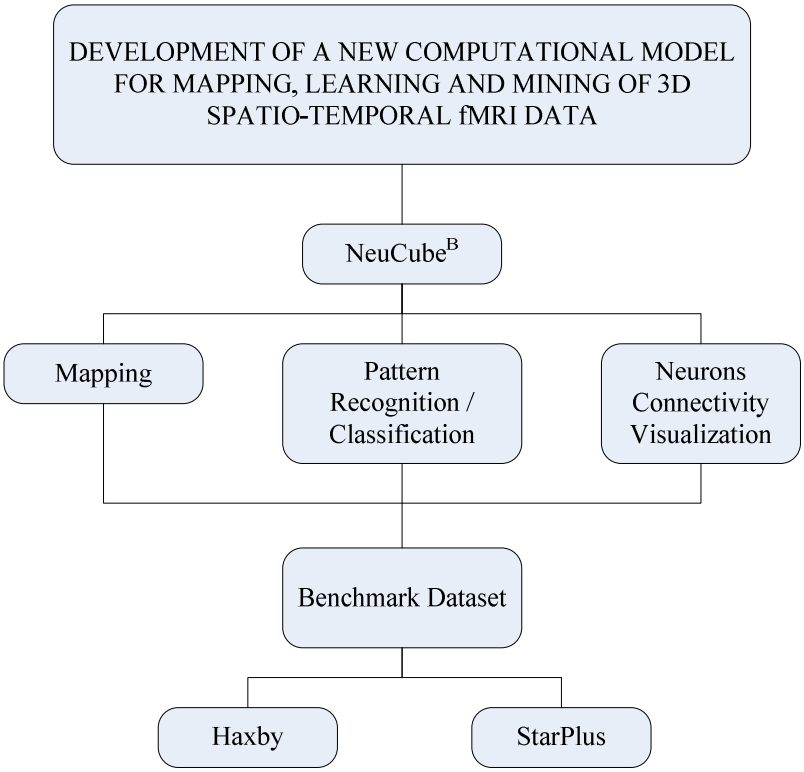

Figure 1-1 : A visual summary of contributions of the thesis in terms of data mapping, classification and connectivity visualization.

1.7 PUBLICATION

Murli, N., Kasabov, N., &Handaga, B. (2014, January) Classification of fMRI Data in the NeuCube^{B} Evolving Spiking Neural Network Architecture. In *Neural Information Processing* (pp. 421-428). Springer International Publishing

Murli, N., Kasabov, N., &Handaga, B. (2015). Evolving Spiking Neural Network for Spatio-temporal Pattern Classification with a Case Study on Visual Object Recognition. Submitted to Journal of Neural Networks.

Kasabov, N., Scott, N, Tu, E., Othman, M., Doborjeh, M., Marks, S., Sengupta, N., **Murli, N.**, Hartono, R., Espinosa-Ramos, I., Capecci, E., Zhou, L., Alvi, F., Wang, G., Taylor, D., Feigin, V., Gulyaev, S., Mahmoud, M., Hou, Z. and Yang, J. (2015). Evolving Spatio-Temporal Data Machines Based on the NeuCube Neuromorphic Framework: Design Methodology and Selected Applications. Submitted to Neural Networks

1.8 CHAPTER SUMMARY

This chapter introduces and describes the study which includes the general view of the problem, the objectives, the motivations for carrying out this study, and the specific research questions need to be answered and justified throughout the whole study period. It also reviews the structure of the thesis i.e. the topics that will be covered in the other chapters of the thesis. Thesis contributions and publication are also outlined.

In the next chapter, a review on spiking neural networks, which is the most important model used in this whole study, is discussed in detail. Topics of spiking neurons that will be the focus of the chapter include neuron models, in particular Hodgkin-Huxley, Leaky Integrate-and-Fire, Izhikevich, Spike Response, Thorpe and Probabilistic Spiking neuron models; and learning models specifically SpikeProp, One-Pass Algorithm and Spike Time Dependent Plasticity. The principles of Liquid State Machine are also discussed in the later chapter, together with a review on applications and tools of SNN.

Chapter 2

REVIEW OF SPIKING NEURAL NETWORKS

This chapter reviews the basis of spiking neural networks (SNN). It will discuss the fundamental components of SNN that includes data encoding methods, neuron models and learning algorithms.

2.1 INTRODUCTION TO SPIKING NEURAL NETWORKS

The human brain is made approximately of 86 billion cells known as neurons (Koch & Reid, 2012) which are connected together in a vast and complex network. These neurons are of various types, shapes and functions, and communicate with other neurons mostly through the generation and propagation of output or action potentials (or better known as spikes). Generally a typical neuron (Figure 2-1) is made of dendrites that act as the collecting points for inputs and propagate them to the cell body/soma. The soma acts as the central processing unit that actually computes all the inputs and makes decision either to generate an output or not. Meanwhile the axon acts as the link to transfer output signals to other neurons. Communicating neurons are linked together through a synapse in which the sending neuron is called as pre-synaptic neuron and the receiving neuron is called as post-synaptic neuron.

Most biological neurons are capable of generating spikes when the average potential of the membrane reaches its normal resting state (i.e. threshold). The study of artificial neural networks (ANN) in computer science is in general inspired by the structures and functions of these biological neurons (Arbib, 1995; Hodgkin & Huxley, 1952; Kandel, Schwartz, & Jessel, 2000). Currently most developed artificial neurons are very simple as compared to its biological counterpart. For instance, the well-known McCulloch-Pits neuron model (McCulloch & Pitts, 1943) has either active or inactive state (binary states), in which a simple threshold function is used to compute the neuron state, based on the weighted sums of other neurons connected to it. Another example is the one modelled by Hebb (Hebb, 1949) which define neuron's synapse to increase when the pre-synaptic and post-synaptic elements tend to be coactive (Arbib, 1995). Then followed by the development of Rosenblatt perceptron neuron model (Rosenblatt,

1961) which is having synchronous inputs and generating real-valued output between 0 and 1, computing the average firing rate of the neuron.

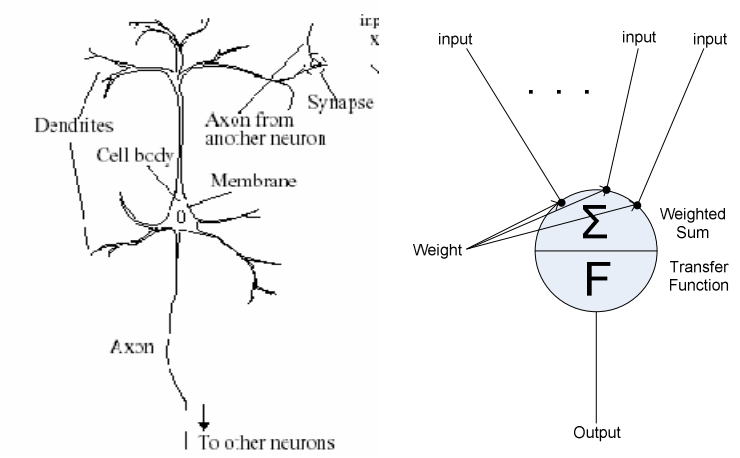

Figure 2-1: Biological neuron (Hemming, 2003) (left) and artificial neuron model (right)

Generally each neuron in ANN, e.g. in a multilayer perceptron network, will fire in each propagation cycle, thus resulting in a deterministic system i.e. the new state and output of a neuron is computed from the state-transition and output function. Despite its' simplicity, this model has successfully solved complex problems in many studies such as for data classification, information retrieval and time series data prediction.

However, several limitations of the model have been pointed out for instance in (Li, 1994; Sharma, Rai, & Dev, 2012) that include the network being a black box, greater computational burden for the machine and only good at solving certain complex numerical computation of linear/nonlinear equations. The ANNs developed so far are not suitable in handling computation of large scale and noisy data, as well as stochastic and dynamic processes (Kasabov, 2010). Other than the parameters already recognized from the ANNs, several other equally important variables need to be considered in the computation of spiking neurons (Kasabov, 2010) such as the probability of synapses to accept spikes and the emission of neurotransmitters (Lauger, 1995), synaptic scaling, STDP and synaptic redistribution (Abbott & Nelson, 2000), physical properties of neurons' connections (Huguenard, 2000), new information about neuronal information processing in biological neural networks (Gerstner & Kistler, 2002) and gene and protein expression (Kojima & Katsumata, 2008). These few examples of concerns are the

parameters defined for the third generation of neural networks or better known as SNN, which is made up of artificial neurons that use spikes trains to represent and process information.

Models of spiking neurons can be traced back to (Lapicque, 1907) and many variations of SNN models have been proposed since then, such as Hodgkin & Huxley model (Hodgkin & Huxley, 1952), Spike Response models (Gerstner & Kistler, 2002; Gerstner, Kreiter, Markram, & Herz, 1997; Gerstner, 1995), Integrate-and-Fire models (Gerstner & Kistler, 2002; Maass & Bishop, 1999), Izhikevich model (Izhikevich & Edelman, 2008; Izhikevich, 2004, 2007) and others (Brette et al., 2007; Thorpe, Delorme, & Van Rullen, 2001). It is then further developed with evolving connectionist paradigm and evolving spiking neurons (eSNN) (Kasabov, 2007) that learn data incrementally in a one-pass propagation by creating and combining the spiking neurons.

Although the computational model is still considered simple as compared to the biological neuron, it provides significantly more realistic computation than the first and second generation of neural networks. The models from the first and second generation disregard the time (i.e. either through an assumed synchronization or through an assumed stochastic asynchronicity (Maass, 1997) whereas this timing information plays a major role in the networks of spiking neurons. To demonstrate a spiking neuron model (Figure 2-1), similar to its biological counterparts, each neuron will receive inputs from a number of pre-synaptic-neurons. Through each synapse, which has its own weight, these inputs are weighted and summed. If the summation of Post-synaptic Potential (PSP) exceeds certain threshold, an output spike will be emitted and propagated by using a transfer/activation function, to another connected post-synaptic neurons via the axon. The transfer function that could be used includes a step, sigmoid or hyperbolic tangent function.

In spiking neurons architecture, there are three important elements that have to be taken care of: data encoding method, spiking neuron model and learning algorithm. Each of these elements will be elaborated in the next few sub topics.

2.2 DATA ENCODING METHODS

Since data in SNN are communicated in terms of spike and spike sequence, a method that can represent/encode data into spikes is of a substantial step in creating spiking neurons architecture. The two main categories of neuron encoding schemes are rate code and pulse code that generate different spike characteristics.

2.2.1 Rate Code

Rate encoding or also known as firing rate is to encode the spikes based on the average number of spikes (or spikes count) over time i.e. how many spikes are emitted within a time encoding window. There are three different views of rate code, referring to different averaging methods: an average over time (single neuron, single run); or an average over several experiment repetitions (single neuron, repeated runs); or average over a populations of neuron (several neurons, single run) (Gerstner & Kistler, 2002).

The study of rate encoding has started with Adrian (1926) in which he showed that as the force applied to the muscle increased, the frequency of neurons being emitted was also increased. In the experiment, identical stimulus is presented to a muscle and the neuronal responses are recorded and these neuronal responses are in terms of spikes count. The measured spikes are those emitted within a specified time window that starts at stimulus onset and ends at stimulus termination.

The rate is calculated by dividing the number of spikes (n_{sp}) emitted in the duration (T) with T, as presented in Equation 2.1. However this encoding scheme was only suitable for stimulus which requires slow reaction of the organisms. This slow reaction was usually found in lab experiments, but not in the real biological brain functions. Real biological brain functions usually happen in much faster duration. In addition, any regularities found, will be considered as noise.

$$v = \frac{n_{sp}}{T} \tag{2.1}$$

The second view of rate code involves averaging the spikes over several experiment runs which are best suited for stationary and time-dependent stimulus. The same stimulus is repeated and the neurons' activity is recorded as spike density of Peri-Stimulus-Time Histogram (PSTH) (Gerstner & Kistler, 2002). As shown in Equation 2.2, it is defined as $n_K(t; t + \Delta t)$ to be the total number of spikes in all runs; starting from stimulus sequence time, t; and Δt is in the range of 1 or few milliseconds; divided with the number of repetitions K which is then further divided with length of val Δt. However, this approach was obviously not suitable for neurons encoding in the brain i.e. certain reaction has to be taken immediately and not after some repetitions of stimulus. Yet, in a situation where a population of independent neurons receives the

same stimulus, the mean firing rate is easier to record from a single neuron and average over N repeated runs.

$$p(t) = \frac{1}{\Delta t}\frac{n_K(t;t+\Delta t)}{K} \tag{2.2}$$

The third view rooted from the notion of neurons population explained earlier which define rate code as the average of spikes over several neurons i.e. neurons with the same characteristics and which respond to the same stimulus. As explained in Gerstner & Kistler (2002) the rate $A(t)$ with units s^{-1} is computed as in Equation 2.3 where N is the population size of neurons, $n_{act}(t; t + \Delta t)$ is total number of spikes emitted between t and $t + \Delta t$ in the neuron population and Δt is a small time interval.

$$A(t) = \frac{1}{\Delta t}\frac{n_{act}(t;t+\Delta t)}{N} = \frac{1}{\Delta t}\frac{\int_t^{t+\Delta t}\sum_j\sum_f \delta(t-t_j{}^{(f)})dt}{N} \tag{2.3}$$

This approach solves the issue raised in the first approach, i.e. calculating the average in a single-neuron level; however it is barely realistic to calculate the average of spikes from a population of neurons with the same properties and connections. Nevertheless, the rate code is still practical in modeling the spikes in many brain areas and has been used in many successful experiments.

2.2.2 Pulse Code

Another encoding approach is based on the exact timing of the spikes or better known as pulse or spike code. The idea of spike time in describing input stimulus has been the interest of many researchers such as in (Optican & Richmond, 1987; Lestienne, 1996; Mainen & Sejnowski, 1995; Thorpe, Fize, & Marlot, 1996). In particular, the first spike carries the most significant information and carries the most weight compared to the later spikes in the sequence (Thorpe et al., 1996). Based on this theory, there are two versions of encoding techniques have been derived which are Rank Order Coding (ROC) (Thorpe & Gautrais, 1998) and Population Rank Order Coding (POC) (Bohte, Kok, & Poutrã, 2002).

In ROC, spikes are ordered according to its arrival, in which the first spike which arrives will be the first in the population, followed by the second spike and so on.

To demonstrate this coding scheme, for neurons A, B, C, D and E in Figure 2-2, the spikes are ranked as C > E > D > A > B.

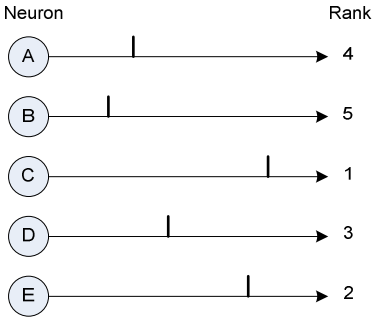

Figure 2-2: Ranking of spikes in a population of neurons applied in ROC method

In contrast, POC is generated based on the firing time identified and calculated using intersection of sensitivity profiles such as Gaussian function (Bohte, Kok, et al., 2002). In this scheme, a single input value i is distributed into multiple input neurons or input spikes M which has its own firing time. Equation 2.4 is the Gaussian function used to calculate the firing time for the input neuron, where its centre μ_i and its width σ are calculated in Equation 2.5 and Equation 2.6 respectively. $[I_{min}, I_{max}]$.$[I_{min}, I_{max}]$ is the maximum and minimum range of input variable and β (values between 1.0 and 2.0) controls the width of each Gaussian receptive field.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{2.4}$$

$$\mu_i = I_{min} + \frac{2i-3}{2} . \frac{I_{max}-I_{min}}{M-2} \tag{2.5}$$

$$\sigma = \frac{1}{\beta} . \frac{I_{max}-I_{min}}{M-2} \tag{2.6}$$

As demonstrated and explained in (Abdul Hamed, 2012; Schliebs, Defoin-Platel, & Kasabov, 2009) as shown in Figure 2-3, the input value is 0.7 with five firing times generated, where $\beta = 2$ and input interval $[I_{min}, I_{max}]$ is $[-2.0,2.0]$.

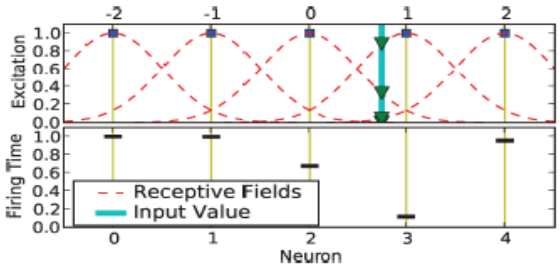

Figure 2-3: Illustration of population rank order coding (Abdul Hamed, 2012; Schliebs, Defoin-Platel, & Kasabov, 2009)

Both ROC and POC have been successfully implemented in many experiments, for instance stroke classification and prediction (Kasabov et al., 2014), visual pattern recognition (Dhoble, Nuntalid, Indiveri, & Kasabov, 2012; Thorpe, Delorme, & Van Rullen, 2001; Wysoski et al., 2006), feature and parameter optimization (Schliebs, Defoin-Platel, & Kasabov, 2009), string pattern recognition (Abdul Hamed, Kasabov, Michlovský, & Shamsuddin, 2009), audio recognition (Wysoski, Benuskova, & Kasabov, 2007b), text-independent speaker authentication (Wysoski, Benuskova, & Kasabov, 2007a).

2.3 NEURON MODEL

There is a wide range of already established mathematical spiking neuron models that imitate the biological neurons to some extent which serve different functions and needs. The following subsections, will discuss some regular spiking neuron models.

2.3.1 Hodgkin-Huxley Model (HHM)

This model was introduced by Hodgkin and Huxley (Hodgkin & Huxley, 1952) who conducted the experiment on the giant axon of a squid. From the experiment, they have concluded that there are three ion channels in the neuron, which are Sodium (Na), Potassium (K) and leakage (L) channel with resistance. To calculate the total of ionic current I_{ion}, which is the sum of all participating channels; the formula in Equation 2.7 and 2.8 are used. In Equation 2.7, G_k represents all channels involved, E_k represents the equilibrium potential and V_m is the membrane potential. In addition, as elaborated by

(Gerstner & Kistler, 2002; Nelson & Rinzel, 1995) three gates of type "m" and one gate of type "h" are used to control the Sodium channel; and four gates of type "n" to control the Potassium channel. These gating variables are calculated using Equation 2.9, 2.10 and 2.11 where the transition rate for each gate from non-permissive to permissive states are represented by $\alpha_m(V)$, $\alpha_h(V)$ and $\alpha_n(V)$ and transition rate for each gate from permissive to non-permissive states are represented by $\beta_m(V)$, $\beta_h(V)$ and $\beta_n(V)$.

$$I_{ion} = \sum_k I_k = \sum_k G_k(V_m - E_k) \tag{2.7}$$

$$I_{ion} = G_{Na} m^3 h(V_m - E_{Na}) + G_K n^4 (V_m - E_K) + G_L(V_m - E_L) \tag{2.8}$$

$$\frac{m}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \tag{2.9}$$

$$\frac{h}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h \tag{2.10}$$

$$\frac{n}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n \tag{2.11}$$

This neuron model only describe the channels and flow of ions in the neuron in generating spikes, which is far from the complex biological neuron, thus bearing several weaknesses as reviewed by Meunier & Segev (2002) which include ignored events that may affect neuron's computation (Strassberg & Defelice, 1993) and inaccurate prediction of the inactivation of the Sodium channel (Bezanilla & Armstrong, 1977). Despite HHM's limitation, it has become the fundamental and starting point for the development of many other simplified neuron models that will be discussed in the following sub topics.

2.3.2 Leaky Integrate-and-Fire Model (LIFM)

As compared from HHM that deals with ion channels and ion flows, LIFM view neuron as a leaky integrator, which will output a spike if the input voltage reaches a threshold and then reset to a resting state. Modelled by a differential equation, integrate-and-fire neuron which can be traced back from (Lapicque, 1907) is represented by a basic circuit that combine a capacitor (C) and a resistor (R) to produce current $(I(t))$. Equation

2.12 is the standard form of LIFM in which $u(t)$ is the membrane potential and $\tau_m = RC$ is the neuron's membrane time constant.

$$\tau_m \frac{du}{dt} = -u(t) + RI(t) \tag{2.12}$$

Spikes are described as the formal events (Gerstner & Kistler, 2002) indicated by a firing time $t^{(f)}$ defined by a threshold value (Equation 2.13) and the potential will be reset to a new value $u_r < \vartheta$ (Equation 2.14).

$$t^{(f)}: u\left(t^{(f)}\right) = \vartheta \tag{2.13}$$

$$lim_{t \to t^{(f)}; t > t^{(f)}} u(t) = u_r \tag{2.14}$$

This model is viewed as the best-known instance of spiking neuron model because of its simplicity and low in computational cost.

2.3.3 Izhikevich Model (IM)

In IM (Izhikevich, 2003), a simple spiking neuron is formulated by combining biologically plausibility of HHM and computational efficiency in LIF neurons. The model is defined as in Equation 2.15 where v is the membrane voltage, u is a recovery variable used to adjust v, $I(t)$ is input currents, and a and b are adjustable parameters.

$$\frac{dv}{dt}(t) = 0.04v^2 + 5v + 140 - u + I(t) \tag{2.15}$$

$$\frac{du}{dt}(t) = a(bv - u) \tag{2.16}$$

A threshold value is set to 30 mV and if the voltage v is bigger than this threshold, v and u are reset (Equation 2.17).

$$if\, v \geq 30\, mV, then \begin{cases} v = c \\ u = u + d \end{cases} \tag{2.17}$$

2.3.4 Spike Response Mode (SRM)

In SRM, the neuron's membrane potential is summarized in terms of response kernel as described in Equation 2.18 (Gerstner & van Hemmen, 1992). It is based on the integrated effects of the incoming spike arriving on the neuron i with its neuron tial $u_i(t)$, and the emission of spike from the neuron if $u_i(t)$ reaches a threshold ϑ (Gerstner, 1998). The potential $u_i(t)$ is the total of the influence of the spikes from pre-synaptic neurons and the spike from its own.

$$u_i(t) = \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta_i\left(t - t_i^{(f)}\right) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in \mathcal{F}_j} w_{ij} \epsilon_{ij}\left(t - t_i^{(f)}\right) \tag{2.18}$$

Although it uses the same concept as the LIFM, the threshold ϑ in SRM is adjustable, which is increased (or decreased) after each spike occurrence. In this model, $t_i^{(f)}$ is the firing time of the last output spike, η_i is a kernel function that describes spike emissions after action potential exceeds the threshold ϑ and its after-potential spikes, ϵ_{ij} is a kernel function that describes the response of the post-synaptic neuron when receiving the spike from pre-synaptic neuron $j \in \Gamma_i$ and w_{ij} is the response weight.

2.3.5 Thorpe Model (TM)

Inspiring from the integrate and fire capabilities of a neuron, TM describes that the first incoming spike carries the most information, because of the argument that the brain only can process one spike from each neuron at one particular processing step (Thorpe et al., 1996). In this model the relation between stimulus saliency and spike relative timing plays a major role i.e. the first spike in the population is the most important in defining meaningful information. The membrane potential $u_i(t)$ is summarized as in Equation 2.19 which will be reset to 0 after each spike emission. In the equation, w_{ij} is the weight of the pre-synaptic neuron, Mod is modulation factor within the val [0,1], and $order(j)$ is the spike rank of neuron j. The threshold $\vartheta = cu_{max}$ where $0 < c < 1$ and u_{max} is maximum potential that a neuron can reach. Simulation software of this model, SpikeNET (Delorme & Thorpe, 2003), has successfully simulated and modelled millions of LIF neurons.

$$u_i(t) = \begin{cases} 0 \quad if\ fired \\ \sum w_{ji} Mod_i^{order(j)} else \end{cases} \tag{2.19}$$

2.3.6 Probabilistic Spiking Neuron Model (pSNM)

The pSNM (Kasabov, 2010) is a further extension of LIFM that include three other probability parameters which are: a probability that a spike will arrive at post-synaptic neuron n_i from pre-synaptic neuron n_j, ($p_{cj,i}(t)$), a probability that a synapse contributes to a spike potential after it receives spike from neuron n_j, ($p_{sj,i}(t)$), and a probability that neuron n_i generates an output spike if the total post-synaptic potential (PSP) reaches the threshold ($p_i(t)$). A simplified representation of pSNM with one synaptic connection together with the probability parameters is shown in Figure 2-4.

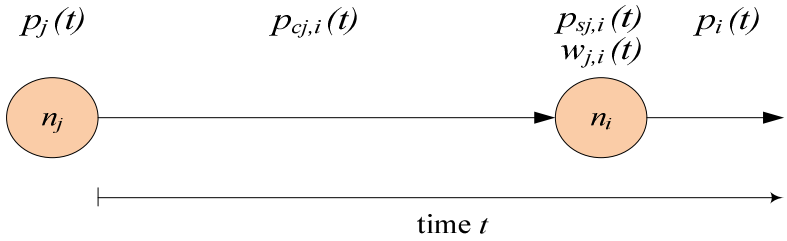

Figure 2-4: Simplified representation of pSNM with all probabilistic parameters and one synaptic connection

The state of post-synaptic neuron n_i is described as the total of inputs received from all m synapses i.e. the post-synaptic potential ($PSP_i(t)$). The model is calculated using Equation 2.20 where $e_j = 1$ if spike is emitted from neuron n_j and $e_j = 0$ if otherwise; $g\left(p_{cj,i}(t)\right) = 1$ with a probability $p_{cj,i}(t)$, and 0 otherwise; $f\left(p_{sj,i}(t)\right) = 1$ if the synapse contributes to the potential with a probability $p_{sj,i}(t)$ and 0 wise; $w_{j,i}(t)$ is the connection weight; t_0 is the time of the last spike emitted by ron n_i; and $\eta(t - t_0)$ is the decay.

$$PSP_i(t) = \sum_{p=t_0,..,t} \sum_{j=1,..,m} e_j g\left(p_{cj,i}(t-p)\right) f\left(p_{sj,i}(t-p)\right) w_{j,i}(t) + \eta(t - t_0) \tag{2.20}$$

If all probability parameters are equal to 1, the model is simplified to be similar to some well-known spiking neuron models, such as LIFM (Gerstner & Kistler, 2002).

2.4 LEARNING

As compared to typical neural computational models, spike precise timing is one of the most important factors in SNN data coding and computation (Bohte, 2004; Maass, 1998) in order to generate efficient processing of information in the neural system. As explained earlier, information is represented and encoded into spikes which are very dependent on the exact firing timing, thus make learning in SNN is a very complex process. In general, learning is defined as the process of parameter adaptation and learning rule is defined as the procedure of adjusting the connection weights. It is divided into reinforcement, supervised and unsupervised learning similar to the learning in traditional neural networks. The next few sections discuss the learning algorithms already designed for SNN in which some of them have been reviewed by Kasinski & Ponulak (2006).

2.4.1 SpikeProp

Similar to backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986) designed for traditional ANN, SpikeProp (Bohte, Kok, & Poutre, 2000) is designed to determine a set of desired firing times (t_j^d) of all output neurons, at the post-synaptic neurons for a given set of input pattern. This is achieved by applying the error function E in the particular least mean squares error function as presented in Equation 2.21, i.e. to minimize the error of squared difference between training output times t_j and desired output times t_j^d. Nevertheless, two assumptions are mentioned: each neuron can fire only once in each processing step and the time course of the neuron's membrane potential after the firing is ignored. Weight w_{ij}^k connecting pre-synaptic neuron and post-synaptic neuron is determined to minimize the error (Equation 2.22) in which η is the learning rate.

$$E = \frac{1}{2}\sum_j \left(t_j - t_j^d\right)^2 \tag{2.21}$$

$$\Delta w_{ij}^{k} = -\eta \frac{\partial E}{\partial w_{ij}^{k}} \quad (2.22)$$

2.4.2 One-Pass Algorithm

In one-pass learning algorithm (Seguier & Mercier, 2002), new output neuron is produced for each training sample. This output neuron is saved in the neuron repository together with the trained threshold values and the weight patterns. A new trained neuron is created if the weight pattern differs from a neuron in the repository. On the other hand, the trained neuron is merged with a neuron in the repository if their weight patterns are similar. The weight pattern and the threshold values of the merged neurons are modified to the average value. The trained network is said as capable to learn new samples incrementally without having to retrain the trained samples (Schliebs, Defoin-Platel, & Kasabov, 2009).

2.4.3 Spike-Time Dependent Plasticity (STDP)

Another well-known unsupervised learning paradigm inspired by the Hebbian learning principle is STDP (Bell, Han, Sugawara, & Grant, 1997; Bi & Poo, 1998; Markram, Lubke, Frotscher, & Sakmann, 1997) in which the synaptic weights are adjusted based on the temporal order of the incoming spike (pre-synaptic) and the output spike (post-synaptic). This synaptic weight adjustment determines synaptic potentiation known as long term potential (LTP) if the synaptic weight is increasing (positive change) and synaptic depression known as long-term depression (LTD) if the synaptic weight is decreasing (negative change). A particular connection is said to potentiate if a pre-synaptic spike arrives before a post-synaptic spike; and is said to depress if it arrives after a post-synaptic spike (Markram et al., 1997).

STDP is expressed in terms of STDP learning window $W(t_{pre} - t_{post})$ in which the difference between arrival time of the pre-synaptic spike and the arrival time of post-synaptic spike will determine the synaptic weight (Equation 2.23). In the tion, τ_+ and τ_- refer to the pre-synaptic and post-synaptic time interval; and A_+ and A_- refer to the maximum fraction of synaptic adjustment if $t_{pre} < t_{post}$ approaches to zero.

$$W(t_{pre} - t_{post}) = \begin{cases} A_+ \ exp\left(\frac{t_{pre}-t_{post}}{\tau_+}\right) & if t_{pre} < t_{post}, \\ A_- \ exp\left(-\frac{t_{pre}-t_{post}}{\tau_-}\right) & if t_{pre} > t_{post}, \end{cases} \quad (2.23)$$

2.4.4 Spike-Driven Synaptic Plasticity (SDSP)

SDSP, a variant of Spike Timing Dependent Plasticity (STDP), is a semi-supervised learning rule (Fusi, Annunziato, Badoni, Salamon, & Amit, 1999) that directs the change of the synaptic plasticity V_{w0} of a synapse w_0 depending on the spike's time of the pre- and post-synaptic neurons. If a pre-synaptic spike arrives at the synaptic terminal while the post-synaptic neuron's membrane potential is higher than a given threshold value (i.e. normally shortly before a post-synaptic spike is emitted), the synaptic efficacy is increased (potentiation). However, when a pre-synaptic spike arrives at the synaptic terminal while the post-synaptic neuron's membrane potential is low (i.e. normally shortly after a spike is emitted), the synaptic efficacy is decreased (depression). Where Δt_{spk} is the pre- and post-synaptic spike time window, this synaptic change can be expressed as:

$$\Delta V_{w0} = \begin{cases} \frac{I_{pot}(t_{post})}{C_p} \Delta t_{spk}, & if t_{pre} < t_{post} \\ \frac{I_{dep}(t_{post})}{C_d} \Delta t_{spk}, & if t_{post} < t_{pre} \end{cases} \quad (2.24)$$

SDSP introduces a dynamic 'drift' of the synaptic weights either to be 'up' or 'down', depending on the value of the weight itself (Kasabov, Dhoble, Nuntalid, & Indiveri, 2013). If the weight is higher than the threshold value, then the weight is slowly driven (by the learning algorithm) to a fixed high value. On the contrary, the weight is slowly driven to a fixed low value if the weight is lower than the threshold value. These two values represent the two stable states and at the end of the learning process, the final weights can be encoded with 1 single bit (Mitra, Fusi, & Indiveri, 2009).

2.5 LIQUID STATE MACHINE

Two major types of reservoir computing is liquid state machine (LSM) (Maass, Natschlager, & Markram, 2002) and echo state network (Jaeger & Haas, 2004). LSM consists of many randomly interconnected recurrent neurons where each neuron receives input from other neurons in different times. This computational model is inspired from the idea of water ripples (output), which are generated after certain objects (input) are dropped on the still water. In an ideal setting, LSM that is constructed with a precise mathematical framework, promises universal computational power, for real-time computing on analogue function in continuous time. LSM is characterized as a model for adaptive computational system, which provides a method for employing randomly connected circuits, a theoretical context where various processors increase the computational power of a circuit and a method for multiplexing different computations within the same circuit (Maass, 2010).

During LSM simulation, the synaptic weights, neurons connectivity and neurons parameters are predefined and predetermined. Referring to Figure 2-5, the continuous stream of input (e.g. trains of spikes) $u(t)$ is transmitted into liquid and will cause the neurons to respond and generate the liquid activity. The state of the liquid $x(t)$ that can be recorded at different time points is simply the current output of some operator or filter that maps input functions $u(t)$ onto function $x(t)$. This state is then passed to the memory-less readout function f that will transform into output $v(t)$.

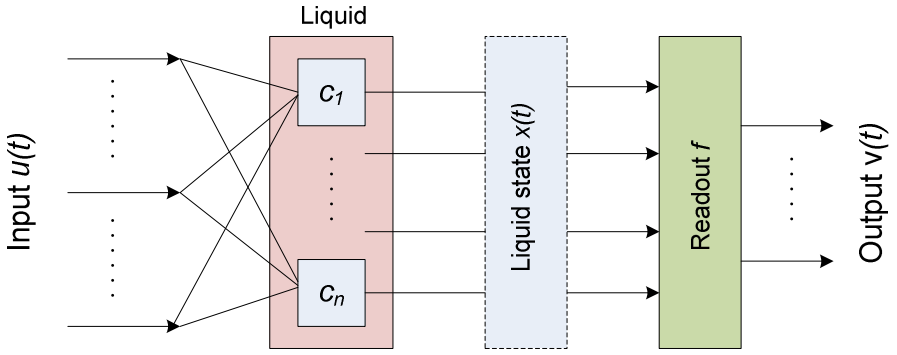

Figure 2-5: The LSM architecture consists of three main layers: input neuron layer, liquid state layer and readout function layer.

This computational model is capable of generating different responses from different input patterns (separation property) that depends on the complexity of the liquid. In addition, it also has 'approximation property' that depends on the adaptation ability of the readout function that can distinguish responses, their generalization and relation with the given output (Grzyb, Chinellato, Wojcik, & Kaminski, 2009). Any statistical analysis or classifier can be used to define the readout function. As the inputs received can be in the form of continuous stream of data, it can be used to solve spatiotemporal problems such as in patterns recognition (Goodman & Ventura, 2006; Schliebs, Nuntalid, & Kasabov, 2010), optimization (Yanduo & Kun, 2009) and classification problem (Ju, Xu, & VanDongen, 2010)

2.6 APPLICATION OF SNN AND TOOLS

Viewed as biologically more plausible model of distributed computation, SNN has been applied in numerous real world applications from various fields either in an unsupervised or in a supervised learning mode. The following few subtopics will discuss the applications according to the learning paradigms.

2.6.1 SNN applications with supervised learning paradigm

Spiking networks trained according to the supervised paradigm have also been used in a number of studies such as in medical field which involve medical data for instance EEG signals and fMRI data as well other medical related data (Ghosh-Dastidar & Adeli, 2007, 2009; Kasabov et al., 2014; McGinley et al., 2010; Nuntalid, Dhoble, & Kasabov, 2011), human movement pattern recognition (Dhoble et al., 2012) that conducted the experiment using different types of SNN classifiers, speech patterns recognition (Hopfield & Brody, 2000; Verstraeten, Schrauwen, Stroobandt, & Van Campenhout, 2005; Wysoski et al., 2007a), pattern recognition (Abdul Hamed et al., 2009; Goodman & Ventura, 2006) and, spike patterns classification (Ponulak & Kasinski, 2010; Schliebs, Defoin-Platel, Worner, & Kasabov, 2009). Apart from that, SNN has also been applied in robot movement control (Joshi & Maass, 2004; Pearson et al., 2007; Rocke, McGinley, Morgan, & Maher, 2007; Trhan, 2010), in economics particularly as a prediction tool (Reid, Hussain, & Tawfik, 2014; Sharma & Srinivasan, 2010; Yang & Zhongjian, 2011) and in general image processing (Meftah, Benyettou, Lezoray, &

Qing Xiang, 2008; Meftah, Lezoray, & Chaturvedi, 2013; Perrinet & Samuelides, 2002; Wysoski et al., 2006).

2.6.2 Applications Using Unsupervised Learning Paradigm

SNN is also trained to learn in an unsupervised mode such as for data clustering and classification (Bohte, La Poutre, & Kok, 2002), image processing and recognition (Masquelier & Thorpe, 2007; Perrinet, Samuelides, & Thorpe, 2004; Srinivasa & Cho, 2014; Thorpe et al., 2001), sensory recognition models (Finelli, Haney, Bazhenov, Stopfer, & Sejnowski, 2008; Martinez & Hugues, 2004), hand writing recognition (Querlioz, Bichler, & Gamrat, 2011), remote sensing classification (Cawley et al., 2011; Silva & Wilcox, 2007; Silva, 2009; Tao & Michel, 2005), feature extraction (Lemoine & Maida, 2013) and robot controller (Xiuqing Wang, Hou, Zou, Tan, & Cheng, 2008).

Other than the typical artificial intelligence or engineering application of SNN, there are other research areas that can take advantage of spiking neurons in particular in modeling and analyzing the structures of biological neurons (Nakagawa et al., 2014; Ponulak & Kasiński, 2010). In fact, many efforts have been conducted to model the spiking neurons in hardware based with neuromorphic circuits (Furber, 2012; Indiveri et al., 2011; Lichtsteiner & Delbruck, 2005; Querlioz et al., 2011) for better analysis and processing of the neuron functions.

2.6.3 SNN Tools

There are many spiking neuron simulators that have been implemented to better simulate, analyze and visualize neuronal networks, including in large-scale networks. One of the most comprehensive spiking neuron simulator is Brian (Goodman & Brette, 2009). It can be used to simulate both integrate-and-fire and Hodgkin-and-Huxley neuron models which can be defined directly by their equation. Neurons connectivity can be defined directly or with predefined functions including the transmission delay. It is more flexible particularly in dealing with non-standard neuron models. Another comprehensive simulator, NEST (Diesmann & Gewaltig, 2002), is dynamic, and can be defined in any size and structure, i.e. not limited to the exact architecture of individual neurons. Other characteristics of NEST include the provision of more than 50 neuron models, more than 10 synapse models that comprise different variants of STDP, allow-

ing neuron state and connectivity modification at any time during the simulation and is fast and memory efficient. General Neural Simulation System (GENESIS) is a general purpose of simulator developed to simulate sub-cellular processes, complex models of individual neurons, simulations of large networks, and systems-level models. In addition, NEURON (Carnevale & Hines, 2006) provides a spiking neurons environment for modeling single neurons as well as networks of neurons. Users can easily create and manipulate neuron model with different complexity through the graphical user interface provided. And lastly neuroConstruct (Gleeson, Steuber, & Silver, 2007), is another SNN simulator that is developed to simplify the development of complex networks of biologically realistic neurons. Several important characteristics include the integration with other simulators such as GENESIS and NEURON, the creation of neurons in a three-dimensional space for more realistic view and the connectivity between neurons can be specified for the networks.

2.7 CHAPTER SUMMARY

This chapter has reviewed the principles of SNN extensively including the encoding methods, neuron models and learning models. In addition, liquid state machine as one of the reservoir types used in the architecture is also presented. The areas where this spiking neuron networks can be applied are also described, as well as examples of tools that can be used to visualize and simulate the network.

The next chapter provides a review on NeuCube architecture: the architecture that will be used in the study. The chapter starts with a general introduction of NeuCube and continues with the three levels defining the architecture.

Chapter 3

REVIEW OF THE NEUCUBE ARCHITECTURE

The general principles of NeuCube architecture (Kasabov, 2012) is shown in Figure 3-1. It is a three-dimensional evolving spiking neural networks architecture developed as a basis of constructing a concrete model to map, learn and understand spatio-temporal brain data (STBD) that may include fMRI, EEG and other brain data as well as SSTD. The architecture consists of four integrated modules: input encoding module, NeuCube module, output module and gene regulatory networks (GRN) module. These modules work in an integrated manner, where different modules may have different types of neurons and different types of learning rules.

Based on the modules stated before, NeuCube architecture is divided into three levels: GRN (lowest level); NeuCube (middle level) and Output (highest level). As the name suggest, NeuCube module is the core processing module in this architecture. NeuCube receives control parameters from the lowest level (GRN) and these parameters will affect how the NeuCube behaves, and in return spiking activity of neurons influence the genes. On the other half of the architecture, neurons in the Cube are connected with those neurons in the highest level (Output module) so that whatever state that the Cube is, interpretation/classification can be made and this state influence further activity of the NeuCube. Thus, the NeuCube structure and functionality evolves over time. Neurons and learning rules of different types can be used in different levels of the architecture.

For any NeuCube model developed, there are two stages of learning involved: unsupervised training and supervised training. During unsupervised training, STBD is entered into appropriate locations in the Cube over time and the neurons' connection weights are calculated and initialized. In the next stage (supervised learning) the same STBD used for training are propagated in the trained Cube and into the Output module. This is to determine the class of the trained data into already predefined classes. Since neurons from middle level are connected to every neuron at the highest level, feedback connection can be created for reinforcement learning.

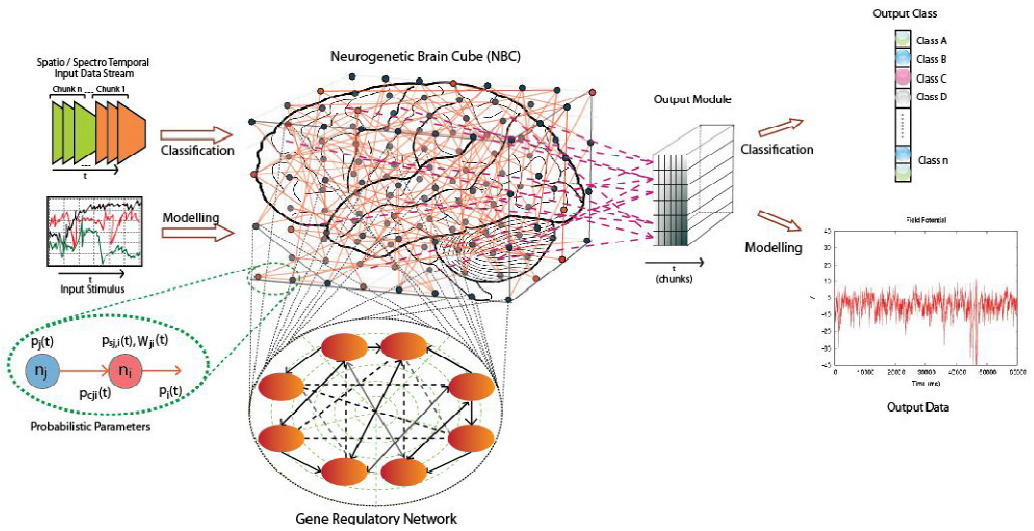

Figure 3-1: General architecture of NeuCube that consists of four integrated modules: Input, NeuCube, GRN and Output module

3.1 LOWEST LEVEL: THE INPUT MODULE

In the lowest level (Input), continuous SSTD is converted into trains of spikes and mapped into corresponding areas in the NeuCube. Conversion of data into spikes can be achieved by using techniques such as Population Rank Coding and Address Event Representation (AER).

3.1.1 Population Rank Coding

As shown in Figure 2-3, continuous input data is encoded into spikes in such a way that the input variable is entered into a population of neurons that release spikes based on their receptive fields. For instance, one channel of EEG data is mapped into several pre-synaptic input neurons in which each of these input neurons hold a spike, which is calculated based on the intersection of Gaussian function. If for example there are 16 channels of EEG data to be encoded and each channel is to be mapped into a population of 5 receptive fields. Thus, there will be a total of 80 pre-synaptic input neurons to be processed. While this figure is still acceptable to be processed by the NeuCube, encoding of fMRI STBD which consists of thousands of voxels may involve a lot of computer processing time and power.

3.1.2 AER Data Encoding

There are a few encoding techniques can be used to convert the continuous data into spikes sequences. For this model, the encoding technique used is AER which is based on the difference between two consecutive voxel values of the same input variable. If the difference is greater than certain threshold value, a spike will be generated.

Depending on the type of data to be measured, whether it is continuous brain data or other continuous data such as ecological/medical/etc., the location of converted data will be determined differently. Different mapping will be used for different types of SSTD. Trains of spikes sequences are continuously entered into the middle level based on their temporal order.

3.2 MIDDLE LEVEL: THE NEUCUBE MODULE

The most important processes take place in the middle level of the architecture or the NeuCube level.

3.2.1 The NeuCube Structure

As shown in Figure 3-1, NeuCube is a 3D SNN structure of approximate map of brain areas following the standard Talairach Atlas. For many years, the standard 1998 Talairach Atlas brain (Talairach & Tournoux, 1988) has served as the casual standard for reporting brain activation locations in the functional and structural brain mapping studies. They have created a co-planar 3D stereotaxic atlas of the human brain that can be used to study it from different subjects and collected using different methods. A software called Talairach Daemon (Figure 3-2) is publicly available for download can be used to calculate (x, y, z) Talairach coordinates of any given point on the brain image together with its corresponding Brodmann area. By using this software, brain areas can be labelled accordingly in different visualization colours as depicted in Figure 3-3.

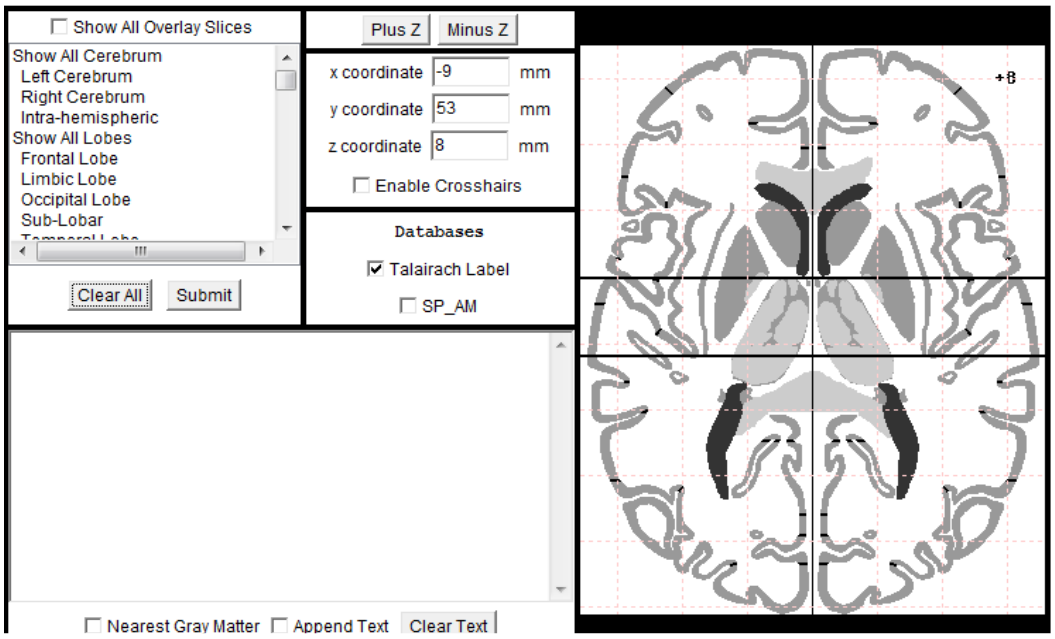

Figure 3-2: The Talairach Daemon Software for brain areas visualization. (http://www.talairach.org/applet/)

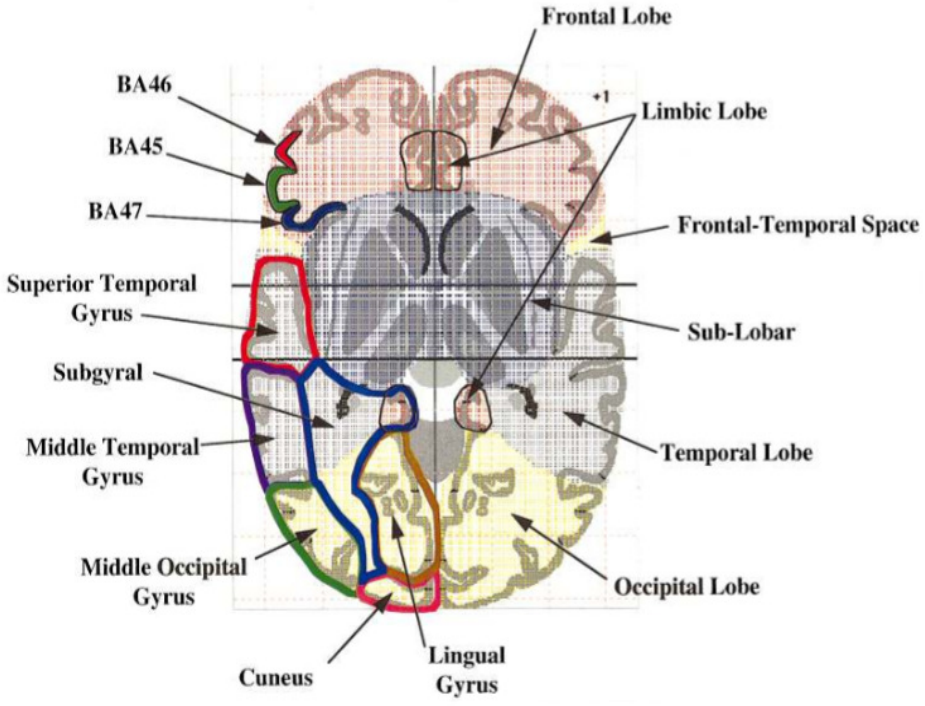

Figure 3-3: The Talairach atlas with lobe labels (illustrated with patterned colour fills), gyral structures (illustrated with bold colour outlines), and several Brodmann areas (illustrated with solid colour fills) (Lancaster et al., 2000).

