A brain-computer interface based on a spiking neural network architecture – NeuCube and neuro-feedback

A case study on memory enhancement of a subject by providing neuro-feedback through a simple gaming environment

Wriju Bhattacharya

A thesis submitted to Auckland University of Technology in partial fulfilment of the requirements for the degree of Master of Computer and Information Sciences (MCIS) 2015

Faculty of Design & Creative Technologies

School of Computing and Mathematical Science

Supervisors:

Professor Nikola Kasabov

Doctor Grace Wang

Abstract:

In an average human brain, there are about 100 billion neurons connected by synapses that transmits electro-chemical signals to the different parts of the body. Brain Computer Interfacing (BCI) can be used to study and record these signals and has gained a lot of interest over the last few decades. Many different methods are used for BCI, EEG being the most common of these methods. Although we now understand more than ever before about how a brain functions through collecting the spatial and temporal brain data and studying it, little work has been done towards actually using this data to enhance brain signals or treat brain related disorders such as Attention Deficit Hyperactivity Disorder (ADHD) in small children. This thesis identifies this need and proposes neuro-feedback as a process to provide positive feedback to a subject based on their brain activities in real time. Although some studies have been conducted in the past in the field of neuro-feedback, it has failed to gain credibility in the larger scientific community. This thesis outlines a gaming environment called NUN developed in which a subject is connected to an EEG machine and their brain data is collected while providing a visual stimulus via a computer game. The gaming environment is used to train a subject with a certain predefined change in patterns while expecting them to remember the same. The subject is then given feedback in terms of a score. The proposed idea is that the subject can keep playing the game until they have attained a perfect score and while doing so enhance their memory. The developed game was only tested on the developer, so future work is required to perform experiments on more than one subject to prove the concept. This paper also suggests that BCI and neuro-feedback experiments can be undertaken via the use of a very inexpensive, light and easy-to-use EEG machines and that BCI technology is not limited to larger more complex laboratory setups.

Contents

List of Figures

List of Abbreviations

ADHD: Attention Deficit Hyperactivity Disorder

AI: Artificial Intelligence

BCI: Brain Computer Interface

BOLD signals: Blood Oxygenation Level Dependent signals

BP: Berishaftspotential

CFT: Continious Fourier Transform

CNV: Contingent Negative Variation

deSNN: Dynamic Evolving Spiking Neural Networks

DFT: Discrete Fourier Transform

ECoG: Electrocorticography

EEG: Electroencephalography

EOG: Electrooculography

ERP: Event Related Potentials

eSNN: Evolving Spiking Neural Networks

FFT: Fast Fourier Transform

fMRI: Functional Magnetic Resonance imaging

fNIR: Functional Near-Infrared Spectroscopy

HCA: Hierarchical Cluster Analysis

HMM: Hidden Markov Model

LDA: Linear Discriminant Analysis

LIFM: Leaky Integrate and Fire Model

LTD: Long-term Depression

LTP: Long-term Potentials

MEG: Magnetoencephalography

MLP: Multilayer Perceptron

MMN: Mismatch Negativity

MNI: Montreal Neurological Institute

MRI: Magnetic Resonance imaging

MVFS: Multivoxel Functional Spectroscopy

NN: Neural Networks

NUN: neuro-feedback Using NeuCube

SMA: Supplementary Motor Area

SNN: Spiking Neural Networks

SQUID: Superconducting Quantum Interference Device

STBD: Spatio-temporal Brain Data

STDP: Spike Timing Dependent Plasticity

SVM: Support Vector Machine

Attestation of Authorship

"I hereby declare that this submission is my own work 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 extent has been submitted for the award of any other degree or diploma of an university or other institution of higher learning".

Wriju Bhattacharya

July, 2015

Acknowledgements

I am writing this thesis with the hope of being awarded a Master's degree in Computer and Information Science. It reflects all my years work in the field of BCI, and I certainly could not have done it alone.

I would like to take this opportunity to thank a few exceptional people that I have had the privilege to know and to work with and who have stood by me during this journey.