Following this template, NeuCube is developed using the 1 cm^3 resulting in 1,471 distinct x, y, z coordinate locations. These coordinates are mapped approximately into the 3D cube of spiking neurons of the NeuCube structure. However different mapping resolution is also possible to define the structure e.g. 1 mm^3 may result in 2,000,000 and more of coordinate locations. While Talairach template is derived from the analysis of a single brain, another template which is referred to as Montreal Neurological Institute (MNI) coordinates is derived from the average of MRI data across individuals, for instance MNI152 and MNI1305 (Evans et al., 1993). Another well-known brain template is proposed by the International Consortium for Brain Mapping (ICBM) and its few releases include ICBM452, ICBMChinese56, ICBM AD for Alzheimer Disease and ICBM MS for Multiple Sclerosis (Toga et al., 2006).

The NeuCube initial structure is preliminarily defined based on the available data and the problem being considered. Regardless of data type whether it is spatio-temporal based or vector based, each data/pixel/voxel will be mapped in the NeuCube according to its appropriate spatial location and not just on random allocation. For instance, tapping a finger or any body movement will trigger motor cortex, and thus these data will be spatially mapped into the neurons in the NeuCube that correspond to the motor cortex, i.e. somewhere around the parietal lobe. Neurons in this area will be heavily connected (as compared to other areas of the brain) and these connections will be initialized using small-world connection (SWC) paradigm. The main reason for suggesting a type of small-world connectivity for the NeuCube initial structure is derived from the fact that clusters of neurons correspond to structural and functional areas related to the brain data across time (Bullmore & Sporns, 2009). The creation of an appropriate initial structure of the NeuCube is significant in order for the model to learn the spatio-temporal data and to capture functional connectivity properly (Honey, Kotter, Breakspear, & Sporns, 2007). It is also means that the higher the connectivity between the neurons indicates that the brain areas are more densely interconnected and are closer to each other (Braitenberg & Schüz, 1998).

Based on evolving connectionist system standards (Kasabov, 2007), new neurons are created if existing neurons could not accommodate the new data coming in to the NeuCube and the new connections, connecting new neurons with the existing neurons, are created using the same small world connectivity principle. Current NeuCube implementation has leaky-integrate-and-fire model (LIFM) of spiking neurons with recurrent connections. Input features are propagated through the SNN where STDP unsupervised learning is applied. Probabilistic neuron model with probabilistic parameters

attached to the connection, the synapse and the output of the spiking neurons are also shown in Figure 3-1. The networks accumulate all input spike trains temporal information and convert them into dynamic states that can be classified over time. The recurrent reservoir creates unique accumulated neuron spike time responses for different classes of input spike trains (Kasabov, 2014).

3.2.2 The Neuron Model

Basically, any spiking neuron model could be used in the NeuCube. The current NeuCube development uses LIFM spiking neuron model. In LIFM (Gerstner, 1995; Gerstner & Kistler, 2002; Gerstner, 2001), an output pulse (a spike) is emitted from a neuron whenever several pre-synaptic pulses arrived at the neuron in a short interval of time, and when at a certain point of time $t_i^{(f)}$, the total voltage from the pulses exceeds the threshold value, θ. In this case, with every input spike at time t multiplied to the synaptic weight, the neuron is said to have increased its PSP or also known as membrane potential $u(t)$ up until it reaches a threshold θ and thus a spike is emitted. Following the biological neuron, after a neuron spikes, the neuron's membrane potential is in its refractory period and the voltage is reset to zero. Figure 3-4 illustrates the spiking neuron and its functionality.

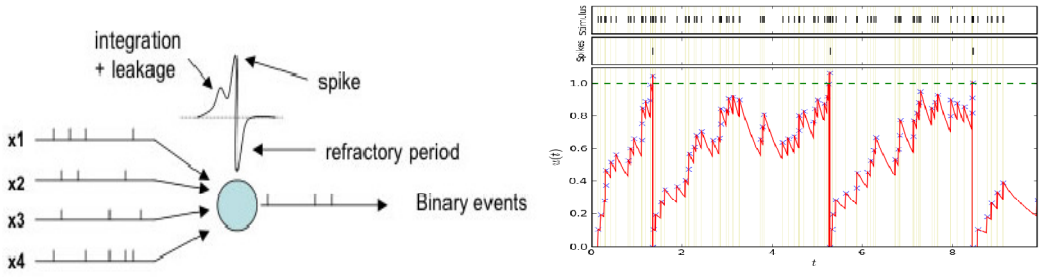

Figure 3-4: (a) Schematic representation of LIFM of a spiking neuron and (b) Functionality of LIFM in which the red line represents the membrane potential, the middle row represents the output spikes and on top is the input train of spikes.

The total probabilistic spiking neuron $PSP_i(t)$ of neuron n_i is calculated using the following formula (Kasabov, 2010):

$$PSP_i(t) = \sum_{p=t_0,..,0}\left(\sum_{j=1,..,m} e_j\, f_1(p_{cj,i}(t-p)) f_2 p_{sj,i}(t-p)\right) w_{j,i}(t) + \eta(t - t_0)) \quad (3.1)$$

where: $e_j = 1$, if a spike has been emitted from neuron n_j, otherwise $e_j = 0$; $f_1\left(p_{cj,i}(t)\right) = 1$ with a probability $p_{cj,i}(t)$, otherwise $f_1\left(p_{cj,i}(t)\right) = 0$; $f_2 p_{sj,i}(t) = 1$ with a probability $p_{sj,i}(t)$, otherwise $f_2 p_{sj,i}(t) = 0$; t_0 is the time of the last spike emitted by neuron n_i; and $\eta(t - t_0)$ is decay in the PSP_i. The probabilistic model will be simplified when all or some of the probabilistic parameters are fixed to '1' and this will resemble the well-known IFM.

Besides connection weights $w_{j,i}(t)$ a probabilistic spiking neuron model (Kasabov, 2010) has the following three probabilistic parameters:

- A probability $p_{cj,i}(t)$ that a spike emitted by neuron n_j will reach neuron n_i at atime moment t through the connection between n_j and n_i. If $p_{cj,i}(t) = 1$ the probability for propagation of spikes is 100%. Otherwise $p_{cj,i}(t) = 0$, i.e. no connection and no spike propagation exist between neurons n_j and n_i.

- A probability $p_{sj,i}(t)$ for the synapse $s_{j,i}$ to contribute to the $PSP_i(t)$ after it has received a spike from neuron n_j.

- A probability $p_i(t)$ for the neuron n_i to emit an output spike at time t once the total $PSP_i(t)$ has reached a value above the PSP threshold (a noisy threshold).

3.2.3 The Learning Rule

The current implementation uses STDP learning rule that make use of Hebbian plasticity (Hebb, 1949) in the form of long-term potentiation (LTP) and depression (LTD). Synapses' weight is increased or decreased based on the timing of post-synaptic action potential in relation to pre-synaptic action potential. If pre-synaptic neuron fires first, the connection weight between the two neurons increases (the difference in the spike time between the pre-synaptic and post-synaptic neurons is negative), otherwise it decreases. Through STDP, connected neurons learn consecutive temporal associations from the data and new connections are also evolved. Pre-synaptic activity that comes

first before post-synaptic firing can encourage LTP, and reversing this temporal order causing LTD.

3.2.4 The Evolvability of the NeuCube

Neurons in a functional or structural brain areas are densely interconnected and the nearer these area are, the more the connectivity between them occur (Braitenberg & Schüz, 1998). These structural and functional connectivity of the brain, either measured through MRI (Chen, He, Rosa-Neto, Germann, & Evans, 2008) or EEG and MEG (Stam, 2004), manifests small-world organization. Hence, this is the key inspiration for proposing a type SWC for the initial structure of NeuCube, where neurons clusters represent structural and functional areas related to the brain data (Bullmore & Sporns, 2009).

Following ECoS principles (Kasabov, 2007), the NeuCube structure also evolves during the training through the creation of new neurons and new connections. New neurons are created to accommodate data that is not possible to accommodate in the existing neurons and these new neurons are connected with the rest of the neurons following initially the SWC principle. Moreover, ECoS could perform training very fast and is capable of processing large amount of spatio-temporal data. The advantages that ECoS and deSNN (which will be elaborated in the next section) have to offer make NeuCube the best architecture to map, train and learn STBD.

3.3 HIGHEST LEVEL: THE OUTPUT MODULE

After the data have been trained in the middle level of the architecture, the patterns generated are learned to be associated with known classes defined in the Output Module. Feedback connections from the Output Module to the NeuCube are possible to establish reinforcement learning. In the current implementation, all spiking neurons in the NeuCube are connected to each of the output neurons. The classifier used is dynamic evolving Spiking Neural Networks (deSNN).

3.3.1 Dynamic evolving spiking neural networks (deSNN)

The deSNN (Dhoble et al., 2012) training algorithm utilizes both RO and SDSP temporal learning rules. While RO learning determines the initial connections weights values for a STPR of AER dataset, SDSP/STDP adjusts these connections in an unsupervised manner utilizing the drift parameter, D, which is based on the following spikes as part of the same spatio-temporal pattern.

In Rank Order (RO) (Thorpe & Gautrais, 1998) learning rule, the rank of the first incoming spikes on the synapse is very important where inputs priority is established based on the order of the spike arrival on that particular synapse. The main advantages of this learning rule are: it is fast and one-pass learning (uses the order of the incoming spikes as additional information); and is asynchronous data entry (synaptic inputs are accumulated into the neuronal membrane potential in an asynchronous way). For each training sample, only neuron with the highest PSP value has its weight updated.

$$PSP_{max} = \sum mod^{order(j)} w_{j,t} \tag{3.2}$$

Where mod is a modulation factor; j is the index for the incoming spike at synapse j, i and $w_{j,t}$ is the corresponding synaptic weight; $order(j)$ is the order of the spike at the synapse j, i among all spikes arriving from all m synapses to neuron i. For the first spike, the value for $order(j)$ is 0 and it increases according to the input spike order. An output spike is generated by neuron i if the $PSP(i,t)$ becomes higher than a threshold $PSP_{Th}(i)$. This threshold is calculated as a fraction of $C \in [0.1]$ of the maximum PSP, generated with a propagation of the training sample into the updated weights.

$$PSP_{Th} = C \,.\, PSP_{max} \tag{3.3}$$

In each training of an input pattern of a classifier, the connection weights are calculated based on the order of the incoming spikes (Thorpe & Gautrais, 1998; Wysoski, Benuskova, & Kasabov, 2010):

$$\Delta w_{j,i}(t) = mod^{order\,(j,i(t))} \tag{3.4}$$

RO learning in eSNN involves the adjustment of the connection weight of each synapse only once i.e. based on the rank of the first spike arrival on this synapse, which may be appropriate for static pattern recognition, but not efficient for complex SSTD like fMRI and EEG (Kasabov et al., 2013). For these type of data, the connection weights need to be further adjusted based on the following spikes arriving on the same synapse over time, and this is where dynamic synapses are introduced, by implementing SDSP (Fusi et al., 1999).

As in the learning phase of eSNN, for each training input pattern, a new output neuron i is created and its connection weights $w_{j,i}$ to the input (feature) neurons are initially calculated as $w_{j,i}(0)$ based on the order of the incoming spikes on the corresponding synapses using the RO learning rule. Once a synaptic weight $w_{j,i}$ is initialized (based on first spike arriving at synapse j), the synapse becomes dynamic and adjusts its weight through SDSP algorithm. It increases its value with a small positive value (positive drift parameter) at any time t a new spike arrives at this synapse and decreases its value (negative drift parameter) if there is no spike at this time. The deSNN training algorithm is as follows:

Table 3-1: deSNN training algorithm

The deSNN Training Algorithm
input: Spike trains, Set deSNN parameters (including: Mod, C, Sim, and the SDSP parameters)
for *every input spatio-temporal spiking pattern* P_j **do**
Create a new output neuron i for this pattern and calculate the initial values of connection weights $W_{j,i}(0)$ using the RO learning formula (Equation 3.4).
Adjust the connection weights $W_{j,i}$ for consecutive spikes on the corresponding synapses using the SDSP learning rule formula (Equation 2.24).
Calculate PSP_{max} using formula (Equation 3.2).
Calculate the spiking threshold of the ith neuron using formula (Equation 3.3).
if (*the new neuron weight vector* $w_{j,i}$ *is similar in its initial* $w_{j,i}(0)$) **then**
merge the two neurons (as a partial case only initial or final values of the connection weights can be considered or a weighted sum of them)
else
Add the new neuron to the output neurons repository.
end if
end for (Repeat for all input spatio-temporal patterns for learning)

3.4 NEUCUBE^{B} STRUCTURE FOR SPATIO-TEMPORAL MODELLING AND PATTERN RECOGNITION OF BRAIN SIGNALS

3.4.1 Spatio-Temporal Brain Data (STBD) and Spatio-Temporal Pattern Recognition (STPR)

The most common types of spatio-temporal data collected including in the field of Neuroinformatics (e.g. fMRI and EEG), engineering (e.g. audio and video data), ecology (e.g. species establishment), environment (e.g. pollution and volcanic formation), medicine (e.g. risk of a disease and recovery data), economics (e.g. financial time series data), bioinformatics (e.g. gene and protein expression data), etc. These data are characterized by a strong temporal (time) as well as spatial (space) components. The most common types of STBD collected at the cognitive level are fMRI, EEG and MEG; and these data can be very complex and very hard to interpret.

An fMRI with the use of MRI scanner can capture images of brain activity, which is based on the brain areas that receive the most activation, either stimulated externally or internally (Mitchell et al., 2003). Active brain areas have higher level of oxyhemoglobin than less active areas. In a single experiment run, a whole-brain functional image is acquired in terms of slices (e.g. 21) and each slice is captured in a single time moment, defined as the sampling or repetition time (TR). Brain slices captured within each TR will be the sample to be modelled, mapped and learned overtime in the NeuCube^{B}.

NeuCube^{B} has brain-inspired eSNN architecture is anticipated to be a promising paradigm able to handle the complex STP of fMRI data. The data which is represented in many dimensions, i.e. in space, time, frequency and phase, and is usually collected in large volumes, acquires so much knowledge to be extracted. Not just only to recognize the simple brain state, either active or inactive, the pattern presented in the brain data could also be used to recognize the other complex physiological as well as psychological state of the brain. Conventional classifiers, such as SVM, MLP and other well known classifiers, are sufficiently capable of handling static pattern recognition, but not for dynamic and spatio-temporal brain processes and recognition. A computational model which is capable of STPR of fMRI and other STBD such as EEG and MEG is very much required. This NeuCube^{B} model will be further elaborated in the next section as well as in Chapter 6.

3.4.2 The NeuCube^{B} Template

Figure 3-5 shows the 1,471 NeuCube^{B} neurons structure displayed in blue dots. The cyan dots represent the mapped fMRI data just in time before the initialization process starts (left) while the yellow dots represent the mapped fMRI after the initialization process (right).

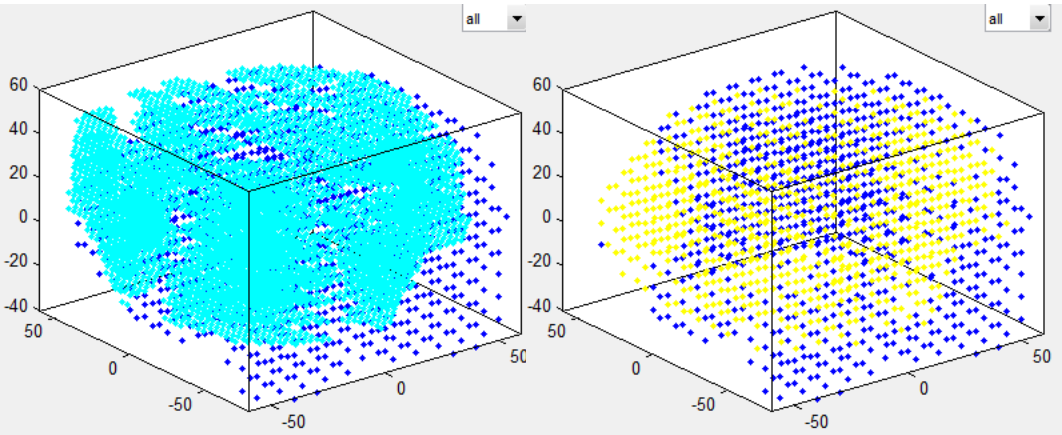

Figure 3-5: Approximate mapping of 1,471 NeuCube neurons displayed in dark blue dots. The cyan and yellow dots are the fMRI input data before initialization (left) and after initialization (right) respectively.

In general, the NeuCube architecture is able to receive and process STBD such as fMRI and EEG as well as other type of spatio-temporal data like sound, image, weather and stroke data. With fMRI data, this mapping technique is required to determine which NeuCube^{B} neurons will be the input neurons and which fMRI data are to be considered as part of the input neurons. The mapping technique which is one of the novel contributions of this study will be elaborated in detail in Chapter 6.

3.5 CHAPTER SUMMARY

This chapter describes the NeuCube architecture that was proposed to be a unifying approach for STBD and other spatio-temporal data learning. NeuCube allows for the combination in one model of various brain data, information and knowledge related to a single subject or to a population of subjects. It can be used not only to discover functional connectivity from data, but also as a predictive system of brain activities. In addi-

tion to this, further analysis of the internal structure of the model after learning process can reveal important spatio-temporal relationships ‘hidden’ in the data.

In the next chapter, a review on the fMRI is presented. Topics discussed will include the general definition of fMRI, other imaging techniques, existing methods for fMRI data analysis and feature i.e. voxel selection in processing the vast and noisy fMRI data.

Chapter 4

REVIEW OF FUNCTIONAL MAGNETIC RESONANCE IMAGING (fMRI)

This chapter reviews the type of data that will be used in the study, which is known as fMRI. fMRI is composed of voxels, in its smallest form, that will be used as features to be experimented and studied. These voxel values are in fact the intensities of the fMRI images that actually reflect the brain functional activities in different parts of the brain. Thus, these values can be directly exploited as features for classification purposes (Mitchell et al., 2003, 2004; Mourão-Miranda, Bokde, Born, Hampel, & Stetter, 2005). The chapter will start with types of brain imaging techniques used in the medical field and continue with functional magnetic resonance imaging in a greater detail, in the next sub-section. Then readers will be presented with previous studies and methods for classifying fMRI data, mainly using conventional classifiers and several studies on selecting relevant features as samples for the experiment.

4.1 INTRODUCTION TO IMAGING TECHNIQUES

Neuroimaging produces images of the structure or activity of the brain or other parts of the nervous systems that let physicians and researchers observe activation of the brain or other nerves. Some may refer neuroimaging as imaging technique that focuses only on human brain. The images are used not only to diagnose and assess brain diseases but also used to study how the brain works, to study which brain regions are activated when certain stimulus are presented to the subject and to investigate which brain locations are affected psychologically (involved with emotion and feelings of the subject) and other functions that may be of potential practice in the future. There are a few safe neuroimaging techniques that are widely available in many hospitals and research facilities around the globe, which include Magnetic Resonance Imaging (MRI) and functional MRI (fMRI), Computed Tomography (CT), Positron Emission Tomography (PET), EEG, MEG and NIRS.

In contrast to PET and CT scans that expose subjects with short-lived radioactive material and x-rays respectively, MRI/fMRI, EEG, MEG and NIRS measure the brain activity without having to expose patients with radiation or injecting substance to

patients' blood flow. However, modalities such as EEG and MEG lack the capability of detecting spatial location of activation area accurately. NIRS have lower signal-to-noise ratio but are however highly correlated with fMRI measurement (Cui, Bray, Bryant, Glover, & Reiss, 2012) and thus gaining the attention for researchers as a complement to fMRI.

4.2 FUNCTIONAL MAGNETIC RESONANCE IMAGING (fMRI)

fMRI can be used to visualize hemodynamic response in relation to neurons' activities in certain part of the brain (Buxton, Uludağ, Dubowitz, & Liu, 2004). This hemodynamic response is indicated by the increasing amount of blood flow to that particular activated neurons area. The components of hemodynamic response include the changes in the oxyhemoglobin and deoxyhemoglobin concentration, in the cerebral blood volume (CBV) per unit of brain tissue and in the cerebral blood flow rate. There are different fMRI techniques that can capture the functional signals generated from the different components of hemodynamic response. One of the most common techniques is based on the concentration of oxyhemoglobin-deoxyhemoglobin component and it is known as blood-oxygen-level-dependent (BOLD) technique (Ogawa, Lee, Kay, & Tank, 1990).

While MRI provides structural mapping of a brain, fMRI imaging technique combined with blood-oxygen-level-dependent (BOLD) technique (Ogawa et al., 1990), produced a better set of brain images, i.e. with excellent temporal as well as spatial information. In addition to structural mapping, fMRI generates functional mapping of the brain that actually takes advantage of iron in the blood-carrying-oxygen and blood-vessels-dilation physiological principle that occurs in activated regions. It is used to measure neural activity changes in the brain resulting from stimuli triggered externally or internally (Haxby et al., 2001). More precisely, fMRI measures the ratio of oxygenated haemoglobin to deoxygenated haemoglobin in the blood with respect to a control baseline, at many individual locations within the brain. It is widely believed that blood oxygen level is influenced by local neural activity, and hence this blood BOLD response is generally taken as an indicator of neural activity (Mitchell et al., 2004).

fMRI imaging technique is non-invasive and radiation-free thus providing a safe environment to the subjects involved. The images are recorded in sequence either vertically or horizontally (Figure 4-1), and over time, in a matrix of intensity values. They are captured in slices through the organs; generally in 8 or 16-bit (Figure 4-1 right).

There are a number of common formats in which the images are saved such as in DICOM, ANALYZE, NIFTI format or in raw voxel intensity values in MATLAB.

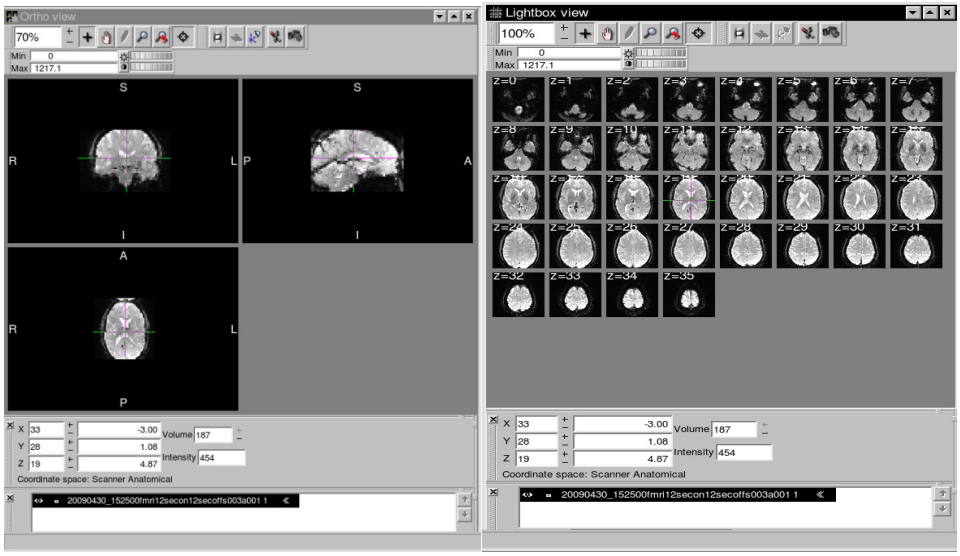

Figure 4-1: Brain images in vertical and horizontal slice: in sagittal, coronal and axial views (left). Slices of brain taken over time i.e. 32 images for a volume of brain (images are viewed using FSLView (FSLView, 2012) software (right).

The images are constructed from two components – spatial/spectral (or spatio) and temporal. The first component is identified as the volume of a brain that can be further sub-divided into smaller 3D cuboids, known as voxels (volume element). In a typical fMRI study, a series of brain volumes are collected in quick succession and the value of BOLD response at all points in a 3D grid are recorded. A general 3D brain image typically contains 10,000 to 15,000 voxels, and each voxel consists of on the order of hundreds of thousands of neurons. Spatial image resolutions can be set either to have low or high resolution. As in Figure 4-2 while high-resolution image provides more accurate information (e.g. voxels with dimensions of 1 mm x 1 mm x 1 mm) more CPU processing power is required and is not feasible at the moment. Typical spatial resolution is 3 mm x 3 mm x 5 mm, corresponding to image dimensions in the order of 64 x 64 x 30 (Lindquist, 2008) and still this resolution is relatively high compared to other imaging techniques.

The temporal component is acquired while scanning the whole volume of a brain that will take a few seconds to complete. In a single run of an experiment, 100 or more brain volumes are usually scanned and recorded for a single subject doing a particular

sensorimotor or cognitive task. Temporal component depends on the time between acquisitions of each individual image, or the time of repetition (TR). In a typical experiment, TR ranges from 0.5 to 4.0 seconds and TR values in the range of 2 seconds are generally considered adequate (Lindquist, 2008).

The combination of this spatial and temporal information of the brain images will be the main concern investigated in this study.

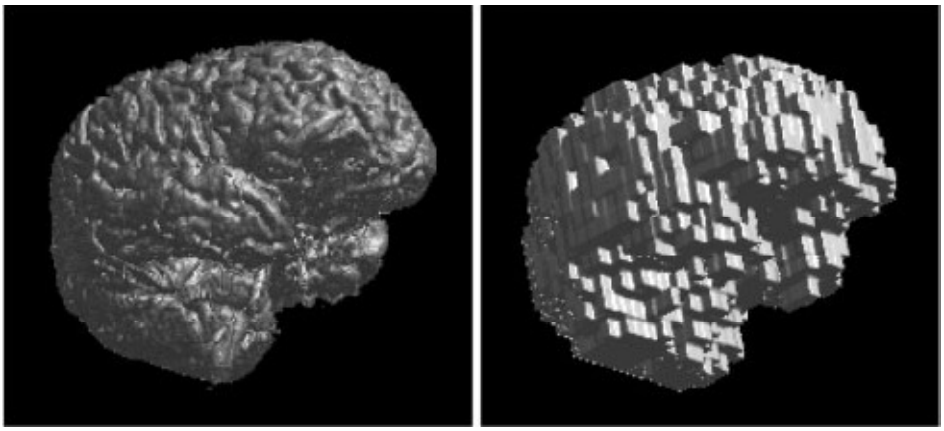

Figure 4-2: Surface renderings of 3D brain images. Small voxels (left, with 1 mm x 1 mm x 1.5 mm) versus large voxels (right, with 7 mm x 7 mm x 10 mm) (Smith, 2004)

4.3 EXISTING METHODS FOR fMRI DATA ANALYSIS

Choosing the best technique for fMRI data analysis is still a question that needs to be answered properly. There are many variables that have to be considered, for instance the weak signal of voxel of interest, the voxels being distributed among various spatial locations of the brain, different brain mapping of different brain sizes and spatially distributed noise. A popular analysis approach is pattern classification, where the brain patterns are observed to forecast the task being performed by the subject.

Naturally, brain activities are captured as fMRI data in a spatio-temporal format. In conducting the analysis, researchers often treat fMRI data classification either in univariate or multivariate; linear or non-linear; or as static or spatio- temporal approach.

In early years, the standard fMRI data analysis approach examines each brain voxel in isolation (univariate) as static data as suggested in Statistical Parametric Mapping (SPM) (Friston, Frith, Frackowiak, & Turner, 1995), which completely disregard the inherent spatio-temporal characteristics of fMRI data. Univariate approach processes fMRI voxels as independent individuals, thus no interaction and no relationship

are measured among the voxels. This approach has been experimented with Gaussian Naïve Bayes method (Mitchell et al., 2004; Schmah, Zemel, Hinton, Small, & Strother, 2010). However this approach neglects the collective information encoded by voxels patterns (Carroll, Cecchi, Rish, Garg, & Rao, 2009).

Multivariate analysis on the other hand, evaluates the correlation of brain patterns across the brain regions rather than examining them on a voxel-by-voxel basis. Haxby et al. (2001) illustrated how multi-voxel patterns of activity can be used to distinguish between cognitive states when subjects were shown faces, houses and a variety of object categories. As different brain locations are triggered with the same (or different) stimulus, experiments should consider all relevant voxels instead of just considering a single particular voxel. This multi-voxel pattern analysis has been adapted by many researchers with various classifiers using either linear or non-linear classifiers: SVM (Cox & Savoy, 2003; Kamitani & Tong, 2005; Misaki, Kim, Bandettini, & Kriegeskorte, 2010; Mourão-Miranda et al., 2005), Gaussian Naïve Bayes (GNB) (Mitchell et al., 2003; Rustandi, 2007a, 2007b), neural networks without a hidden layer (Polyn et al., 2005), non-linear SVM (Fan, Shen, & Davatzikos, 2006; Misaki et al., 2010) and neural networks with hidden layers (Hanson, Matsuka, & Haxby, 2004). All these studies only consider data at a single time or time interval.

Another fMRI classifier approach is whether they are linear or non-linear. While linear classifier divides the classes with a linear plane, non-linear classifier separates the classes using a more complex function (Yourganov et al., 2014). Works related to fMRI on linear classifier includes (Haynes & Rees, 2005; Kamitani & Tong, 2005; Ku, Gretton, Macke, & Logothetis, 2008; Misaki et al., 2010; Mitchell et al., 2004). Although this approach is more biased and less flexible than the non-linear classifier, several studies suggest that they could still generate accurate results (Ku et al., 2008; Misaki et al., 2010). Non-linear classifier on the other hand has also produced good analysis (Fan et al., 2006; Schmah et al., 2010) although some other studies suggest that it produces the worst result (LaConte, Strother, Cherkassky, Anderson, & Hu, 2005; Misaki et al., 2010). However, for robust classification, non-linear classifier requires larger training set (Morch et al., 1997).

In recent years, researchers are moving towards brain analysis that embed both spatial and temporal behaviour such as spatial-temporal SVM (Mourão-Miranda, Friston, & Brammer, 2007), Bayesian formulation(van Gerven, Cseke, de Lange, & Heskes, 2010) and Generalized Sparse Classifier (GSC) (Ng & Abugharbieh, 2011b). A research conducted by Avesani, Hazan, Koilis, Manevitz, & Sona (2011) selects a set

of relevant voxels using General Linear Model (GLM) and then incorporates liquid state machine and Multi-Layer Perceptron (MLP). These researches focus on spatio-temporal classification, where multiple brain volumes within a trial are treated as a sample.

In conclusion, the study of fMRI characteristics and its relation with the behaviour of a classifier is still not well comprehended. Typically, fMRI datasets are ill-posed datasets that require massive computational power to process their voxels. In addition to this, the interaction of the classifier properties with BOLD signal properties of fMRI is still not well and treated. To regularize this ill-conditioned problem (i.e. limited number of samples together with severe noise), this study proposes to better exploit the spatio-temporal structure of fMRI data using probabilistic Spiking Neural Networks (pSNN) (Kasabov, 2010). In particular, this study will employ ESNN to incorporate spatio-temporal characteristics of fMRI data into the classifier.

4.4 SELECTING FEATURES OF fMRI DATA

In a typical fMRI experiment, a sequence of images related to the subject's brain activity every half seconds will be produced. The experiment usually consists of a set of trials and each trial produces many brain volumes over time. Each brain volume is comprised of voxels in the order of thousands and these voxels' intensities are the features to be classified. Learning this brain data poses many challenges especially in terms of the data being extremely sparse noisy data and high dimensional. This would cause over-fitting problem for the classifier. Hence it is necessary to apply feature selection methods to make learning tractable and to prevent over-fitting.

In selecting relevant features (voxels) that respond from a stimulus, it can either be done in a univariate or multivariate manner. Apart from the standard univariate approach (Friston, Holmes, Worsley, Poline, Frith, and Frackowiak, 1995) multivariate pattern analysis approaches towards detection of ROI from fMRI data have been gaining a lot of attention recently. The advantage of multivariate method stem from the fact that even voxels with weak individual response may carry important cognitive information when analysed together.

Evolutionary feature selection is an algorithm that is based on evolutionary techniques. This approach was proven effective in feature subset selection that detects which number and combination of individual voxels that carry information relevant to a stimulus (Åberg, Löken, & Wessberg, 2008). These voxels are used as features in mul-

tiple linear regression (MLR) classifier and they proved that even the simple classification scheme can detect and distinguish relevant cortical information in noisy fMRI data. Although it considers voxels in multivariate way (analyse voxels collectively), voxels are only on a single volume and not tested on multiple volumes over time.

Another approach uses particle swarm optimization (PSO) based fMRI brain state classification algorithm, specifically designed to efficiently extract a subset of voxels optimal for classification task (Niiniskorpi, Bj, & Wessberg, 2009). PSO is a stochastic optimization method (Kennedy & Eberhart, 1995) loosely based on the behaviour of swarming animals such as fish and birds. A number of particles, representing potential solution to the problem, are released in the search space of potential solutions. Each particle has a position and a velocity, and is free to fly around the search space. And in the case of feature selection, this standard PSO is modified as proposed by Wang, Yang, Teng, Xia, & Jensen (2007) and not only achieves high performance scores but also identifies functionally relevant ROI (Niiniskorpi et al., 2009).

In addition, methods that simultaneously select relevant voxels have been proposed which extend traditional classifiers by incorporating sparse regularization, which controls over fitting by encouraging zero weights to be assigned to irrelevant voxels (Carroll et al., 2009; Ryali, Supekar, Abrams, & Menon, 2010; Yamashita, Sato, Yoshioka, Tong, & Kamitani, 2008). And these works have been improved in the recently proposed Generalized Sparse Classifiers (GSC) (Ng, Vahdat, Hamarneh, & Abugharbieh, 2010) that permits more general penalties, such as spatial smoothness in addition to it being sparse and to be seamlessly integrated. Another improvement was Generalized Group Sparse Classifiers (GGSC) (Ng & Abugharbieh, 2011a) that permits associations between features within predefined groups to be modelled.

4.5 CHAPTER SUMMARY

This chapter presented a review of fMRI. It first discussed the imaging techniques that exist in medical science which include fMRI, MRI, PET, EEG, MEG and NIRS. In contrast to other imaging technique, the advantages offered by fMRI are: high visualization quality of the brain coupled with good temporal characteristic; motivate researches in brain analysis and brain understanding. Numerous studies have been, and still are, conducted to better understand the brain function: not to just detect brain disease or illness but also to have a greater understanding of how the brain works, such as to locate the brain area affected the most by a certain stimulus or to classify brain images as to be

in a certain class or the other e.g. healthy or not healthy; or doing task A or doing task B etc.

In the next chapter, StarPlus data (Mitchell, 2005) downloaded from Carnegie Mellon University's Centre for Cognitive Brain Imaging (CCBI) is described, used and tested on conventional classification models, which include SVM and MLP. Results from both experiments will be explained as well.

Chapter 5

REVIEW OF EXISTING TECHNIQUES TO STARPLUS fMRI DATA

The document continues with the review of existing technique that has been applied to the benchmark dataset. It first describes the dataset used in the study which is StarPlus data that are freely made available by Carnegie Mellon University (CMU). This dataset is experimented using conventional classifiers, which are Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). Preliminary result will also be described later in the chapter.

5.1 ABOUT STARPLUS DATA

Publicly available StarPlus dataset is utilized to evaluate the performance of the proposed framework; which was used by many researchers before (Hutchinson, Niculescu, Keller, Rustandi, & Mitchell, 2009; Mitchell et al., 2004; Ng & Abugharbieh, 2011a; Shinkareva et al., 2008).

In a StarPlus experiment, the experiment is set into 2 settings and the brain volumes captured are divided into a set of trials. In each trial, subjects are required to look at a Picture (or Sentence) followed by a Sentence (or Picture), and to decide whether the Sentence (or Picture) correctly describe the Picture (or Sentence). In the first setting, the subject first sees a sentence (Semantic stimulus) such as “The plus sign is above the star sign” for 4 seconds followed by a blank screen for another 4 seconds and finally another stimulus (Picture stimulus) is displayed for the next 4 seconds, during which the subject must press a button for “yes” or “no”, depending on whether the sentence matches the picture seen or not. Brain snapshots are taken every 500 ms (or TR is 500 ms). The subject then has 15 seconds of rest period before proceeding to the next trial. The subject is instructed to rehearse the sentence in his/her brain until the picture is presented rather than to visualize the sentence immediately. All experiment conditions are labelled with 0, 1, 2 and 3 referring to Ignore, Rest, Sentence is Not Negated and Sentence is Negated as tabulated in Table 5-1. In the second experiment setting, the presentation of sentence and pictures are switched, and the subject is instructed to remember

the picture until the sentence is presented. For both settings, the dataset is labelled as SP (Sentence-Picture) and PS (Picture-Sentence).

The brains volumes are acquired from 6 normal subjects (Mitchell et al., 2004) and they are labelled as data-starplus-04799-v7.mat, data-starplus-04820-v7.mat, data-starplus-04847-v7.mat, data-starplus-05675-v7.mat, data-starplus-05680-v7.mat and data-starplus-05710-v7.mat. For the preliminary test, data for a single subject data-starplus-04820-v7.mat is chosen and analysed. There are 54 trials in which the experiment conditions are labelled as follows:

Table 5-1: Experiment Conditions

Condition label	Experiment Condition
0	Ignore
1	Rest (fixation)
2	Sentence is not negated
3	Sentence is negated

For each trial there are approximately 54 volumes snapshots x 5,015 voxels. To account for delay in the hemodynamic response, only 8 volumes collected 4 seconds after stimulus start were used (Ng & Abugharbieh, 2011b).

This experiment involves certain part of the brain: visual cortex for reading and seeing the Sentence, Broca's area for language processing, Intra Parietal Sulcus for spatial visualization, motor cortex for pressing the button and so on. The accurate way in which part of the brain is activated varies across subjects, based on their mental and verbal abilities and also based on the experiment setting (i.e. mentally rehearsing the sentence until he/she sees the picture or remembering the picture as he/she reads the sentence and then matching the picture being shown).

Each dataset (per subject per trial) comprised approximately 5,015 voxels which are within the 25 ROIs that are chosen by neuroscience experts which are Calcarine Sulcus (CALC), Bilateral Dorsolateral (BD), Frontal Eye Field (FEF), Inferior Parietal Lobule (IPL), Intraparietal Sulcus (IPS), Inferior Temporal Lobule (ITL), Opercularis (OPER), Posterior Precentral Sulcus (PPREC), Supramarginal Gyrus (SMG), Superior Parietal Lobule (SPL), Temporal Lobe (T), Triangularis (TRIA) and Supplementary Motor Area (SMA) (as in Figure 5-1). Out of these 25 ROIs, only regions labelled as CALC, LDLPFC, LFEF, LIPL, LOPER, LT and LTRIA as in Table 5-2 will be considered in the classification task.

Table 5-2: ROIs considered in the classification task and the number of features in each region for subject S04820

Label	Region	Main function	Number of features
CALC	Calcarine Sulcus	primary visual cortex	408 x 8 = 3264
LDLPFC	Left Bilateral Dorsolateral Prefrontal Cortex	motor planning, working memory, short-term memory	501 x 8 = 4008
LIPL	Left Inferior Parietal Lobule	perception of emotions, information interpretation	60 x 8 = 496
LIPS	Left Intraparietal Sulcus	perceptual-motor coordination (for directing eye movements and reaching), visual attention, working memory	155 x 8 = 1240
LOPER	Left Opercularis	speech-language production	103 x 8 = 824
LT	Left Temporal Lobe	auditory perception, speech and vision processing, memory	484 x 8 = 3872
LTRIA	Left Triangularis	Speech-language production	175 x 8 = 1400

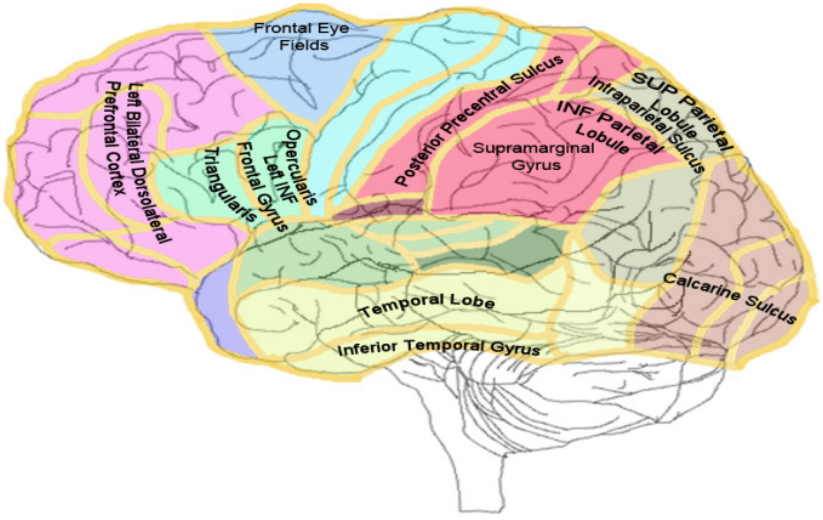

Figure 5-1: 25 anatomically defined region of interest in human brain used in the study

5.2 EXPERIMENT SETTING

In this study, intensity value for each voxel, at each time point is treated as feature, and all brain volumes within the same trial of each experimental condition as sample. In other words, where temporal information is disregarded, the 8 volumes (i.e. 8 time points) are concatenated and treated as a single sample (Ng & Abugharbieh, 2011b). As in Table 5-2, for example, from the downloaded data, there are 3,264 features (408 features x 8 time points) for CALC region and 4,008 features (501 features x 8 time points) for LDLPFC region. For both SVM and MLP experiments, brain volume (i.e. in terms

of features or voxels) extraction procedure for SVM and MLP classifications is as follows:

- Select trials with experiment condition labelled as 2 (Sentence is not Negated) and 3 (Sentence is Negated) (Table 5-1). These conditions are when the brain is activated from seeing a Sentence or Picture stimulus.

- Select CALC, LDLPFC, LFEF, LIPL, LOPER, LT and LTRIA regions and discard other voxels from other regions. These regions receive the most activation from the stimulus (Mitchell et al., 2004).

- As there are 54 trials, for the first 27 trials where the first stimulus is Picture (i.e. PS setting), select brain volumes of 9-16 as Class 1 (Picture Class) and 25-32 as Class 2 (Sentence Class) and for the next 27 trials where the first stimulus is Sentence (i.e. SP setting), select brain volumes of 25-32 as Class 1 (Picture Class) and 9-16 as Class 2 (Sentence Class).

- Concatenate all brain volumes within the same trial to become a sample. Thus the resulting matrix will be 80 samples x 3,264 features for CALC region and 80 samples x 4,008 features for LDLPFC region. There are 40 samples for Class 1 (subject looking at a Picture) and 40 samples for Class 2 (subject looking at a Sentence).

- Normalize the voxels intensity values

- Select 50 and 100 voxels of highest priority by using Signal-Noise-Ratio (SNR) technique. All features from each ROI were ranked from higher priority to lower priority by selecting only the 50 and 100 most significant features (Refer to Figure 5-2 and Figure 5-3). This feature extraction process is carried out using NeuCom©.

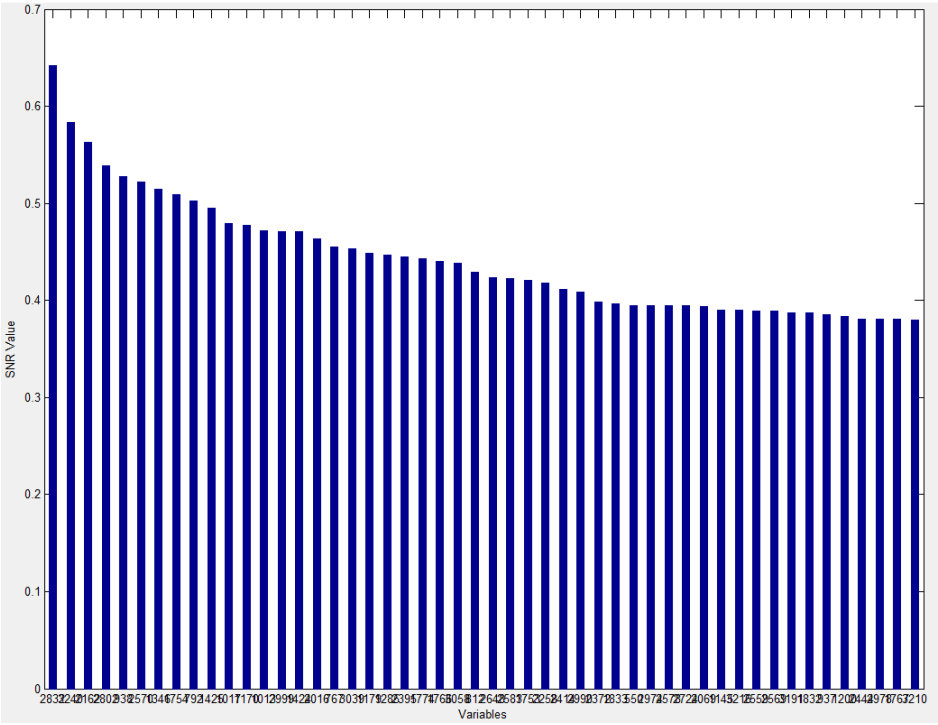

Figure 5-2: 50 most significant features are selected using SNR data analysis for CALC region.