First and foremost I would like to thank my mother. I had to take a break from study and work for few years. My mother always maintained that I return to finish my study. Now that I have, I am ever so grateful that I did.

I would like to thank my supervisor and teacher Professor Nikola Kasabov for always being there and extending a helping hand whenever I needed it. Throughout this journey he has guided me as a mentor, corrected me whenever I made a mistake and shared his knowledge on the topic with me. Whenever I had the chance to brainstorm with him, it was always very productive toward my work. Thank you, Professor, for being there.

I would like to specially thank my good friend and colleague Akshay Gollahali for his support. He has always encouraged me to keep working even when I felt that I was not up to it.

I would like to extend my warm regards and thanks to Joyce D'Mello, who has been like a mother, guiding not only me, but everyone else here at KEDRI with her kind and inspiring words.

Special thanks to Dr Grace Wang for her comments on my thesis and suggestions for its improvement.

Thanks to Catriona Carrathers for her proof reading services.

And last, but not least, I want to thank God for providing me with this opportunity to work amongst such reputable people at such a reputable institution.

1.1 Introduction to the topic

The human brain is considered one of the most complex structures known to mankind. Compared to other primates, the human brain is much larger in size and much more sophisticated. It is the ‘central computer’ of the human nervous system and because of it can we undertake all our day to day functions that we so take for granted. If we were to just stop and ponder the complexity of the brain we would be amazed.

An average human brain weighs about 1.5 kg and makes up about two percent of a person’s body weight. It consists of 100 billion neurons connected by synapses which transmit electro chemical signals from the brain to the different parts of the body. It is highly adaptive and gives us our consciousness. It is therefore only natural that the subject of human brain has eluded and amazed scientists for centuries and has been the subject of study since.

The study of the human brain dates back to even before the early Egyptian mummification period. Perhaps Egyptian neuro-surgery was the first of its kind. Texts from old papyruses have been found explaining their knowledge of how to treat a head injury with surgery. Neuroscience since then has become far more sophisticated and advanced. We now have knowledge of many functions of the brain, which parts of the brain control which actions in the human body for example. However there are still some gaps in our understanding of the human brain. In spite of all the advances in the field, we still do not have a complete understanding of the entire functioning of the brain. How it can perform complex tasks such composing music or writing poetry or even have consciousness itself is still a mystery.

Scientists have for years studied the brain with the hope that they can understand its functionality and possibly replicate it. This has given rise to artificial intelligence by merely copying various ways a brain functions. One branch of this field is particularly pertinent to this study, BCI or Brain-computer Interface.

BCI or Brain-computer Interface, sometimes called mind-machine interface (MMI), is a technology that records electro-chemical signals from the brain, associates (classifies) the signal with some sort of action, such as thinking about moving the arm to the left and then sends this data to a computer in order to interact with it without using any kind of physical movements. The brain signals are thus converted into machine comprehensible commands so that a certain kind of action can be performed, for example, the moving of a robotic arm. The full implications of this technology can revolutionise our current technologies. It has implications in many fields such as medicine, military, augmented reality and many more.

There are two main forms of BCI, namely Invasive and Non-invasive.

Invasive BCI involves planting electrodes inside the skull or inside the grey matter of the brain. Since the electrodes are closer to the brain, the data collected is more accurate and contains less noise. However, this involves complex brain surgery and raises serious ethical issues.

In non-invasive BCI, electrodes are placed around the skull to measure the brain activities from different areas of the brain. The subject generally wears a cap in which the electrodes are fitted. Conductive gels are used to increase impedance between the electrodes and the scalp.

1.2 History of BCI

The rhythmic oscillation in the brains of mammals was first demonstrated in the 1870s by a psychologist from Liverpool, Richard Caton. However the first structured and systematic study of the brain signals can only be traced back to one man and one man only, Hans Berger, a German neurologist. He used Electroencephalography (EEG) to successfully record human brain signals in 1929. His early work included the recording of the brain signals of many of his patients, including his son and himself. He was convinced that the same action carried out by different human beings should result in the same EEG readings. He also recorded EEG readings of patients who had neuro-psychological disorders like dementia, epilepsy etc. to compare them with ‘normal’ brain readings. After studying EEG recordings for many years he was filled with doubts about his findings through EEG and strictly restricted his research findings to facts and figures rather than hypotheses and speculation.