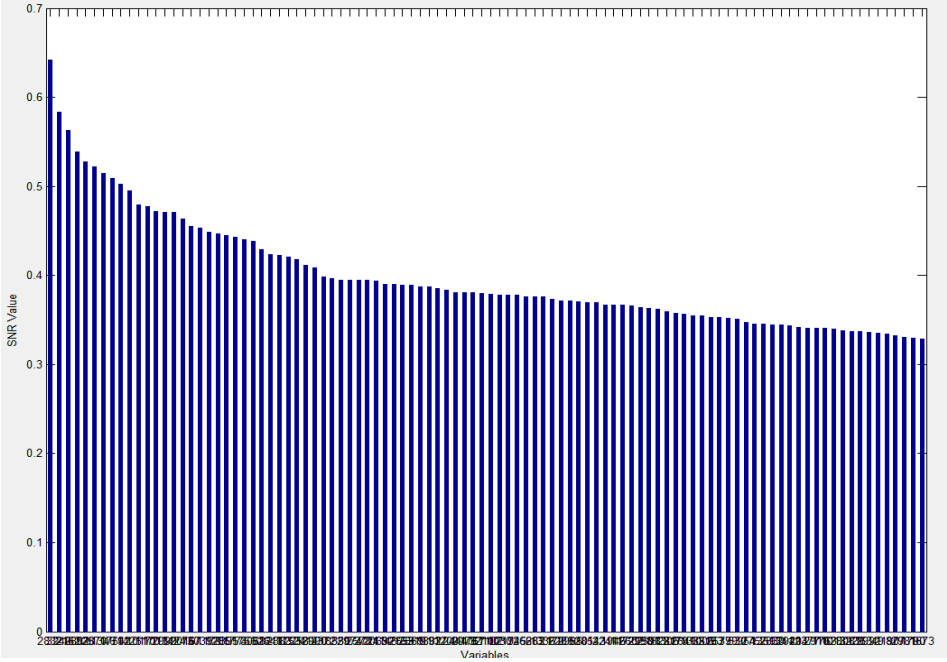

Figure 5-3: 100 most significant features are selected using SNR data analysis for CALC region

The parameters are tuned using a blind search approach yielding the final parameter settings as the following:

- Polynomial kernel with degree 2, 5-fold cross validation mode for SVM;
- 80 hidden neurons and 300 training cycles for MLP.

5.3 EXPERIMENT RESULT

The experiment is performed to measure the classification accuracy when a subject is presented with a static stimuli i.e. either a subject is looking at a Picture or a Sentence by running a conventional classifier: SVM, or MLP. As in Table 5-3, in general MLP results in the highest accuracy for every region, with the average 94% accuracy, except for CALC being only78%, as compared to 89% accuracy for SVM.

Table 5-3: Classification accuracy percentage for each region using SVM and MLP classifiers (A=SVM classifier, B=MLP classifier)

	CALC		LDLPFC		LIPL		LIPS		LOPER		LTRIA		LT	
Model	A	B	A	B	A	B	A	B	A	B	A	B	A	B
50 features	89	78	91	93	90	91	95	91	91	90	93	93	98	96
100 features	91	89	98	99	84	88	94	98	89	94	96	98	95	98

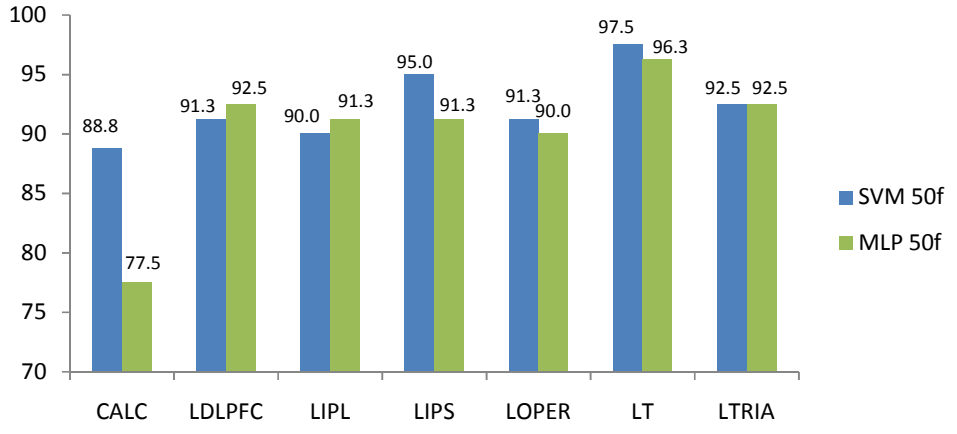

Figure 5-4: Percentage of classification accuracy for 50 features selected based on SNR

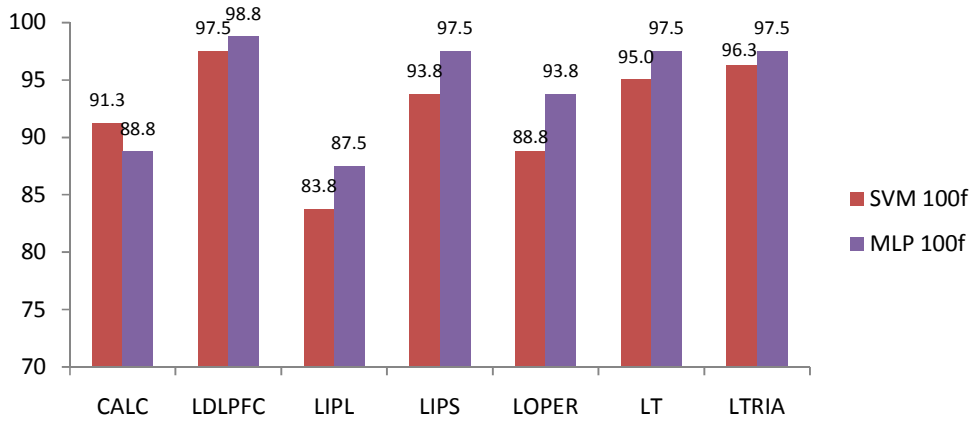

Figure 5-5: Percentage of classification accuracy for 100 features selected based on SNR

For comparison purposes, for the experiment involving only 50 most significant features (Figure 5-4), SVM produces higher classification accuracy as opposed to MLP in almost all regions, except for LDLPFC and LIPL regions. This is because LIPL region involves only in the perception of emotions and in the interpretation of sensory information, whereas the experiment involves more on rehearsing and remembering. The highest classification accuracy is for LT region for both SVM and MLP classifiers (98% and 96% respectively). This is because LT region involves processing vision semantics i.e. the brain is processing information when presented with a Picture/Sentence stimuli. In contrast, CALC region produced the lowest accuracy for both classifiers. This is because even though CALC is where the primary visual cortex is concentrated, the experiment setting highly involve the subject to read and to remember the Picture or Sentence that he/she is looking at rather than just looking at the stimulus.

As features were increased to 100 (Figure 5-5), different patterns of classification accuracies were generated. Generally, MLP algorithm generated higher classification accuracy for all ROIs, except for CALC region in which SVM outperformed the result (the same situation in 50 features experiment). LDLPFC region produced the highest classification accuracy (98% and 99%), whereas LIPL region produced the lowest accuracy (84% and 88%), for both SVM and MLP respectively.

From both experiments conducted, in some situations, increasing the number of features does not necessarily mean the classification accuracy will be increased too, e.g. in the case of LIPL, for both types of classifiers. In other region (e.g. LT), increasing the number of features will increase the classification accuracy for a particular classifier

(MLP) but will decrease the classification accuracy for another classifier (SVM). MLP performed better than SVM if the number of features is increased from 50 to 100 features for almost all ROIs with the average accuracy classification increases from 90% to 94%.

However, accuracy generalization could not be concluded because of the following issues:

- The experiment was conducted based on individual ROI and did not consider collective information on all ROIs i.e. some regions may be stimulated by the same task condition.

- Only 50 or 100 most significant voxels were selected whereas in real situation, even voxels with the weakest intensity value (irrelevant voxels) may have significant impact if they are to be considered (in the classification procedure) with other voxels with the strongest intensity values.

- The classification method does not allow any new knowledge discovery.

5.4 CHAPTER SUMMARY

This chapter discussed a preliminary experiment on the benchmark dataset using the standard computational intelligent techniques, SVM and MLP. The experiment is for 2 Class problems selected based on condition 2 (Sentence is not Negated) and condition 3 (Sentence is Negated), provided from the StarPlus, involving 7 regions labelled as CALC, LDLPFC, LFEF, LIPL, LOPER, LT and LTRIA. From these regions, 40 samples for Class 1 and 40 samples for Class 2 from each region are selected and tested.

In general, increasing the number of features will increase the classification accuracy for almost all regions except for LIPL, for both classifiers. This is anticipated because the voxels that are selected for the experiment are voxels that are most active (selected based on SNR) i.e. 50 or 100 voxels with the highest intensity values. Voxels having weak intensity values are considered irrelevant in this test. However, for LIPL region, adding more active voxels add more noise rather than information thus resulting in lower percentage accuracies as compared to the other regions. This indicates that not only active voxels carry valuable information; the less active voxels also may have sig-

nificant impact when evaluated collectively with active voxels from the same or other regions.

This test is only a preliminary experiment conducted on the benchmark dataset downloaded from the StarPlus project. In the next chapter, the same dataset will be tested on the proposed framework involving NeuCube^{B} structure as a reservoir, STDP as a learning rule and deSNN as the classifier. deSNN classifier will be trained in a supervised mode to classify the spike trains of the NeuCube^{B} into the pre-defined classes.

Chapter 6

NEUCUBE BASED METHODOLOGY FOR fMRI DATA MODELLING

This chapter explains the newly proposed methodology used in the research which is based on the networks of spiking neuron of the NeuCube. Previous experiments involving NeuCube and spatio-temporal data have been conducted successfully and further exploration on its capability towards other STBD learning and classification are very much desirable. The same architecture and principles of NeuCube which include the unsupervised and supervised mode of learning are used in the study. However new modifications and new methods specifically in terms of data mapping, which is one of the main contributions in this thesis, is implemented to ensure that the intended STBD i.e. fMRI, could be recognized, learned and classified by the architecture. We also proposed a new method for visualization and interpretation of NeuCube-based model of the fMRI data. This novel and new version of NeuCube is identified as NeuCube^{B} (Figure 6-1) to represent the specific NeuCube version that purposely implemented and tested for fMRI brain data learning.

The existing NeuCube is designed to model, map, learn and classify many types of STD and it has been tested on stroke (Othman et al., 2014) as well as fMRI (Doborjeh, Capecci, & Kasabov, 2014) and EEG data (Doborjeh, Wang, Kasabov, Kydd, & Russell, 2015). In the experiment involving stroke occurrences (Othman et al., 2014), all features of stroke (i.e. input neurons) are automatically mapped close to each other on a 3D SNN architecture and this architecture size is flexible according to the input neuron. As for the fMRI experiment, only certain voxels are selected as input feature and they are mapped into the NeuCube 1,471 neurons, while maintaining the spatial information (Doborjeh et al., 2014). In the EEG experiment, a 3D brain-like SNN architecture is created in which input neurons are mapped to the same (x, y, z) correspondingly variables' coordinate of the brain template to preserve the spatial information. As for the new version of NeuCube, it is built in a 3D brain-like architecture consisting of 1,471 neurons, which is based on the 1 cm^3 of the standard Talairach Template, where all voxels will be considered as input features to the networks of spiking neurons.

The following sections describe the overall proposed definition of NeuCube[B] and continue with the reviews on the three levels that defines the architecture. The levels are the input module, the NeuCube[B] module and the output module. In the lowest level (input), possible data encoding methods that may be used in the architecture are described. Next, the module where most of the crucial processing is performed is explained (NeuCube[B] module), especially in terms of its structure, the neuron model used, the learning rule employed and evolvability nature of the cube. And finally the highest module is explained, which involves the output module where all the trained data in the NeuCube[B] module is pumped into a SNN based classifier to be classified into predefined classes.

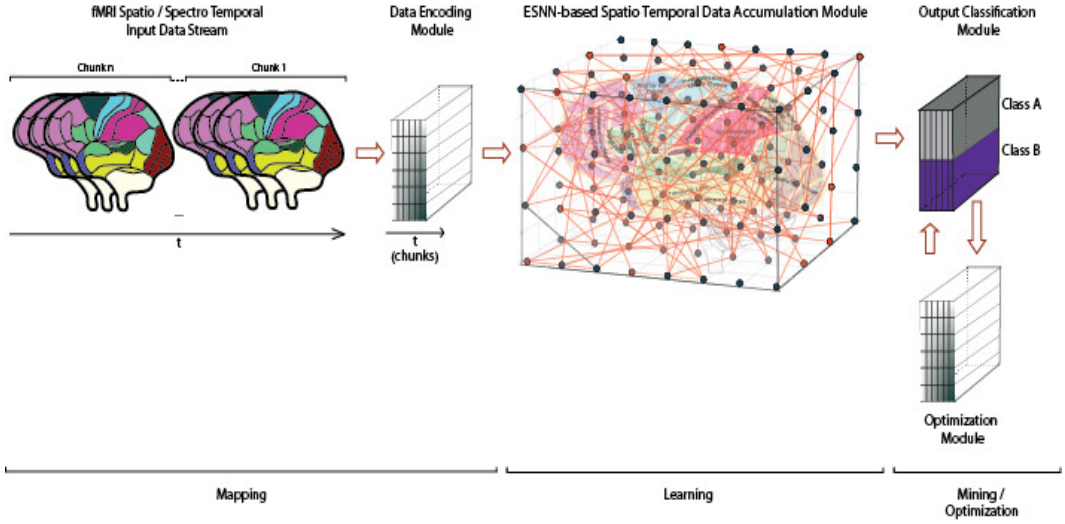

Figure 6-1: Schematic diagram of NeuCube[B] architecture involving mapping, learning and mining phases

6.1 INDIVIDUAL INPUT NEURON MAPPING

Different mapping approaches were used in previous experiments involving spatio-temporal data of fMRI, EEG and stroke (Doborjeh et al., 2014, 2015; Othman et al., 2014). In this study, we proposed a new method for STBD fMRI data mapping that will determine the input features based on every voxels comprising a brain volume. The first step in STDP NeuCube[B] learning is to map each voxels' coordinates into the NeuCube[B] architecture. Voxels coordinates are first extracted from the metadata of the fMRI data. Since NeuCube[B] architecture is built within the Talairach coordinate space,

it is important for the fMRI voxel coordinates to be on the same coordinate space and to have the same centre point. Voxel coordinates mapping involves the following steps:

1. Determine the mean of fMRI voxel coordinates $x_{fMRI\ data}$, $y_{fMRI\ data}$ and $z_{fMRI\ data}$, denoted as x_{mean}, $y_{mean}$and z_{mean}

2. Determine stretch factors d_x, d_y and d_z in each $x_{fMRI\ data}$, $y_{fMRI\ data}$ and $z_{fMRI\ data}$ directions respectively

$$d_x = \frac{x_{minNeuCube}}{x_{minimum} - x_{mean}} \tag{6.1}$$

$$d_y = \frac{y_{minNeuCube}}{y_{minimum} - y_{mean}} \tag{6.2}$$

$$d_z = \frac{z_{minNeuCube}}{z_{minimum} - z_{mean}} \tag{6.3}$$

where $x_{minNeuCube}$, $y_{minNeuCube}$ and $z_{minNeuCube}$ are x, y and z minimum of NeuCube[B], $x_{minimum}$,$y_{minimum}$ and $z_{minimum}$ are x, y and z minimum of fMRI voxel coordinates.

3. Calculate the newly mapped coordinates $x_{newcoordinates}$, $y_{newcoordinates}$ and $z_{newcoordinates}$

$$x_{newcoordinates} = d_x * (x_{fMRI\ data} - x_{mean}) \tag{6.4}$$

$$y_{newcoordinates} = d_y * (y_{fMRI\ data} - y_{mean}) \tag{6.5}$$

$$z_{newcoordinates} = d_z * (z_{fMRI\ data} - \mathrm{z}_{mean}) \tag{6.6}$$

where $(x_{fMRIdata}, y_{fMRIdata}, z_{fMRIdata})$ are the original fMRI voxels coordinates and $(x_{newcoordinates}, y_{newcoordinates}, z_{newcoordinates})$ are the new mapped NeuCube coordinates.

Stretch factors d_x, d_y and d_z are calculated from each of x, y and z coordinates of fMRI raw data. For instance, in determining d_x of StarPlus data, the centre value of $x_{fMRIdata}$ is determined from the mean value of x, such that this center value will be mapped to the center of $x_{newcoordinates}$ of NeuCube[B] and in this case is 0. Let us consider the following situation. StarPlus $x_{fMRIdata}$ are ranging from 12 to 52 while NeuCube[B] $x_{newcoordinates}$ are ranging from -60 to 60 (as depicted in Figure 6-3). Based

on Equation 6.1, the $x$stretch factor is $d_x = \frac{-60}{12-32} = 3$ and this value will be multiplied to each $x_{fMRIdata}$ coordinate as in Equation 6.4 resulting in a new set of $x_{newcoordinates}$. For example StarPlus $x_{fMRIdata} = 32$ will be mapped to NeuCube^B $x_{newcoordinates} = 3 * (32 - 32) = 0$. The same mapping concept is applied in order to determine d_y and d_z and hence to determine y and z coordinates of NeuCube^B.

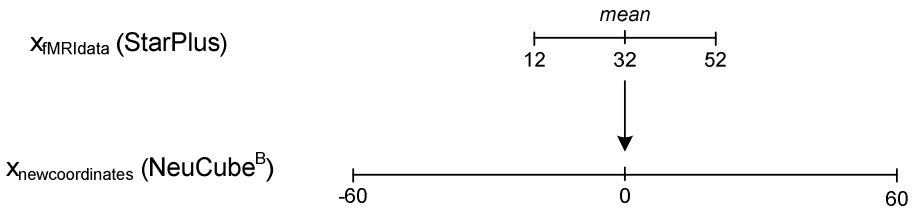

Figure 6-2: Determination of d_x value that will be used in mapping of $x_{fMRIdata}$ into $x_{newcoordinates}$ of NeuCube^B.

The newly calculated voxel coordinates are used in the next step, which is to find those coordinates that are close to NeuCube^B points i.e. to be within a certain radius unit. This is to ensure that, not a single coordinate of the newly calculated mapped coordinates is located outside of the NeuCube^B brain. This mapping is used to determine the sample set. The exact set of sample (i.e. input neurons), which is later used for learning, is generated based on the 1,471 neurons of NeuCube^B.

6.2 MAPPING ILLUSTRATION OF INDIVIDUAL NEURON AS INPUT NEURON

The main and novel feature proposed in the NeuCube^B model is that, both NeuCube^B and fMRI neurons can be visualized in 3D view. Specifically, we proposed a method for visualizing the fMRI data as input neurons in the Cube of NeuCube^B. Visualization is calculated and derived based on the 1,471 of NeuCube^B neurons. Coordinate mapping from the previous section is used to determine the exact set of input neurons to the SNN. Based on each these 1,471 mapped neurons of NeuCube^B, the voxel inputs are defined as those fMRI voxels that are located within a certain radius unit from these neurons. For illustration purposes, in this NeuCube^B model, the radius is assumed to be 7 mm. For example, as illustrated in Figure 6-3, for a NeuCube^B coordinate (indexed as

50) there are 6 fMRI voxel coordinates that are located within the specified 7 mm radius. The voxel values from these 6 coordinates are averaged and this value is used as the exact input to the learning process. Certainly this radius value can be changed i.e. selecting smaller/bigger radius will generate a different number of input features.

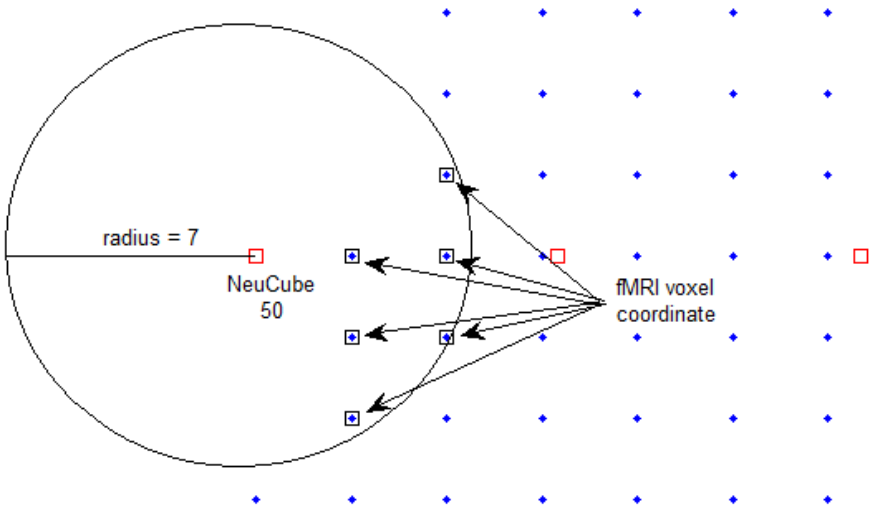

Figure 6-3: NeuCube^{B} 50 and the neighbouring 6 fMRI voxel coordinates that are within 7 mm radius.

6.3 MAPPING ILLUSTRATION OF NEURON CLUSTER AS INPUT

The previous step generated 1,471 clusters of neurons in which not all of these neurons were the input neurons since some of the clusters were empty i.e. without neighbouring fMRI voxel. For mapping illustration, Figure 6-4 shows NeuCube^{B} neurons of 500, 1009 and 1105 from which both neurons 500 and 1105 were the input neurons because there are 8 and 16 neighbouring fMRI voxels around them respectively. Meanwhile neuron 1009 will not be the input neuron because there is no fMRI voxel within the specified radius. The mapping from different subjects produces different numbers of input neurons, which is according to the number of voxels defined for each subject.

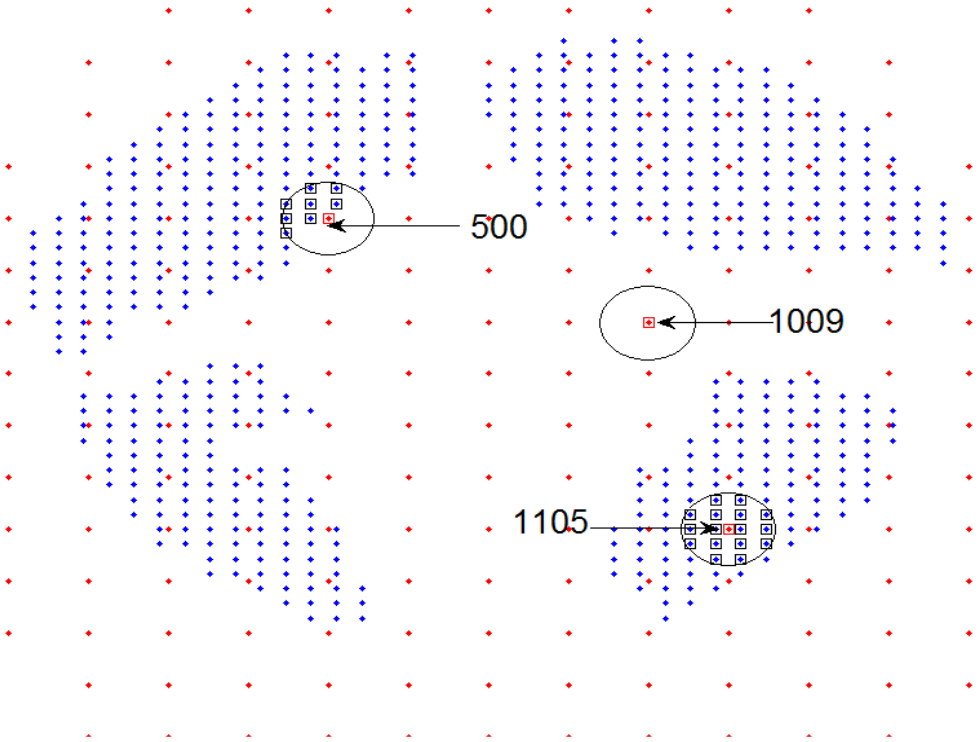

Figure 6-4: NeuCube^{B} 500, 1009 and 1105 with their neighbouring fMRI neurons viewed from xy direction.

6.4 VISUALIZATION

We also proposed that the neurons that are mapped previously (both NeuCube^{B} neurons and fMRI input neurons), are visualized in the same cube. For illustration purposes Figure 6-5 shows these neurons mapping in which the black dots represent NeuCube^{B} neurons while the blue dots represent the fMRI neurons. From the application, we can also easily rotate the NeuCube^{B} to see the neurons from any angle. It is also possible to visualize these fMRI neurons i.e. input neurons in all three directions (in xy, xz and yz views) (Figure 6-6). NeuCube^{B} also provides visualizations of neurons in a separate layer (i.e. z −slice), where each layer is captured in different times. Figure 6-7 shows the NeuCube^{B} neurons which are represented with cyan squares while fMRI neurons are represented with blue squares where each sub-figure displays a slice of neurons. From this figure, we can see that NeuCube^{B} is composed of 11 layers of $z - slice$ while fMRI data (i.e. StarPlus data), is composed of only 8 layers of $z - slice$. These different visualizations are important to ensure that the fMRI neurons are properly mapped within the specified 1,471 NeuCube^{B} neurons, and these mappings were not implemented in the previous studies (Doborjeh et al., 2014, 2015; Othman et al., 2014).

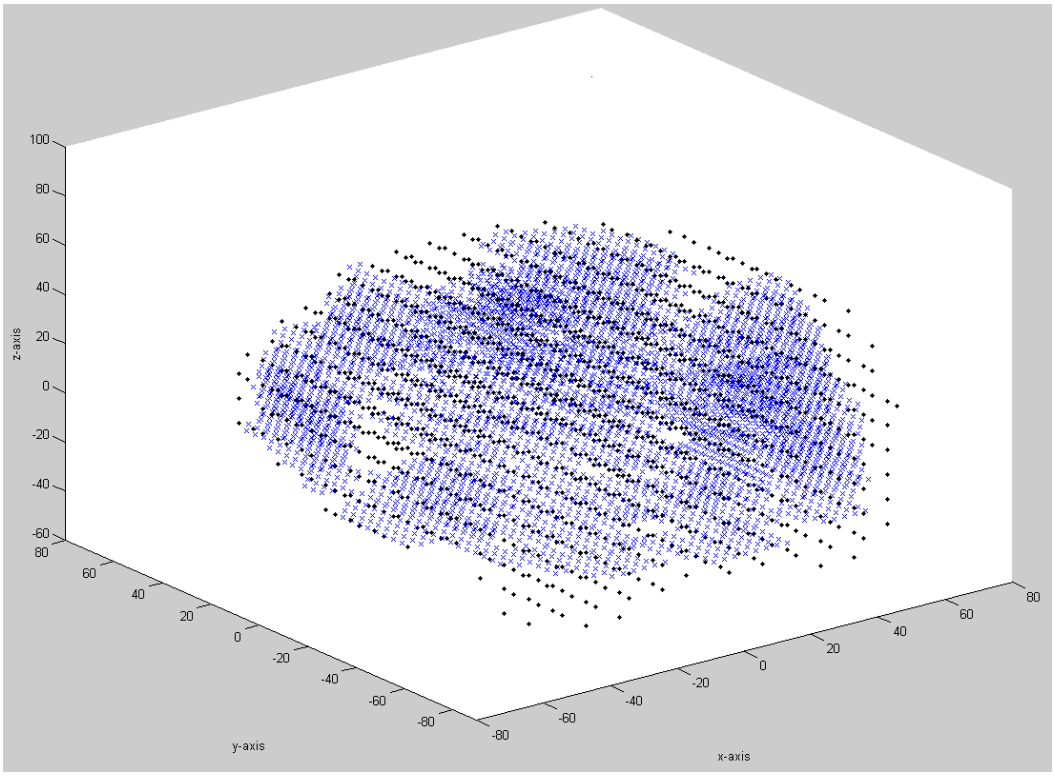

Figure 6-5: NeuCube^{B} neurons (black dots) and fMRI input neurons (blue crosses) mapped and visualized together in 3D view in initialization stage.

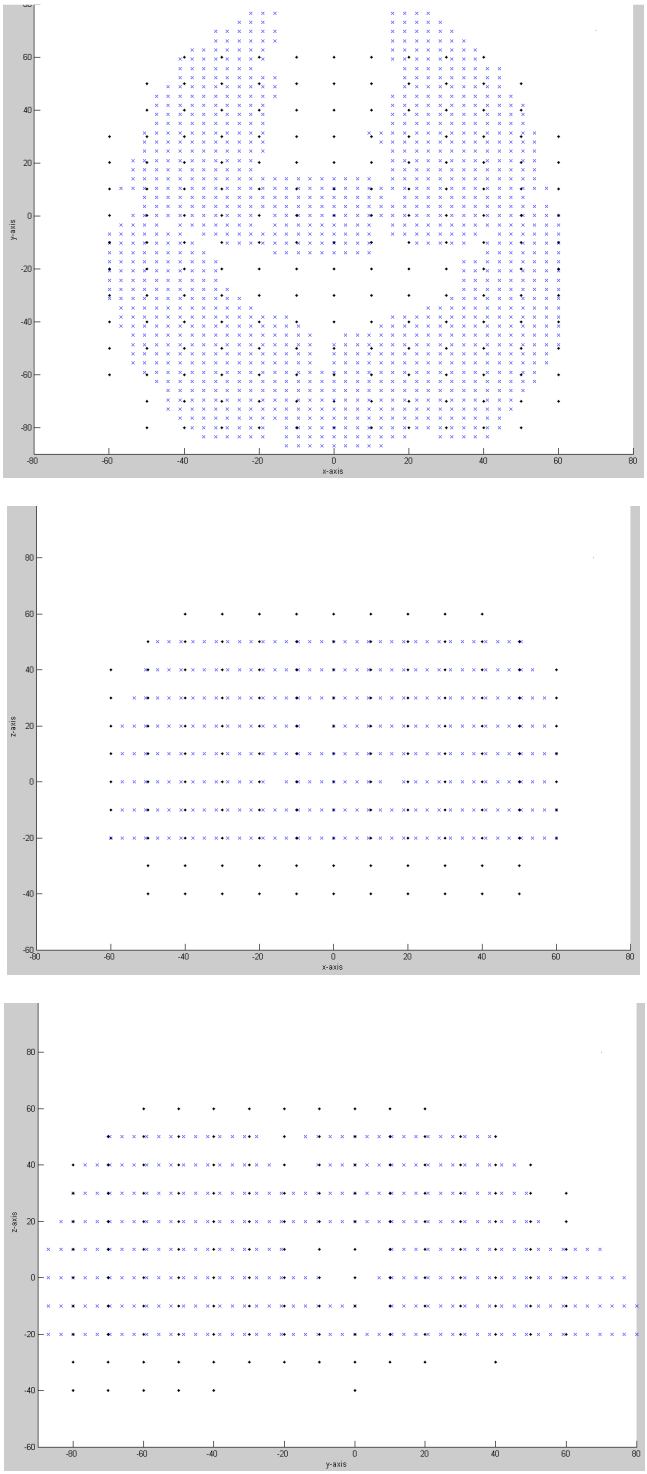

Figure 6-6: NeuCube^{B} neurons (black dots) and fMRI neurons (blue crosses) in three views (in xy, xz and yz views)

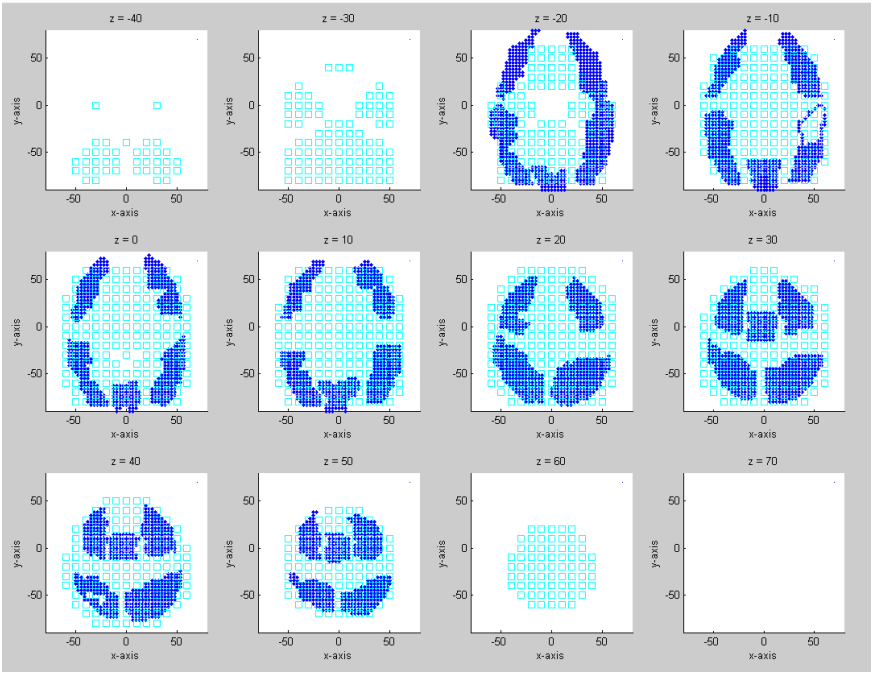

Figure 6-7: NeuCube^{B} (cyan squares) and fMRI (blue squares) input neurons mapping, in each z-slice, in which each sub-figure represent a single slice.

One of the main advantages of the NeuCube^{B} model is that it also allows for visualization in terms of the neurons' state i.e. the state of the neurons before (NeuCube^{B} initial structure); and after unsupervised STDP learning as well as after supervised deSNN learning. This is another contribution to the thesis in which we visualize the neuron states through the connections between the neurons, which are actually small world recurrent connections, created after each training cycles. Following the evolving connectionist system paradigm (Kasabov, 2007), eSNN learn the fMRI data incrementally by one pass propagation by creating and merging the spiking neurons.

In STDP learning, synapses' weight is increased or decreased based on the timing of the post-synaptic action potential in relation to the pre-synaptic action potential. If a pre-synaptic neuron fires first, the connection weight between the two neurons increases (i.e. the difference in the spike time between the pre-synaptic and post-synaptic neurons is negative), otherwise it decreases. Through STDP, connected neurons learn consecutive temporal associations from the fMRI data and new connections are also evolved.

In addition, in the second stage of the model, we proposed that these spatio-temporal fMRI data are fed into deSNN learning (Dhoble et al., 2012), and weights are further calculated and adjusted, using RO and STDP. RO learning will initialize the

connection weights and STDP will make adjustments to those connection weights based on the spikes that follow after that.

6.4.1 Model Interpretation

We also proposed a new method for interpreting the neuron connections created inside the brain-like model of eSNN Cube in correlation with the fMRI spiking activity during the NeuCube^{B} learning. As shown in Figure 6-8, spiking neurons activity before and after unsupervised (or supervised) learning is captured and visualized through connections between neurons that are presented with blue and red lines. Blue lines represent positive weight connections (connection weight > 0) and red lines represent negative weight connections (connection weight < 0). These connections are initialized, created and calculated in each training cycle that actually represents the spatio-temporal connections between input variables in the network. Active neurons are presented with blue dots and inactive neurons are presented with red dots.

During initialization stage, NeuCube^{B} connections and their connection weights are initialized randomly. After training processes, more densely interconnected neurons could be seen in different areas of the NeuCube^{B} resulted from higher positive spikes activity (higher connectivity weight) that have been created and calculated during the training. Neuron connections with the stronger spikes can also be visualized through the model.

Areas that are highly interconnected with positive blue lines are those areas that are highly activated when seeing certain stimulus. Connection with maximum weights which implies stronger spikes can be visualized and the affected brain areas can be identified and labelled. This extraction of new knowledge (stimulated brain areas) is definitely impossible to be gained from the traditional machine learning techniques. For illustration purposes Figure 6-8 shows neurons connectivity before and after STDP learning.

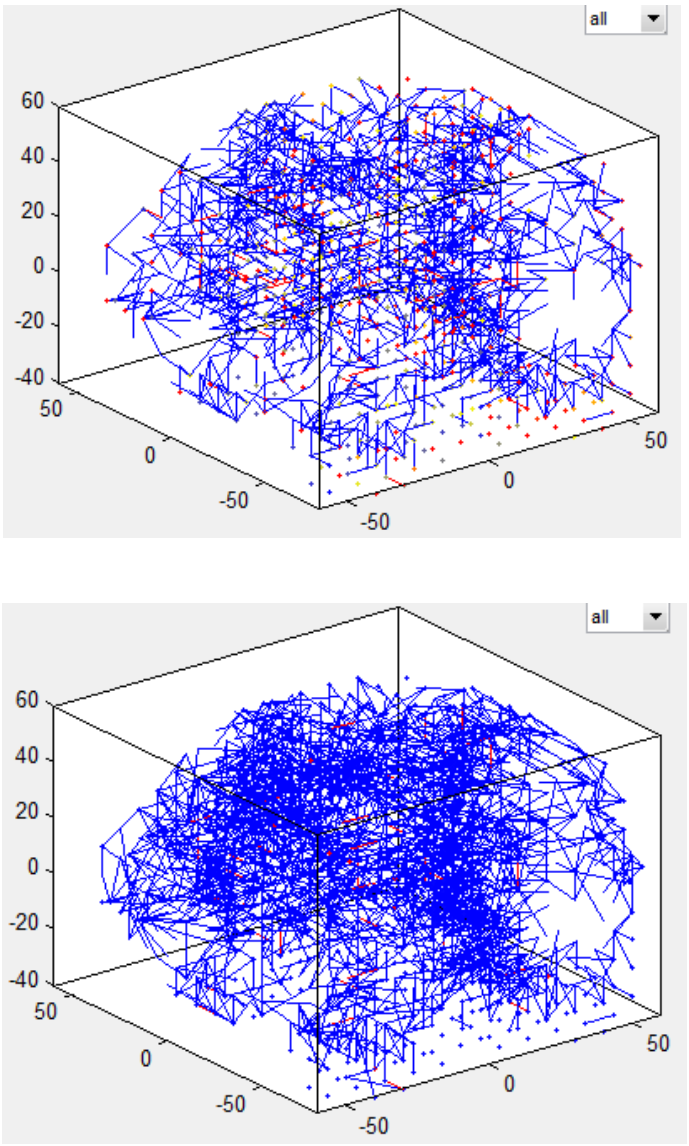

Figure 6-8: Visualization of neurons connectivity for fMRI data stimulus before (top) and after (down) STDP learning in 3D view. Positive connections are in blue and negative connections are in red.

6.5 CHAPTER SUMMARY

This chapter has elaborated the recently proposed NeuCube^{B} as a generic metholdogy that can be used to model and to learn spatio-temporal fMRI data. From this model, the generic and novel contributions of this study can be identified in terms of Talairach template fMRI mapping (i.e. individual and cluster fMRI data mapping into the NeuCube^{B} structure), unsupervised and supervised learning of spatio-temporal fMRI data, classification, visualization of the model before and after unsupervised (or supervised) learning and finally in interpretation of the model.

In the next chapter, the first case study involving NeuCube^{B} and StarPlus data will be explained in detail. It starts with the experimental setup, mapping of the data into the 3D structure, learning of the data in unsupervised mode, and learning in supervised deSNN mode. The classification results from the deSNN classifier is tabulated and analyzed. The dynamic brain state in terms of the neuron connections during the NeuCube^{B} learning is also captured, visualized and interpreted.

Chapter 7

CASE STUDY 1: APPLICATION OF NEUCUBE^{B} ON STARPLUS DATA

This chapter presented our first original case study that uses the NeuCube^{B} model in mapping, learning, classifying and interpreting the StarPlus dataset. We fed in the spatio-temporal StarPlus fMRI data into the NeuCube^{B} over time (i.e. in its temporal order) and the coordinates of StarPlus fMRI were mapped into the architecture. After the encoding process, these corresponding voxel values are the input spikes to the system that will then be used in the learning phase. The output from this phase will be trained again using a classifier to determine to which class these data belongs. Each phase will be explained in detail in the next sections.

7.1 EXPERIMENTAL SETUP

The experiment (Mitchell et al., 2003) is conducted by executing 2 sets of trials in which half of the trials, (a subject is first presented with a Picture stimulus for 4 seconds, then a rest period for 4 seconds and finally the subject is presented with a Sentence stimulus for another 4 seconds), are identified as PS trials. For the other half of trials, a subject is first presented with a Sentence stimulus, followed by a rest period and then Picture stimulus (SP trials). For our experiment, we took the whole lot of 1-24 time points of PS dataset to be Class 1 and the same time points of SP dataset to be Class 2, to differentiate and classify whether the subject is viewing a Picture or viewing a Sentence. There are 10 samples for each class. Experiment was conducted for each 6 subjects who are labelled as S04799, S04820, S04847, S05675, S05680 and S05710.

7.2 MAPPING OF 3D STARPLUS fMRI DATA INTO NEUCUBE^{B}

As explained in Chapter 6, we have to map the StarPlus voxels' coordinates into the 3D NeuCube^{B} architecture. Voxels coordinates are first extracted from the metadata and this voxel extraction is according to the steps explained in the previous chapter.

7.2.1 Neuron Mapping for Each Subjects

The newly calculated StarPlus coordinates are used in the next step, which is to find the coordinates that are close to NeuCube^{B} points i.e. to be within a certain radius unit. NeuCube^{B} neurons are constructed in 11 slices while StarPlus neurons are constructed in 8 slices ($z_{NeuCube} = 11$ and $z_{StarPlus} = 8$). In this experiment, we set the radius to 7. This is to ensure that, not a single coordinate of new StarPlus is located outside of the NeuCube^{B} brain.

Figure 7-1 until Figure 7-26 show NeuCube^{B} and fMRI StarPlus neuron mapping for all 6 subjects in different views; and their details are presented in Table 7-1. For instance for subject S04799, its original fMRI volume has 4,949 voxels and Figure 7-1 shows these 4,949 voxels mapping in 3D view while Figure 7-3 shows the same subject mapping but in xy, xz and yz views respectively. In Figure 7-1 NeuCube neurons are in black dots while StarPlus neurons are in blue dots. In Figure 7-3, neuron mapping in xz and yz views provide visualization as to which 8 $z-$slices of StarPlus are mapped into the 11 $z-$slices of NeuCube^{B}. This analysis ensures that the each $z-$slice of StarPlus is within the NeuCube^{B} slices.

We can also visualize the different regions of the brain in the NeuCube^{B} model (Figure 7-2) and each region is labelled and illustrated with different representation as tabulated in Table 7-2. Another possible visualization is in terms of brain slices (Figure 7-4) where each sub-figure represents a single slice ($z-$slice) of the brain data. Sub-figures which have only cyan squares mean that no StarPlus coordinates i.e. no input neurons are mapped into that particular $z-$slice (for example in $z = \{-40, -30, 60\}$). The subsequent figures (Figure 7-5 to Figure 7-24) show the neuron mapping for the other 5 subjects.

This mapping method is proposed to be used in determining the sample set. For instance, as tabulated in Table 7-1, for subject S04799, the number of voxels specified is 4949 and thus implying 4,949 x 3 coordinates. The exact set of sample, which is later used for learning, is generated based on the 1,471 neurons of NeuCube^{B} i.e. 24 x 1,471 x 20, calculated from the base sample of 24 x 4,949 x 20. From the specified 1,471 NeuCube^{B} neurons, only 817 neurons were used as input neurons for subject S04799 i.e. other unselected neurons were those NeuCube^{B} neurons without mapped StarPlus neurons. These 817 input neurons were actually StarPlus neurons in clusters, which held the average of StarPlus neurons that were located within the radius.

Table 7-1: Voxel details for each subject

Subject/ Item	S04799	S04820	S04847	S05675	S05680	S05710
Number of voxel	4,949	5,015	4,698	5,135	5,062	4,634
StarPlus co-ordinates	4,949 x 3	5,015 x 3	4,698 x 3	5,135 x 3	5,062 x 3	4,634 x 3
Base sample	24 x 4,949 x 20	24 x 5,015 x 20	24 x 4,698 x 20	24 x 5,135 x 20	24 x 5,062 x 20	24 x 4,634 x 20
Exact sample	24 x 1,471 x 20	24 x 1,471 x 20	24 x 1,471 x 20	24 x 1,471 x 20	24 x 1,471 x 20	24 x 1,471 x 20
Input neuron	817	840	821	847	803	747

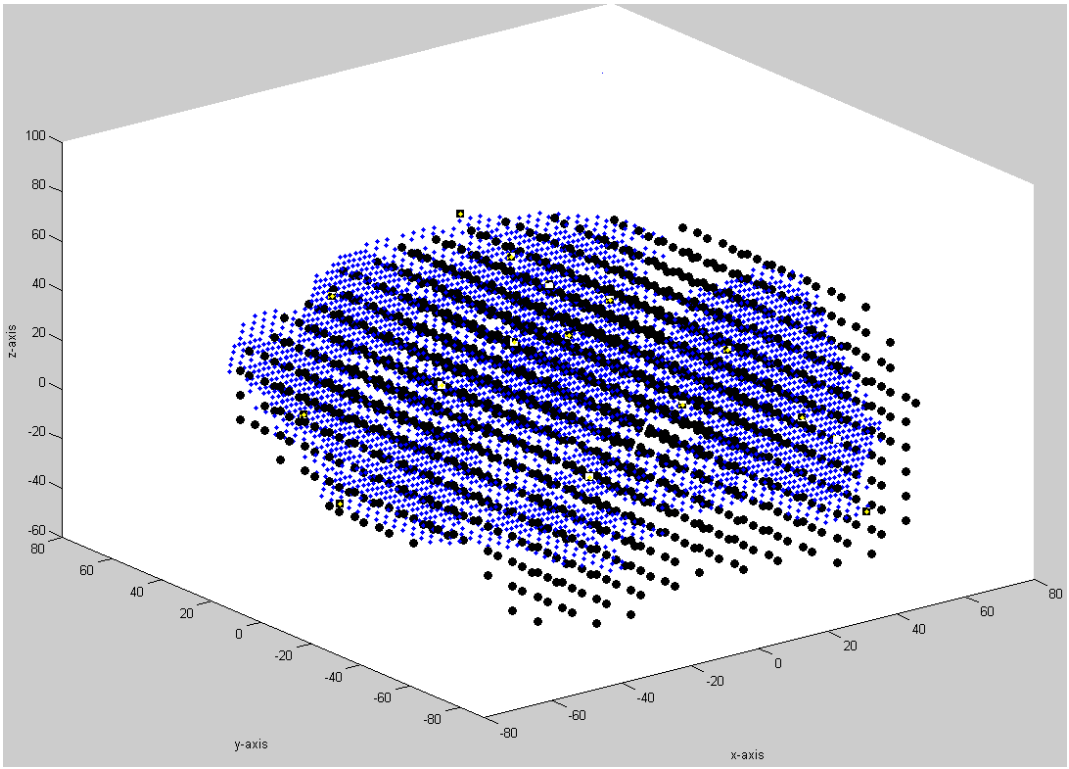

Figure 7-1: Mapping of 4,949 fMRI neurons (blue crosses) into NeuCube^{B} input neurons (black dots) for subject S04799 in 3D view.

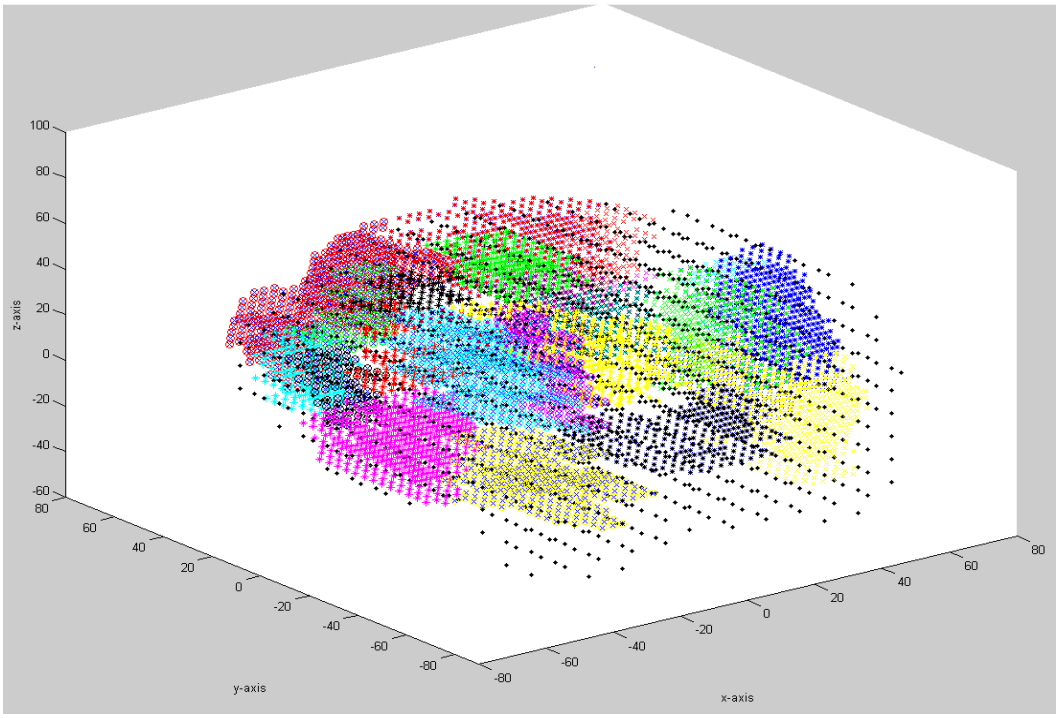

Figure 7-2: NeuCube[B] and segmented StarPlus fMRI neurons for subject S04799. Different colours represent different brain areas.