Modern day BCI and EEG is based on Berger's research and his rudimentary EEG machine.

1.3 Motivation

"The day science begins to study non-physical phenomena, it will make more progress in one decade than in all the previous centuries of its existence." - Nikola Tesla

There are many phenomena occurring in the universe which are nothing short of miracles. How a single celled organism evolved into a being as complex as humans and how an entire universe was formed out of the Big Bang are just to name a few.

We, as a species have always been inspired by fantasies of thought control, telepathy etc. and these ideas have been the basis of countless science-fiction movies. In my childhood I used to watch Star Trek, where Mr Spock would able to able to read another person's mind just by touching him. How fascinating would it be if we could do something similar in reality? Now I am not saying that we start building machines which will be able to read another human being's thoughts. The very idea is daunting. But consider the example of a paraplegic man. If he were use a machine which could understand his thoughts and perform some mechanical actions by simply thinking about it, wouldn't that be nothing short of a miracle for that person. He would be able to go about his daily business without anyone's help. This is where the true miracle of science would lie.

1.4 Research questions

A well-structured research is incomplete without research questions. It is the questions that a researcher ponders that takes the research down a definite path. This research has been stimulated by few questions and, while some has been answered, others are probably opportunities for future researches.

Some of the questions which have been answered by this thesis are:

- How can we convert thoughts into machine understandable commands which can be used to perform certain mechanical actions for example the moving of a robotic arm?
- What are the different classification algorithms currently used by scientists and neurologists to perform BCI actions?
- How can BCI be made more affordable for ordinary people with neuro-psychological disorders?
- How sophisticated are existing BCI technologies that the scientists are currently working with, and can they be made better?
- Each human being has unique brain signals. To perform same action, for example driving a car, the brain signals in person A will be different from brain signals in person B. How can a machine be devised which will uniquely identify brain patterns and perform the same mechanical action for person A as well as person B?
- Can BCI be used to cure psychological disorders such as epilepsy or ADHD?

Questions which may be solved in the future are as follows:

- In the future, can BCI bridge the gap that is currently in the field of Artificial Intelligence?
- During darker eras of human history, such as the World War II, a lot of human experimentation was performed to harness information about the functioning of brain and how it works. To what degree can experiments be ethically performed on humans to understand brain functioning in the development of BCI technology?

1.5 Research scope

The field of BCI is quite new and the field of neuro-feedback is even more so. Neuro-feedback is a method by which subjects with neuro-psychological disorders are given some form of positive reinforcement/feedback based on the activities in their brain in order to self-regulate and enhance their brain activity. Neuro-feedback has in the past been used successfully to ease epileptic seizures in patients (Kotchoubey, Schneider, Uhlmann, Schleichert, & Birbaumer, 1997). It has also been used to treat ADHD (Attention deficit Hyperactive Disorder) in children (Geller, 2007).

However scientists working with neuro-feedback have always faced the challenge of being taken seriously by the larger scientific community.

This research aims to prove that neuro-feedback can be used as a viable solution for treating neuro-psychological disorders by providing a score based on how well they have performed. The subject can then try to improve their thought and thereby increase their memory.

1.6 Research contribution

Below are outlined some of the contributions this research will add to the field:

- A poster presented in the NCEI 2015 conference along with demonstration of the project using NeuCube framework to provide neuro-feedback.
- A detailed literature review of the field of BCI and various BCI technologies and the human brain and how they fit into the whole BCI picture.
- A gaming software developed to provide bio-feedback. This software also demonstrates real time EEG signal acquisition and real-time signal classification.
- A suggestion to the BCI community to consider BCI with neuro-feedback as a viable technique to provide psychological support and possible treatment for patients with neuro-psychological disorders.
- A gateway for future researches on the topic of BCI and neuro-feedback which could extend the software into a more sophisticated gaming environment with more complexity and better results.

1.7 Research outline

This thesis is divided into seven chapters.