Table 7-2: Brain region representation

Region	Labelled Region	Representation
Calcarine Sulcus	CALC	Black dot (.)
Left Inferior Parietal Lobe	LIPL	Blue dot (.)
Left Temporal Lobe	LT	Yellow dot (.)
Left Triangularis	LTRIA	Magenta dot (.)
Left Opercularis	LOPER	Cyan dot (.)
Left Intraparietal Sulcus	LIPS	Green dot (.)
Left Bilateral Dorsolateral Prefrontal Cortex	LDLPFC	Red dot (.)
Left Frontal Eye Field	LFEF	Red cross (x)
Left Inferior Frontal Gyrus	LIFG	Black cross (x)
Left Inferior Temporal Lobe	LIT	Yellow cross (x)
Left Posterior Precentral Sulcus	LPPREC	Magenta cross (x)
Left Supramarginal Gyrus	LSGA	Cyan cross (x)
Left Superior Parietal Lobe	LSPL	Green cross (x)
Right Bilateral Dorsolateral Prefrontal Cortex	RDLPFC	Red circle (o)
Right Frontal Eye Field	RFEF	Green circle (o)
Right Inferior Parietal Lobe	RIPL	Cyan circle (o)
Right Intraparietal Sulcus	RIPS	Magenta circle (o)
Right Inferior Temporal Lobe	RIT	Yellow circle (o)
Right Opercularis	ROPER	Black circle (o)
Right Posterior Precentral Sulcus	RPPREC	Black asterisk (*)
Right Supramarginal Gyrus	RSGA	Red asterisk (*)
Right Superior Parietal Lobe	RSPL	Yellow asterisk (*)
Right Temporal Lobe	RT	Magenta asterisk (*)
Right Triangularis	RTRIA	Cyan asterisk (*)
Supplementary Motor Area	SMA	Green asterisk (*)

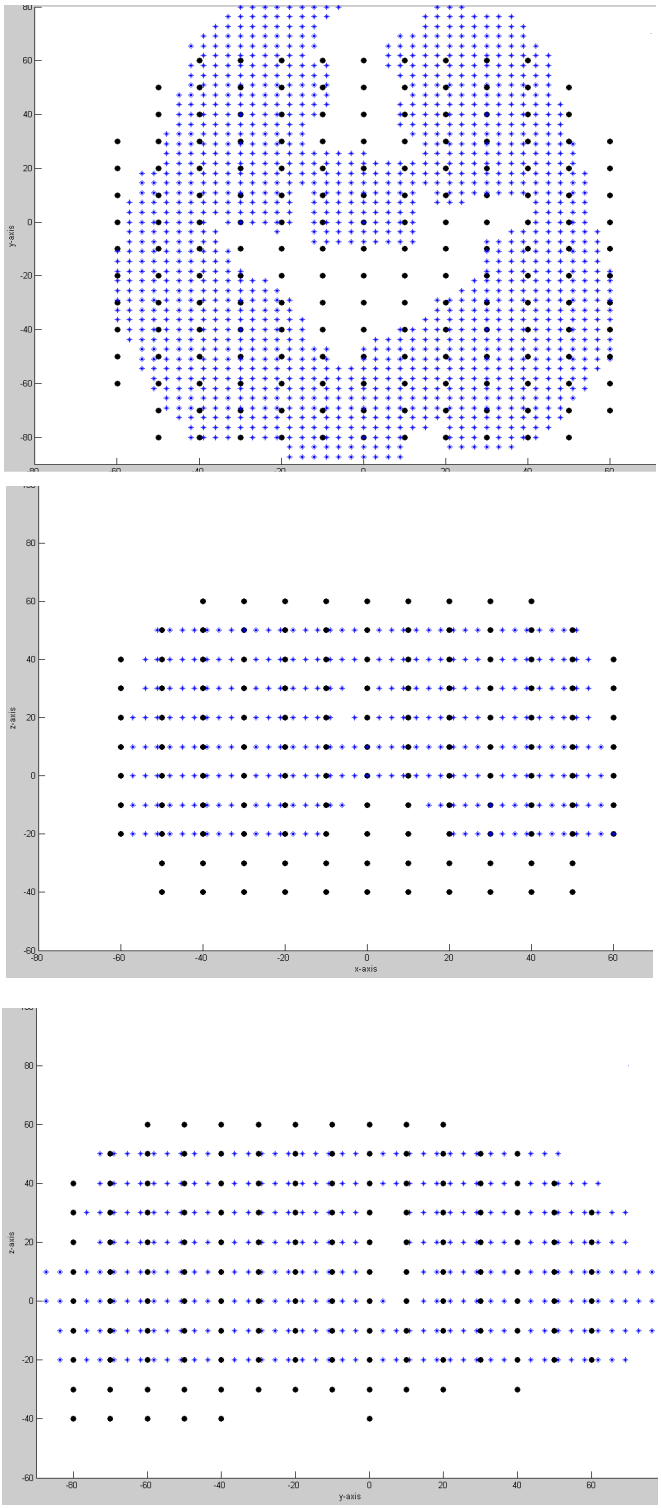

Figure 7-3: NeuCube^{B} neurons (black dots) and fMRI S04799 input neurons (blue dots) in three views (in xy, xz and yz views).

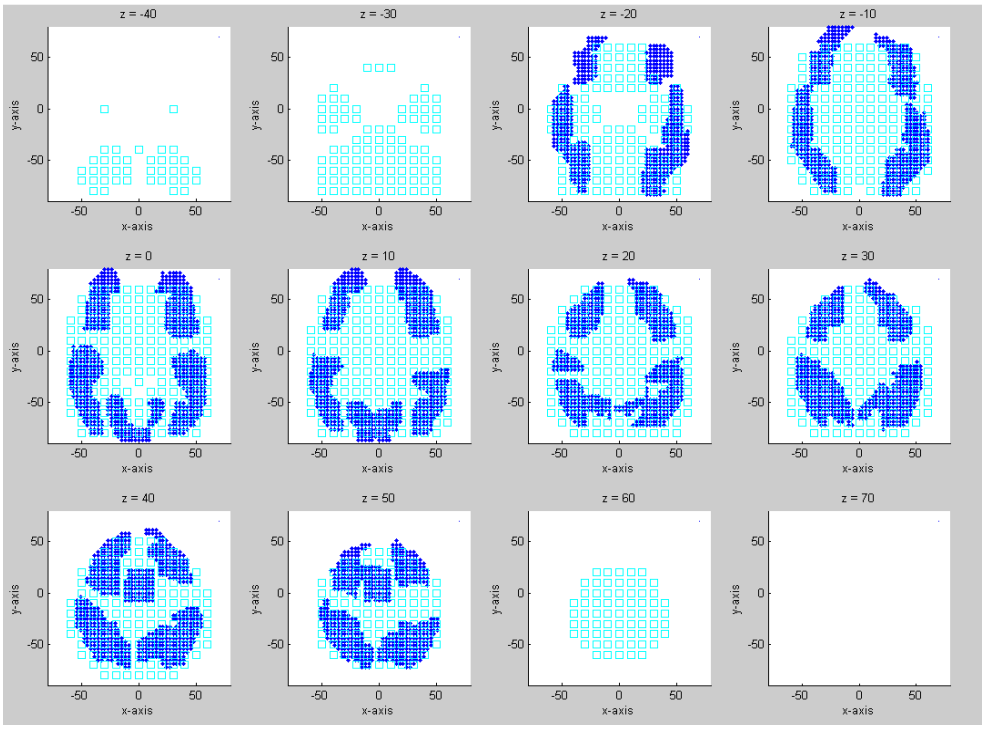

Figure 7-4: NeuCube[B] neurons (cyan squares) and fMRI S04799 input neurons (blue dots) mapping, in each 11 z −slice, in which each sub-figure represents a single slice.

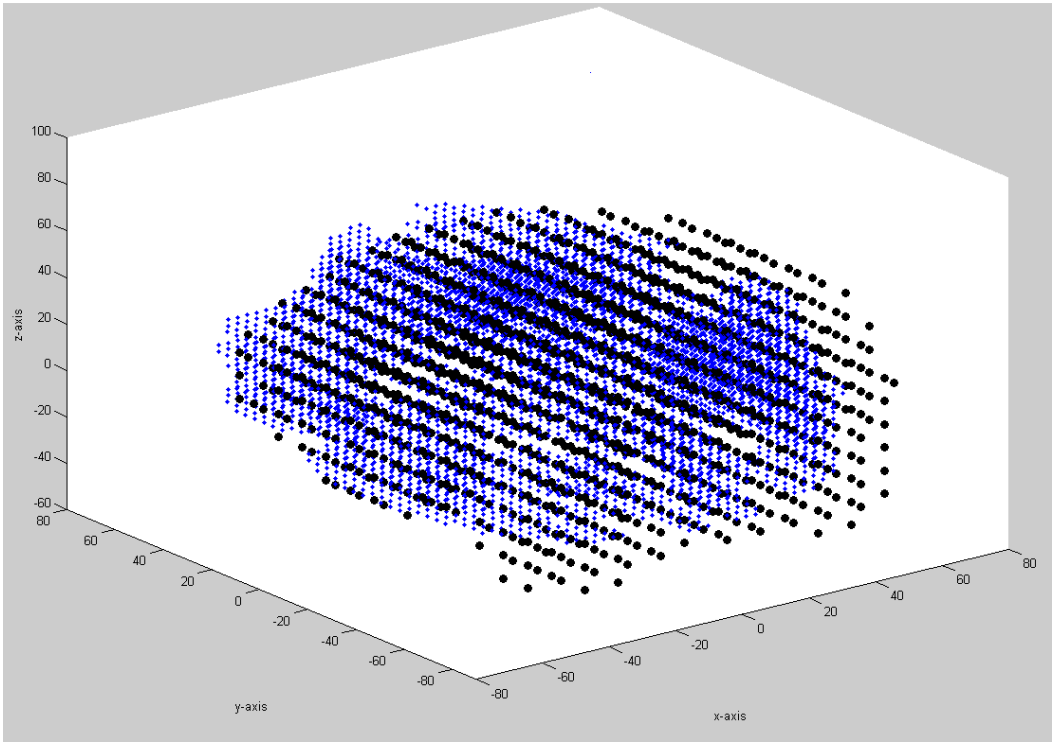

Figure 7-5: Mapping of 5,015 fMRI neurons (blue crosses) into NeuCube[B] input neurons (black dots) for subject S04820 in 3D view.

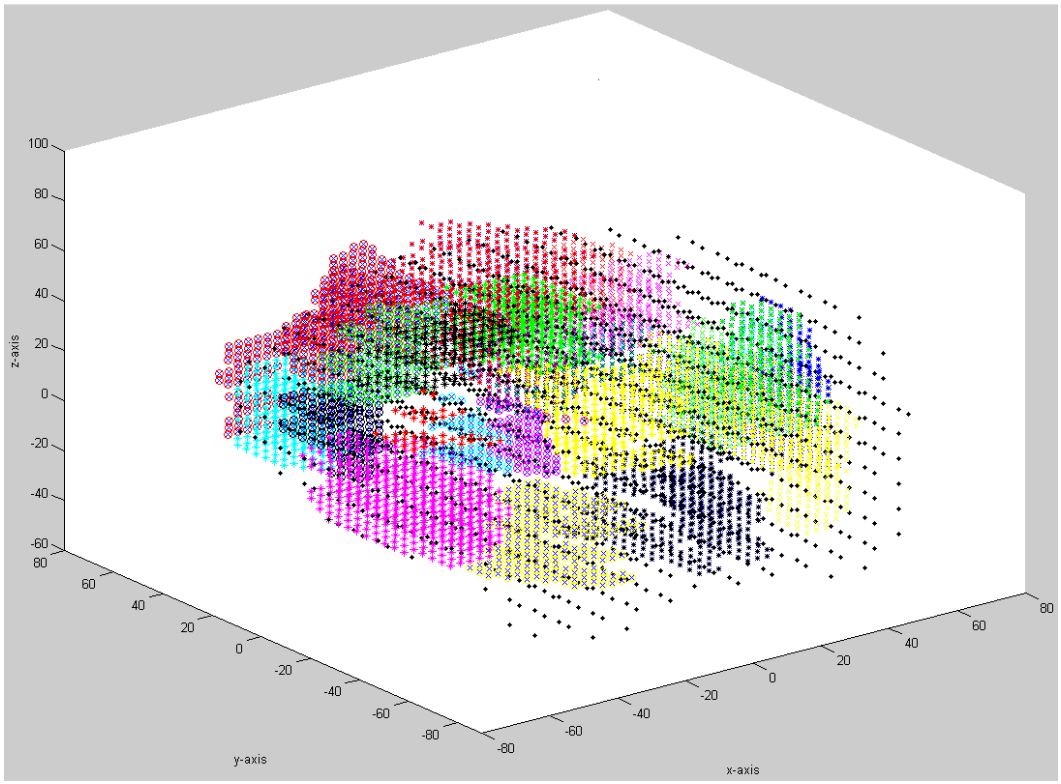

Figure 7-6: NeuCube^{B} and segmented StarPlus fMRI neurons for subject S04820. Different colours represent different brain areas.

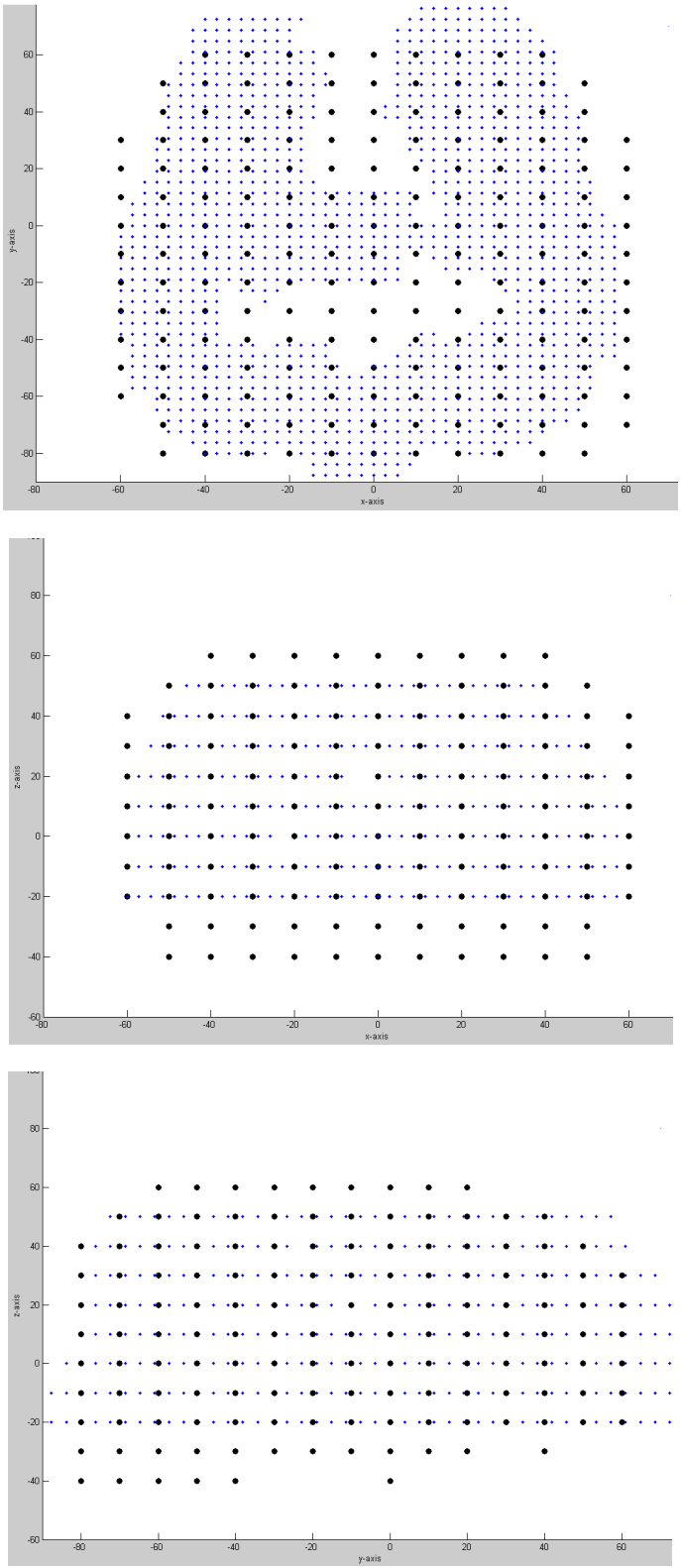

Figure 7-7: NeuCube^{B} neurons (black dots) and fMRI S04820 input neurons (blue dots) in three views (in xy, xz and yz views).

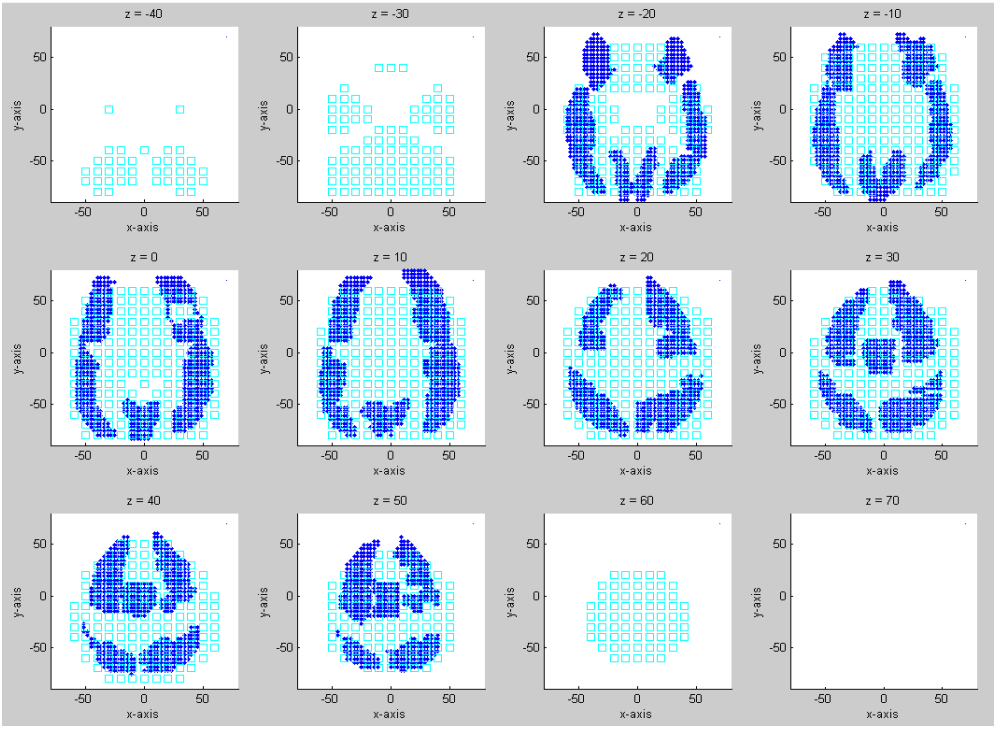

Figure 7-8: NeuCube^{B} neurons (cyan squares) and fMRI S04820 input neurons (blue dots) mapping, in each 11 z −slice, in which each sub-figure represents a single slice.

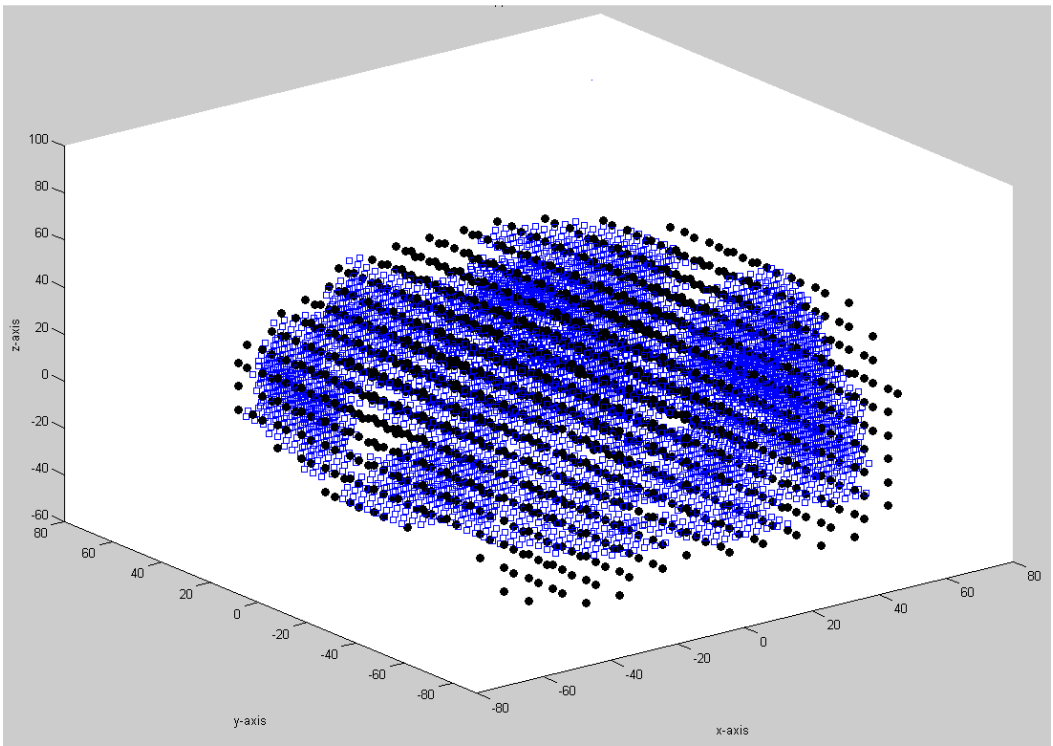

Figure 7-9: Mapping of 4,698 fMRI neurons (blue crosses) into NeuCube^{B} input neurons (black dots) for subject S04847 in 3D view.

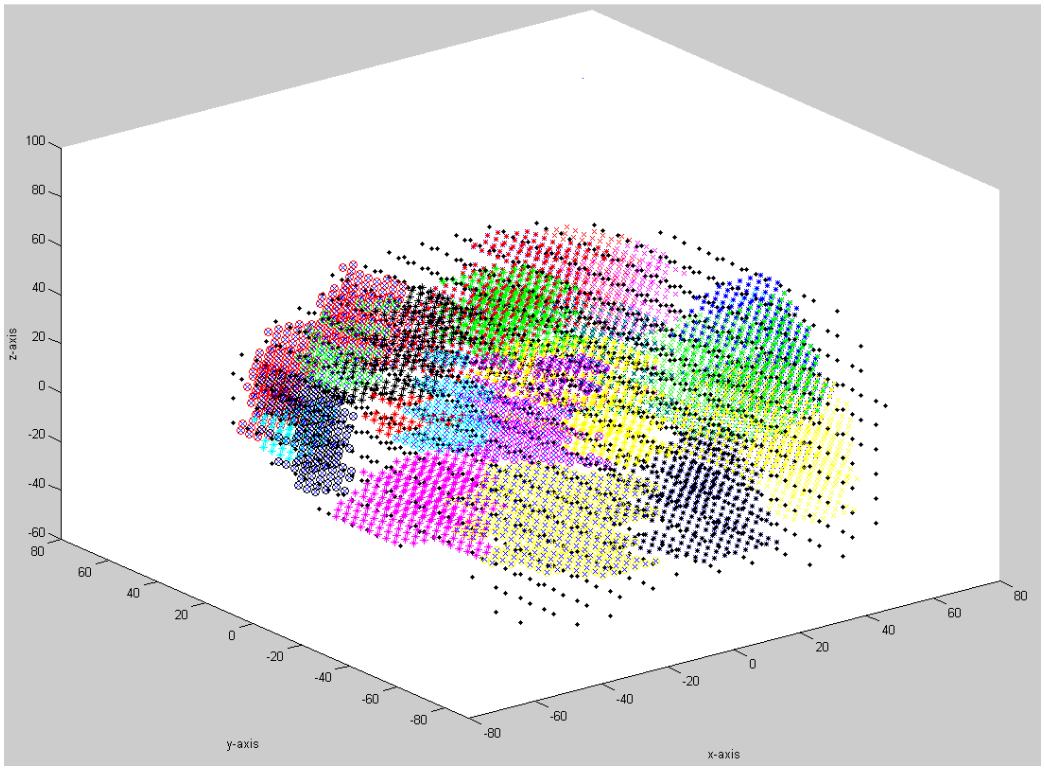

Figure 7-10: NeuCube[B] and segmented StarPlus fMRI neurons for subject S04847. Different colours represent different brain areas.

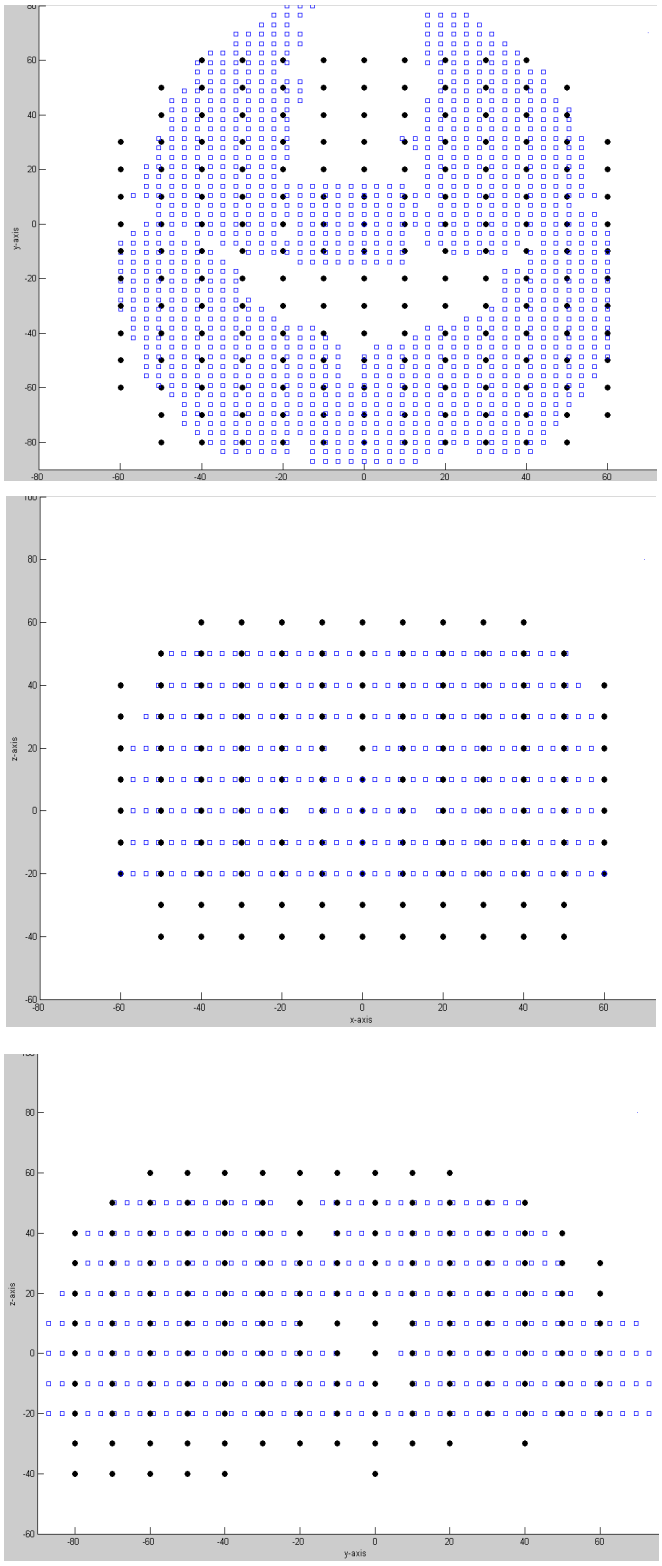

Figure 7-11: NeuCube[B] neurons (black dots) and fMRI S04847 input neurons (blue dots) in three views (in xy, xz and yz views).

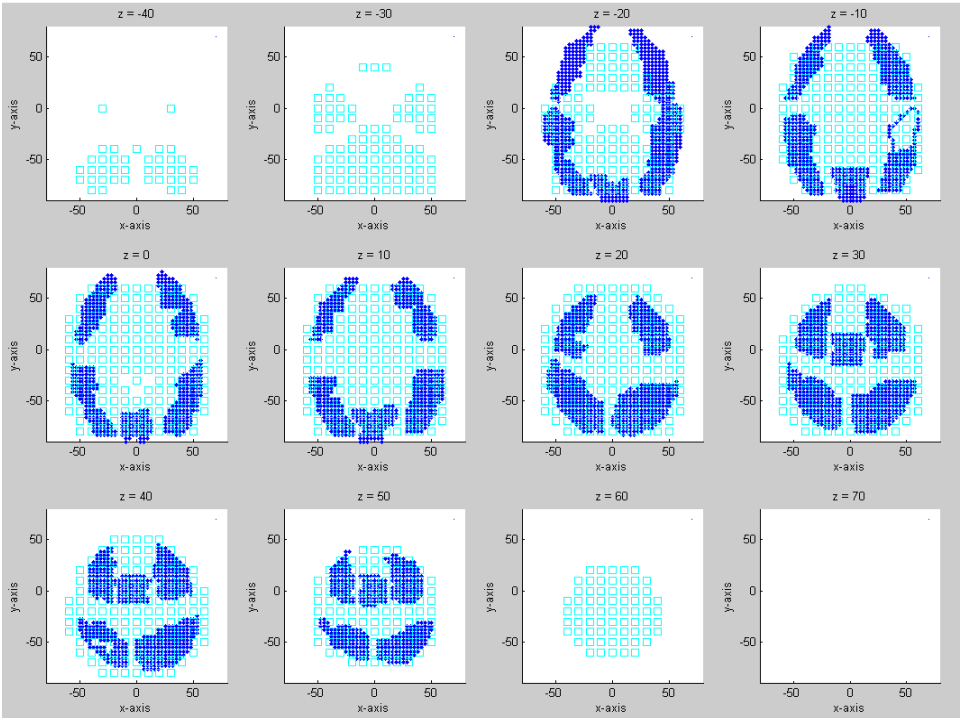

Figure 7-12: NeuCube^{B} neurons (cyan squares) and fMRI S04847 input neurons (blue dots) mapping, in each 11 z −slice, in which each sub-figure represents a single slice.

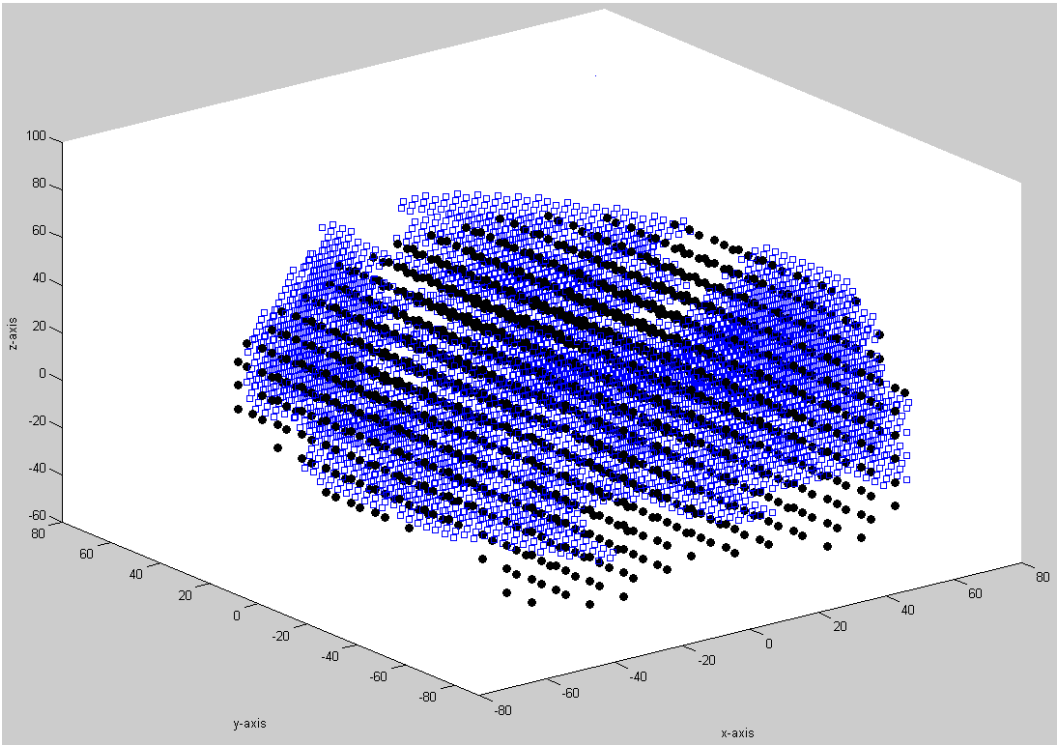

Figure 7-13: Mapping of 5,135 fMRI neurons (blue crosses) into NeuCube^{B} input neurons (black dots) for subject S05675 in 3D view.

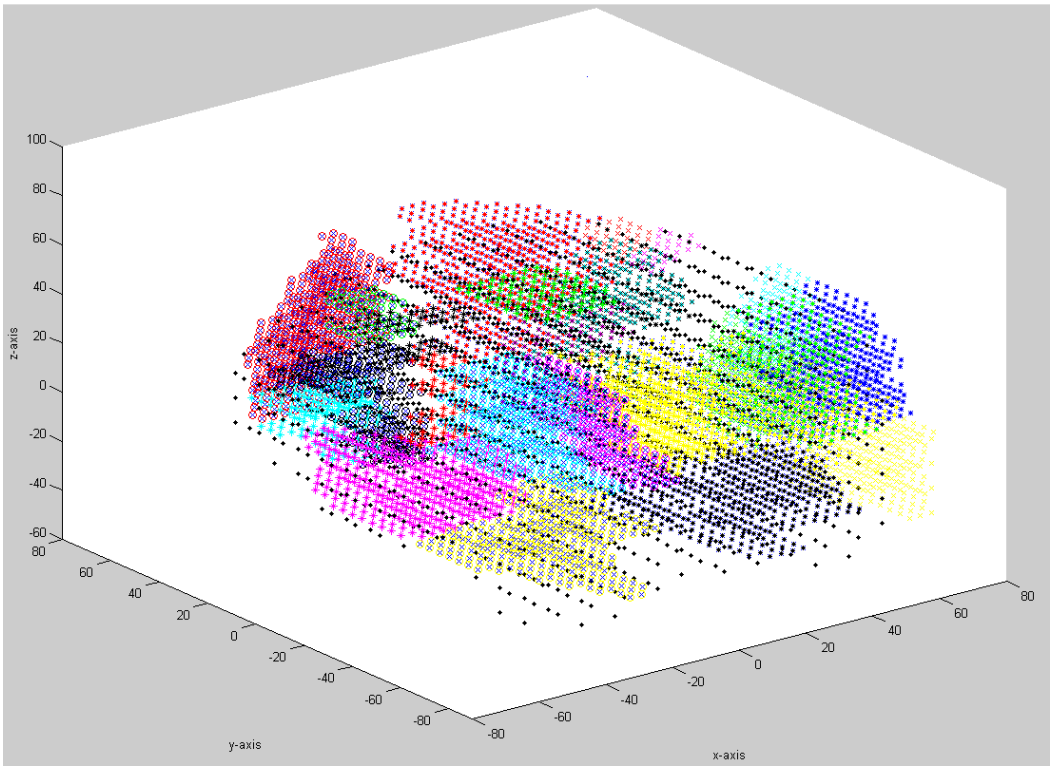

Figure 7-14: NeuCube[B] and segmented StarPlus fMRI neurons for subject S05675. Different colours represent different brain areas.

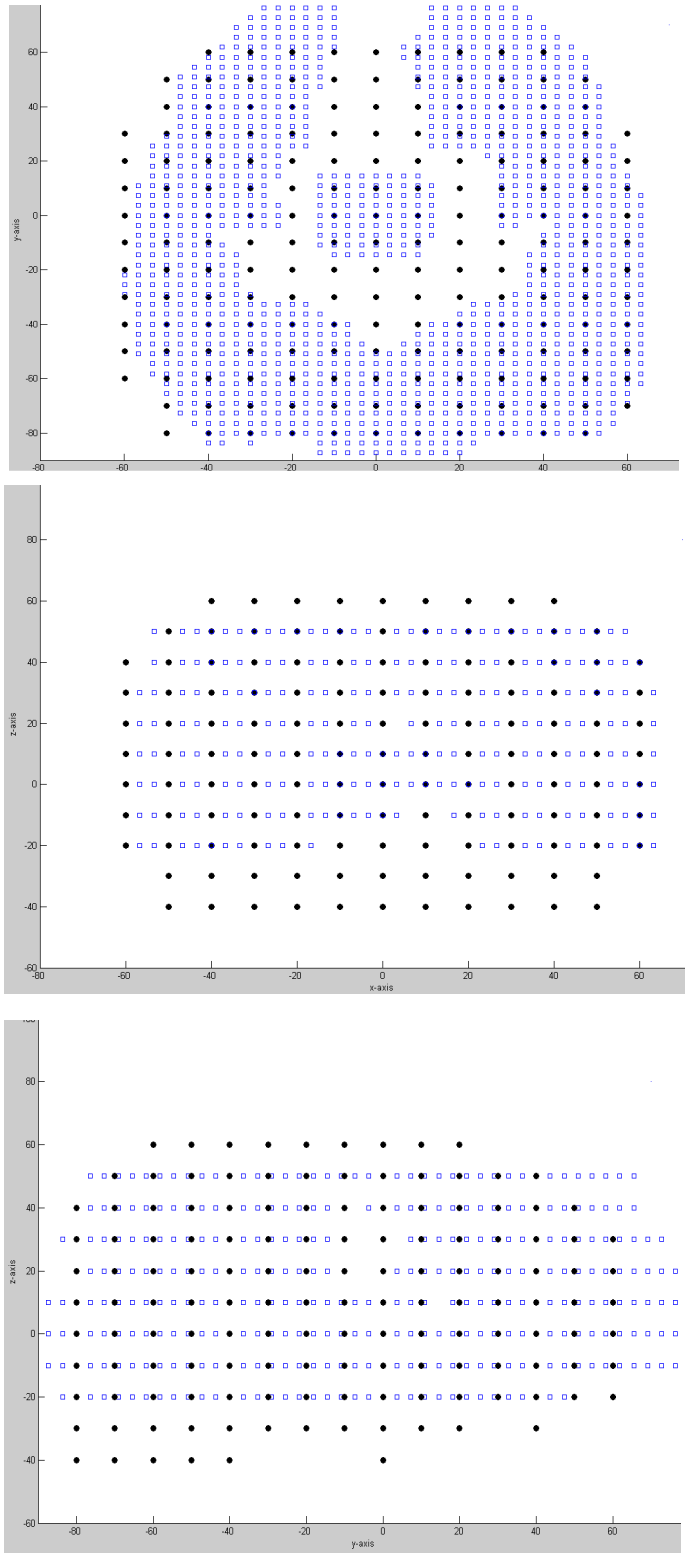

Figure 7-15: NeuCube^{B} neurons (black dots) and fMRI S05675 neurons (blue dots) in three views (in xy, xz and yz views).

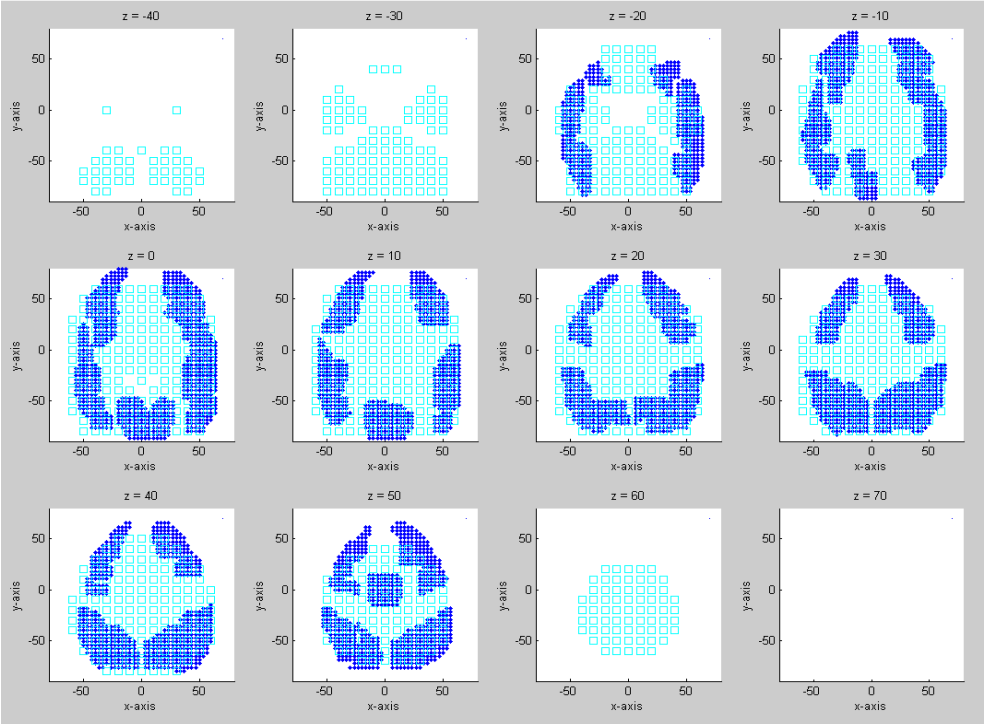

Figure 7-16: NeuCube^{B} neurons (cyan squares) and fMRI S05675 input neurons (blue dots) mapping, in each 11 z −slice, in which each sub-figure represents a single slice.

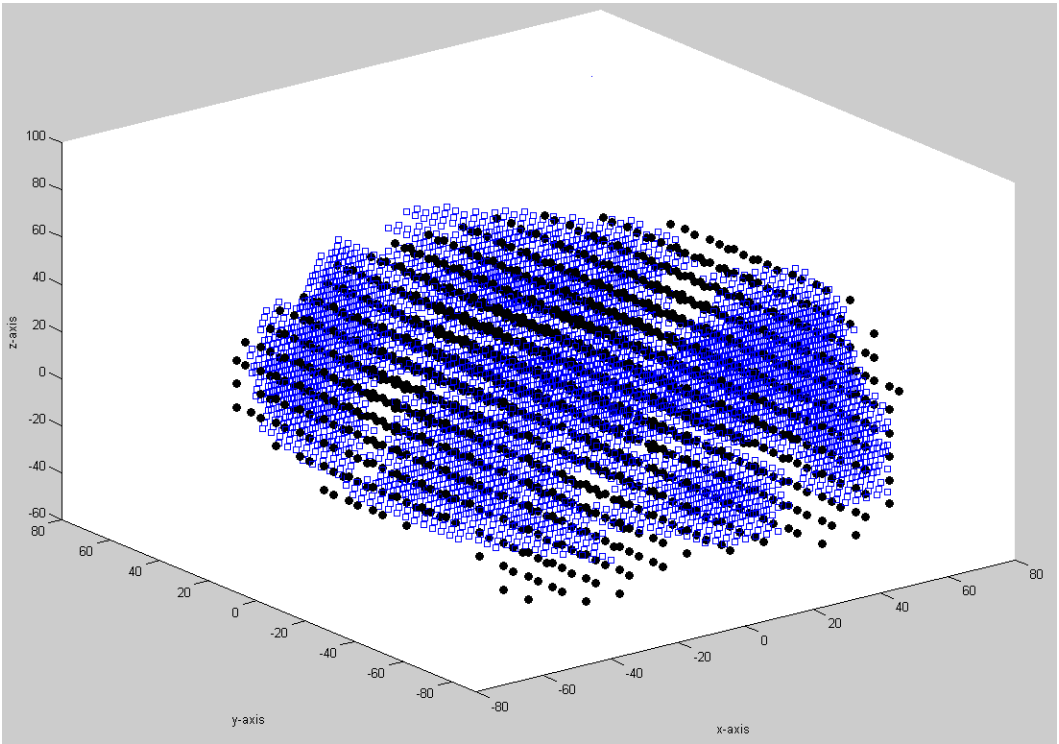

Figure 7-17: Mapping of 5,062 fMRI neurons (blue crosses) into NeuCube^{B} input neurons (black dots) for subject S05680 in 3D view.

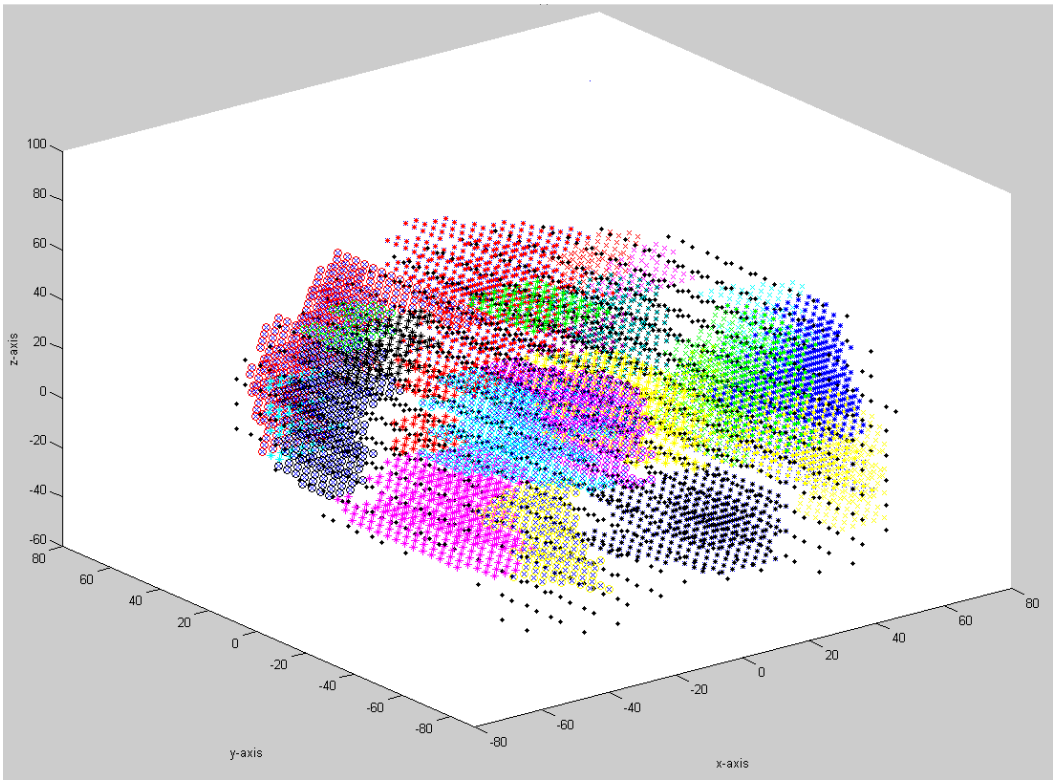

Figure 7-18: NeuCube^{B} and segmented StarPlus fMRI neurons for subject S05680. Different colours represent different brain areas.