Chapter 2 outlines a literature review of the brain in relation to BCI.

Chapter 3 mentions some of the most common BCI techniques and applications while chapter 4 outlines some of the BCI signal processing techniques and sheds some light on some of the common BCI classification methods.

Chapter 5 introduces a new method of measuring spatio-temporal brain data called Spiking Neural Network architecture implemented in an architecture called NeuCube.

The research methodology and design are presented in Chapter 6.

Chapter 7 presents the conclusion and future work related to BCI.

Appendix A is the code used to provide neuro-feedback in this research.

Chapter II – A literature review of the human brain in relation to BCI

In this chapter, a detailed literature review of human brain and how it fits into the field of BCI and neuro-feedback are given.

BCI, if simply put, is interfacing a computer 'brain' with a human brain. It is the process or technology in which human thought signals are converted into machine commands which are then used to perform mechanical actions such as the moving of a robotic arm or the moving of a cursor on a computer screen.

In order to understand BCI, it is necessary to understand the anatomy of the human brain and its functioning.

2.1 The Human Nervous System

The human nervous system is an intrinsic network of tissues and cells which is responsible for "response to stimuli". It is comprised of sensory receptors which conduct nerve impulses from the receptors to the brain and the spinal cord and also brings back impulses to the receptor organs. The entire human nervous system can be broadly classified into two parts:

[1] The central nervous system – comprising of the brain and the spinal cord.

[2] The peripheral nervous system – this consists of the nerve cells which carry impulses to and from the central nervous system.

2.1.1 The Central Nervous System

The human central nervous system consists of the brain and the spinal cord. Both the spinal cord and the brain develop from the neural tube at the embryonic stage. They are surrounded by cerebrospinal fluid and have a covering of protective membranes called meninges. The brain is encased in a thick, hard and bony structure called the cranium or skull. The spinal cord, on the other hand is lies within the vertebral column which comprises of vertebrae joined by ligaments.

2.1.1.1 Human Brain

The human brain roughly weighs about 3 pounds and comprises of 2% of the entire body weight (Encyclopaedia Britannica).

The brain can be broadly divided into three regions

- ❖ The left and right hemispheres comprising the cerebrum.
- ❖ The brain stem. This includes the thalamus, hypothalamus, midbrain, medulla oblongata, pons etc.
- ❖ The cerebellum.

<u>2.1.1.1. A: The Cerebrum</u>

The cerebrum can be distinctively divided into two regions; namely the left hemisphere and the right hemisphere. A deep groove called the longitudinal fissure partially divides the cerebrum into the respective hemispheres. A thick band of white matter lies at the base of the longitudinal fissure called the corpus callosum. Studies have proven that the side of the hemisphere which receives sensory stimuli controls motor functions on the opposite side i.e. the Left Hemisphere is responsible for controlling the right side of the body and the Right Hemisphere is responsible for controlling the left (Nieuwenhuys, Voogd, & Huijzen, 2008).

For the purpose of anatomical classification, the cerebrum is divided into 4 lobes which perform distinctive functions of the body as shown in Figure 2.1.

1. The Frontal Lobe - This area consists of a primary motor cortex called the "precentral gyrus" (Fedorenko, Fillmore, Smith, Bonilha, & Fridriksson, 2015) , the premotor and supplementary motor areas, Broca area and the prefrontal cortex. Giving an electrical charge to parts of the gyrus in patients under anaesthesia has shown involuntary motor actions in the other side of the body. Injury to this gyrus can cause paralysis. The Broca area is responsible for speech. Other functions of the frontal lobe include voluntary motor actions, language production, and determination of personality.

2. The Parietal Lobe – parts of the parietal lobe receive sensory inputs from the opposite side of the body. This lobe performs the function of comprehending language, and providing a sense of spatial position.

3. The Temporal Lobe – This constitutes of the middle fossa (hollow area) of the skull. This lobe is believed to be linked to hearing of sounds.

4. The Occipital Lobe – This lobe extends from the parieto-occipital sulcus to calcarine sulcus and has a Y shaped structure. It is the primary area for performing visually related functions.

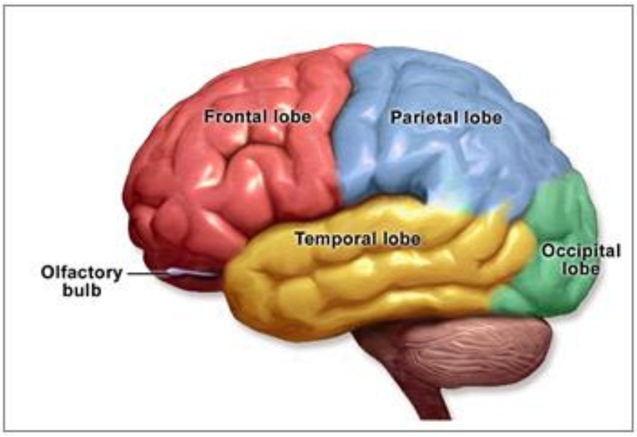


Figure 2. 1 Diagrammatic representation of the brain outlining the 4 lobes

(nyest.hu, 2015)

<u>2.1.1.1. B: The Brain Stem</u>

The brain stem comprises all structures which are responsible for connecting the cerebrum and the spinal cord. These structures includes the hypothalamus, thalamus, epithalamus and the subthalamus and can be collectively called as diencephalon. Just below the diencephalon, sits the mid brain and below that sits the pons and medulla oblongata (the hindbrain). The structure of the brain stem is outlined in Figure 2.2

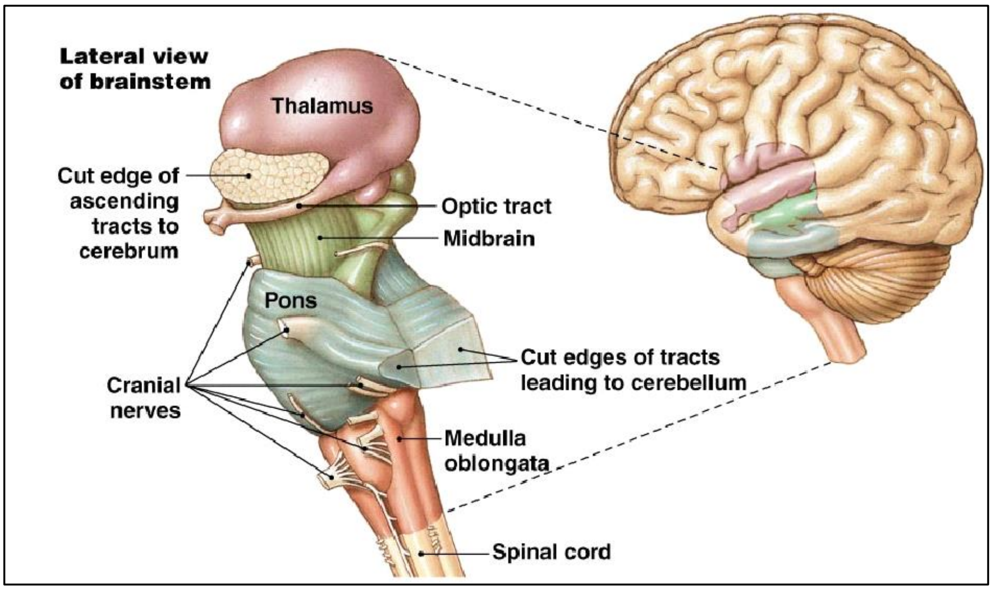


Figure 2. 2 Lateral view of the brain stem

(Fenner, 2013)

2.1.1.1. C: The Cerebellum

As shown in Figure 2.3, the cerebellum, also sometimes referred to as the small brain lies to the posterior of the pons and the medulla oblongata. The cerebellum is comprised of two lateral lobes or hemispheres separated by a midline called the vermis. The cerebellum is composed of an outer surface called grey matter and an inner core called white matter. Three bundles of fibre (superior, middle and inferior peduncles) connect the cerebellum with the midbrain, pons and medulla oblongata respectively (Nolte, 1988).