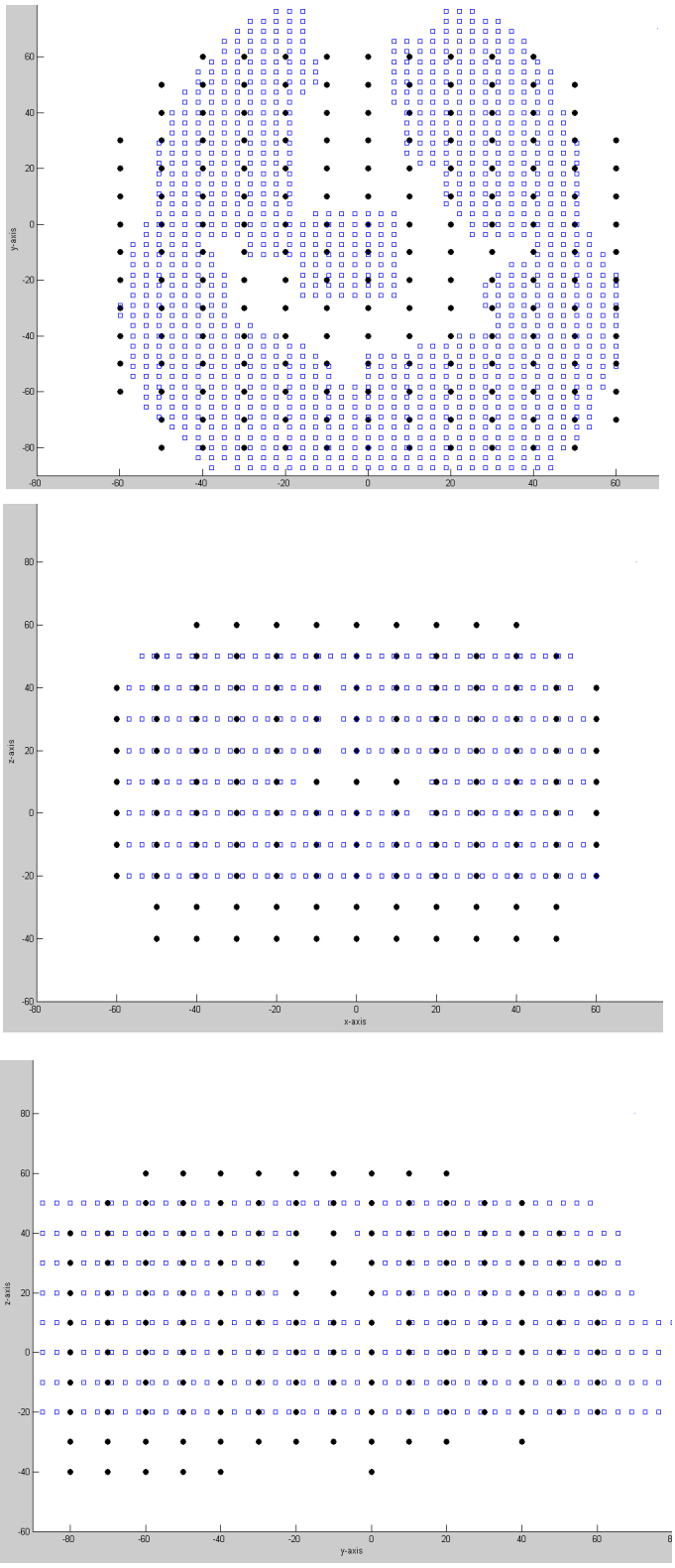

Figure 7-19: NeuCube^{B} neurons (black dots) and fMRI S05680 neurons (blue dots) in three views (in xy, xz and yz views).

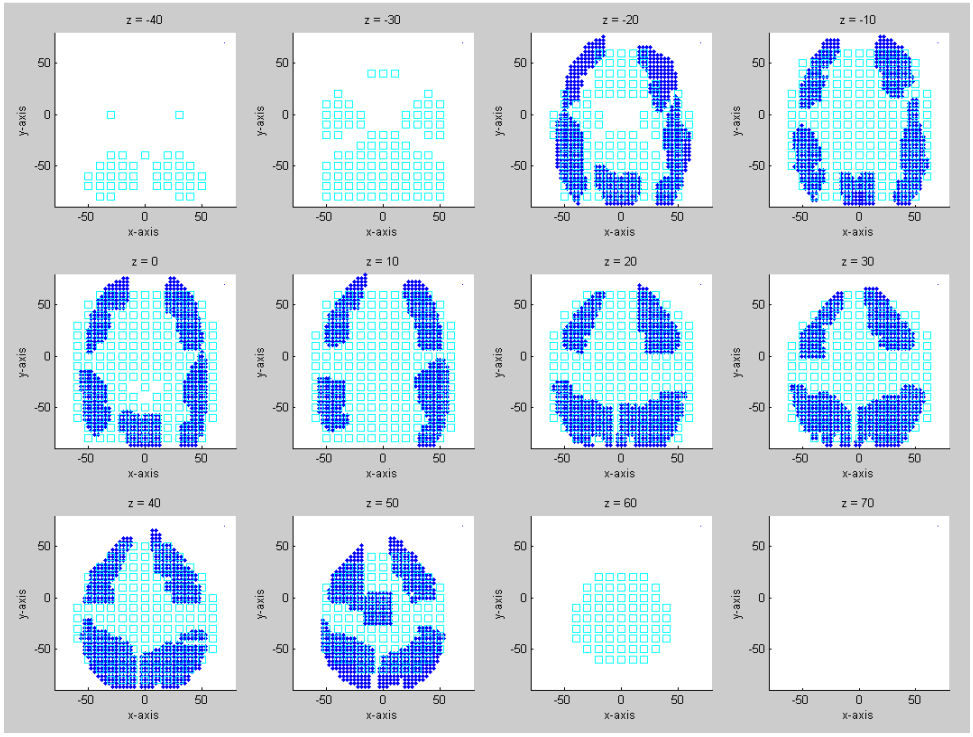

Figure 7-20: NeuCube^{B} neurons (cyan squares) and fMRI S05680 input neurons (blue dots) mapping, in each 11 z −slice, in which each sub-figure represents a single slice.

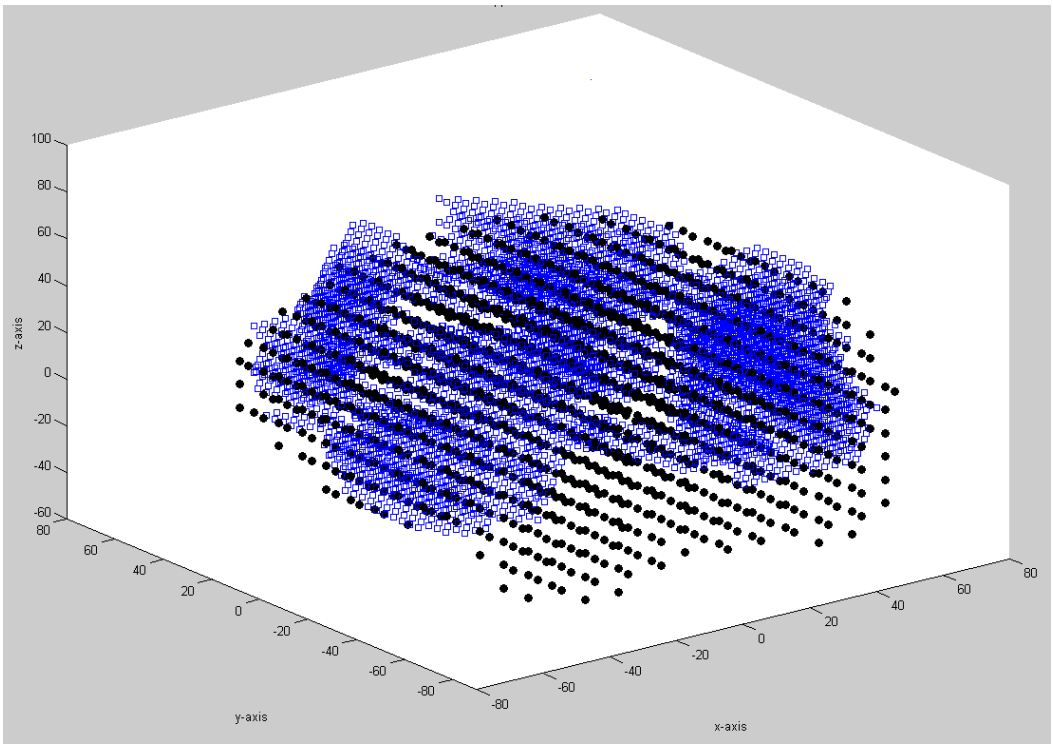

Figure 7-21: Mapping of 4,634 fMRI neurons (blue crosses) into NeuCube^{B} neurons (black dots) for subject S05710 in 3D view.

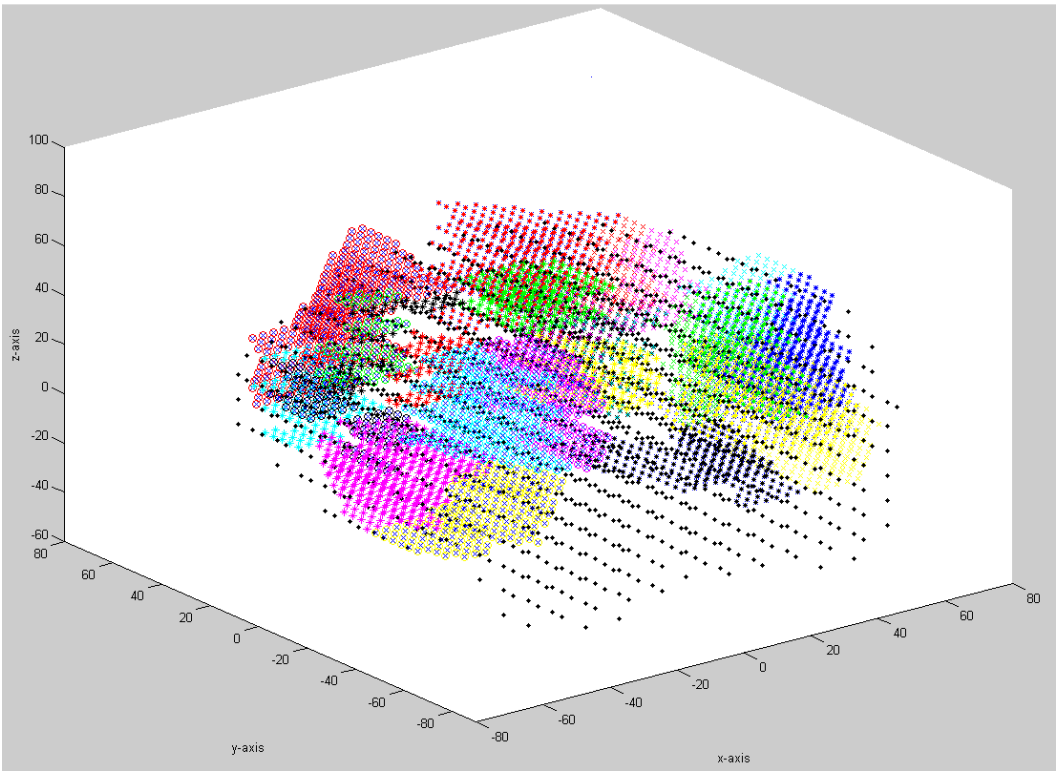

Figure 7-22: NeuCube^{B} and segmented StarPlus fMRI neurons for subject S05710. Different colours represent different brain areas.

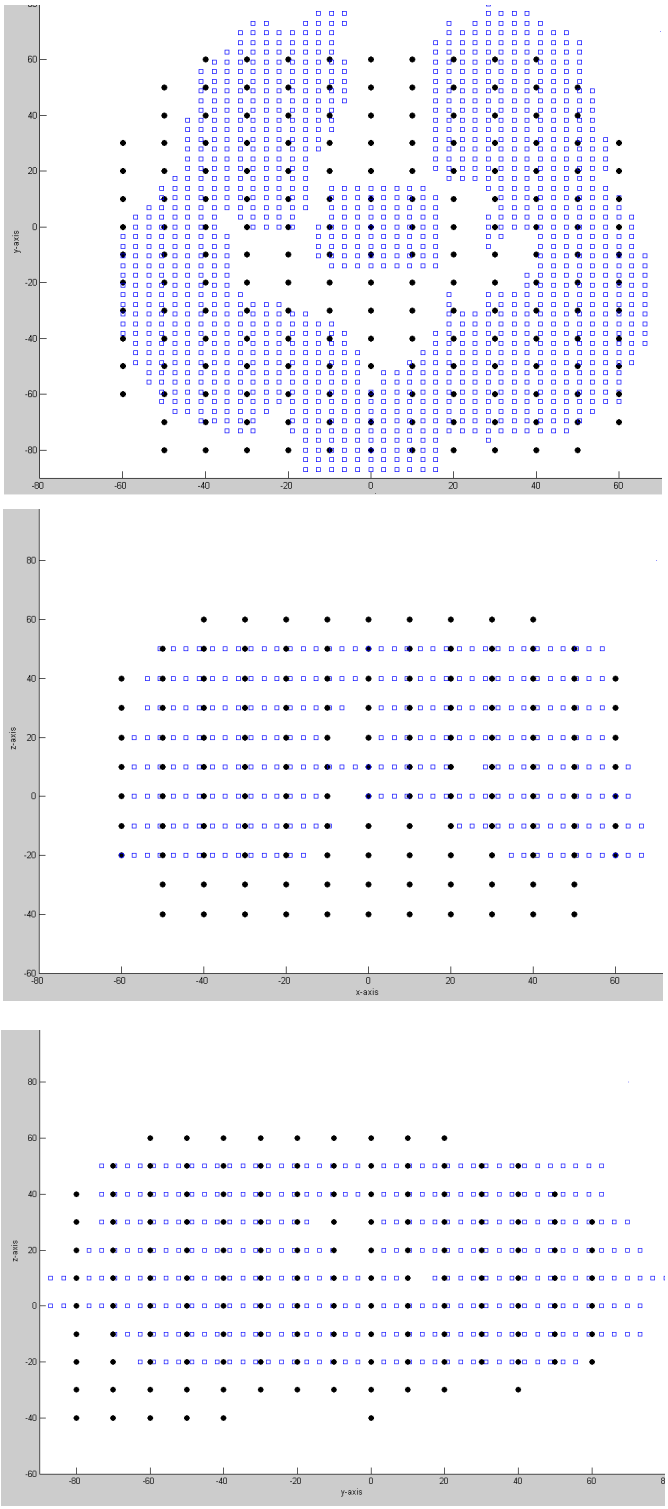

Figure 7-23: NeuCube[B] neurons (black dots) and fMRI S05710 input neurons (blue dots) in three views (in xy, xz and yz views).

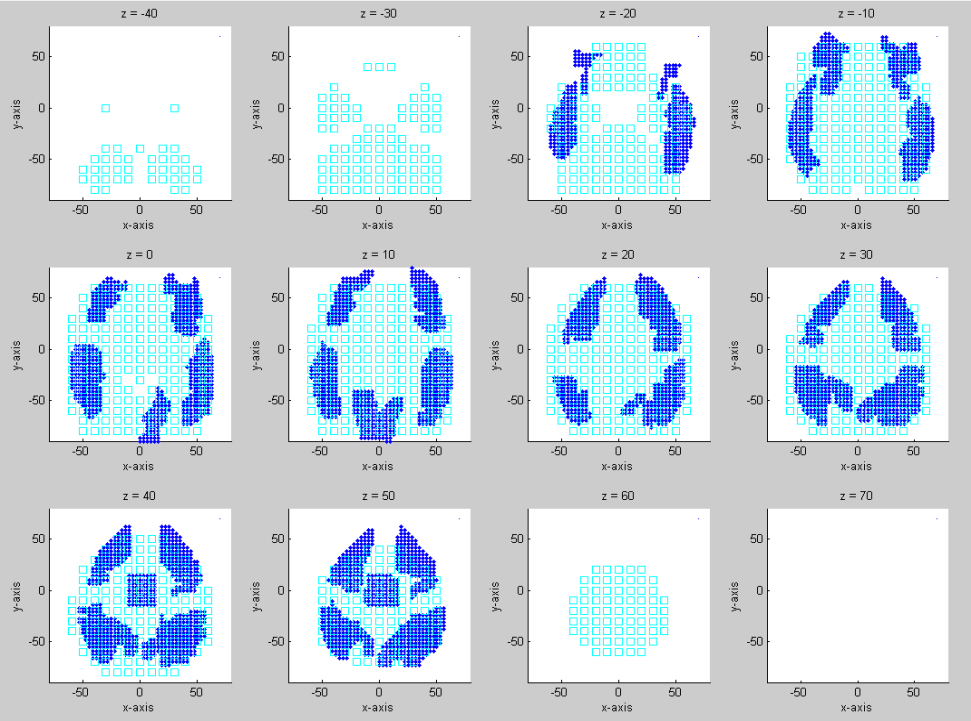

Figure 7-24: NeuCube^{B} neurons (cyan squares) and fMRI S05710 input neurons (blue dots) mapping, in each 11 z −slice, in which each sub-figure represents a single slice.

It is also possible to visualize the 7 ROIs of StarPlus neurons that were used in the classification procedures. To better visualize the different brain regions, each ROI was segmented and plotted in different colours as depicted in Figure 7-25. Figure 7-27 is used to visualize the segmented fMRI StarPlus data according to the regions. The spatio-temporal relationship between these input neurons can be segmented as an identical region in terms of the spatial correlation between the neighbouring neurons and their temporal spiking activities. Different regions of the brain are presented in different colours and each region is labelled appropriately.

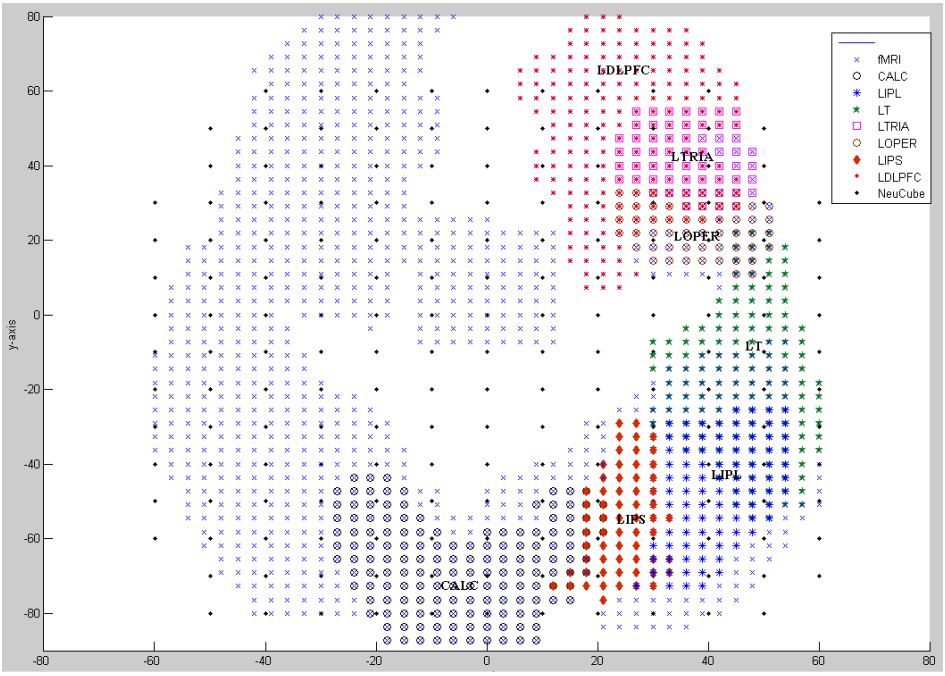

Figure 7-25: NeuCube^{B} and StarPlus 7 ROIs neurons mapping in xy view. Different colour represents different ROI. Black dots are NeuCube^{B} coordinates and blue crosses are the other StarPlus neurons.

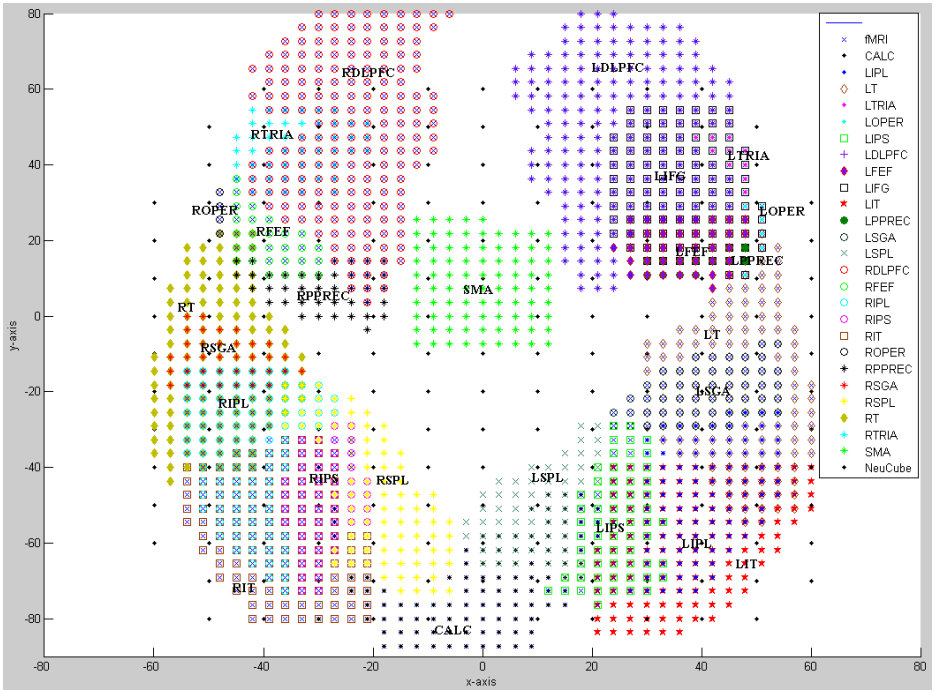

Figure 7-26: NeuCube^{B} and StarPlus 25 ROIs neurons mapping in 2D view.

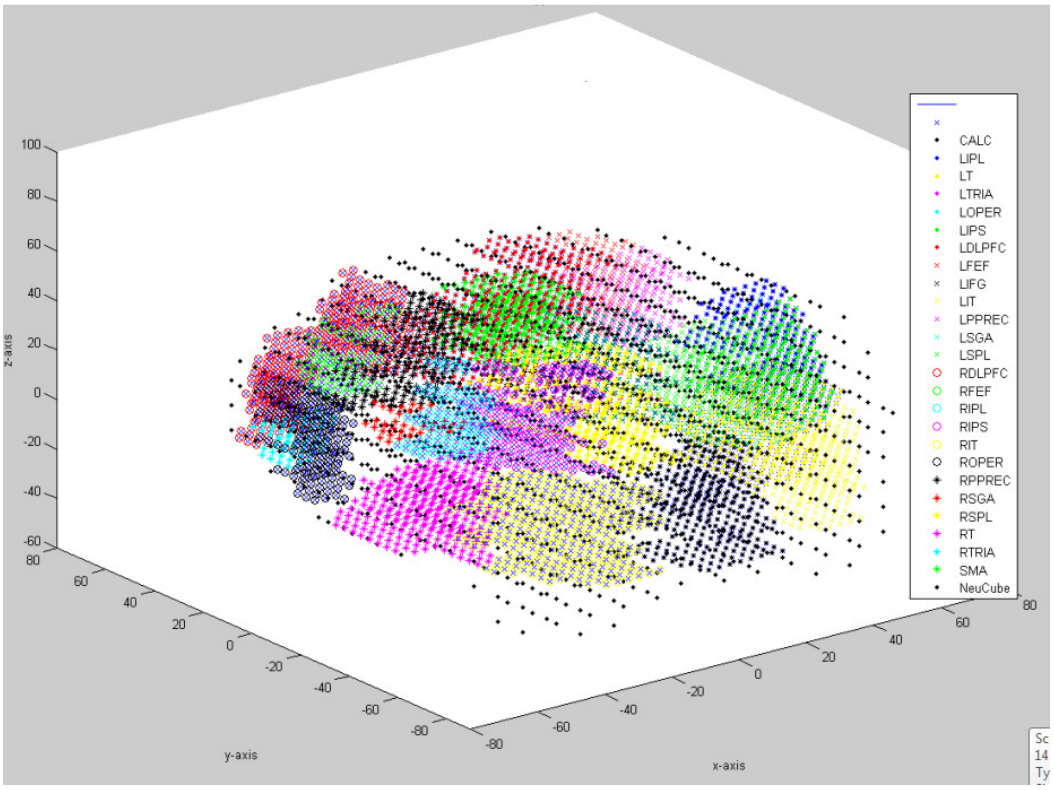

Figure 7-27: NeuCube^{B} and segmented StarPlus 25 ROIs neurons mapping in 3D view.

We can conclude that our proposed mapping technique provides a new insight of learning and classifying spatio-temporal StarPlus fMRI data which was not studied before (Kasabov, 2013). We maintained and made full use of the spatio-temporal information. Pervious approaches had either disregard one of the spatio-temporal components or treated the data as a static data (Brodersen et al., 2012; Georgieva & Torre, 2013; Rustandi, Mitchell, & Xing, 2010). Visualization of the data across the subjects also presents a new method in analyzing the input neurons, which will be used later.

7.3 DATA ENCODING

fMRI input data that have already been mapped as described in previous section is encoded using AER method which is based on the difference between 2 consecutive voxel values of the same input variable. If the difference is greater than certain threshold value, which is defined through a grid search, a spike will be generated. For instance in Table 7-3, for subject S04799, after mapping process there are 817 input neurons generated. From this 817 input neurons, AER encoding method will produce 1,634 spike states in which 240 x 1,634 spike states are for training and another 240 x 1,634 spike

states are for validation (i.e. assuming 50-50 split of data for training and validation respectively). The spike state length which is 240 is generated based on the following formula: time length (24s) x number of samples (10).

Table 7-3: Total number of neurons for each subject, the actual input neurons and the corresponding spike states for training and validation

Subject	Total Neurons	Actual Input Neuron	Spike States for Training	Spike States for Validation
S04799	4,949	817	240 x 1,634	240 x 1,634
S04820	5,015	840	240 x 1,680	240 x 1,680
S04847	4,698	821	240 x 1,642	240 x 1,642
S05675	5,135	847	240 x 1,694	240 x 1,694
S05680	5,062	803	240 x 1,606	240 x 1,606
S05710	4,634	747	240 x 1,494	240 x 1,494

7.4 DATA LEARNING

The previous experimental setting in (Kasabov et al., 2014) was designed and tested to access the feasibility of NeuCube in predicting stroke occurrences for a specific person and further experiment is done to access on other SSTD namely fMRI data. In our experiment, for each six subjects from the StarPlus data, 24 time points are extracted from the raw data with selected meta.cond=2; which indicates the interval is a Sentence/Picture trial in which the sentence is not negated e.g. “Is it true that the plus is below the dollar?” These sets of image sequences are labelled into classes of person seeing a Picture or seeing a Sentence. There were 20 samples generated, 10 samples for each class.

STDP learning method is applied in the NeuCube^{B} to initialize, modify and retain connection weights (memory) throughout the unsupervised learning stage. As illustration, the following parameter values were selected for classification accuracy:

- NeuCube^{B} is constructed with 1,471 neurons

- The threshold for the AER is 2.375 i.e. when the input value variable is greater (smaller) than 2.375, there will a positive (or negative) spike generated and entered into the SNN. Data is not normalized to minimize error or loss of information.

- SWC is used to initialize neurons connection in the NeuCube^{B} with the radius of initial connection is set to 0.15. Initial connections between neurons are generated probabilistically.

- The threshold of LIFM neurons is 0.5

- Leak parameter of the LIFM neurons is 0.002;

- STDP learning rate is 0.01;

- Number of training is 5 times;

- Mod parameter is 0. 4 and drift is 0.25.

The combination of parameter values stated above is determined after a thorough grid search from countless experiment runs, which resulted with highest percentage accuracy. For illustration purposes Figure 7-27 represents the 1,471 neurons of the brain-like cube were mapped according to the Talairach brain atlas. After STDP learning is completed, the dynamic brain activities were captured inside the Cube, while the spiking neurons were trained with the time series data (Picture and Sentence spike sequences). The brain states are captured before and after unsupervised learning (Figure 7-28 left and right). The number of neurons connection has increased and different neuron connections are created and added. In certain areas of the NeuCube^{B}, the connections are more condensed as compared to the other area suggesting the increase in the calculated synaptic weight and consequently an increase in the number of connections between neurons. This implies that more spikes are generated in that area as compared to areas with no connections i.e. no spike. The connections between neurons are represented by blue lines if the calculated weights are greater than 0 (i.e. positive connections), while the weights that are less than 0 are represented by red lines (i.e. negative connections). The connection weight is increased by STDP rate if pre-synaptic neurons fire first before the post-synaptic neurons. As a result, spikes are emitted if the calculated neuron potential has reached the stated threshold, thus resulting in connected neurons. Figure 7-28 demonstrates connections between neurons after initialization process and after unsupervised learning is completed.

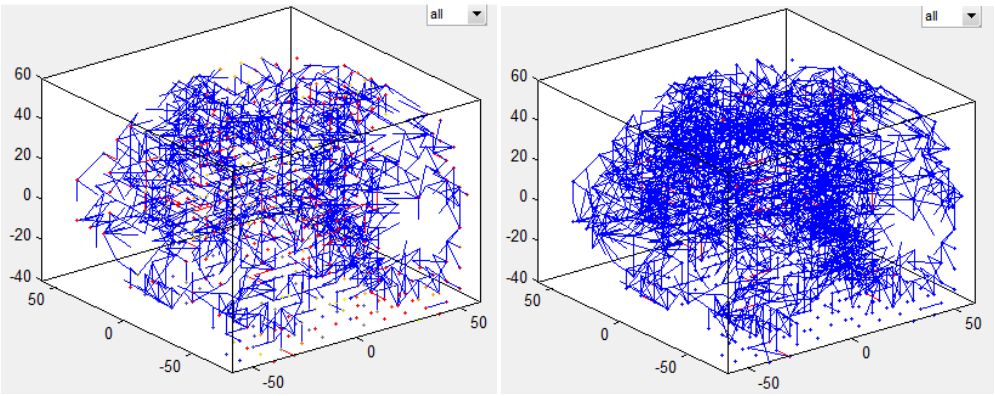

Figure 7-28: Neuron connections after initialization (left) and neuron connections after unsupervised learning (right).

From the figure above, we can conclude that more positive weight connections (blue lines) i.e. more connections were formed between the neurons after the unsupervised learning than before the learning process (i.e. after initialization). Different brain areas are also activated. For visualization purposes, different connection weights can be specified to show areas of the brain with less/more spikes activities, ranging from the connection weight > 0 to the maximum connection weight determined before/after the unsupervised/supervised learning.

7.5 MINING USING DYNAMIC EVOLVING SPIKING NEURAL NETWORKS (deSNN)

In our study, the potential of deSNN as a training algorithm is further explored using spatio-temporal StarPlus fMRI data. deSNN as described in (Dhoble et al., 2012) achieve fast and accurate learning of AER data for STPR as compared to eSNN and SNN that uses only SDSP. With deSNN, for every fMRI input, the connection weights are initialized using the RO learning rule, the connections weights for consecutive spikes are adjusted using SDSP learning rule, and the maximum of post synaptic potential and threshold value are calculated. The detail algorithm is as presented in Section 3.3.1. For our experiment, we use two classes of fMRI data of a subject - either that subject is viewing a Picture or viewing a Sentence. The parameters used in the experiment are presented in the previous section. The training and testing is conducted using 50:50 split.

The SVM experiment uses Polynomial Kernel of first degree while MLP experiment uses 20 hidden nodes and one output, with learning rate of 0.001 and 500 iterations. The data design for both NeuCube^{B} and standard classifiers are as in Figure 7-29. In NeuCube^{B} model, brain volumes within the same trial are learned as one spatio-temporal pattern i.e. one sample and is propagated one after another. The signal intensity of each brain volume at time point t_k within a trial is treated as a voxels M, and all brain volumes within a trial of the same experimental condition as a sample X_i. Each sample will be 24 x 1,471 voxel and thus generating 24 x 1,471 x 20 samples. Whereas in standard classifiers, brain volumes within the same trial are concatenated and taken as a single sample i.e. each sample will be 35,304 voxels generating 35,304 x 20 samples for StarPlus data.

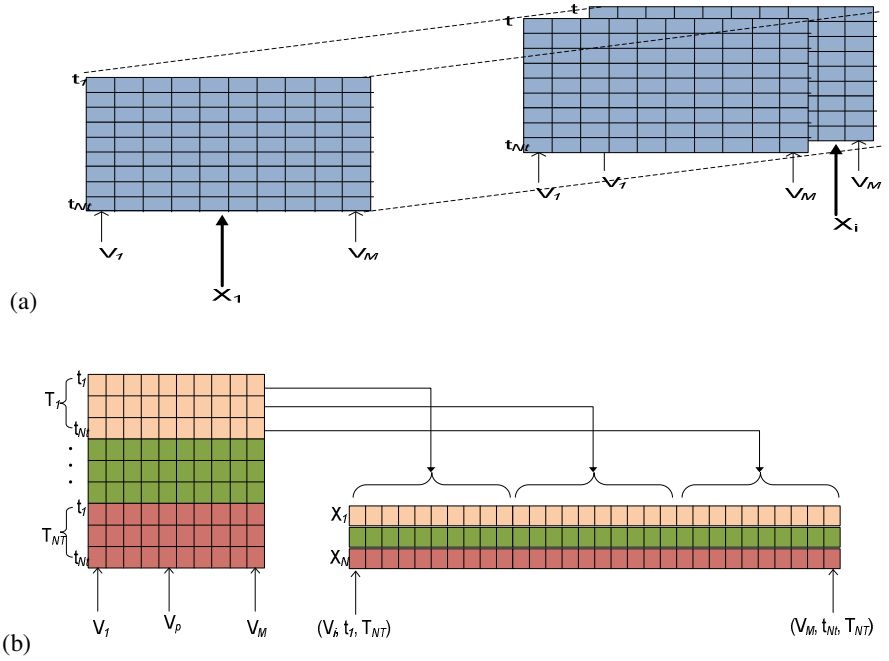

Figure 7-29: Data design for: (a) NeuCube^{B} model. Brain volumes within the same trial are learned as one spatio-temporal pattern i.e. one sample and propagated one after another. (b) Standard classifiers model. Brain volumes within the same trial are concatenated and taken as a single sample.

7.6 RESULT AND DISCUSSION

7.6.1 Classification Results

For the experiment, we have used the NeuCube^{B} model of 1,471 spiking neurons with AER as the input data encoding method, STDP as the learning method and deSNN as the classifier. A snapshot of the NeuCube^{B} software implementation after a successful classification is presented in Figure 7-30. The best percentage accuracies obtained for the designed and trained NeuCube^{B} against eSNN and standard classifiers (SVM and MLP) for 20 samples for each subject are as depicted in Table 7-4. As for the NeuCube^{B} and eSNN models, the networks are trained in unsupervised and supervised stages. For example for subject S04847 the overall classification accuracy obtained in NeuCube^{B} was 90% (100% for Class 1 and 80% for Class 2), in eSNN the overall classification accuracy was 60% (40% for Class 1 and 80% for Class 2), in SVM was only 45% (60% for Class 1 and 30% for Class 2) and in MLP was 65% (70% for Class 1 and 60 for Class 2). For the experiment involving subject S04820, eSNN and deSNN produce a comparatively similar result which was 90% of accuracy. In other experiment involving subject S05675, eSNN produced a slightly better accuracy performance compared to deSNN which was 90% for eSNN and 80% for deSNN. However, for the other experiments involving the other 4 subjects i.e. S04799, S04847, S05680 and S05710, deSNN produced better results compared to eSNN with the overall classification accuracies of 90%. From these results, we can conclude NeuCube^{B} model produced higher accuracy percentage results than eSNN, SVM or MLP in almost all experiments.

Figure 7-31(a) shows neurons in the Cube in which blue dots represent NeuCube^{B} neurons and yellow dots represent StarPlus input neurons. The Cube is in initialization stage. Figure 7-31(b) shows the spiking activity in the Cube during the learning process. Active neurons are represented in red, inactive neurons are represented in blue, positive input neurons are represented in magenta, negative input neurons are represented in cyan and zero input neurons are represented in yellow. Neurons connectivity before and after training are displayed in Figure 7-31(c) and (d), in blue lines (i.e. positive spikes) and red lines (i.e. negative spikes). More neuron connections were created after the learning process than before the learning process.

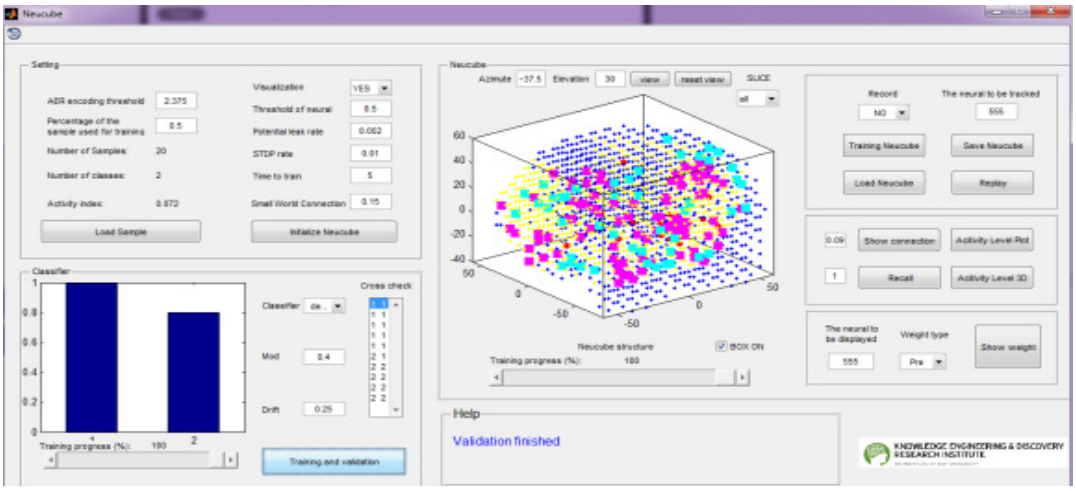

Figure 7-30: A snapshot of a software implementation of the NeuCube^{B} architecture for classification of 2 class fMRI data for subject S04847. The parameter values are as in the Setting parameter box. The classification model used is deSNN an accuracy of 100% for Class 1 and 80% for Class 2

Table 7-4: Classification results between classical machine learning and NeuCube^{B}. The dataset consists of 20 samples of 1,471 pixel value variables

Subject /Method	SVM	MLP	eSNN	NeuCube^{B}
S04799	50 (20, 80)	35 (30, 40)	60(60, 60)	90 (100, 80)
S04820	40 (30, 50)	75 (80, 70)	90 (100, 80)	90 (80, 100)
S04847	45 (60, 30)	65 (70, 60)	60 (40, 80)	90 (100, 80)
S05675	60 (40, 80)	30 (20, 40)	90 (100, 80)	80 (100, 60)
S05680	40 (70, 10)	50 (40, 60)	70 (80, 60)	90 (80, 100)
S05710	55 (60, 50)	50 (50, 50)	50 (60, 40)	90 (100, 80)

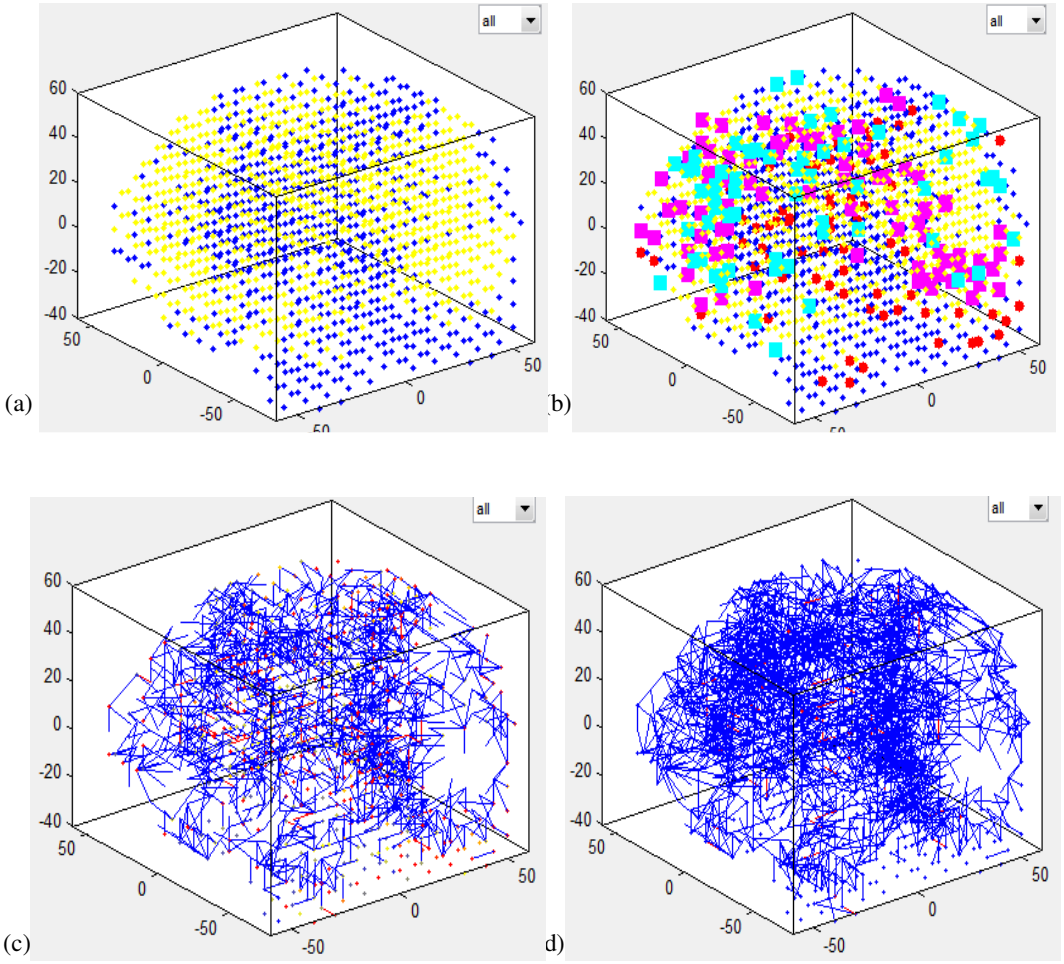

Figure 7-31: Visualization of fMRI data model and connectivity between neurons of eSNN: (a) no spiking activity yet, inactive neurons are in blue, fMRI data neurons (input neurons) are in yellow; (b) spiking activity: active neurons are represented in red, inactive neurons are represented in blue, positive input neurons are represented in magenta, negative input neurons are represented in cyan and zero input are represented in yellow; (c) neurons connectivity before training (small world connection), positive connections are in blue and negative connections in red; (d) neurons connectivity after training.

The results clearly show that NeuCube^{B} model is much more applicable to handle complex data because without filtering any noise from the data, the results are improved than classical machine learning methods mentioned before. This proves that noise also carries valuable information in defining association between SSTD samples, but failed to be recognized and processed in conventional methods. The deSNN model that combines RO and SDSP learning offers a fast one pass propagation of fMRI data, on-line supervised and unsupervised learning, modelling and pattern recognition of the StarPlus fMRI data. In particular, the synaptic drift parameter makes the synapse dy-

namic i.e. it is modifiable during the learning process and this parameter is important although the value specified is small. The model is also capable to evolve and merge neurons and connections in an incremental, adaptive and 'life-long' learning. Additionally, classical machine learning methods clearly not suitable in analyzing complex SSTD that have spatial and temporal information, because their capability is only limited to only vector and static based kind of data.

7.7 NEURONS CONNECTIVITY

We also proposed the interpretation of the trained data in terms of neurons connectivity in the NeuCube^{B} before and after the learning process. Taking subject S04847 for illustration, during the unsupervised training, the data is mapped into relevant areas of the NeuCube^{B} over time and unsupervised learning is used to calculate the connection weights between neurons. Figure 7-32 to Figure 7-35 show the neurons connectivity (positive and negative connections) before and after 5 times of unsupervised trainings for data Sentence and Picture. Blue lines represent positive connections (i.e. connection weight > 0) and red lines represent negative connections (i.e. connection weight < 0). Neurons in blue indicate active neurons while red neurons indicate inactive neurons. Neuron connections are created and calculated in each training cycle that actually represents spatio-temporal relationship between input neurons (i.e. input voxels) in the network.

7.7.1 NeuCube^{B} Visualization for Spatio-temporal Connections Based on the fMRI Spiking Activity for Sentence stimulus

After the NeuCube^{B} learning processes, the spatio-temporal relationship between the neurons was analyzed with respect to the data collected. More densely interconnected neurons could be seen in different areas of the NeuCube^{B} resulted from higher connectivity that has been created i.e. more spikes emission, during the learning process. During initialization stage, NeuCube^{B} connections and their connection weights are initialized randomly (Figure 7-32). The left and right figures show the connections with connection weight >= 0.19 (i.e. stronger spikes) and weight >= 0.09 (i.e. weaker spikes) respectively. For this experiment, the maximum connection weight calculated was 0.2 before the learning process and 0.2079 after the learning process. Areas that are highly interconnected with positive blue lines are those areas that are highly activated when

seeing Sentence stimulus i.e. more spikes are generated from the input neurons. Higher NeuCube^{B} weights which imply stronger spikes are initialized somewhere in LIFG, LDLPFC, LPPREC, LTRIA, LSPL, CALC, RTRIA and RIT areas.

After 5 times of STDP unsupervised learning these connections has increased and more positive spikes are created. More and stronger NeuCube^{B} connections are generated on areas specifically on LSGA, RSGA, LTRIA, RTRIA, ROPER, LFEF, RFEF and CALC (Figure 7-33). Higher NeuCube^{B} weights indicating stronger spikes (Figure 7-33 left) are identified in LTRIA, RTRIA, and RSGA suggesting input neurons in these areas emit stronger spikes than neurons in other areas.

It can be concluded that LTRIA, RTRIA and RSGA are most activated and stimulated with the Sentence stimulus. These brain areas are known to involve among all in semantic definition, semantic working memory and language processing tasks (LTRIA and RTRIA) (Gabrieli, Poldrack, & Desmond, 1998), and language perception and processing (RSGA) (Gazzaniga, Ivry, Mangun, & Steven, 2009).

7.7.2 NeuCube^{B} Visualization for Spatio-temporal Connections Based on the fMRI Spiking Activity for Picture stimulus

The same procedures as in the previous section are taken for the Picture stimulus. The dynamic brain state was captured inside the NeuCube^{B} and more spikes were emitted after the learning process (Figure 7-35) than before the learning process (Figure 7-34). The left and right figures in Figure 7-34 shows the NeuCube^{B} input neuron connections created with connection weight >= 0.19 (stronger spikes) and weight >= 0.09 (weaker spikes) respectively. Blue lines represent positive connections while red lines represent negative connections, and blue neurons represent active neuron while red neuron represent inactive neuron. Higher NeuCube^{B} weights (i.e. more spikes are emitted) are initialized in distributed locations that include CALC, LDLPFC, RDLPFC, LT, RT, LSGA, RSGA, LTRIA, RTRIA, ROPER, LFEF, RFEF,LIPL and RIPL areas.

More neuron connections and stronger spikes are created after 5 times of STDP learning (Figure 7-35). Higher NeuCube^{B} weights are identified in CALC, LIPL, LT, LSGA, LFEF, RFEF, ROPER and RIPL areas, suggesting the areas that receive the most signals from the Picture stimulus i.e. more positive spikes are emitted.

In short, neurons in CALC, LIPL, LT, LSGA, LFEF, RFEF, ROPER and RIPL areas emitted stronger spikes than neurons in other brain areas. These brain areas are

known to involve activities such as in visual processing (CALC) (Wandell, Brewer, & Dougherty, 2005), language processing (LSGA) (Gazzaniga et al., 2009), continuing attention (RIPL and LIPL) (Singh-Curry & Husain, 2009), visual memories (LT) (Kolb & Whishaw, 1996), visual attention and eye movement (RFEF and LFEF) (Schall, 2004), and language processing (ROPER) (Foundas, Eure, Luevano, & Weinberger, 1998).

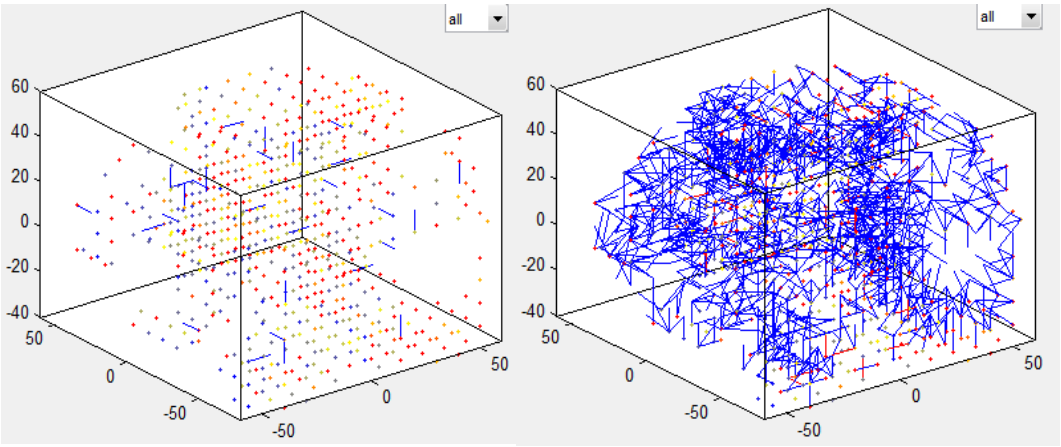

Figure 7-32: Visualization of neurons connectivity for data Sentence stimulus after initialization in 3D view. For visualization purposes, displayed connections are based on weight more than 0.19 (left) and weight more than 0.09 (right).

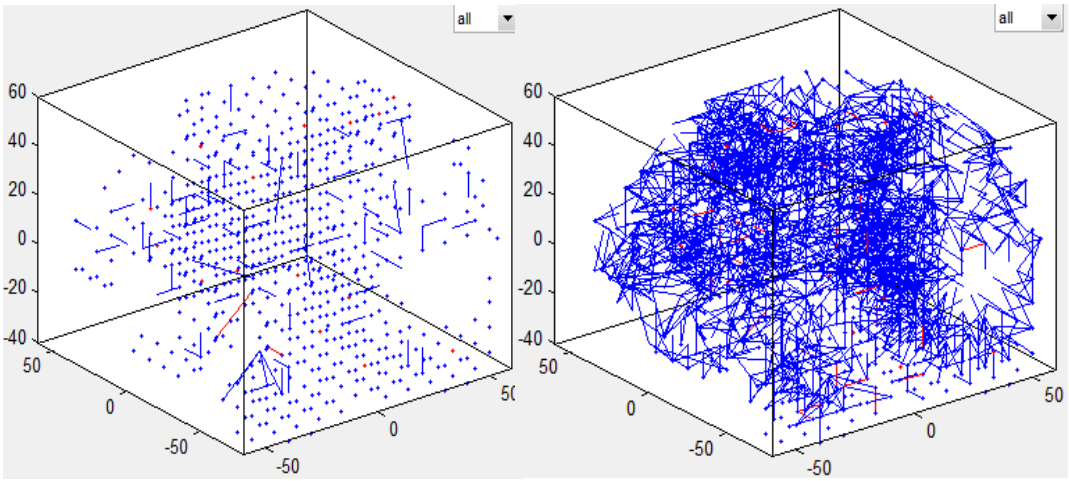

Figure 7-33: Visualization of neurons connectivity for data Sentence stimulus after STDP learning in 3D. For visualization purposes, displayed connections are based on weight more than 0.19 (left) and weight more than 0.09 (right).

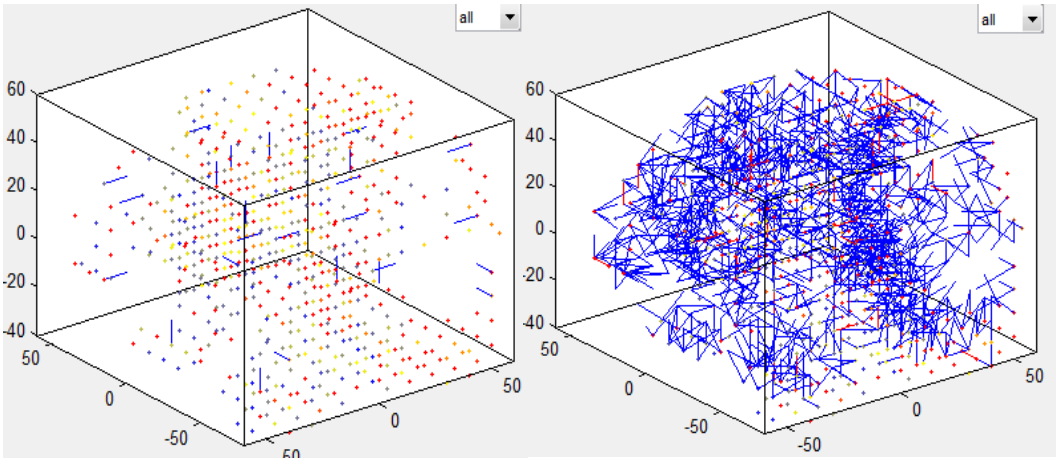

Figure 7-34: Visualization of neurons connectivity for data Picture stimulus after initialization in 3D view. For visualization purposes, displayed connections are based on weight more than 0.19 (left) and weight more than 0.09 (right).

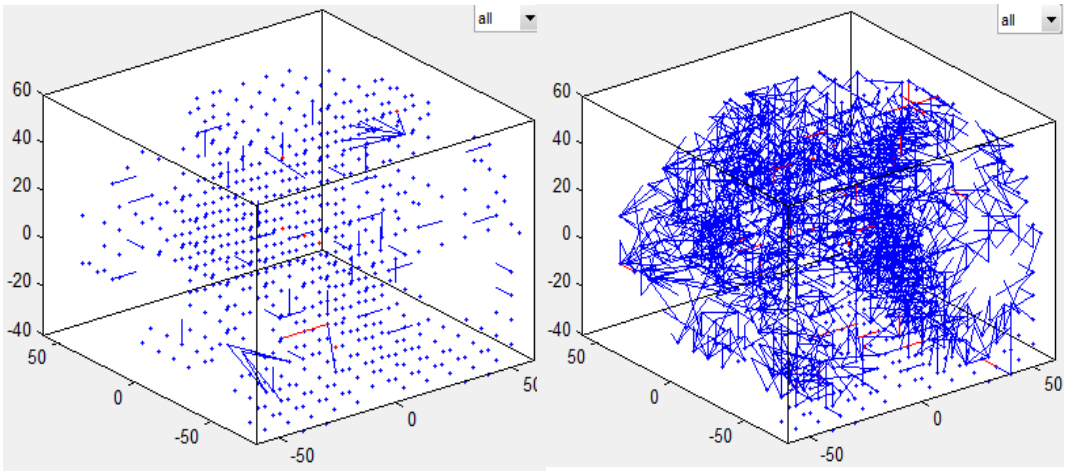

Figure 7-35: Visualization of neurons connectivity for data Picture stimulus after STDP learning in 3D view. For visualization purposes, displayed connections are based on weight more than 0.19 (left) and weight more than 0.09 (right).

From these observations, we can conclude that seeing a Sentence (or Picture) involves activation of neurons (emission of spikes) in many distributed and sometimes overlapped areas. For instance seeing a Picture activates similar neuron areas as seeing a Sentence which are CALC, ROPER, LSGA, LFEF and RFEF. This suggests that both experiment conditions created more and stronger connections in those areas than other

brain areas. The connectivity was especially enhanced between the input neurons (i.e. input voxels) in these areas.

Neurons in RSGA, LDLPFC, RDLPFC, LT and RT regions are more densely connected in Sentence stimulus as compared to the Picture stimulus suggesting activities of visual memory processing (LT and RT) (Kolb & Whishaw, 1996), information processing and perception (LDLPFC and RDLPFC) (Miller, Freedman, & Wallis, 2002) and language perception and processing (RSGA) (Gazzaniga et al., 2009). Areas with fewer or no connected lines are those areas in the NeuCube^{B} that are not really affected by either stimulus i.e. no spike was emitted such as in region SMA which are basically activated by physical voluntary/control movement.

In a nutshell, the analysis on the neurons connectivity provides a new method in analyzing the StarPlus data especially in visualizing the neuron connections which is based on the spiking activity of the data. By using the neuron connection weight, we can visualize from the model which neuron connections are stronger and which connections are weaker. We can also identify the regions with more spiking activities through the density of the connected neurons. This visualization is not provided in the previous studies involving the StarPlus data.

7.8 CHAPTER SUMMARY

The outcome from the first classification experiment through the application of NeuCube^{B} model with StarPlus fMRI data validates the NeuCube^{B} feasibility in analyzing complex spatio-temporal data. Applying this model on fMRI data have greatly improved classification accuracy as compared to other standard classifiers in determining different patterns of voxels. In addition to this, the output from the second experiment suggests that Sentence or Picture stimulus activates different but sometimes overlapped brain regions. More densely connected neurons could be seen in certain regions as compared to other regions suggesting more neurons activities (i.e. more spikes) had occurred during STDP learning in that particular regions. From both experiments, we can also conclude that NeuCube^{B} model offers a better understanding and better analysis of the StarPlus data compared to the previous studies involving the same dataset. Specifically in terms of visualization and neurons connectivity interpretation of the spatio-temporal data, the proposed NeuCube^{B} model has presented other benefits that conventional classifiers have failed to provide.

In the next chapter, NeuCube^{B} is tested on the second dataset involving visual object classification. The purpose of this experiment is to classify a stimulus against another stimulus i.e. Face versus Scrambled Pictures, and to classify a stimulus against the other 7 stimuli (Face versus Not Face and Scrambled Pictures versus Not Scrambled Pictures). The experimental setup, voxels mapping, unsupervised learning and supervised learning will be explained in detail. The classification results from the deSNN classifier technique is also tabulated and analyzed, together with the analysis about neurons connectivity that were created in the network before and after the learning process.

Chapter 8

CASE STUDY 2: APPLICATION OF NEUCUBE^{B} METHODOLOGY ON HAXBY VISUAL OBJECT RECOGNITION DATA

Our new NeuCube^{B} model is experimented with another set of fMRI data which is in a different format from the StarPlus. This chapter presented our second original case study that uses the NeuCube^{B} model in mapping, learning, classifying and interpreting the Haxby dataset. Although the same procedures (as in the first case study) are applied to the data, the complexity of this experiment, especially in the mapping process, poses a new challenge to our research. This is because of the NII format that this data hold. NII file type is in principal associated with NIfTI-1 data format defined by Neuroimaging Informatics Technology Initiative, to save volumetric fMRI data. A single volumetric data consists of a header, with extension .hdr to store meta-information, the actual data, with extension .*img* and other extra information.

The same generic methodology explained and used in Chapter 6 is used again in this second experiment involving a different set of complex SSTD i.e. named as Haxby's visual object recognition (Haxby et al., 2001) downloaded from OpenfMRI (Haxby et al., 2011). The successful results in the previous experiment motivate this new experiment. The generic NeuCube^{B} architecture as displayed in Figure 6-1 will be used. Our second mapping approach is different from the previous experiment because of the different brain dimension and size, as well as in the number of voxels.

8.1 EXPERIMENTAL SETUP

This experiment involves taking the neural response, as reflected in hemodynamic changes, in six different subjects (five female and one male). However, only five subjects (SUB001, SUB002, SUB003, SUB004 and SUB006) are considered for the experiments because the data for SUB005 is incomplete. They were presented with one-back repetition of Faces, Houses, Cats, Bottles, Scissors, Shoes, Chairs, and control Nonsense Patterns stimulus, all in gray-scaled colours as shown in Figure 8-1. Each stimulus is labelled as COND001 (House), COND002 (Scrambled Pictures), COND003 (Cat), COND004 (Shoe), COND005 (Bottle), COND006 (Scissors), COND007 (Chair)

and COND008 (Face). All stimuli from the category have the same base level name and the nonsense patterns were phase-scrambled images. For each subject, there are 12 time-series obtained where each time series began and ended with 12 seconds of rest and contained 8 stimulus blocks of 24 seconds duration, one for each category, separated by 12seconds intervals of rest. Each stimulus is displayed for 0.5 seconds with 1.5 seconds interval between the stimuli. The same face or object stimulus is photographed in 4 different angles, and there are 12 different exemplars for each stimuli category. The fMRI scanning is set on a GE 3T scanner (General Electric, Milwaukee, WI) [repetition time (TR) = 2500 millisecond, 40 3.5-mm-thick sagittal images, field of view (FOV) = 24 cm, echo time (TE) = 30 millisecond, flip angle = 90^{o}] where high-resolution images were obtained for each subject to provide detailed anatomy.

In this second experiment, we extracted the raw data from the file body.nii specified in the 40 x 64 x 64 x 121 .img structure. The .img structure is extracted and tabulated for each of the 5 subjects from the first stimuli displayed until the eighth stimuli, for each 12 experiment runs (time series). In every experiment run, a subject is presented with a stimulus of different exemplars for 0.5 seconds, another 1.5 seconds for an inter-stimulus interval, thus producing 24 seconds of a particular stimulus block. In between stimulus presentation, there are 12 seconds intervals of rest. For instance, Table 8-1 shows an example of the first time series for subject 1 (SUB001), starting with Scissors stimulus (COND006), followed by Face stimulus (COND008), Cat (COND003), Shoe(COND004), House (COND001), Scrambled Pictures (COND002), Bottle (COND005) and finally Chair stimulus (COND007). Each stimulus block is identified with unique IDs representing different time-points, which generates 108 time-series i.e. 9 IDs x 12 runs for each stimulus for each subject.

We conducted the experiment in 2 approaches: to classify a stimulus against another stimulus (Face versus Scrambled Pictures) and to classify a stimulus against the other 7 stimuli (Face versus Not Face and Scrambled Pictures versus Not Scrambled Pictures) for each subject. For the first approach, the classification is between 2 stimuli within the same subject, 12 samples for Class 1 (Face) and 12 samples for Class 2 (Scrambled Pictures). For the second approach, the samples for the classification is 96 samples in which 12 samples are for Class 1 (Face) and 84 samples are for Class 2 (Not Face). Figure 8-2 shows an example of Haxby data for Subject 1 (SUB001) when the subject is presented with Face stimulus for ID=1 and run=1.

Figure 8-1: Examples of stimulus used in the Haxby visual experiment. Subjects performed a one-back repetition detection task i.e. the same object is presented with different views. Picture is taken from Haxby et al., (2001).

Table 8-1: Example of task run 1 for SUB001 in which the subject is presented first with Scissors, followed by Face, Cat, Shoe, House, Scrambled Picture, Bottle and finally Chair stimulus, and the brain image is captured in every 2.5 seconds (0.5 seconds to display stimulus and another 1.5 seconds are the interval between stimuli)

ID	TR (s)	Task Run	ID	TR (s)	Task Run	ID	TR (s)	Task Run	ID	TR (s)	Task Run
1	0		31	75		61	150		91	225	
2	2.5		32	77.5		62	152.5		92	227.5	
3	5		33	80		63	155		93	230	
4	7.5		34	82.5		64	157.5		94	232.5	
5	10		35	85		65	160		95	235	
6	12.5		36	87.5		66	162.5		96	237.5	
7	15		37	90		67	165		97	240	Bottle
8	17.5		38	92.5		68	167.5	House	98	242.5	
9	20		39	95	Cat	69	170		99	245	
10	22.5	Scissors	40	97.5		70	172.5		100	247.5	
11	25		41	100		71	175		101	250	
12	27.5		42	102.5		72	177.5		102	252.5	
13	30		43	105		73	180		103	255	
14	32.5		44	107.5		74	182.5		104	257.5	
15	35		45	110		75	185		105	260	
16	37.5		46	112.5		76	187.5		106	262.5	
17	40		47	115		77	190		107	265	
18	42.5		48	117.5		78	192.5		108	267.5	
19	45		49	120		79	195		109	270	
20	47.5		50	122.5		80	197.5		110	272.5	
21	50		51	125		81	200	Scrambled Picture	111	275	Chair
22	52.5		52	127.5		82	202.5		112	277.5	
23	55		53	130	Shoe	83	205		113	280	
24	57.5		54	132.5		84	207.5		114	282.5	
25	60	Face	55	135		85	210		115	285	
26	62.5		56	137.5		86	212.5		116	287.5	
27	65		57	140		87	215		117	290	
28	67.5		58	142.5		88	217.5		118	292.5	
29	70		59	145		89	220		119	295	
30	72.5		60	147.5		90	222.5		120	297.5	
									121	300	

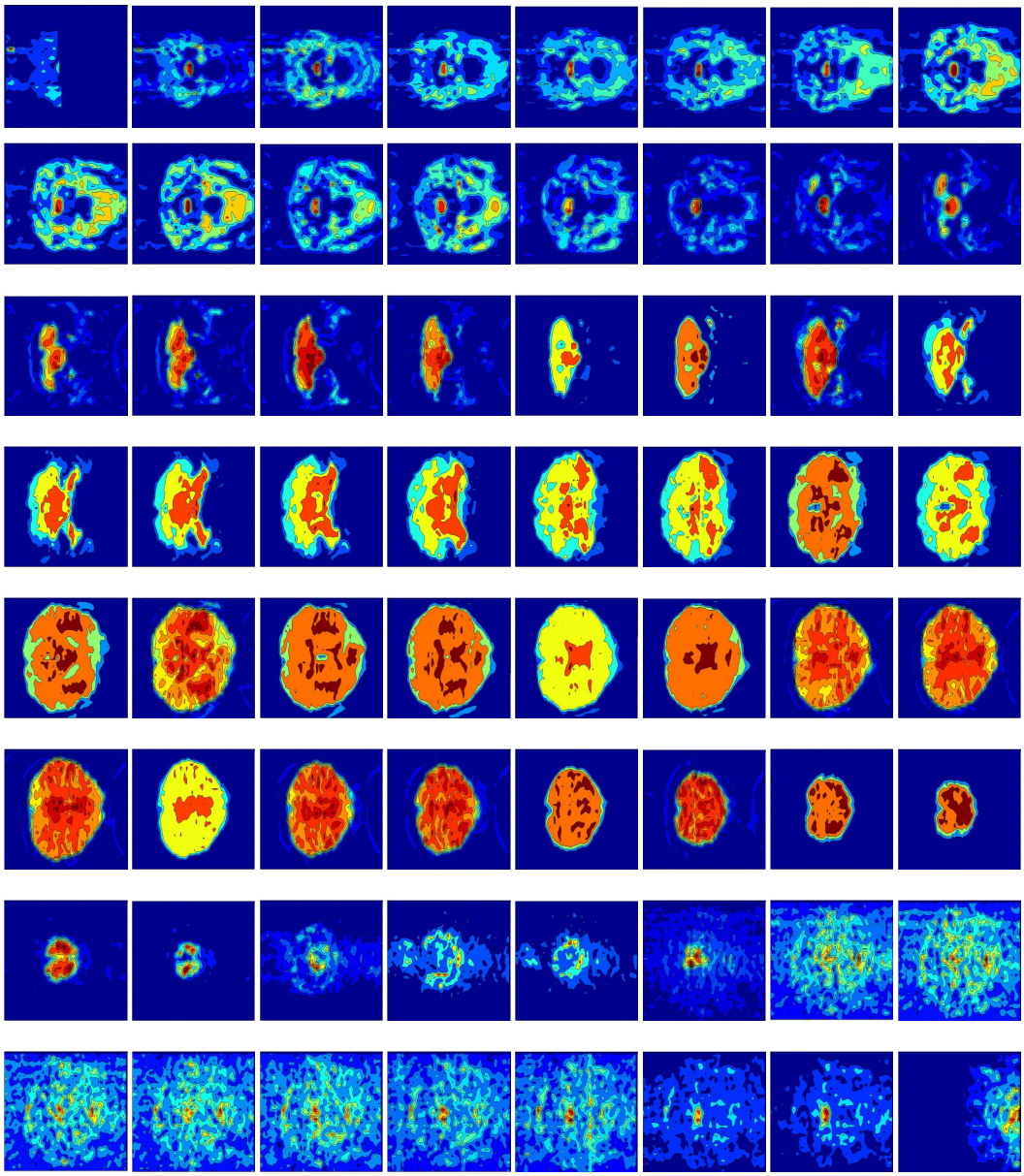

Figure 8-2: Illustration of Haxby data (Haxby et al., 2011) for SUB001 when the subject is presented with Face stimulus for ID=1 and Run=1

8.2 MAPPING OF NIfTI-1 fMRI DATA INTO NEUCUBE^{B}

Firstly, we have to map the Haxby voxels' coordinates into the 3D of NeuCube^{B} architecture. The same steps explained in Chapter 6 are followed which is to extract the coordinates and then to adjust the coordinates according to the NeuCube^{B} coordinate. The size of the NeuCube^{B} is built in Talairach coordinate space with 1,471 defined neuron locations, while Haxby's size has 40 x 64 x 64 defined neuron locations and thus generating163,840 neuron locations with possible input neurons.

8.2.1 Neuron Mapping

Figure 8-3 shows the direct coordinates mapping of Haxby's fMRI neuron locations into NeuCube^{B} neurons in 3D space for subject SUB001 in which blue dots represent NeuCube^{B} neurons while cyan dots represent Haxby neurons. Figure 8-4, Figure 8-5 and Figure 8-6 show the different 3 views of the direct mapping i.e. in xy, xz and yz views. To be able to use the voxels for learning, these neurons will be mapped according to the mapping procedure explained in Chapter 6. The visualizations of this mapping are as depicted in Figure 8-7, Figure 8-8 and Figure 8-9.

After mapping, the Haxby neurons' coordinates are reduced to 19,930. The next step is to select those important voxels that will be the input to the NeuCube^{B} learning based on certain threshold value. In this experiment, the threshold is set to 1,000 and the calculated significant voxels are 19,930. The lower the threshold value is, the more voxels are identified. From these 19,930 voxels, their respective voxel coordinate locations are determined. Next, every coordinates that are close to the NeuCube^{B} coordinates that are within the specified radius are also identified, thus reducing the coordinates from 19,930 to 1,471 following the 1,471 NeuCube^{B} neuron locations. Based on each calculated neuron locations, all voxels that are within each NeuCube^{B} neuron and within the specified radius are averaged (radius is set to 7 mm). However, only 854 neuron locations are with neurons, while the rest are empty location i.e. no Haxby coordinate is mapped to the NeuCube^{B} coordinate. The 854 computed voxels values are the actual input voxel to the framework. The other details for the other conditions for each subject are tabulated in Table 8-2. Because of the imaging experiments were conducted based on each experiment condition, the number of voxels (of the same subject) was not the same thus producing a slightly different number of input neurons. The input voxels are then encoded into spike sequences that will be elaborated in the next section.

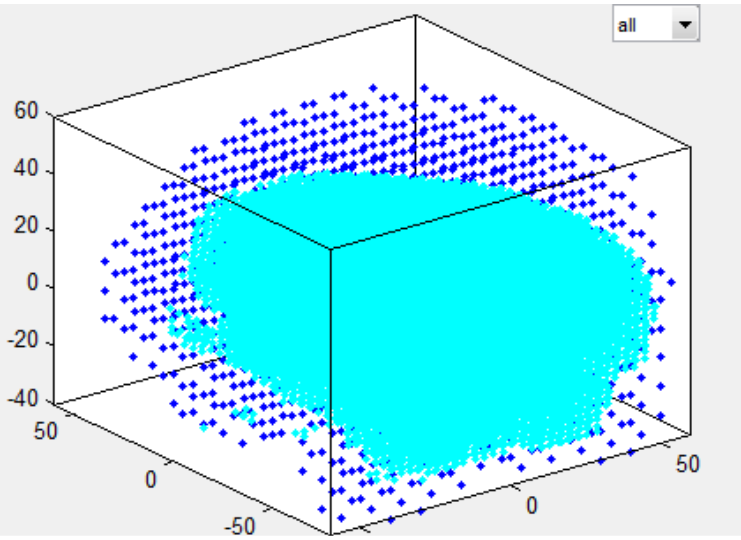

Figure 8-3: Visualization of coordinates that are mapped into the NeuCube^{B}. Cyan dots represent the coordinates of Haxby fMRI brain data while blue dots are coordinates of NeuCube^{B}, in 3D view for subject SUB001.

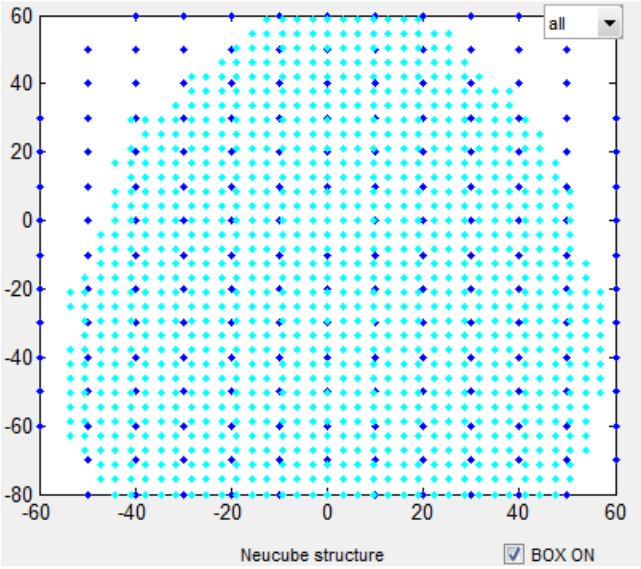

Figure 8-4: Visualization of coordinates that are mapped into the NeuCube^{B}. Cyan dots represent the original voxels of Haxby fMRI brain data while blue dots are coordinates of NeuCube^{B}, in xy view for subject SUB001.

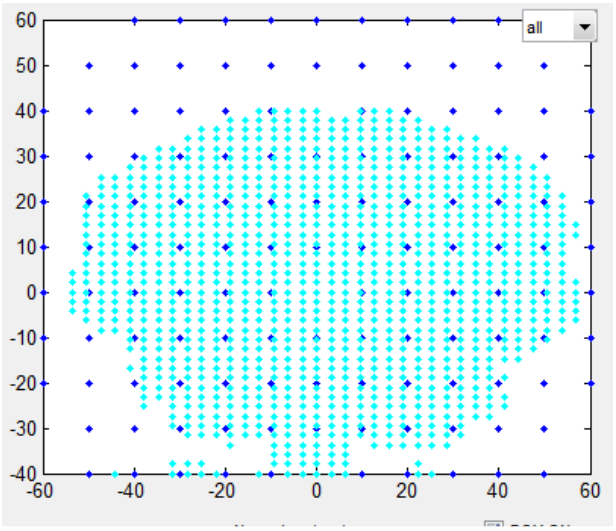

Figure 8-5: Visualization of coordinates that are mapped into the NeuCubeB. Cyan dots represent the original voxels of Haxby fMRI brain data while blue dots are coordinates of NeuCubeB, in xz view for subject SUB001.

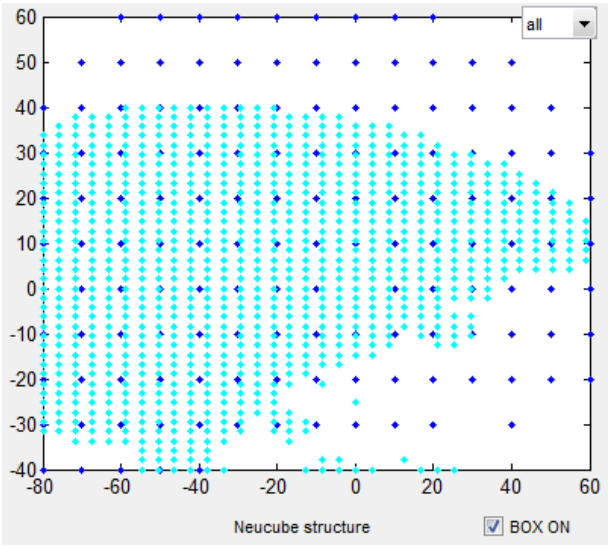

Figure 8-6: Visualization of coordinates that are mapped into the NeuCube^{B}. Cyan dots represent the original voxels of Haxby fMRI brain data while blue dots are coordinates of NeuCube^{B}, in yz view for subject SUB001.

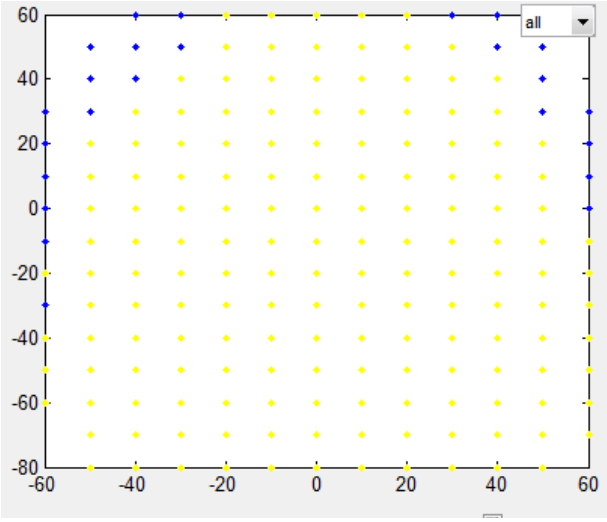

Figure 8-7: Visualization of mapped Haxby coordinates into NeuCube[B] coordinates, in xy view for subject SUB001 after the mapping procedure.

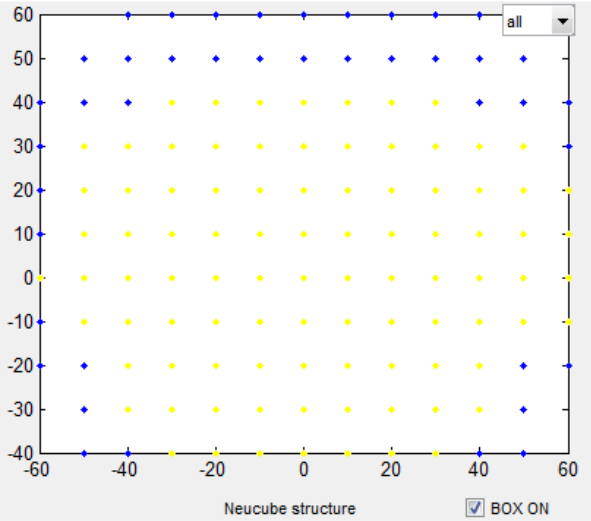

Figure 8-8: Visualization of mapped Haxby coordinates into NeuCube[B] coordinates, inxz view for subject SUB001 after the mapping procedure.

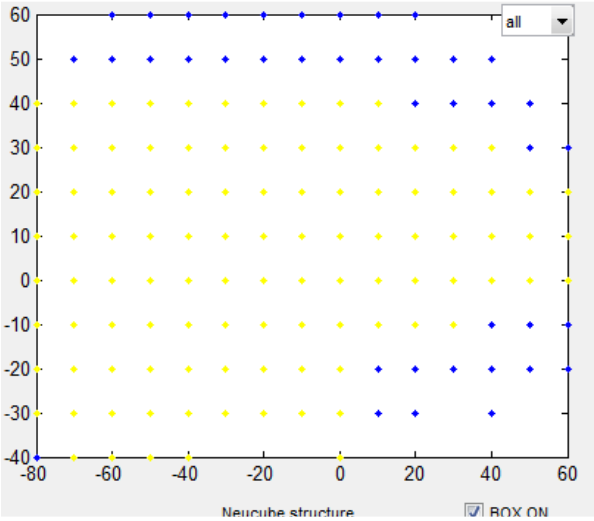

Figure 8-9: Visualization of mapped Haxby coordinates into NeuCube[B] coordinates, in *yz* view for subject SUB001 after the mapping procedure.

8.3 DATA ENCODING

Input fMRI data that have already been mapped as described in the previous section is encoded using AER method which is based on the difference between two consecutive voxel values of the same input variable. If the difference is greater than a certain threshold value, a spike will be generated. For instance in Table 8-2, for 854 input voxels, 1,708 spike states are generated: 108 x 1,708 spike states for training and 108 x 1,708 spike states for validation. The spike states length (108) is computed from the time length (9 seconds) times the number of samples (12 samples i.e. 6 samples for Class 1 and 6 samples for Class 2) with the assumption of 50% of samples is used for training and 50%samples are used for validation. These spike states are for 12 samples of Class 1 (Face) and 12 samples of Class 2 (Scrambled Pictures).

For the experiment involving classification of a stimulus versus the other 7 stimuli, the spike states details are displayed in Table 8-3. For illustration purposes, spike states generated for training and validation for SUB001 are 432 x 1,708. The spike states length (432) is calculated from the time length (seconds) x the number of samples (i.e.:9 x 48) in which the 48 samples are composed of 6 samples for Class 1 and 42 samples for Class 2.

Table 8-2: Total number of neurons for each subject, the actual input neurons and the corresponding spike states for training and validation for a stimulus against the other stimulus

Subject	Total Neurons	Actual Input Neuron	Spike States for Training	Spike States for Validation
SUB001	19,930	854	108 x 1,708	108 x 1,708
SUB002	8,853	603	108 x 1,206	108 x 1,206
SUB003	7,999	571	108 x 1,142	108 x 1,142
SUB004	18,778	823	108 x 1,646	108 x 1,646
SUB006	24,841	1,051	108 x 2,012	108 x 2,012

Table 8-3: Total number of neurons for each subject, the actual input neurons and the corresponding spike states for training and validation for a stimulus against the other 7 stimuli

Subject	Total Neurons	Actual Input Neuron	Spike States for Training	Spike States for Validation
SUB001	19,930	854	432 x 1,708	432 x 1,708
SUB002	8,853	603	432 x 1,206	432 x 1,206
SUB003	7,999	571	432 x 1,142	432 x 1,142
SUB004	18,778	823	432 x 1,646	432 x 1,646
SUB006	24,841	1051	432 x 2,012	432 x 2,012

8.4 DATA LEARNING

We have applied STDP learning method in the NeuCube^{B} to initialize, modify and retain connection weights (memory) throughout the unsupervised learning stage. In learning the data, we have taken two approaches. The first approach is to classify the different brain patterns for each subject. The idea is to differentiate brain areas that are affected by the Face stimulus and the brain areas that are affected by the Scrambled Picture for each subject. The second approach is to classify the brain patterns whether the subject is looking at a Face stimulus or that the subject is looking at the other 7 stimuli. The choice of approaches is based on the study of activation patterns that are distributed and overlapped in the ventral temporal cortex and ventral occipital cortex of the human brain (Haxby et al., 2001; Ishai, Ungerleider, Martin, & Haxby, 2000).

8.4.1 Stimulus by Stimulus Learning for Each Subject Approach

Taking the same approach as (Haxby et al., 2001) which distinguish different brain activation patterns for each stimulus, we apply the same data formulation to run on STDP learning. For each subject, the classification of brain patterns is based on a stimulus

versus the other one stimulus (i.e. Face against Scrambled Pictures). There are a total of 24 samples generated for both classes. For illustration, the following parameter values, determined after a grid search, were selected for classification accuracy:

- NeuCube^{B} is constructed with 1,471 neurons
- Threshold for the AER is 2.
- SWC is set to 0.15, which is used to initialize neurons connection.
- Threshold of LIFM neurons is 0.5
- Leak parameter of the LIFM neurons is 0.002;
- STDP learning rate is 0.01;
- Number of training is 2 times;
- Mod parameter is 0. 4 and drift is 0.25.

For illustration, after completing the STDP learning, neuron connections have increased and new neurons connections have evolved. In certain part of the SNN Cube, the connections are more concentrated as compared to other areas suggesting the increase in the calculated synaptic weight and consequently increasing the number of connections. The connections between neurons are represented by blue (or red) lines if the calculated weights are greater (or less) than 0. Spikes emissions are determined by the calculated neuron potentials, which are based on the connection weights. Figure 8-5 demonstrates connections between neurons after initialization process and after unsupervised learning is completed. Blue lines represent positive connections while red lines represent negative connections. Neurons in blue are active neurons whereas neurons in red are inactive neurons.

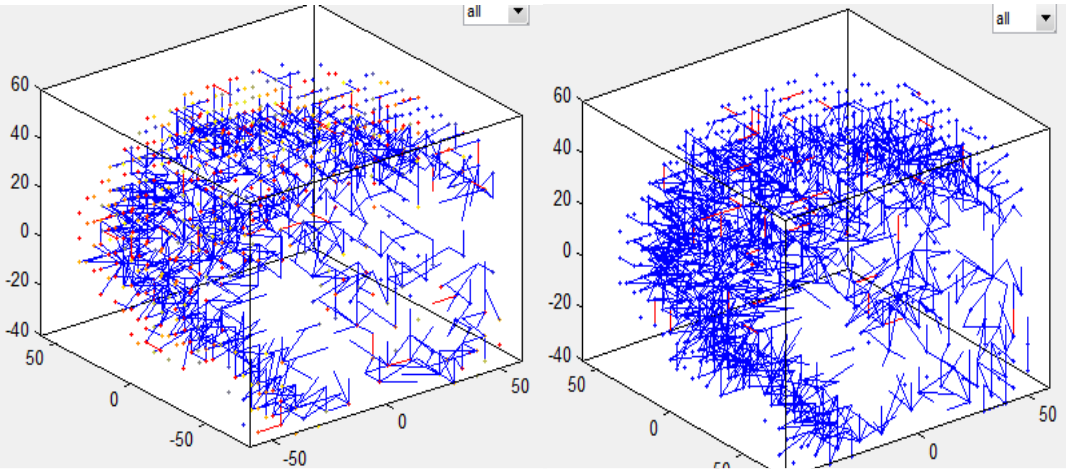

Figure 8-10: Neuron connections after initialization (left) and neuron connections after unsupervised learning (right).

8.4.2 Subject by Subject (Personalized) Approach

Another approach is more on personalized way of learning the brain data i.e. for a particular subject, classification of brain patterns is based on 1 stimulus (i.e. Face) versus the other 7 stimuli (the other stimuli except Face stimulus). We have generated 8 different experiment combinations. As illustration, the following parameter values were selected for classification accuracy:

- NeuCube^{B} is constructed with 1,471 neurons
- Threshold for the AER is 4.25;
- SWC is set to 0.15;
- Threshold of LIFM neurons is 0.8;
- Leak parameter of the LIFM neurons is 0.002;
- STDP learning rate is 0.01;
- Number of training is 2 times;

- Mod parameter is 0. 4 and drift is 0.25.

The parameter setting stated above is determined after a thorough grid search from countless experiment runs, which resulted with highest percentage accuracy.

8.5 MINING USING DYNAMIC EVOLVING SPIKING NEURAL NETWORKS (deSNN)

In our second selected dataset i.e. Haxby visual object recognition, the potential of deSNN as a training algorithm is further explored. After going through STDP learning in the previous stage, the already trained fMRI data is fed into the second stage of learning i.e. deSNN for classification. In the deSNN (Dhoble et al., 2012) algorithm as presented in Section 3.3.1, for every fMRI input, the connection weights are initialized using the RO learning rule, connections weights for consecutive spikes are adjusted using SDSP learning rule, and the maximum post synaptic potential and threshold value are calculated. For our experiment, two classes of stimuli are formulated i.e. Class 1 (Face) versus Class 2 (Scrambled Pictures), to differentiate which brain area is activated by the different stimulus. Parameters used in the experiment are presented in the previous section. The training and testing are conducted using 50:50 split i.e. 50% of samples is used as training set and the rest 50% is used for testing.

The SVM experiment uses Polynomial Kernel of first degree while MLP experiment uses 5 hidden nodes, with learning rate of 0.001 and 100 iterations.

8.6 RESULT AND DISCUSSION

The following section presents and discusses the results on a stimulus-by-stimulus basis for each subject experiment and followed by the results on the subject-by-subject basis i.e. personalized experiment.

8.6.1 Stimulus by Stimulus for Each Subject Classification Results

In this experiment, we have trained the network with parameters as shown in section 8.4.1. For instance, for SUB004, there are 823 input neurons for Class 1 (Scrambled Pictures) and Class 2 (Face) trained in unsupervised and supervised mode. From these inputs, the calculated spike states for training are 108 x 1,708 and the spike states for

validation are 108 x 1708. A snapshot of the NeuCube^{B} software implementation after a successful classification is presented in Figure 8-6. The best percentage accuracies obtained for the designed and trained NeuCube^{B} against eSNN and standard classifiers (SVM and MLP) for 24 samples for each subject (Table 8-4). As for the NeuCube^{B} and eSNN model, the networks are trained in unsupervised and supervised stages. The best classification accuracy was obtained for NeuCube^{B} involving subject SUB004 which was 83% (83% for Class 1 and 83% for Class 2) as compared to 75% for eSNN (67% for Class 1 and 83% for Class 2), 67% for SVM and 57% for MLP. For the experiment involving subject SUB003, eSNN outperformed deSNN of NeuCube^{B} and other conventional classifiers in classifying the stimuli by achieving 75% classification accuracy (67% for Class 1 and 83 for Class 2) as compared to NeuCube with only 67%. For all other experiments, conventional classifiers produced inferior classification accuracies than NeuCube^{B} and eSNN.

Figure 8-7(a) shows neurons in the NeuCube^{B} in which blue dots represent NeuCube^{B} neurons and yellow dots represent Haxby input neurons. The NeuCube^{B} is said to be in initialization stage. Figure 8-7(b) shows the spiking activity in the NeuCube^{B} during the learning process. Active neurons are represented in red, inactive neurons are represented in blue, positive input neurons are represented in magenta, negative input neurons are represented in cyan and zero input neurons are represented in yellow. Neuron connectivities before and after training are displayed in blue lines, i.e. positive connections, and red lines i.e. negative connections (Figure 8-7(c) and (d)). More neuron connections are created after the learning process than before the learning process.

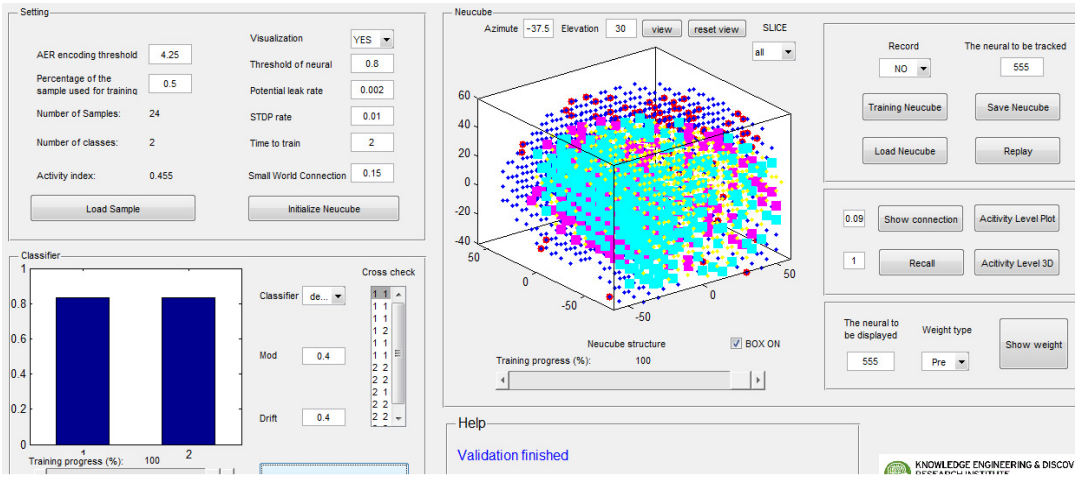

Figure 8-11: A snapshot of a software implementation of the NeuCube^{B} architecture for classification of 2 class fMRI data for SUB004. The parameter values are as stated in section 7.5.2. The classification model used is deSNN with an accuracy of 83.3% for Class 1 and 83.3% for Class 2.

Table 8-4: Classification accuracy percentage of Class1 (Face) and Class 2 (Scrambled Pictures) stimulus for all subjects

Subject	SVM	MLP	eSNN	NeuCube^{B}
Sub001	64	52	67 (83, 50)	75 (100,50)
Sub002	57	54	67 (50, 83)	75 (77,73)
Sub003	57	56	75 (67, 83)	67 (43, 90)
Sub004	67	57	75 (67, 83)	83 (83, 83)
Sub006	65	52	67 (67, 67)	67 (50, 83)

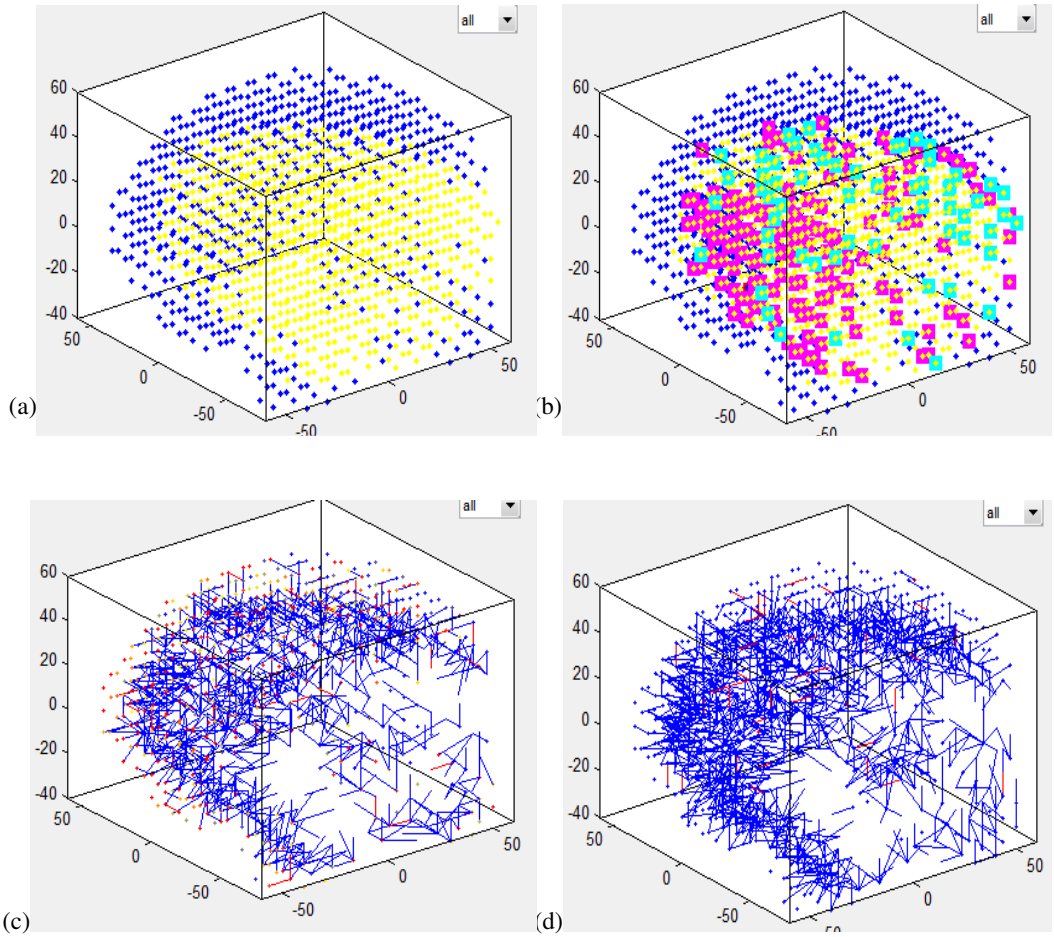

Figure 8-12: Visualization of fMRI data model and connectivity between neurons of eSNN: (a) no spiking activity yet, inactive neurons are in blue, fMRI data neurons (input neurons) are in yellow; (b) spiking activity: active neurons are represented in red, inactive neurons are represented in blue, positive input neurons are represented in magenta, negative input neurons are represented in cyan and zero input are represented in yellow; (c) neurons connectivity before training (small world connection), positive connections are in blue and negative connection in red; (d) neurons connectivity after training.