The cortex of the cerebellum consists of three layers. The outer synaptic layer or the molecular layer, the intermediate Purkinje layer and the inner granular layer or the receptive layer. The receptive layer consists of numerous number of nerve cells from which axons are projected into the synaptic layer. The dendrites of the Purkinje cells are excited by these axons. Due to the presence of the Purkinje cells, all sensory inputs into the cerebellum result in inhibitory actions upon the cells and nucleus of the cerebellum. Conversely, the deep cerebellar nuclei cells project excitatory actions upon the thalamus.

The cerebellum is the main part of the brain which computes quick responses to sensory signal inputs.

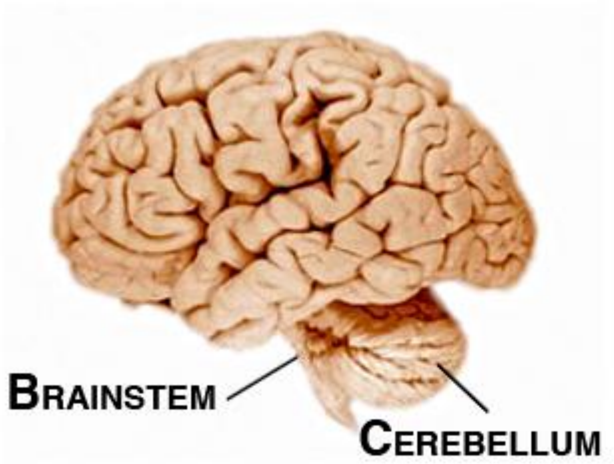


Figure 2. 3 The cerebellum

(Serendip Studio, 2015)

2.1.1.2 The Spinal Cord

The human spinal cord is a cylindrical-shaped, elongated structure roughly 18 inches long extending from the medulla oblongata to the first and second lumbar vertebrae. The spinal cord comprises of grey matter and white matter. The grey matter contains motor neuron fibres which connect the dorsal and the ventral ganglia. It is responsible for reflex actions and also the modulation of signals. The grey matter is made up of three horn like structures namely the dorsal horns, comprised of sensory neurons, the lateral horns comprised of visceral neurons, and the ventral horns comprised of motor neurons. There are 31 pairs of spinal nerves (Figure 2.4) associated with the spinal cord and on the basis of that, the spinal cord is divided into the below segments:

- 8 pairs of cervical nerves
- 12 pairs of thoracic nerves
- 5 pairs of lumbar nerves
- 5 sacral nerves
- 1 coccygeal nerve

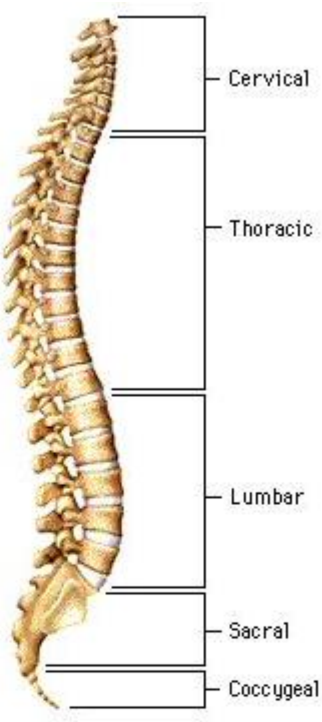


Figure 2. 4 The spinal cord

(Spinal Injury Network, 2009)

2.1.2 The Peripheral Nervous System

The peripheral nervous system is comprised of numerous nerve cells and ganglia which are used as a channel for sending impulses from the central nervous system to the muscles and organs of the body and vice versa. As we can see in Figure 2.5, the peripheral nervous system is comprised of the following:

- Spinal nerves
- Cranial nerves
- Parts of the autonomous nervous system.

The pathway of the peripheral nervous system is largely comprised of neurons which are nothing but nerve cells with axons and dendrites.