From the results, it clearly shows that the NeuCube^{B} model once again is more applicable to handle complex and noisy data that reside in STBD fMRI as compared to eSNN and conventional classifiers. NeuCube^{B} also offers a display on the neuron connectivities, indicating as to which areas of the brain are most activated by the Face or Scrambled Picture stimulus.

8.6.2 Subject by Subject (Personalized) Classification Result

The network is trained with parameters as shown in Section 8.4.2 to the 96 samples of each subject in unsupervised and supervised mode. Samples are fed into the network one at a time to maintain the temporal information. Figure 8-8 shows a snapshot of the NeuCube^{B} software implementation after successful classification.

The overall percentage accuracies obtained using 50% of the data for training and 50% of data for testing, for the designed and trained NeuCube^{B} against eSNN and standard classifiers (SVM and MLP) for subject SUB001, SUB002, SUB003, SUB004 and SUB006 are as depicted in Table 8-5. Values in brackets are the average of accuracy (in percent) obtained for Class 1 and Class 2 from 5 test runs. From all experiment conditions formulated for each subject, NeuCube^{B} has moderately outperformed eSNN, MLP and SVM, with an average accuracy achieved at 85% as compared to 80% for eSNN, 76% for MLP and 66% for SVM. The highest accuracy is attained from SUB001 Face against Not Face test run (90%) of that, 23% correctly recognizing brain patterns generated from Face stimulus (Class 1) and that 99% correctly recognizing brain patterns generated from the other 7 stimulus (Class 2).

In all test runs conducted, Class 1 achieved very small accuracy percentage as compared to Class 2. For instance the lowest result for Class 1 of MLP classifier is 0% which is achieved in SUB001 and SUB002 for Scrambled against Not Scrambled experiments. As for SVM classifier, the lowest result for Class 1 is 12% which is achieved for SUB001 for Scrambled against Not Scrambled and for Face against Not Face experiments. Low accuracy result is also achieved by NeuCube^{B} with 13% of Class 1 for Scrambled against Not Scrambled experiment and 17% of Class 1 for Face against Not Face experiments for SUB001 and SUB002 respectively. However, in all experiments we can conclude that NeuCube^{B} performed better in discriminating Class 1 against Class 2 for all 5 subjects.

Figure 8-9(a) shows neurons in the Cube in which blue dots represent NeuCube^{B} neurons and yellow dots represent Haxby input neurons. The Cube is said to be in initialization stage. Figure 8-9(b) shows the spiking activity in the Cube during the learning process. Active neurons are represented in red, inactive neurons are represented in blue, positive input neurons are represented in magenta, negative input neurons are represented in cyan and zero input neurons are represented in yellow. Neuron connectivities before and after training are displayed in Figure 8-9(c) and (d), displayed in blue lines, which indicate positive connections, and red lines, which indicate negative con-

nections. More neuron connections were created after the learning process than before the learning process.

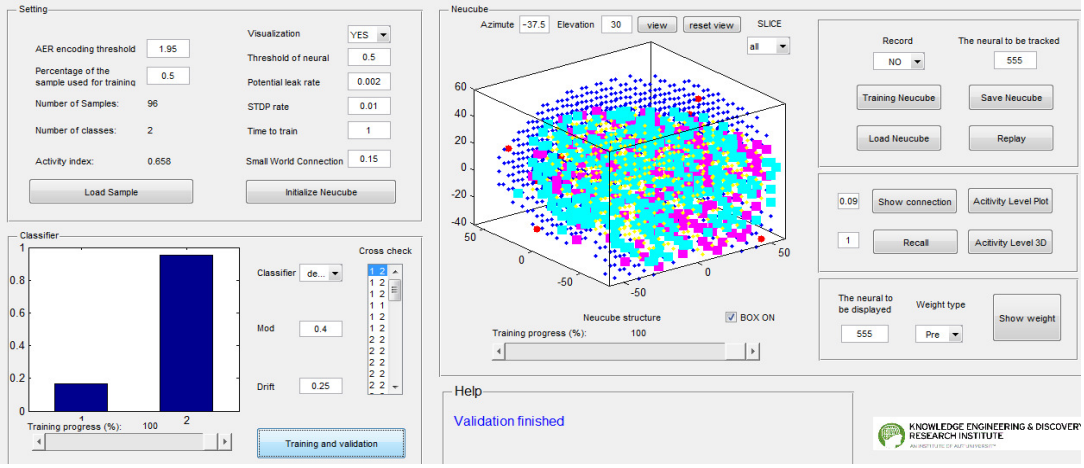

Figure 8-13: A snapshot of a software implementation of the NeuCube[B] architecture for classification of 2 class fMRI data for SUB001. The classification model used is deSNN with accuracy at 17% for Class 1 and 97% for Class 2

Table 8-5: Overall percentage of classification results for each subject for Face versus Not Face and Scrambled Pictures versus Not Scrambled Pictures.

Subject / Item	MLP	SVM	eSNN	NeuCube[B]
SUB001				
Scrambled VS Not Scrambled	87 (0, 87)	76 (12, 87)	76 (7, 86)	88 (13, 97)
Face VS Not Face	86 (10, 88)	74 (12, 87)	85 (47, 92)	90 (23, 99)
SUB002				
Scrambled VS Not Scrambled	71 (0, 73)	55 (17, 73)	80 (10, 91)	83 (33, 91)
Face VS Not Face	71 (6, 75)	62 (23, 75)	80 (17, 89)	86 (17, 96)
SUB003				
Scrambled VS Not Scrambled	77 (49, 80)	70 (31, 75)	78 (37, 84)	87 (50, 92)
Face VS Not Face	74 (27, 76)	64 (33, 79)	79 (27, 86)	76 (17, 85)
SUB004				
Scrambled VS Not Scrambled	74 (8, 75)	69 (30, 77)	75 (17, 83)	81 (20, 90)
Face VS Not Face	72 (3, 74)	68 (31, 78)	77 (3, 88)	83 (17, 92)
SUB006				
Scrambled VS Not Scrambled	73 (3, 75)	68 (27, 75)	81 (7, 92)	85 (33, 93)
Face VS Not Face	71(21, 76)	58 (20, 73)	85 (33, 92)	90 (17, 100)

From these results, we can conclude that the low percentage accuracies for Class 1 are resulted from the low number of samples (i.e. 12 samples only; 6 samples for training and 6 samples for testing) in which the network has not enough sample data (and time) to learn the pattern that may resides within the data. As more samples are

extracted and used during the STDP learning, the more the network learns from the brain pattern. Consequently, a higher percentage of correctly classified stimuli are achieved. This situation can be reflected from the high percentage accuracies achieved in NeuCube^{B} model for Class 2 involving 96 samples (i.e. 48 samples for training and 48 samples for testing) which resulted as high as 100% accuracy for SUB006 i.e. the network could perfectly distinguish that the data is Not-Face data. In general, NeuCube^{B} model provides better results as compared to eSNN and the conventional classifiers.

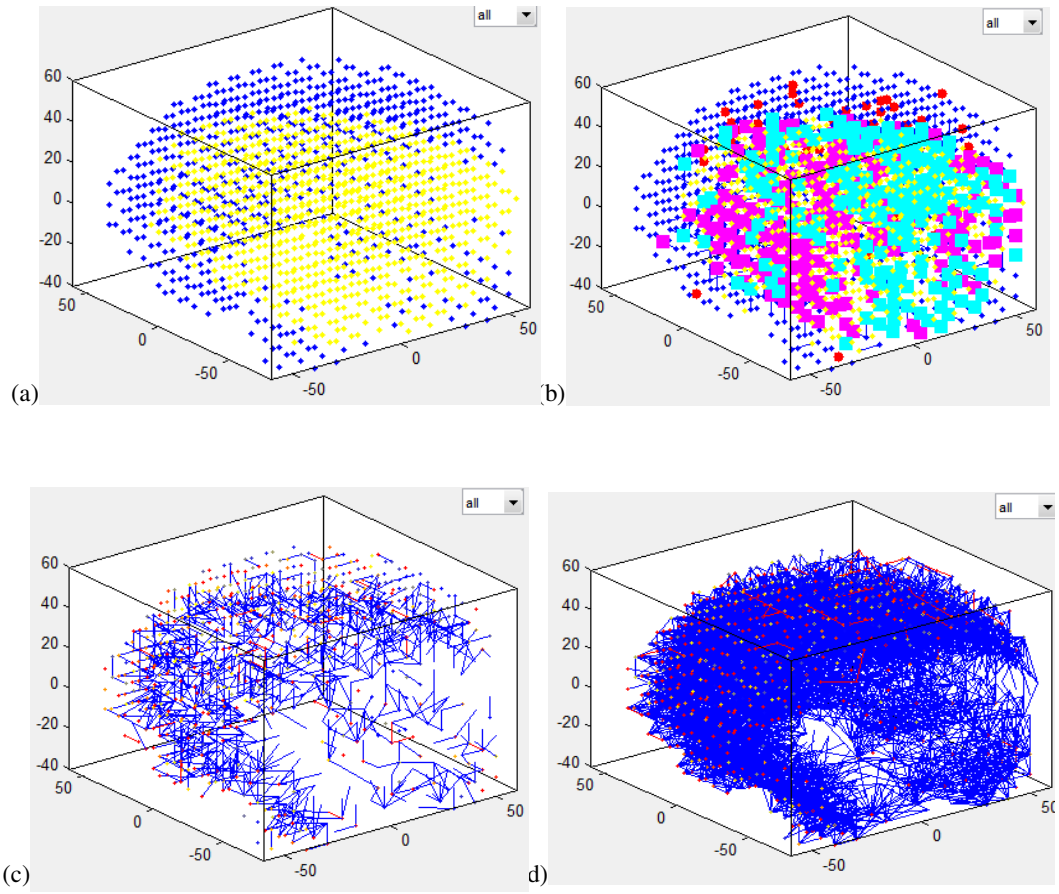

Figure 8-14: Visualization of fMRI data model and connectivity between neurons of eSNN for SUB001 for Face Versus Not Face test run: (a) no spiking activity yet, inactive neurons are in blue, fMRI data neurons (input neurons) are in yellow; (b) spiking activity: active neurons are represented in red, inactive neurons are represented in blue, positive input neurons are represented in magenta, negative input neurons are represented in cyan and zero input are represented in yellow; (c) neurons connectivity before training (small world connection), positive connections are in blue and negative connection in red; (d) neurons connectivity after training.

In addition, neuron connectivities of the network after initialization, after unsupervised training and after supervised training could be visualized and studied. From these connections, NeuCube^{B} offers visualization as to which part of the brain is activated most by the stimulus. More details on these neuron connectivities are presented in Section 8.7.

8.7 NEURONS CONNECTIVITY

For illustration, during the unsupervised training, data for Face stimulus and Scrambled Pictures of SUB001 are entered into the model over time, where connection weights between neurons are created and calculated. Figure 8-10 presents neuron connectivities before and after 5 times of STDP learning, which actually represent spatio-temporal interaction between input voxels (i.e. input neurons) in the network. Blue lines represent positive connections (i.e. connection weight > 0) and red lines represent negative connections (i.e. connection weight < 0). Neurons in blue indicate active neurons while red neurons indicate inactive neurons. The more and stronger connections were formed between the neurons after the learning process than before the learning process.

8.7.1 NeuCube^{B} Visualization for Spatio-temporal Connections Based on the fMRI Spiking Activity for Face Stimulus

In initialization stage, NeuCube^{B} connections and their connection weights are initialized randomly (Figure 8-10(a)). For visualization purposes, the left figures (Figure 8-10(a) and Figure 8-10(c)) show the neuron connections that were created with connection weight >= 0.09 (i.e. weaker spikes) while the right figures (Figure 8-10(b) and Figure 8-10(d)) show neuron connections that have weight >= 0.19 (i.e. stronger spikes). NeuCube^{B} active neurons which weights are more than 0.09 are initialized somewhere in left and right Superior Frontal Gyrus (SFG), left and right Medial Frontal Gyrus (MFG), left Inferior Frontal Gyrus (IFG), left and right Superior Parietal Lobe (SPL), left Inferior Parietal Lobe (IPL), Superior Temporal Gyrus (SPG), Inferior Temporal Gyrus (ITG) and right and left Occipital Lobe (OL) regions. Higher NeuCube^{B} weights (weight >= 0.19) which indicate stronger spikes are in MFG, IFG, SPL and OL regions. Figure 8-11 shows these approximated spatial locations of neuron connections in the NeuCube^{B} after initialization (i.e. before learning process). Regions with highly

interconnected positive blue lines were areas that were with most spikes activities and were highly activated with the Face stimulus.

After initialization, NeuCube^{B} neuron potential is calculated and updated, whether to spike or not to spike, assuming that each sample of data is taken in different times. Therefore the potential will not continue from one sample to the next sample i.e. NeuCube^{B} neuron potential is reset to 0 for each incoming sample. Each neuron potential is increased if NeuCube^{B} pre-synaptic neuron (i) and its post-synaptic neuron (j) are equal to 1 and NeuCube^{B} refractory neuron (j) is equal to 0. Each neuron's state is also determined, whether or not it has reached the threshold. For neurons that still have not reached the threshold, their neuron potentials are leaked. NeuCube^{B} neuron connections (memory) and weights are also calculated and updated hence resulting in more densely interconnected neurons.

After 5 times of STDP unsupervised learning, as shown in Figure 8-10(c), more positive spikes were emitted (i.e. more connections are created) and connection weights are also updated. Figure 8-11 and Figure 8-12 show a different view of NeuCube^{B} dynamic activities through a visualization of positive and negative connections. NeuCube^{B} connections are more concentrated (i.e. more positive connections) on the same areas as mentioned before. However higher NeuCube^{B} weight connections (weight>= 0.21), thus implying stronger spikes, were identified in left SFG, left ITG and right MFG as shown in Figure 8-13.

From these connections, we can conclude that these areas received the most activation (i.e. stronger spikes) during the whole learning process. These areas among all are responsive to visual stimulus as well as to sensory system (e.g. vision/smell/touch/auditory/taste) coordination (SFG) (Goldberg, Harel, & Malach, 2006), facial/body recognition (ITG) (Haxby et al., 2001) and visual stimulus (MFG) (Talati & Hirsch, 2005).

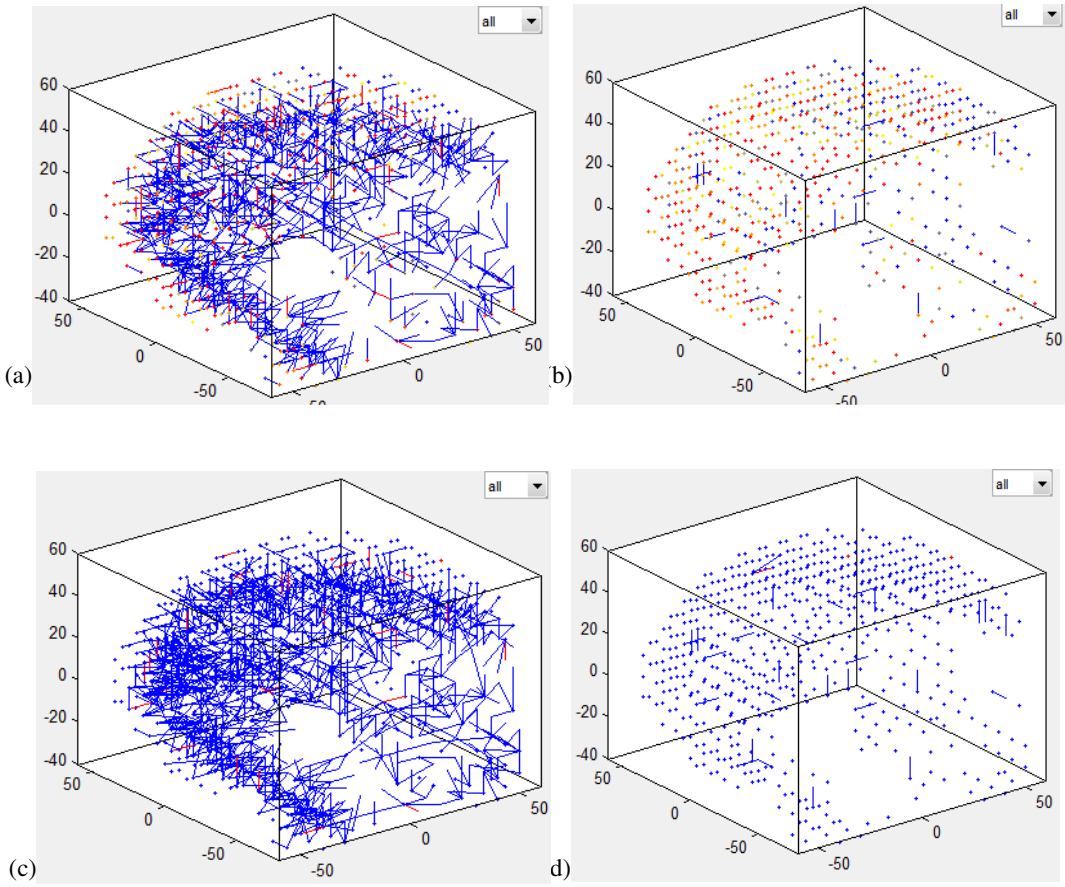

Figure 8-15: Visualization of neurons connectivity for Face stimulus (a) before and (c) after STDP learning with (b) weight >= 0.09 and (d) weight >= 0.19. Positive connections are in blue and negative connections are in red.

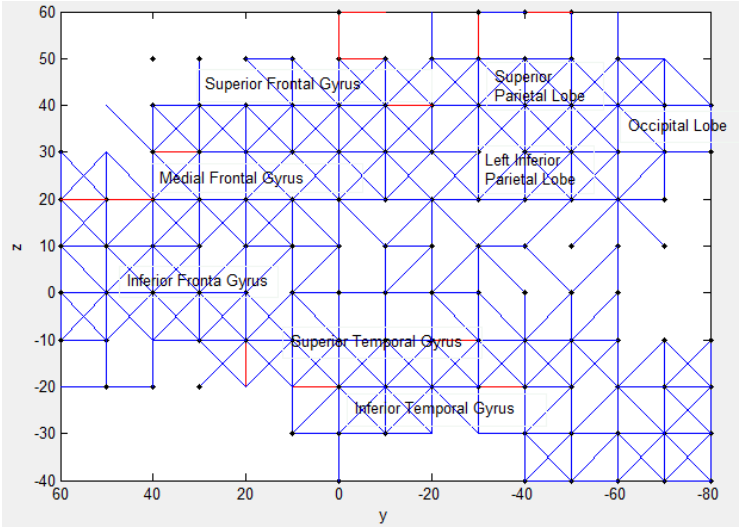

Figure 8-16: Approximated location of densely interconnected neurons that are activated when seeing Face stimulus before training.

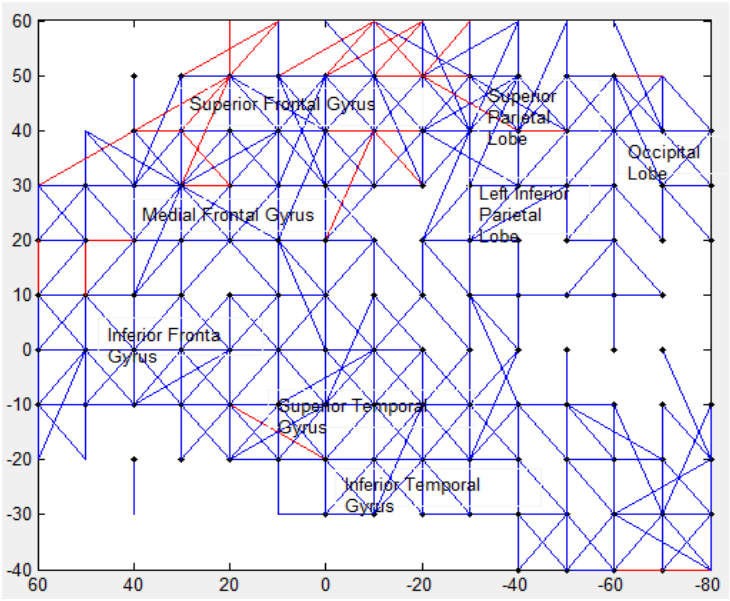

Figure 8-17: Approximated location of densely interconnected neurons that are activated when seeing Face after the learning process. More connections are created in areas SFG, MFG, IFG, STG, IFG, SPL, IPL and OL.

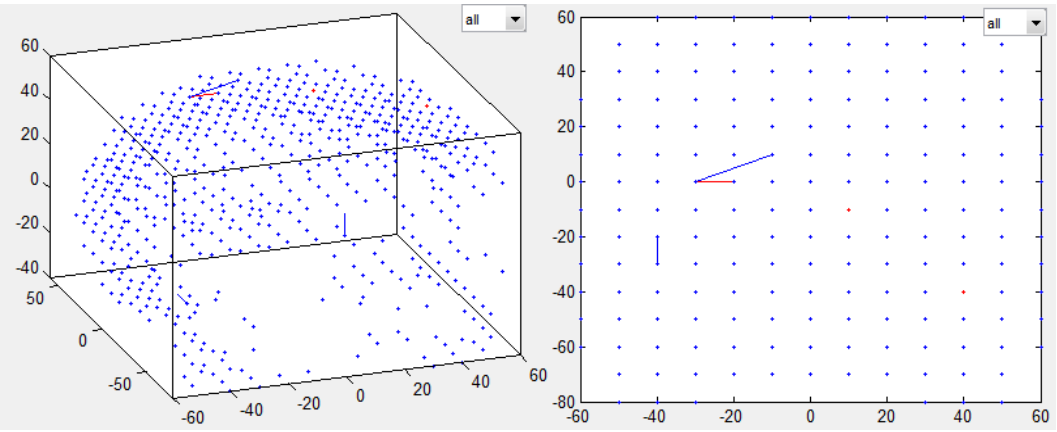

Figure 8-18: Visualization of neurons connectivity for Face stimulus when weight is greater than 0.21 in 3D view (left) and in xy view (right).

8.7.2 NeuCube^{B} Visualization for Spatio-temporal Connections Based on the fMRI Spiking Activity for Scrambled Picture Stimulus

In the initialization stage, NeuCube^{B} connections and their connection weights are initialized randomly (Figure 8-14(a)). For visualization purposes, Figure 8-14(a) shows the connections that were created with connection weight >= 0.09 (i.e. weaker spikes) while Figure 8-14(b) shows neurons connections that have weight >= 0.19 (i.e. stronger spikes). NeuCube^{B} active neurons which weights are more than 0.09 are initialized somewhere in the left and right Superior Frontal Gyrus (SFG), left and right Medial Frontal Gyrus (MFG), left Inferior Frontal Gyrus (IFG), left and right Superior Parietal Lobe (SPL), left Inferior Parietal Lobe (IPL), Superior Temporal Gyrus (SPG), Inferior Temporal Gyrus (ITG) and right and left Occipital Lobe (OL) regions. Higher NeuCube^{B} weights (weight >= 0.19) which indicate stronger spikes are in the left IFG, left and right MFG and left and right ITG. Regions with highly interconnected positive blue lines are areas that were highly activated with the Scrambled Picture stimulus. Figure 8-15 shows these approximated spatial locations of neurons connections in the NeuCube^{B} after initialization (i.e. before the learning process).

After initialization, NeuCube^{B} neuron potential is calculated and updated, whether to spike or not to spike, assuming that each sample of data is taken in different times. Therefore the potential will not continue from one sample to the next sample i.e. NeuCube^{B} neuron potential is reset to 0 for each incoming sample. Each neuron potential is increased if NeuCube^{B} pre-synaptic neuron (i) and its post-synaptic neuron (j) are equal to 1 and NeuCube^{B} refractory neuron (j) is equal to 0. Each neuron's state is also determined, whether or not it has reached the threshold. For neurons that still have not reached the threshold, their neuron potentials are leaked. NeuCube^{B} neuron connections (memory) and weights are also calculated and updated hence resulting in more densely interconnected neurons.

After 5 times of STDP unsupervised learning, more positive spikes were emitted i.e. more connections are created and connection weights are also updated (Figure 8-14(c)). Stronger spikes were also generated as in Figure 8-14(d) as compared to Figure 8-14(b) which was before the learning process. Figure 8-15 and Figure 8-16 show a different view of NeuCube^{B} dynamic connections through a visualization of positive and negative connections before and after the learning process respectively. NeuCube^{B} connections are more concentrated (i.e. more spikes) on the same areas as mentioned

above. However, higher NeuCube[B] weights (weight>= 0.21) which indicate stronger spikes and represented with blue lines, were identified in the left MFG, left IPL and SPL as shown in Figure 8-17. From these connections, it can be suggested that these areas received the most activation (stronger spikes) during the whole learning process. These areas among all are responsive to visual stimulus (left MFG) (Talati & Hirsch, 2005), continuing attention (RIPL and LIPL) (Singh-Curry & Husain, 2009), and audio-visual integration (SPL) (Molholm et al., 2006).

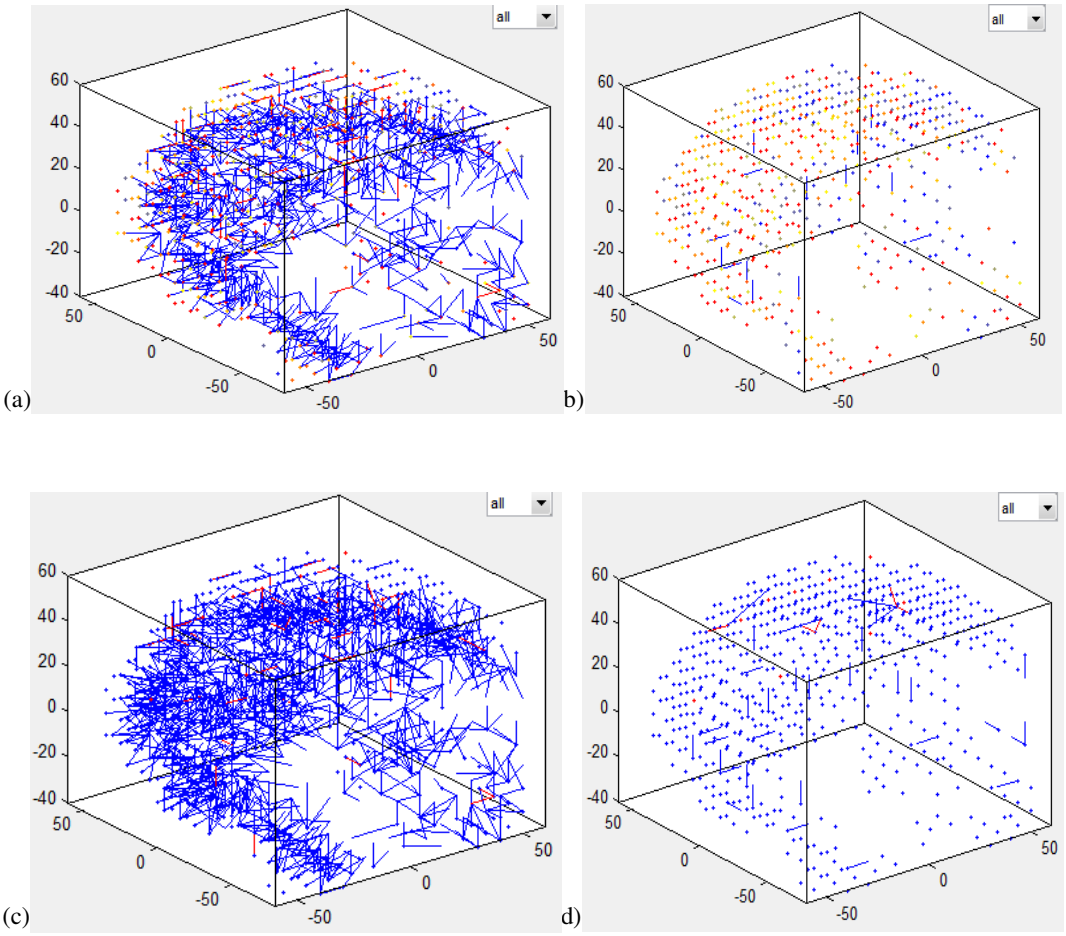

Figure 8-19: Visualization of neurons connectivity for Scrambled Picture stimulus (a) before and (c) after STDP learning with (b) weight >= 0.09 and (d) weight >= 0.19. Positive connections are in blue and negative connections are in red.

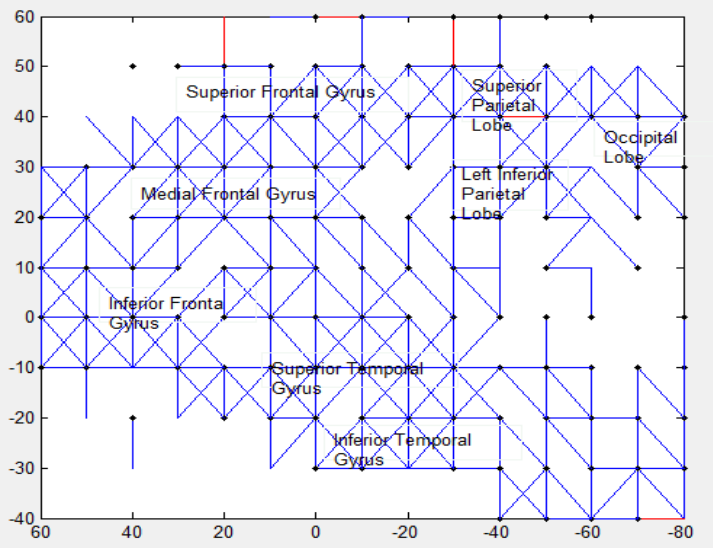

Figure 8-20: Approximated location of densely interconnected neurons before training.

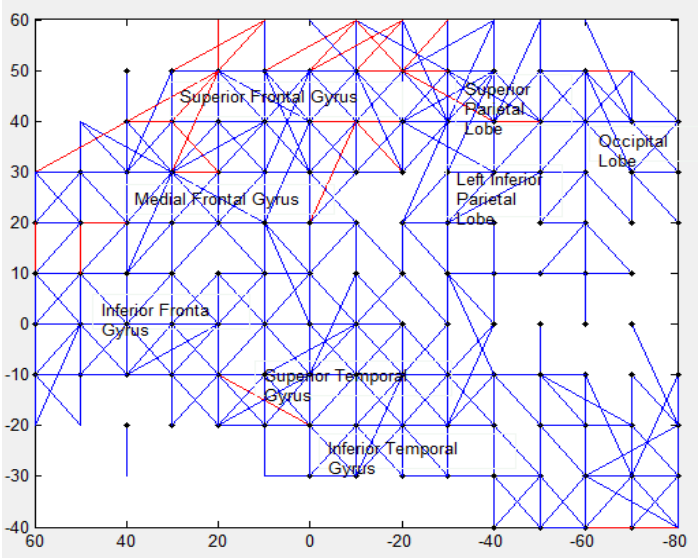

Figure 8-21: Approximated location of densely interconnected neurons after training. More connections are created in areas SFG, MFG, IFG, STG, IFG, SPL, IPL and OL.

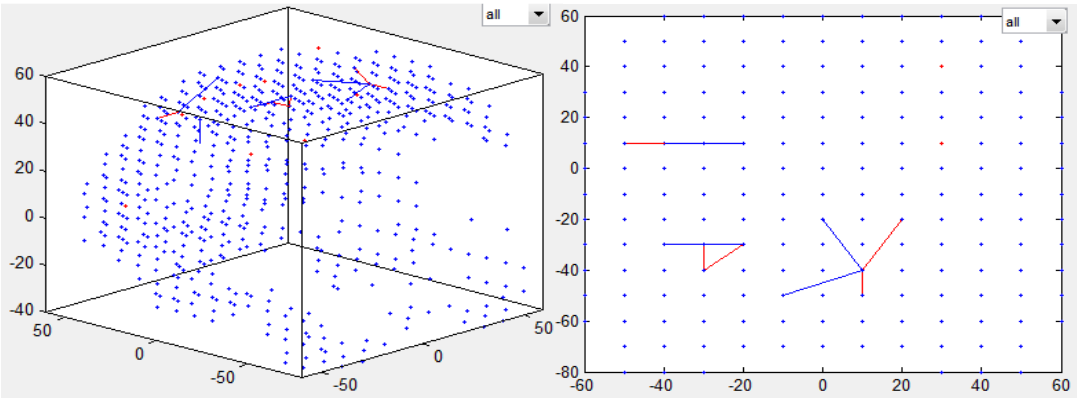

Figure 8-22: Visualization of neurons connectivity for Scrambled Picture stimulus when weight is greater than 0.21 in 3D view (left) and in xy view (right).

In short, the analysis on the neurons connectivity provides a new method in analyzing the Haxby data especially in the extraction of new knowledge. This is achieved by visualizing the neuron connections which is based on the spiking activity of the data. There are certain areas of the NeuCube^{B} that are densely interconnected with blue lines while other areas are with sparsely or no interconnected neurons. Regions in the NeuCube^{B} with more connections indicate that more spike activities had occurred during the learning process in that particular neuron areas compared to the other areas of neurons, which also imply how important these voxels are in determining the areas which receive the most activation when a stimulus is presented to the subject. Thus the areas that received the most activation can be identified from the NeuCube^{B} architecture.

Stronger connections and weaker connections which imply stronger spikes and weaker spikes respectively can be visualized using the neuron connection weight as specified by the user. Again, areas with higher connection weight i.e. more activation can be identified. This visualization is not provided in the previous studies involving the Haxby data.

8.8 CHAPTER SUMMARY

This chapter described the second experiment involving the use of different set of visual stimulus dataset using the NeuCube^{B} model. From the experiments, using our proposed new method, different patterns of brain activation could be classified moderately using the model as compared to the standard machine learning techniques. However, further analysis could be conducted to the Haxby brain data, i.e. in terms of the connectivity of the neurons. Different areas of the brain are activated and are able to be displayed, and

these activations are identified by the densely interconnected neuron connections. This kind of analysis could not be performed in standard machine learning techniques, which is more of black-box in nature. From both experiments, we can also conclude that NeuCube^{B} model offers a better understanding and better analysis of the Haxby data compared to the previous studies involving the same dataset.

In the next chapter, a conclusion of this study is presented. It starts with a general conclusion, research directions and finalizes with some future works that may be carried out.

Chapter 9

CONCLUSION AND FUTURE DIRECTION

This chapter summarizes the experiments that have been conducted to achieve the research objectives and to answer the research questions stated in Chapter 1. Several ideas and recommendations for future work are also discussed in this chapter.

9.1 CONCLUSION

This thesis presents and explains a new NeuCube^{B} computational model which was implemented apart from NeuCube (Kasabov, 2010) framework. Previous implementation of NeuCube has been successfully experimented on other spatio-temporal data namely EEG and stroke data, and this study extends the work by implementing it on a large, complex and noisy spatio-temporal fMRI data. Two different datasets with different formats have been chosen to test the model. Not only are the results generated from the experiments comparatively superior than standard machine learning techniques, but it also allows visualization of model activity and model connectivity that can be used to determine spatio-temporal pattern. In addition, the functional pathways created from the connectivity can also be used to better understand the brain process that happened during the learning process.

Although the contributions are relatively small to the fields of machine learning, it is significant for the community, specifically for the eSNN and brain research teams. These contributions will be explained in the next few paragraphs based on the dataset.

9.2 THESIS CONTRIBUTIONS

Based on the first experiment involving StarPlus data presented in Chapter 7, several contributions could be identified.

(i) A novel StarPlus coordinates mapping into the NeuCube^{B} Talairach coordinate system.

The (x, y, z) coordinates of the spiking neurons in the Cube correspond to the same spatial coordinates of the StarPlus. The main idea of this procedure is to ensure that the newly calculated StarPlus coordinates are within the 1,471 NeuCube^{B} coordinates. Spatial estimation is imposed to each subject due to the different brain size and structure. Coordinates that are within a radius of 7 mm are identified and the corresponding voxels are the input features to the model. These feature voxels are transformed to spike trains and entered into spatially allocated input neurons in the NeuCube^{B} Cube as temporal sequences.

(ii) Visualization of neurons connections.

After completing the unsupervised STDP learning, the spiking activity of the neurons and the updated neural connections can be visualized and examined. In some regions of the brain cube, neuron connections are more densely connected as compared to other areas. This suggests the different but sometimes overlapped areas which are activated to the different stimulus.

Areas labelled CALC, ROPER, LSGA, LFEF and RFEF display higher concentration of connected neurons indicating brain areas are highly activated with visual stimulus regardless of seeing Picture or Sentence. Higher NeuCube^{B} weights are identified in LTRIA, RTRIA, and RSGA areas suggesting these areas are the most activated and stimulated with the Sentence stimulus. These brain areas are known to involve semantic definitions, semantic working memories and language processing tasks. In contrast, higher NeuCube^{B} weights are identified in CALC, LIPL, LT, LSGA, LFEF, RFEF, ROPER and RIPL areas, indicating as the areas that receive the most signals from the Picture stimulus. These brain areas are known to be involved in activities such as visual processing, language processing, continuing attention, visual memories, visual attention, eye movement and language processing.

(iii) Picture to Sentence brain patterns classification.

Classification procedure applied to the spikes patterns determined the class for these patterns. From the experiments, deSNN of NeuCube^{B} model successfully had train and learn the brain patterns and consequently produced superior classification accuracy compared to the standard machine learning approaches. Neu-

NeuCube^{B} has the average of 90% classification accuracy for all 6 subjects while for eSNN, SVM and MLP; it is 70%, 48% and 51% respectively. This indicates that the spatio-temporal characteristic of the brain data has been well taken care by the network of the spiking neurons.

As for the second experiment involving the Haxby dataset presented in Chapter 8, the following contributions are identified:

(i) A novel coordinates mapping of the Haxby fMRI data into the NeuCube^{B} Talairach coordinate system.

The format of this dataset is different from the StarPlus thus posing a new challenge. The main idea of this procedure is to ensure that the newly calculated Haxby dataset coordinates are within the 1,471 eSNN coordinates. Feature voxels are identified based on the voxels that are within the specified radius of the spiking neurons coordinates. These voxels are further selected based on specified voxel threshold value (e.g. threshold = 1,000) and transformed to spike trains and entered into spatially allocated input neurons in the eSNN as temporal sequences.

(ii) Visualization of neurons connections.

Brain regions that are activated by Class 1 (Face) and Class 2 (Scrambled Pictures) stimulus are visualized using the connections that are created before and after the unsupervised STDP learning.

Connections density is identified in similar NeuCube^{B} areas for both classes (SFG, MFG, IFG, SPL, IPL, SPG, ITG and OL). However, higher NeuCube^{B} weight connections are identified differently. For Face stimulus, activation is recognized in the left SFG, left ITG and right MFG while for Scrambled Picture stimulus, brain activation is in the left MFG, left IPL and SPL. Higher NeuCube^{B} weight connections in the stated areas indicate that these areas received the most spikes during the learning process.

(iii) Face against Not Face and Scrambled Pictures against Not Scrambled Pictures brain pattern classifications.

Classification procedure of deSNN is applied to the spikes patterns to determine the class. From the experiments, deSNN of NeuCube^{B} model successfully trained and learned the brain patterns and consequently produced superior classification accuracy compared to the standard machine learning approaches. In deSNN, connection weights are further adjusted dynamically at each time moment of simulation, either by increasing the weight (if there is a spike) or decreasing the weight (if there is no spike) by using the synaptic weight drift parameter. From the classification experiments of Face against Not Face samples and Scrambled Pictures versus Not Scrambled Pictures samples, NeuCube^{B} has an average of 85% of classification accuracy as compared to MLP's 76% and SVM's 66%. Compared to other learning and classification of SNN models, deSNN is computationally inexpensive and boosts the importance of the spike arrival sequence.

Further analysis of the result shows that Class 1 achieved very low accuracy results compared to Class 2. From these results, we can conclude that NeuCube^{B} (as well as MLP and SVM) needs more sample data in order for the network to learn better from the data. Class 1 only has 12 samples (6 samples for training and 6 samples for testing) whereas Class 2 has 96 samples (48 samples for training and 48 samples for testing). As the number of samples is very low for Class 1, the network does not have sufficient sample data (and time) to learn the pattern that may reside within the spatio-temporal fMRI data. In contrast, as more samples are trained for Class 2, the more the network learns from the brain pattern, and thus producing more accurate results. Although the results for Class 1 are very low in all experiments, NeuCube^{B} model still performed better than the other two classifiers.

From the experiments involving the two different fMRI datasets, it can be concluded that NeuCube^{B} could learn the networks of spiking neurons of spatio-temporal fMRI brain data better than eSNN and the conventional classifiers. Although MLP and SVM could process the spatial information quite well, via fairly good results, they fail to recognize and process the temporal information that fMRI data has. The superior results achieved by NeuCube^{B} shows that this additional information (i.e. temporal infor-

mation) has contributed significantly in recognizing the brain patterns. In addition to this, NeuCube^{B} also provides visualization in terms of neurons locations and neurons connectivity that actually illustrate brain areas that receive the most signals i.e. spikes, which again, fails to be recognized by the conventional classifiers.

Neuropsychology is one of the research areas that could largely benefit from this study. This study has only examined brain data for subjects who were looking at pictures i.e. star or plus signs and face or other man-made objects, and thus highly affecting many brain areas especially areas responsive to visual processing and visual memories. However, in reality massive collections of brain data are available to be learned and examined to better understand the structure and functions for each and every part of the brain. With the NeuCube^{B} model, researcher/scientist could easily and accurately detect and identify the brain states, for example, to discriminate a normal brain to a brain with abnormalities from a series of fMRI. In terms of visualization, the model can also show as to which part of the brain is highly activated from the tumour activities. From the experiments that we have conducted, even though involving just a small number of samples, the model have learned and classified the spatio-temporal brain data more accurately compared to the existing eSNN. And this study can be further explored to examine other brain states and functions.

9.3 FUTURE DIRECTION

The proposed NeuCube^{B} framework has successfully tested two different fMRI data formats specifically for solving classification problems. However, several improvements are suggested for better performance particularly in improving the classification accuracy. Below are some recommendations for future research:

9.3.1 Parameters Optimization

As for the current implementation, the user has to manually enter more than five different parameters with a target of achieving the highest classification accuracy. Only a perfect combination could produce good results. The framework has to be tested for each and every possible combination of parameters and the results produced are then compared with the conventional method. Thus it is always a good idea for the framework to automatically determine the combination of parameter values involved so that the overall performance of the framework is further improved. Values for AER encod-

ing threshold and neural threshold are two important parameters that greatly affect the overall classification performance. Furthermore, the threshold for the voxel value also has to be predetermined and decided first. This is to select the most relevant voxels (features) that later are to be used to determine their appropriate encoded AER values.

As explained in Chapter 4, voxels, the main component of fMRI data or the features to be processed by the classifier, are more than ten thousand for a single volume of a single experiment run of a single subject. Thus, in most available fMRI dataset, voxels to be selected and to be processed are huge. One method that can be used is to employ feature optimization in the framework, for instance Quantum-inspired Particle Swarm Optimization (QiPSO), so that relevant features can be selected better. However, a particular issue that may occur is due to the prohibitive processing time.

9.3.2 Automatic voxel mapping

Regardless of the format that defines the spatio-temporal fMRI data, coordinates and voxels of the data could be mapped into the NeuCube^{B} framework automatically and effortlessly. This mapping module should be able to read and process the most widely used fMRI data format, for instance NIfTI-1 and ANALYZE format or at least the module should be able to identify and extract the coordinates and voxel values from the data. The mapping module should also be able to determine and calculate the coordinates of voxels of fMRI data so that the data reside in the NeuCube^{B} cube.

9.3.3 Specialized SNN hardware implementation

One of the current issues in dealing the huge and complex fMRI data (as well as other STBD) is in terms of prohibitive CPU processing power. Current implementation has the NeuCube^{B} built in a single CPU of a personal computer, and sometimes the load is unable to be processed by the computer system, resulting in a suspended system. More CPU processing power that runs parallel, currently is being developed by the research group in KEDRI, is able to accept and process massive data in an eSNN based framework (Kasabov, 2012). NeuCube is currently being implemented on a neuromorphic chip, which is based on Neuro-genetic neuron model that specifically will enable large scale STBD processing and in general will enable any large engineering application to take advantage of the model.

References

Abbott, L. F., & Nelson, S. B. (2000). Synaptic plasticity : taming the beast. *Nature*, *3*(November).

Abdul Hamed, H. N. (2012). *Novel Integrated Methods of Evolving Spiking Neural Network and Particles Swarm Optimazation*. Auckland University of Technology.

Abdul Hamed, H. N., Kasabov, N., Michlovský, Z., & Shamsuddin, S. M. (2009). *String Pattern Recognition Using Evolving Spiking Neural Networks and Quantum Inspired Particle Swarm Optimization* (pp. 611–619). Springer Berlin Heidelberg.

Åberg, M., Löken, L., & Wessberg, J. (2008). An Evolutionary Approach to Multivariate Feature Selection for fMRI Pattern Analysis.

Adrian, E. D. (1926). The impluses produced by sensory nerver endings: Part 1. *The Journal of Physiology*, *61*(1), 49–72.

Arbib, M. A. (1995). *The Handbook of Brain Theory and Neural Networks* (p. 1309). The MIT Press.

Avesani, P., Hazan, H., Koilis, E., Manevitz, L., & Sona, D. (2011). Learning BOLD Response in fMRI by Reservoir Computing. *2011 International Workshop on Pattern Recognition in NeuroImaging*, 57–60. doi:10.1109/PRNI.2011.16

Bell, C. C., Han, V. Z., Sugawara, Y., & Grant, K. (1997). Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, *387*, 278–281. doi:10.1038/387278a0

Bezanilla, F., & Armstrong, C. L. (1977). Inactivation of the Sodium Channel I: Sodium Current Experiments. *The Journal of General Physiology*, *70*(5), 549–566.

Bi, G., & Poo, M. (1998). Synaptic Modifications in Cultured Hippocampal Neurons : Dependence on Spike Timing , Synaptic Strength , and Postsynaptic Cell Type. *The Journal of Neuroscience*, *18*(24), 10464–10472.

Bohte, S. M. (2004). The evidence for neural information processing with precise spike-times: A survey. *Natural Computing*, *3*(2), 195–206. doi:10.1023/B:NACO.0000027755.02868.60

Bohte, S. M., Kok, J. N., & Poutrã, H. La. (2002). Error-backpropagation in temporally encoded networks of spiking neurons, *48*, 17–37.

Bohte, S. M., Kok, J. N., & Poutre, H. La. (2000). SpikeProp : Backpropagation for Networks of Spiking Neurons Error-Backpropagation in a Network of Spik- ing Neurons. *ESANN*, 419–424.

Bohte, S. M., La Poutre, H., & Kok, J. N. (2002). Unsupervised Clustering With Spiking Neurons By Sparse Temporal Coding and Multilayer RBF Networks. *IEEE Transactions on Neural Networks*, *13*(2), 426–435.

Braitenberg, V., & Schüz, A. (1998). *Cortex: statistics and geometry of neuronal connectivity.*

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., … Destexhe, A. (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of Computational Neuroscience*, *23*(3), 349–98. doi:10.1007/s10827-007-0038-6

Brodersen, K. H., Wiech, K., Lomakina, E. I., Lin, C. S., Buhmann, J. M., Bingel, U., … Tracey, I. (2012). Decoding the perception of pain from fMRI using multivariate pattern analysis. *NeuroImage*, *63*(3), 1162–70. doi:10.1016/j.neuroimage.2012.08.035

Bullmore, E., & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, *10*(3), 186–198.

Buxton, R. B., Uludağ, K., Dubowitz, D. J., & Liu, T. T. (2004). Modeling the hemodynamic response to brain activation. *NeuroImage*, *23 Suppl 1*, S220–33. doi:10.1016/j.neuroimage.2004.07.013

Carnevale, N. T., & Hines, M. L. (2006). *The NEURON Book*. *Neuron* (Vol. 30, p. 457). doi:10.1017/CBO9780511541612

Carroll, M. K., Cecchi, G. a, Rish, I., Garg, R., & Rao, a R. (2009). Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, *44*(1), 112–22. doi:10.1016/j.neuroimage.2008.08.020

Cawley, S., Morgan, F., McGinley, B., Pande, S., McDaid, L., Carrillo, S., & Harkin, J. (2011). Hardware spiking neural network prototyping and application. *Genetic Programming and Evolvable Machines*, *12*, 257–280. doi:10.1007/s10710-011-9130-9

Chen, Z. J., He, Y., Rosa-Neto, P., Germann, J., & Evans, A. C. (2008). Revealing modular architecture of human brain structural networks by using cortical thickness from MRI. *Cerebral Cortex (New York, N.Y. : 1991)*, *18*, 2374–81. doi:10.1093/cercor/bhn003

Cox, D. D., & Savoy, R. L. (2003). Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, *19*(2), 261–270. doi:10.1016/S1053-8119(03)00049-1

Cui, X., Bray, S., Bryant, D. M., Glover, G. H. A., & Reiss, A. L. (2012). A quantitative comparison of NIRS and fMRI across multiple cognitive tasks, *54*(4), 2808–2821. doi:10.1016/j.neuroimage.2010.10.069.A

Delorme, A., & Thorpe, S. J. (2003). SpikeNET : an event-driven simulation package for modelling large networks of spiking neurons. *Network: Computation in Neural Systems*, *14*, 613–627.

Dhoble, K., Nuntalid, N., Indiveri, G., & Kasabov, N. (2012). Online spatio-temporal pattern recognition with evolving spiking neural networks utilising address event representation, rank order, and temporal spike learning. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 1–7. doi:10.1109/IJCNN.2012.6252439

Diesmann, M., & Gewaltig, M. O. (2002). NEST: An environment for neural systems simulations. In *In T. Plesser and V. Macho (Eds.), Forschung und wisschenschaftliches Rechnen, Beitrage zum Heinz-Billing-Preis 2001, Volume 58 of GWDG-Bericht* (pp. 43–70). Retrieved from http://www.scholarpedia.org/article/NEST_(NEural_Simulation_Tool)

Doborjeh, M. G., Capecci, E., & Kasabov, N. (2014). Temporal Brain Data with a NeuCube Evolving Spiking Neural Network Model. In *IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)* (pp. 73–80).

Doborjeh, M. G., Wang, G., Kasabov, N., Kydd, R., & Russell, B. R. (2015). A Spiking Neural Network Methodology and System for Learning and Comparative Analysis of EEG Data from Healthy versus Addiction Treated versus Addiction Not Treated Subjects. *IEEE Transactions on Biomedical Engineering*, *9294*(c), 1–1. doi:10.1109/TBME.2015.2503400

Evans, A. C., Collins, D. L., Mills, S. R., Brown, E. D., Kelly, R. L., & Peters, T. M. (1993). 3D statistical neuroanatomical models from 305 MRI volumes. In *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference* (pp. 1813–1817). doi:10.1109/NSSMIC.1993.373602

Fan, Y., Shen, D., & Davatzikos, C. (2006). Detecting Cognitive States from fMRI Images by Machine Learning and Multivariate Classification. In *Conference on Computer Vision and Pattern Recognition Workshop 2006* (pp. 89–89). IEEE. doi:10.1109/CVPRW.2006.64

Finelli, L. A, Haney, S., Bazhenov, M., Stopfer, M., & Sejnowski, T. J. (2008). Synaptic learning rules and sparse coding in a model sensory system. *PLoS Computational Biology*, *4*(4), e1000062. doi:10.1371/journal.pcbi.1000062

Foundas, A. L., Eure, K. F., Luevano, L. F., & Weinberger, D. R. (1998). MRI asymmetries of Broca's area: the pars triangularis and pars opercularis. *Brain and Language*, *64*, 282–296. doi:10.1006/brln.1998.1974

Friston, K. J., Frith, C. D., Frackowiak, R. S., & Turner, R. (1995, June). Characterizing dynamic brain responses with fMRI: a multivariate approach. *NeuroImage*. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9343599

Friston, K. J., Holmes, A.P., Worsley, K.J., Poline, J.P., Frith, C.D., and Frackowiak, R. S. J. (1995). Statistical Parametric Maps in Functional Imaging : A General Linear Approach.

FSLView. (2012). Retrieved from http://fsl.fmrib.ox.ac.uk/fsl/fslview/

Furber, S. (2012). To build a brain. *IEEE Spectrum*, *49*(8), 44–49.

Fusi, S., Annunziato, M., Badoni, D., Salamon, A., & Amit, D. J. (1999). Spike-driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural Computation*, *12*(10), 2227–2258. doi:10.1162/089976600300014917

Gabrieli, J. D., Poldrack, R. A, & Desmond, J. E. (1998). The role of left prefrontal cortex in language and memory. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(February), 906–913. doi:10.1073/pnas.95.3.906

Gazzaniga, M. S., Ivry, R. B., Mangun, G. R., & Steven, M. S. (2009). *Cognitive Neuroscience: The Biology of the Mind* (3rd. ed., pp. 359–401). New York: W.W. Norton.

Georgieva, P., & Torre, F. D. (2013). Robust Principal Component Analysis for Brain Imaging. In *Artificial Neural Networks and Machine Learning – ICANN 2013* (pp. 288–295). Springer Berlin Heidelberg. doi:10.1007/978-3-642-40728-4_36

Gerstner, W. (1995). Time structure of the activity in neural network models. *Physical Review E*, *51*(1), 738–758. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9962697

Gerstner, W. (1998). Spiking Neurons. In W. Maass & C. M. Bishop (Eds.), *Pulsed Neural Networks* (pp. 3–54). MIT Press.

Gerstner, W. (2001). What's different with spiking neurons. *Plausible Neural Networks for Biological Modeling*, (1), 23–48. doi:10.1.1.23.2539

Gerstner, W., & Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons , Populations , Plasticity*. Cambridge University Press.

Gerstner, W., Kreiter, A. K., Markram, H., & Herz, A. V. M. (1997). Neural codes: firing rates and beyond. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(24), 12740–12741. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=34168&tool=pmcentrez&rendertype=abstract

Gerstner, W., Sprekeler, H., & Deco, G. (2012). Theory and simulation in neuroscience. *Science (New York, N.Y.)*, *338*(6103), 60–5. doi:10.1126/science.1227356

Gerstner, W., & van Hemmen, J. L. (1992). Associative Memory in a Network of "Spiking" Neurons. *Network: Computation in Neural Systems*, *3*(2), 139–164.

Ghosh-Dastidar, S., & Adeli, H. (2007). Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering*, *14*(3), 187–212.

Ghosh-Dastidar, S., & Adeli, H. (2009). A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure

detection. *Neural Networks : The Official Journal of the International Neural Network Society*, *22*(10), 1419–31. doi:10.1016/j.neunet.2009.04.003

Gleeson, P., Steuber, V., & Silver, R. A. (2007). neuroConstruct: a tool for modeling networks of neurons in 3D space. *Neuron*, *54*(2), 219–35. doi:10.1016/j.neuron.2007.03.025

Goldberg, I. I., Harel, M., & Malach, R. (2006). When the Brain Loses Its Self: Prefrontal Inactivation during Sensorimotor Processing. *Neuron*, *50*, 329–339. doi:10.1016/j.neuron.2006.03.015

Goodman, D. F. M., & Brette, R. (2009). The brian simulator. *Frontiers in Neuroscience*. doi:10.3389/neuro.01.026.2009

Goodman, E., & Ventura, D. (2006). Spatiotemporal Pattern Recognition via Liquid State Machines. In *IJCNN* (pp. 3848–3853).

Grzyb, B. J., Chinellato, E., Wojcik, G. M., & Kaminski, W. A. (2009). Which Model to Use for the Liquid State Machine ? In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on* (pp. 1018–1024). IEEE.

Hanson, S. J., Matsuka, T., & Haxby, J. V. (2004). Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a "face" area? *NeuroImage*, *23*(1), 156–66. doi:10.1016/j.neuroimage.2004.05.020

Haxby, J. V, Gobbini, M. I., Furey, M. L., Ishai, a, Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science (New York, N.Y.)*, *293*(5539), 2425–30. doi:10.1126/science.1063736

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2011). Visual object recognition. Retrieved January 1, 2014, from https://openfmri.org/dataset/ds000105

Haynes, J. D., & Rees, G. (2005). Predicting the orientation of invisible stimuli from activity in human primary visual cortex. *Nature Neuroscience*, *8*(5), 686–91. doi:10.1038/nn1445

Hebb, D. O. (1949). *The organization of behavior: A Neuropsychological Theory*.

Hemming, C. (2003). Using Neural Networks in Linguistic Resources. Department of Languages, University College of Skövde, Swedish National Graduate School of Language Technology.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve, *117*, 500–544.

Honey, C. J., Kotter, R., Breakspear, M., & Sporns, O. (2007). Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *PNAS*, *104*(24), 10240–10245. doi:10.1073/pnas.0701519104

Hopfield, J. J., & Brody, C. D. (2000). What is a moment ? “‘ Cortical ’” sensory integration over a brief interval. In *Proceedings of the National Academy of Sciences, 97(25)* (Vol. 97, pp. 13919–13924).

Huguenard, J. R. (2000). Reliability of axonal propagation : The spike doesn’t stop here. *Proceedings of the National Academy of Sciences*, *97*(17), 9349–9350.

Hutchinson, R. A., Niculescu, R. S., Keller, T. A., Rustandi, I., & Mitchell, T. M. (2009). Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using Hidden Process Models. *NeuroImage*, *46*(1), 87–104. doi:10.1016/j.neuroimage.2009.01.025

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., … Boahen, K. (2011). Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, *5*(May), 73. doi:10.3389/fnins.2011.00073

Ishai, A., Ungerleider, L. G., Martin, A., & Haxby, J. V. (2000). The representation of objects in the human occipital and temporal cortex. *Journal of Cognitive Neuroscience*, *12 Suppl 2*, 35–51. doi:10.1162/089892900564055

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, *14*(6), 1569–72. doi:10.1109/TNN.2003.820440

Izhikevich, E. M. (2004). Which Model to Use for Cortical Spiking Neurons ? *IEEE Transactions on Neural Networks*, *15*(5), 1063–1070.

Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Computation*, *18*, 245–282. doi:10.1162/089976606775093882

Izhikevich, E. M. (2007). *Dynamical Systems in Neuroscience*. MIT Press.

Izhikevich, E. M., & Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems, *2007*.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science (New York, N.Y.)*, *304*(5667), 78–80. doi:10.1126/science.1091277

Joshi, P., & Maass, W. (2004). Movement Generation with Circuits of Spiking. *Neural Computation*, *17*(8), 1715–1738.

Ju, H., Xu, J., & VanDongen, A. M. J. (2010). Classification of Musical Styles Using Liquid State Machines. *Neural Networks (IJCNN), The 2010 International Joint Conference*, 1–7.

Kamitani, Y., & Tong, F. (2005). Decoding the visual and subjective contents of the human brain. *Nature Neuroscience*, *8*(5), 679–685.

Kandel, E. R., Schwartz, J. H., & Jessel, T. M. (2000). *Principles of Neural Science* (4th ed.). New York: McGraw-Hill.

Kasabov, N. (1998). ECOS : Evolving Connectionist Systems and the ECO learning paradigm. In S. Usui & T. Omori (Ed.), *The Fifth International Conference on Neural Information Processing, ICONIP'98* (pp. 1232– 1235). Kitakyushu, Japan: IOA Press.

Kasabov, N. (2007). *Evolving Connectionist Systems : The Knowledge Engineering Approach*. London: Springer-Verlag.

Kasabov, N. (2010). To spike or not to spike: a probabilistic spiking neuron model. *Neural Networks*, *23*(1), 16–19. doi:10.1016/j.neunet.2009.08.010

Kasabov, N. (2012). NeuCube EvoSpike Architecture for Spatio-temporal Modelling and Pattern Recognition of Brain Signals, 225–243.

Kasabov, N. (2013). Evolving spiking neural networks for spatio- and spectro-temporal pattern recognition. *Neural Networks : The Official Journal of the International Neural Network Society*, *41*, 188–201. doi:10.1016/j.neunet.2012.11.014

Kasabov, N. (2014). NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, *52*, 62–76. doi:10.1016/j.neunet.2014.01.006

Kasabov, N., Dhoble, K., Nuntalid, N., & Indiveri, G. (2013). Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Networks : The Official Journal of the International Neural Network Society*, *41*(1995), 188–201. doi:10.1016/j.neunet.2012.11.014

Kasabov, N., Feigin, V., Hou, Z. G., Chen, Y., Liang, L., Krishnamurthi, R., … Parmar, P. (2014). Evolving Spiking Neural Networks for Personalised Modelling, Classification and Prediction of Spatio-Temporal Patterns with a Case Study on Stroke. *Neurocomputing*, *134*, 269–279. doi:10.1016/j.neucom.2013.09.049

Kasinski, A., & Ponulak, F. (2006). Comparison of Supervised Learning Methods for Spike Time. *International Journal of Applied Mathematics and Computer Science*, *16*(1), 101–113.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942–1948. doi:10.1109/ICNN.1995.488968

Koch, C., & Reid, R. C. (2012). Observatories of the mind. *Nature*, *483*(22 March 2012), 397–398. doi:doi:10.1038/483397a

Kojima, H., & Katsumata, S. (2008). An Analysis of Synaptic Transmission and its Plasticity by Glutamate Receptor Channel Kinetics Models and 2-Photon Laser Photolysis. In *Advances in Neuro-Information Processing* (pp. 88–94). Springer Berlin Heidelberg.

Kolb, B., & Whishaw, I. Q. (1996). *Fundamentals of Human Neuropsychology* (4th ed., pp. 370–390). W. H. Freeman.

Ku, S., Gretton, A., Macke, J., & Logothetis, N. K. (2008). Comparison of pattern recognition methods in classifying high-resolution BOLD signals obtained at high magnetic field in monkeys. *Magnetic Resonance Imaging*, *26*(7), 1007–14. doi:10.1016/j.mri.2008.02.016

LaConte, S., Strother, S., Cherkassky, V., Anderson, J., & Hu, X. (2005). Support vector machines for temporal classification of block design fMRI data. *NeuroImage*, *26*(2), 317–29. doi:10.1016/j.neuroimage.2005.01.048

Lancaster, J. L., Woldorff, M. G., Parsons, L. M., Liotti, M., Freitas, C. S., Rainey, L., … Fox, P. T. (2000). Automated Talairach atlas labels for functional brain mapping. *Human Brain Mapping*, *10*(3), 120–31. doi:10.1002/1097-0193(200007)10:3<120::AID-HBM30>3.0.CO;2-8

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Physiol Pathol Gen*, *9*(1), 620–635. doi:10.1007/s00422-007-0189-6

Lauger, P. (1995). Conformational transition of ion channels. In B. Sakmann & E. Neher (Eds.), *Single-Channel Recording* (2nd ed., pp. 651–662). New York: Springer.

Lemoine, B., & Maida, A. S. (2013). GPU Facilitated Unsupervised Visual Feature Acquisition in Spiking Neural Networks. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–6.

Lestienne, R. (1996). Determination of the precision of spike timing in the visual cortex of anaesthetised cats. *Biological Cybernetics*, *74*, 55–61. doi:10.1007/BF00199137

Li, E. Y. (1994). Artificial neural networks and their business applications, *27*, 303–313.

Lichtsteiner, P., & Delbruck, T. (2005). A 64x64 aer logarithmic temporal derivative silicon retina. *Research in Microelectronics and Electronics, 2005 PhD*, *2*(1), 202–205. doi:10.1109/RME.2005.1542972

Lindquist, M. A. (2008). The Statistical Analysis of fMRI Data. *Statistical Science*, *23*(4), 439–464. doi:10.1214/09-STS282

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671. doi:10.1016/S0893-6080(97)00011-7

Maass, W. (1998). On the role of time and space in neural computation. *Mathematical Foundations of Computer Science 1998*, 72–83.

Maass, W. (2010). Chapter 1 Liquid State Machines : Motivation , Theory , and Applications.

Maass, W. & Bishop, C. M. (1999). *Pulsed Neural Networks*. *Science* (Vol. 275, pp. 1–29). doi:10.1.1.36.4685

Maass, W., Natschlager, T., & Markram, H. (2002). Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, *14*(11), 2531–2560.

Mainen, Z. F., & Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science*, *268*(5216), 1503–1506.

Markram, H. (2006). The Blue Brain Project, *7*(February), 153–161.

Markram, H., Lubke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science*, *275*(January), 213–215.

Martinez, D., & Hugues, E. (2004). A spiking neural network model of the locust antennal lobe : towards neuromorphic electronic noses inspired from insect olfaction. *Electronic Noses & Sensors for the Detection of Explosives*, 209–234.

Masquelier, T., & Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, *3*(2), e31. doi:10.1371/journal.pcbi.0030031

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

McGinley, B., O'Halloran, M., Conceicao, R. C., Morgan, F., Glavin, M., & Jones, E. (2010). Spiking neural networks for breast cancer classification using radar target signatures. *Progress In Electromagnetics Research C*, *17*, 79–94.

Meftah, B., Benyettou, A., Lezoray, O., & QingXiang, W. (2008). Image clustering with spiking neuron network. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 681–685. doi:10.1109/IJCNN.2008.4633868

Meftah, B., Lezoray, O., & Chaturvedi, S. (2013). Image Processing with Spiking Neuron Networks. *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, 525–544.

Meunier, C., & Segev, I. (2002). Playing the Devil's advocate : is the Hodgkin – Huxley model useful? *Trends in Neurosciences*, *25*(11), 558–563.

Miller, E. K., Freedman, D. J., & Wallis, J. D. (2002). The prefrontal cortex: categories, concepts and cognition. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *357*(July), 1123–1136. doi:10.1098/rstb.2002.1099

Misaki, M., Kim, Y., Bandettini, P. A., & Kriegeskorte, N. (2010). Comparions of multivariate classifiers and response normalizations for pattern-information fMRI. *NeuroImage*, *53*(1), 103–118. doi:10.1016/j.neuroimage.2010.05.051.Comparison

Mitchell, T. M. (2005). StarPlus fMRI Data. Retrieved from http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/

Mitchell, T. M., Hutchinson, R., Just, M., Niculescu, R. S., Pereira, F., & Wang, X. (2003). Classifying instantaneous cognitive states from FMRI data. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, 465–9. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1479944&tool=pmcentrez&rendertype=abstract

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., & Newman, S. (2004). Learning to Decode Cognitive States from Brain Images. *Machine Learning*, *57*(1/2), 145–175. doi:10.1023/B:MACH.0000035475.85309.1b

Mitra, S., Fusi, S., & Indiveri, G. (2009). Real-Time Classification of Complex Patterns Using Spike-Based Learning in Neuromorphic VLSI. *IEEE Transactions on Biomedical Circuits and Systems*, *3*(1), 32–42. doi:10.1109/TBCAS.2008.2005781

Molholm, S., Sehatpour, P., Mehta, A. D., Shpaner, M., Gomez-Ramirez, M., Ortigue, S., … Foxe, J. J. (2006). Audio-visual multisensory integration in superior parietal lobule revealed by human intracranial recordings. *Journal of Neurophysiology*, *96*(May 2006), 721–729. doi:10.1152/jn.00285.2006

Morch, N., Hansen, L. K., Strother, S. C., Svarer, C., Rottenberg, D. A., Lautrup, B., … Paulson, O. B. (1997). Nonlinear versus Linear Models in Functional Neuroimaging : Learning Curves and Generalization Crossover. *Information Processing in Medical Imaging*, 259–270.

Mourão-Miranda, J., Bokde, A. L. W., Born, C., Hampel, H., & Stetter, M. (2005). Classifying brain states and determining the discriminating activation patterns: Support Vector Machine on functional MRI data. *NeuroImage*, *28*(4), 980–95. doi:10.1016/j.neuroimage.2005.06.070

Mourão-Miranda, J., Friston, K. J., & Brammer, M. (2007). Dynamic discrimination analysis: a spatial-temporal SVM. *NeuroImage*, *36*(1), 88–99. doi:10.1016/j.neuroimage.2007.02.020

Nakagawa, T. T., Woolrich, M., Luckhoo, H., Joensson, M., Mohseni, H., Kringelbach, M. L., … Deco, G. (2014). How delays matter in an oscillatory whole-brain spiking-neuron network model for MEG alpha-rhythms at rest. *NeuroImage*, *87*, 383–94. doi:10.1016/j.neuroimage.2013.11.009

Nelson, M., & Rinzel, J. (1995). The Hodgkin-Huxley Model. In J. M. Bower & Beeman (Eds.), *The book of Genesis* (pp. 27–51). New York: Springer.

Ng, B., & Abugharbieh, R. (2011a). Generalized group sparse classifiers with application in fMRI brain decoding. *Cvpr 2011*, 1065–1071. doi:10.1109/CVPR.2011.5995651

Ng, B., & Abugharbieh, R. (2011b). Modeling Spatiotemporal Structure in fMRI Brain Decoding Using Generalized Sparse Classifiers. *2011 International Workshop on Pattern Recognition in NeuroImaging*, 65–68. doi:10.1109/PRNI.2011.10

Ng, B., Vahdat, A., Hamarneh, G., & Abugharbieh, R. (2010). Generalized Sparse Classifiers for Decoding Cognitive States in fMRI. In *Machine Learning in Medical Imaging* (pp. 108–115). Springer.

Niiniskorpi, T., Bj, M., & Wessberg, J. (2009). Particle Swarm Feature Selection for fMRI Pattern Classification. In *BIOSIGNALS* (pp. 279–284).

Nuntalid, N., Dhoble, K., & Kasabov, N. (2011). EEG Classification with BSA Spike Encoding Algorithm and Evolving Probabilistic Spiking Neural Network. *Neural Information Processing*, 451–460.

Ogawa, S., Lee, T. M., Kay, A. R., & Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, *87*(24), 9868–9872.

Optican, L. M., & Richmond, B. J. (1987). Temporal encoding of two-dimensional patterns by single units in promate inferior temporal cortex. III. Information theoretic analysis.pdf. *Journal of Neurophysiology*, *57*(1), 162–177.

Othman, M., Kasabov, N., Tu, E., Feigin, V., Krishnamurthi, R., Hou, Z., … Hu, J. (2014). Improved predictive personalized modelling with the use of Spiking Neural Network system and a case study on stroke occurrences data. *2014 International Joint Conference on Neural Networks (IJCNN)*, 3197–3204. doi:10.1109/IJCNN.2014.6889709

Pearson, M. J., Pipe, A. G., Mitchinson, B., Gurney, K., Melhuish, C., Gilhespy, I., & Nibouche, M. (2007). Implementing Spiking Neural Networks for Real-Time Signal-Processing and Control Applications : A Model-Validated FPGA Approach. *IEEE Transactions on Neural Networks*, *18*(5), 1472–1487.

Perrinet, L., & Samuelides, M. (2002). Sparse Image Coding Using an Asynchronous Spiking Neural Network. *ESANN*, 313–318.

Perrinet, L., Samuelides, M., & Thorpe, S. (2004). Sparse spike coding in an asynchronous feed-forward multi-layer neural network using matching pursuit. *Neurocomputing*, *57*, 125–134. doi:10.1016/j.neucom.2004.01.010

Poline, J. B., & Poldrack, R. A. (2012). Frontiers in brain imaging methods grand challenge. *Frontiers in Neuroscience*, *6*(July), 96. doi:10.3389/fnins.2012.00096

Polyn, S. M., Detre, G. J., Takerkart, S., Natu, V. S., Benharrosh, M. S., Singer, B. D., … Norman, K. A. (2005). A Matlab-based toolbox to facilitate multi-voxel pattern classification of fMRI data ., 2005.

Ponulak, F., & Kasinski, A. (2010). Supervised Learning in Spiking Neural Networks with ReSuMe : Sequence Learning , Classification , and Spike Shifting. *Neural Computation*, *510*, 467–510.

Ponulak, F., & Kasiński, A. (2010). Introduction to spiking neural networks : Information processing , learning and applications. *Acta Neurobiologiae Experimentalis*, *71*(4), 409–433.

Querlioz, D., Bichler, O., & Gamrat, C. (2011). Simulation of a Memristor-Based Spiking Neural Network Immune to Device Variations. In *The 2011 International Joint Conference on Neural Networks (IJCNN)* (pp. 1775–1781).

Reid, D., Hussain, A. J., & Tawfik, H. (2014). Financial time series prediction using spiking neural networks. *PloS One*, *9*(8), e103656. doi:10.1371/journal.pone.0103656

Rocke, P., McGinley, B., Morgan, F., & Maher, J. (2007). Reconfigurable hardware evolution platform for a spiking neural network robotics controller. *Reconfigurable Computing: Architectures, Tools and Applications*, 373–378. doi:10.1007/978-3-540-71431-6_36

Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. CORNELL AERONAUTICAL LAB INC BUFFALO NY.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Rustandi, I. (2007a). Classifying Multiple-Subject fMRI Data Using the Hierarchical Gaussian Naïve Bayes Classifier. In *13th Conference on Human Brain Mapping* (pp. 4–5).

Rustandi, I. (2007b). Hierarchical Gaussian Na¨ıve Bayes Classifier for Multiple-Subject fMRI Data. *Submitted to AISTATS*, (1), 2–4.

Rustandi, I., Mitchell, T. M., & Xing, E. (2010). Predictive fMRI Analysis for Multiple Subjects and Multiple Studies (Thesis) Thesis Committee :

Ryali, S., Supekar, K., Abrams, D. A, & Menon, V. (2010). Sparse logistic regression for whole-brain classification of fMRI data. *NeuroImage*, *51*(2), 752–64. doi:10.1016/j.neuroimage.2010.02.040

Schall, J. D. (2004). On the role of frontal eye field in guiding attention and saccades. *Vision Research*, *44*, 1453–1467. doi:10.1016/j.visres.2003.10.025

Schliebs, S., Defoin-Platel, M., & Kasabov, N. (2009). Integrated feature and parameter optimization for an evolving spiking neural network. *Advances in Neuro-Information*, 1229–1236. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-02490-0_149

Schliebs, S., Defoin-Platel, M., Worner, S., & Kasabov, N. (2009). Integrated feature and parameter optimization for an evolving spiking neural network: exploring heterogeneous probabilistic models. *Neural Networks : The Official Journal of the International Neural Network Society*, *22*(5-6), 623–32. doi:10.1016/j.neunet.2009.06.038

Schliebs, S., Nuntalid, N., & Kasabov, N. (2010). Towards Spatio-Temporal Pattern Recognition Using Evolving Spiking Neural Networks. In *Neural Information Processing. Theory and Algorithms* (Vol. 6443, pp. 163–170). doi:10.1007/978-3-642-17537-4_21

Schmah, T., Zemel, R. S., Hinton, G. E., Small, S. L., & Strother, S. C. (2010). Comparing Classification Methods for Longitudinal fMRI Studies. *Neural Computation*, *22*(11), 2729–2762. doi:10.1162/NECO_a_00024

Seguier, R., & Mercier, D. (2002). Audio-visual speech recognition, one pass learning with spiking neurons. In *Artificial Neural Networks - Icann 2002* (Vol. 2415, pp. 1207–1212). Retrieved from <Go to ISI>://WOS:000181441900195

Sharma, V., Rai, S., & Dev, A. (2012). A Comprehensive Study of Artificial Neural Networks, *2*(10), 278–284.

Sharma, V., & Srinivasan, D. (2010). A Spiking Neural Network Based on Temporal Encoding for Electricity Price Time Series Forecasting in Deregulated Markets. *International Joint Conference on Neural Networks*, 1–8. doi:10.1109/ijcnn.2010.5596676

Shinkareva, S. V, Mason, R. A, Malave, V. L., Wang, W., Mitchell, T. M., & Just, M. A. (2008). Using FMRI brain activation to identify cognitive states associated with perception of tools and dwellings. *PloS One*, *3*(1), e1394. doi:10.1371/journal.pone.0001394

Silva, R. G. (2009). Condition monitoring of the cutting process using a self-organizing spiking neural network map. *Journal of Intelligent Manufacturing*, *21*(6), 823–829. doi:10.1007/s10845-009-0258-x

Silva, R. G., & Wilcox, S. (2007). Multi-sensor condition monitoring using spiking neuron networks. In *IADIS International Conference Applied Computing 2007*.

Singh-Curry, V., & Husain, M. (2009). The functional role of the inferior parietal lobe in the dorsal and ventral stream dichotomy. *Neuropsychologia*, *47*(6), 1434–1448. doi:10.1016/j.neuropsychologia.2008.11.033

Smith, S. M. (2004). Overview of fMRI analysis. *British Journal of Radiology*, *77*(suppl_2), S167–S175. doi:10.1259/bjr/33553595

Srinivasa, N., & Cho, Y. (2014). Unsupervised discrimination of patterns in spiking neural networks with excitatory and inhibitory synaptic plasticity. *Frontiers in Computational Neuroscience*, *8*(December), 1–23. doi:10.3389/fncom.2014.00159

Stam, C. J. (2004). Functional connectivity patterns of human magnetoencephalographic recordings: A “small-world” network? *Neuroscience Letters*, *355*, 25–28. doi:10.1016/j.neulet.2003.10.063

Strassberg, A. F., & Defelice, L. J. (1993). Limitations of the Hodgkin-Huxley Formalism: Effects of Single Channel Kinetics on Transmembrane Voltage Dynamics. *Neural Computation*, *5*(6), 843–855.

Talairach, J., & Tournoux, P. (1988). *Co-planar stereotaxic atlas of the human brain: 3-dimensional proportional system: an approach to cerebral imaging*. *Neuropsychologia* (Vol. 39, p. 145). doi:10.1111/mono.12083

Talati, A., & Hirsch, J. (2005). Functional specialization within the medial frontal gyrus for perceptual go/no-go decisions based on “what,” “when,” and “where” related information: an fMRI study. *Journal of Cognitive Neuroscience*, *17*, 981–993. doi:10.1162/0898929054475226

Tao, X., & Michel, H. E. (2005). Novel artificial neural networks for remote-sensing data classification. In *Defense and Security* (pp. 127–138). International Society for Optics and Photonics.

Thorpe, S., Delorme, A, & Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks : The Official Journal of the International Neural Network Society*, *14*(6-7), 715–25. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11665765

Thorpe, S., Delorme, A., & Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*, *14*, 715–725.

Thorpe, S., Fize, D., & Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, *381*(6582), 520–522.

Thorpe, S., & Gautrais, J. (1998). Rank Order Coding. *Computational Neuroscience: Trends in Research*, 113–119.

Toga, A. W., Thompson, P. M., Mori, S., Amunts, K., & Zilles, K. (2006). Towards multimodal atlases of the human brain. *Nature Reviews. Neuroscience*, *7*(12), 952–66. doi:10.1038/nrn2012

Trhan, P. (2010). The application of spiking neural networks in autonomous robot control. *Computing and Informatics*, *29*, 823–847.

Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T. E. J., Bucholz, R., … Consortium, W.-M. H. (2012). The Human Connectome Project : A data acquisition perspective. *NeuroImage*, *62*(4), 2222–2231. doi:10.1016/j.neuroimage.2012.02.018.The

Van Gerven, M. A. J., Cseke, B., de Lange, F. P., & Heskes, T. (2010). Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *NeuroImage*, *50*(1), 150–61. doi:10.1016/j.neuroimage.2009.11.064

Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005). Isolated word recognition with the Liquid State Machine: a case study. *Information Processing Letters*, *95*(6), 521–528. doi:10.1016/j.ipl.2005.05.019

Wandell, B. A, Brewer, A. A, & Dougherty, R. F. (2005). Visual field map clusters in human cortex. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *360*(April), 693–707. doi:10.1098/rstb.2005.1628

Wang, X., Hou, Z. G., Zou, A., Tan, M., & Cheng, L. (2008). A behavior controller based on spiking neural networks for mobile robots. *Neurocomputing*, *71*(4-6), 655–666. doi:10.1016/j.neucom.2007.08.025

Wang, X., Hutchinson, R., & Mitchell, T. M. (2003). Training fMRI Classifiers to Discriminate Cognitive States across Multiple Subjects. In *Advances in neural information processing systems* (p. none).

Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, *28*(4), 459–471. doi:10.1016/j.patrec.2006.09.003

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2006). On-Line Learning with Structural Adaptation in a Network of Spiking Neurons for Visual Pattern, 61–70.

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2007a). Spiking Neural Networks for Text-Independent Speaker Authentication. In *Artificial Neural Networks–ICANN* (Vol. 2, pp. 758–767). Springer Berlin Heidelberg.

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2007b). Text-Independent Speaker Authentication with Spiking Neural Networks, 758–767.

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2010). Evolving spiking neural networks for audiovisual information processing. *Neural Networks : The Official Journal of the International Neural Network Society*, *23*(7), 819–35. doi:10.1016/j.neunet.2010.04.009

Yamashita, O., Sato, M., Yoshioka, T., Tong, F., & Kamitani, Y. (2008). Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns. *NeuroImage*, *42*(4), 1414–29. doi:10.1016/j.neuroimage.2008.05.050

Yanduo, Z., & Kun, W. (2009). The Application of Liquid State Machines in Robot Path Planning. *Journal of Computers*, *4*(11), 1182–1186.

Yang, L., & Zhongjian, T. (2011). Prediction of Grain Yield Based on Spiking Neural Nerworks Model. *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 171–174.

Yourganov, G., Schmah, T., Churchill, N. W., Berman, M. G., Grady, C. L., & Strother, S. C. (2014). Pattern classification of fMRI data: applications for analysis of spatially distributed cortical networks. *NeuroImage*, *96*, 117–32. doi:10.1016/j.neuroimage.2014.03.074

Appendix A

StarPlus Main Source Codes

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% For StarPlus data extraction
%% fmri_data_reading
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% select trials of cond 2: in which interval is a sentence/picture trial in %
which thesentence is not negated?

trials=find([info.cond]==2);
[info1,data1,meta1]=transformIDM_selectTrials(info,data,meta,trials);
% separate P1st and S1st trials
[in-
foP1,dataP1,metaP1]=transformIDM_selectTrials(info1,data1,meta1,find([info1.fi
rstStimulus]=='P'));
[infoS1,dataS1,metaS1]=transformIDM_selectTrials(info1,data1,meta1,find([info1
.firstStimulus]=='S'));

% separate reading P vs. S
[in-
foP2,dataP2,metaP2]=transformIDM_selectTimewindow(infoP1,dataP1,metaP1,[1:24])
;
[infoS2,dataS2,metaS2]=transformIDM_selectTimewindow(infoS1,dataS1,metaS1,[1:2
4]);

% combine examples and create labels.  Label 'picture' 1, label 'sentence' 2
labelsP=ones(size(dataP2,1),1);
labelsS=ones(size(dataS2,1),1)+1;
dataPS=[dataP2;dataS2];

class_label=[labelsP;labelsS];
fmri_data=zeros(24,meta.nvoxels,20); %initialize 3d array of zeros data
fori=1:20
fmri_data(:,:,i)=dataPS{i,1}; %put the data in 3d array data
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% For Mapping the coordinates
%% fmri_talairach_transform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

set(handles.stage,'String','Loading NEUCUBE coordinate system (Talairach)
...'); drawnow;
loadneural_coordinate;

set(handles.stage,'String','the Talairach transformation is begin...');
drawnow;

% the nfMRI is fMRI with Talairach coordinate system
% x-axis
XMID = fix(mean(meta.colToCoord(:,1)));
dx = min(neuron_location(:,1))/min(meta.colToCoord(:,1) - XMID);
nfMRI(:,1) = dx * (meta.colToCoord(:,1) - XMID);
% y-axis
YMID = fix(mean(meta.colToCoord(:,2)));
dy = min(neuron_location(:,2))/min(meta.colToCoord(:,2) - YMID);
nfMRI(:,2) = dy * (meta.colToCoord(:,2)- YMID);
% z-axis
nfMRI(:,3) = 10 * (meta.colToCoord(:,3)-4);
```

```
set(handles.stage,'String','Identify every nfMRI points that close to NEUCUBE
point within RADIUS unit ...'); drawnow;

% identify every nfMRI points that close to NEUCUBE point within RADIUS unit
% the NEUCUBE using Talairach coordinate system with resolution 10 mm.
NEURON_ORIGINAL = 1471;
TAL = cell(NEURON_ORIGINAL,1);
INPUT_NEURON = zeros(NEURON_ORIGINAL,1);
% RADIUS unit is mm
RADIUS = 7;

for N=1:NEURON_ORIGINAL
% CIRCLE
TAL{N} = find( ...
        (sqrt( (nfMRI(:,1)-neuron_location(N,1)).^2 +(nfMRI(:,2)-
neuron_location(N,2)).^2 ) <= RADIUS)&...
        (nfMRI(:,3)==neuron_location(N,3)));
if (~isempty(TAL{N}))
INPUT_NEURON(N,1) = 1;
end
end

set(handles.stage,'String','Calculating the average value of voxel ...');
drawnow;
% Mapping voxel
SAMPLES = size(fmri_data,3);
TIMEPOINTS = size(fmri_data,1);
NTAL = length(TAL); % = NEURON_ORIGINAL
talvoxel=zeros(TIMEPOINTS,NTAL,SAMPLES);
for S=1:SAMPLES
for T=1:TIMEPOINTS
for N=1:NTAL
if (~isempty( TAL{N} ) )
talvoxel(T,N,S)=mean(fmri_data(T,TAL{N},S));
end
end
end
set(handles.stage,'String',sprintf('Calculating the average value of voxel ...
%d of %d samples',S,SAMPLES)); drawnow;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% transformIDM_selectTimewindow (info, data, Meta, snapshots)
%
% Returns a copy of info,data, Meta containing only the specified snapshots
% intime within each trial.  The input parameter 'snapshots' is an array
% listing the indices of snapshots to be included, assuming the index of
% thefirst snapshot of each trial is 1.
%Example:  select just snapshots 3,4,5, and 7 from each trial
% [info2, data2, meta2] =
% transformIDM_selectTimewindow (info, data, Meta, [3, 4, 5, 7])
%History
% - 11/2/02 TMM Created file.
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [rinfo,rdata,rmeta] = transfor-
mIDM_selectTimewindow(info,data,meta,snapshots)
ntrials= length(data);
rdata = cell(ntrials,1);
rmeta=meta;

for j=1:1:ntrials
rdata{j} = data{j}(snapshots,:);
rinfo(j) = info(j);
rinfo(j).len = length(snapshots);
end;
rmeta.nsnapshots= sum([rinfo.len]);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% transformIDM_selectTrials (info,data,meta,trials)
%
% Returns a copy of info, data, meta containing only the specified trials.
The
% input parameter 'trials' is an array listing the indices of trials to be
% included.
%
% Example:  select just trials 3 and 5
% [info2, data2, meta2] = transformIDM_selectTrials (info,data,meta,[3,5])
%
% History
% - 9/1/02 TMM Created file.
% - 5/9/05 indra Update mint and maxt after the selection of trials
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [rinfo,rdata,rmeta] = transfor-
mIDM_selectTrials(info,data,meta,trials)
ntrials=length(trials);

rdata = cell(ntrials,1);
rmeta=meta;
for j=1:1:ntrials
    t=trials(j);  % get trial number
rdata{j} = data{t};
rinfo(j) = info(t);
end;

rmeta.ntrials=length(trials);
rmeta.nsnapshots= sum([rinfo.len]);

% update mint and maxt
tStart = 1;
for j=1:rmeta.ntrials
tEnd = tStart + rinfo(j).len - 1;
rinfo(j).mint = tStart;
rinfo(j).maxt = tEnd;

tStart = tEnd + 1;
end;
```

Appendix B

Haxby Main Source Codes

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% For Haxby data extraction
%% fmri_data_reading
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

set(handles.stage,'String','The first stimulus NII data extraction ...');
drawnow;

load(strcat('COND00',int2str(COND1),'.mat'));
switch COND1
case 1
        COND = COND001; clear COND001;
case 2
        COND = COND002; clear COND002;
case 3
        COND = COND003; clear COND003;
case 4
        COND = COND004; clear COND004;
case 5
        COND = COND005; clear COND005;
case 6
        COND = COND006; clear COND006;
case 7
        COND = COND007; clear COND007;
case 8
        COND = COND008; clear COND008;
end

TIMEPOINT = 9;
SAMPLE = 12;  % number of TASK_RUN
% STIMULUS = COND2;
BASE = (SUBJECT - 1) * 108;

[X,Y,Z] = size(COND(:,:,:,1));
fmri_data = zeros(TIMEPOINT,Z*X*Y,2*SAMPLE);
class_label = zeros(1,2*SAMPLE);

for S=1:SAMPLE
ID = BASE + ((S-1)*TIMEPOINT );
       for ROW = 1:TIMEPOINT
              fornX=1:X
                     fornY=1:Y
                            fornZ=1:Z
              COL = (nX-1)*64*64 + (nY-1)*64 + nZ;
              fmri_data(ROW,COL,S) = COND(nX,nY,nZ,ID+ROW);
              class_label(1,S) = 1;
                            end
                     end
              end
        set(handles.stage,'String',sprintf('The second stimulus: timepoint %d,
        sample %d from SUBJECT: %d', ROW,S,SUBJECT)); drawnow;
       end
end

set(handles.stage,'String','The second stimulus NII data extraction ...');
drawnow;
load(strcat('COND00',int2str(COND2),'.mat'));
switch COND2
case 1
```

```
        COND = COND001; clear COND001;
case 2
        COND = COND002; clear COND002;
case 3
        COND = COND003; clear COND003;
case 4
        COND = COND004; clear COND004;
case 5
        COND = COND005; clear COND005;
case 6
        COND = COND006; clear COND006;
case 7
        COND = COND007; clear COND007;
case 8
        COND = COND008; clear COND008;
end

% STIMULUS = COND2;
BASE = (SUBJECT - 1) * 108;

for S=1:SAMPLE
ID = BASE + ((S-1)*TIMEPOINT );
       for ROW=1:TIMEPOINT
              fornX=1:X
                     fornY=1:Y
                            fornZ=1:Z
       COL = (nX-1)*64*64 + (nY-1)*64 + nZ;
       fmri_data(ROW,COL,12+S) = COND(nX,nY,nZ,ID+ROW);
       class_label(1,12+S) = 2;
                            end
                     end
              end
        set(handles.stage,'String',sprintf('The second stimulus: timepoint %d,
        sample %d from SUBJECT: %d', ROW,S,SUBJECT)); drawnow;
       end
end

%% used for Coordinate Transformation only
COND = COND(:,:,:,1);

fmri_niiTransform

set(handles.stage,'String','PreparingfMRI data for training and validation
...'); drawnow;

length_per_sample = size(fmri_data,1);
number_of_class = length(unique(class_label));
total_sample_number = size(fmri_data,3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
percentage = str2num(get(handles.percentage,'string'));
sample_amount_per_class_for_training =fix(total_sample_number * percent-
age/number_of_class);
sample_amount_per_class_for_validation = total_sample_number/number_of_class -
sample_amount_per_class_for_training;
sample_amount_for_training = sample_amount_per_class_for_training * num-
ber_of_class;
sample_amount_for_validation = sample_amount_per_class_for_validation * num-
ber_of_class;

class_label_for_training = [];
class_label_for_validation = [];

z = 0;
fori = 1:number_of_class
% consider only for INPUT NEURON
```

```
    fmri_data_for_training(:,:,z+1:z+sample_amount_per_class_for_training) =
talvoxel(:,INDEX_INPUT,(i-1)*total_sample_number/number_of_class+1:(i-
1)*total_sample_number/number_of_class+sample_amount_per_class_for_training);
    z = size(fmri_data_for_training,3);
end

z=0;
fori = 1:number_of_class
% consider only for INPUT NEURON
fmri_data_for_validation(:,:,z+1:z+sample_amount_per_class_for_validation) =
talvoxel(:,INDEX_INPUT,i*total_sample_number/number_of_class-
sam-
ple_amount_per_class_for_validation+1:i*total_sample_number/number_of_class);
    z = size(fmri_data_for_validation,3);
end

fori = 1:number_of_class
class_label_for_training =
[class_label_for_trainingones(1,sample_amount_per_class_for_training)*i];
class_label_for_validation =
[class_label_for_validationones(1,sample_amount_per_class_for_validation)*i];
end

clearz
cleari;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% For Mapping the coordinates
%% fmri_niiTransform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load('neural_coordinate.mat');
set(handles.stage,'String','Create fmri_colToCoord ...'); drawnow;

%% create colToCoord
% Z= 12:51>40, Y=20:59>40
fmri_colToCoord = zeros(40*40*40,4);
fornx=1:40
       forny=1:40
              fornz=1:40
                 ROW = (nx-1)*40*40 + (ny-1)*40 + nz;
fmri_colToCoord(ROW,1) = nx;
fmri_colToCoord(ROW,2) = ny;
fmri_colToCoord(ROW,3) = nz;
              end
       end
end

vfMRI = COND(:,20:59,12:51);
NFMRI_LOCATION = 40*40*40;
nfMRI = zeros(NFMRI_LOCATION,4);
% x-axis
XMID = fix(mean(fmri_colToCoord(:,1)));
dx = min(neuron_location(:,1))/min(fmri_colToCoord(:,1) - XMID);
nfMRI(:,1) = dx * (fmri_colToCoord(:,1) - XMID);
% y-axis
YMID = fix(mean(fmri_colToCoord(:,2)));
dy = min(neuron_location(:,2))/min(fmri_colToCoord(:,2) - YMID);
nfMRI(:,2) = dy * (fmri_colToCoord(:,2)- YMID);
% z-axis
% nfMRI(:,3) = 10 * (fmri_colToCoord(:,3)-4);
ZMID = fix(mean(fmri_colToCoord(:,3)));
dz = min(neuron_location(:,3))/min(fmri_colToCoord(:,3) - ZMID);
nfMRI(:,3) = dz * (fmri_colToCoord(:,3)- ZMID);
fornx=1:40
       forny=1:40
              fornz=1:40
                 ROW = (nx-1)*40*40 + (ny-1)*40 + nz;
```

```
nfMRI(ROW,4) = vfMRI(nx,ny,nz);
            end
      end
end

% filter only for important neuron
IDX = find(nfMRI(:,4) > THRESHOLD);
nfMRI = nfMRI(IDX,:);

set(handles.stage,'String','Identify every nfMRI points that close to NEUCUBE
point within RADIUS unit ...'); drawnow;

% identify every nfMRI points that close to NEUCUBE point within RADIUS unit
% the NEUCUBE using Talairach coordinate system with resolution 10 mm.
NEURON_ORIGINAL = 1471;
TAL = cell(NEURON_ORIGINAL,1);
INPUT_NEURON = zeros(NEURON_ORIGINAL,1);
% RADIUS unit is mm
RADIUS = 7;

for N=1:NEURON_ORIGINAL
% Collect nii_neuron around NeuCubearound
% CIRCLE
      TAL{N} = find( ...
               (sqrt( (nfMRI(:,1)-neuron_location(N,1)).^2 +(nfMRI(:,2)-
      neuron_location(N,2)).^2 ) <= RADIUS)&...
               (sqrt( (nfMRI(:,1)-neuron_location(N,1)).^2 +(nfMRI(:,3)-
      neuron_location(N,3)).^2 ) <= RADIUS)&...
               (sqrt( (nfMRI(:,2)-neuron_location(N,2)).^2 +(nfMRI(:,3)-
      neuron_location(N,3)).^2 ) <= RADIUS) ...
               );

      if (~isempty(TAL{N}))
      INPUT_NEURON(N,1) = 1;
      end
end

set(handles.stage,'String','Calculating the average value of voxel ...');
drawnow;

% Mapping voxel
SAMPLES = size(fmri_data,3);
TIMEPOINTS = size(fmri_data,1);
NTAL = length(TAL); % = NEURON_ORIGINAL
talvoxel=zeros(TIMEPOINTS,NTAL,SAMPLES);
for S=1:SAMPLES
      for T=1:TIMEPOINTS
            for N=1:NTAL
                  if (~isempty( TAL{N} ) )
      talvoxel(T,N,S)=mean(fmri_data(T,TAL{N},S));
                  end
            end
      end
set(handles.stage,'String',sprintf('Calculating the average value of voxel ...
%d of %d samples',S,SAMPLES)); drawnow;
end
```