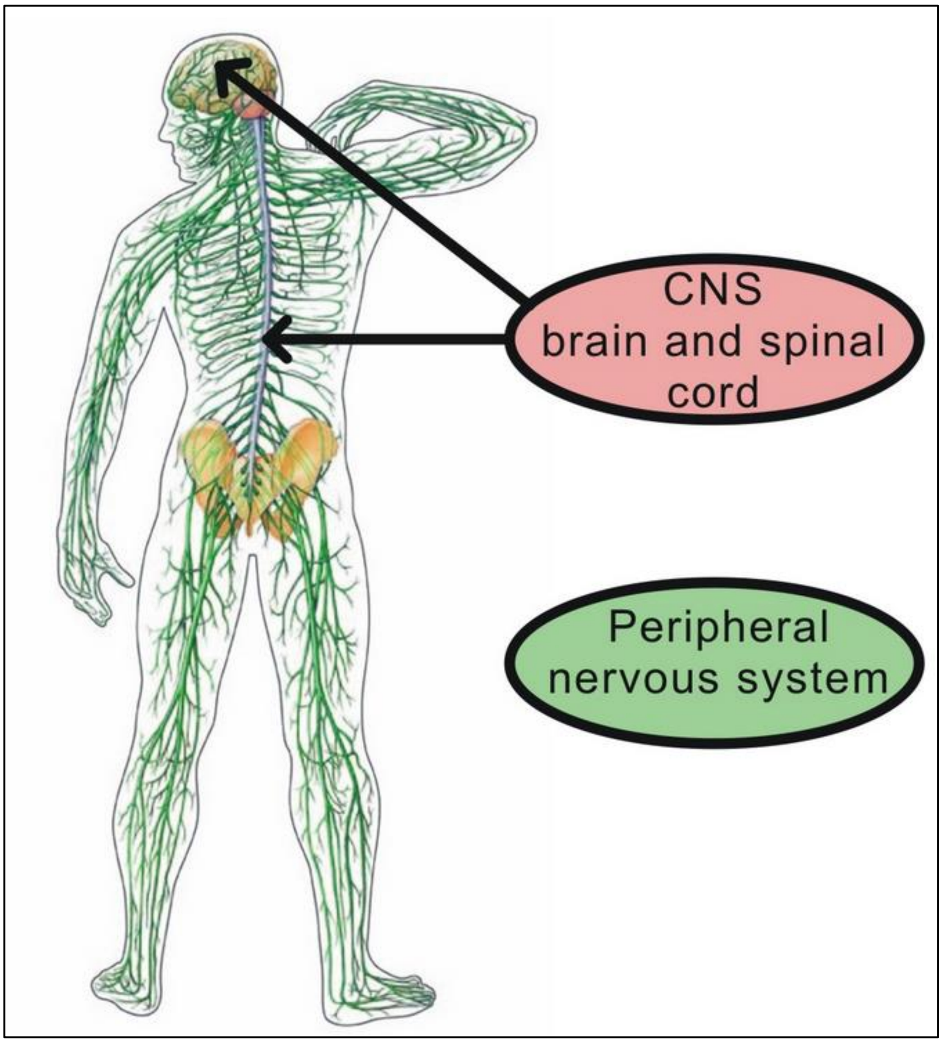


Figure 2. 5 The CNS and PNS

(Bernaciková, 2012)

2.1.2.1 Neurons and Electrical Signalling

The signals for the brain need to be transmitted to the specific part of the body for which the computation is meant. This job is done by nerve cells or neurons. These special cells transmit electro-chemical signals from the brain to different parts of the body. Modern BCI systems uses different techniques to measure this unidirectional flow of electricity for the purpose of interpreting the brain. The thesis exploits the firing potentials of neurons to classify brain signal patterns. Hence some of the neuron firing properties are outlined in this section of the chapter.

A neuron can be distinctly divided into four distinct regions (Figure 2.6).

1. Soma or Cell Body – this is the control centre of the neuron. It manufactures as well as synthesises neuronal proteins.

2. Dendrites – Their job is to capture incoming signals from other neurons.

3. Axons – Axons provide a passage for outgoing neuronal signals.

4. Axon terminals – These contain neurotransmitters which are nothing but chemical mediums which allow signal transmission.

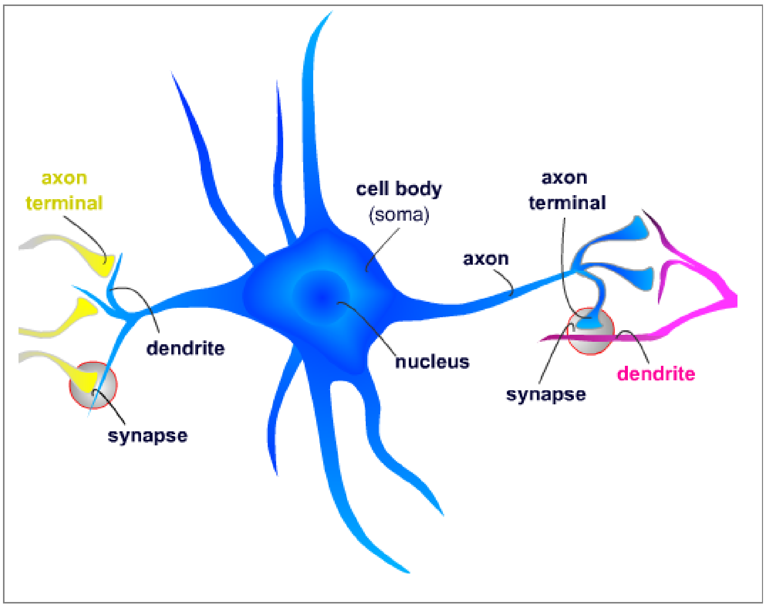


Figure 2. 6 Structure of a neuron

(Braingamereview, 2015)

Neurons and synapses work in a similar fashion to the wires and transistors of electrical devices that we are so familiar with and use in our day-to-day lives. However, in contrast to electrical wires, neurons are surrounded by aqueous mediums. Hence it is quite a difficult task to transmit electrical signals through them. Therefore neurons use the movement of ions rather than electrons to transmit electrical current from the brain to parts of the body for which the spike is intended. It utilises the energy from the concentration of ionic gradients to do so and is controlled by molecular switches.

The firing of signals through neurons is due to an action potential generated near the soma as shown in Figure 2.8 (a). Action potentials are movements of ions across membranes. The ions move across the membranes due to the opening and closing of neurotransmitters. Neurons are bounded by an electrically polarized semipermeable membrane with a resting membrane potential of -65 mV. When a neuron is excited due to neurotransmission, the membrane potential starts increasing. After it reaches the threshold which is achieved when the potential increases by about 10 mV (Figure 2.8 b). It results in the complete depolarization of the membrane. As a result the sodium channels open and sodium ions rush in after which the channels again close back. This is also called as the Rising Phase (Figure 2.8 c).

The next step in the neuron firing phase is the falling phase where the potassium channels open and repelled by the positive charge inside, potassium ions move out through the potassium channels resulting in repolarisation. Due to this action, the membrane potential returns to normal and this the potassium channel also closes (Figure 2.8 d).

During the brief period before the membrane potential returns back to – 65 mv, there is a phase called the refractory phase during which the neuron cannot fire another Action Potential (Figure 2.8 e).

Once the refractory phase passes, the membrane potential returns back to its resting potential, the excess sodium and potassium ions diffuse and the neuron is ready to fire again.

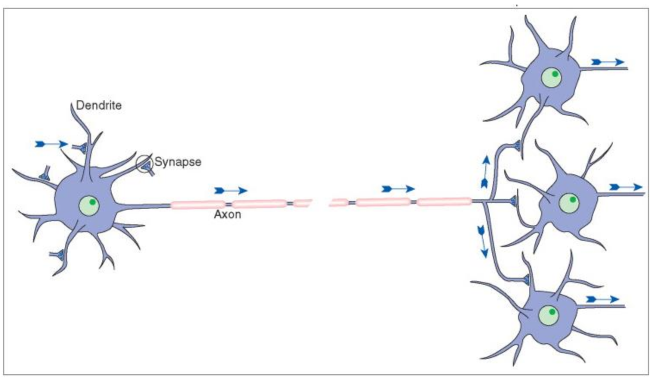
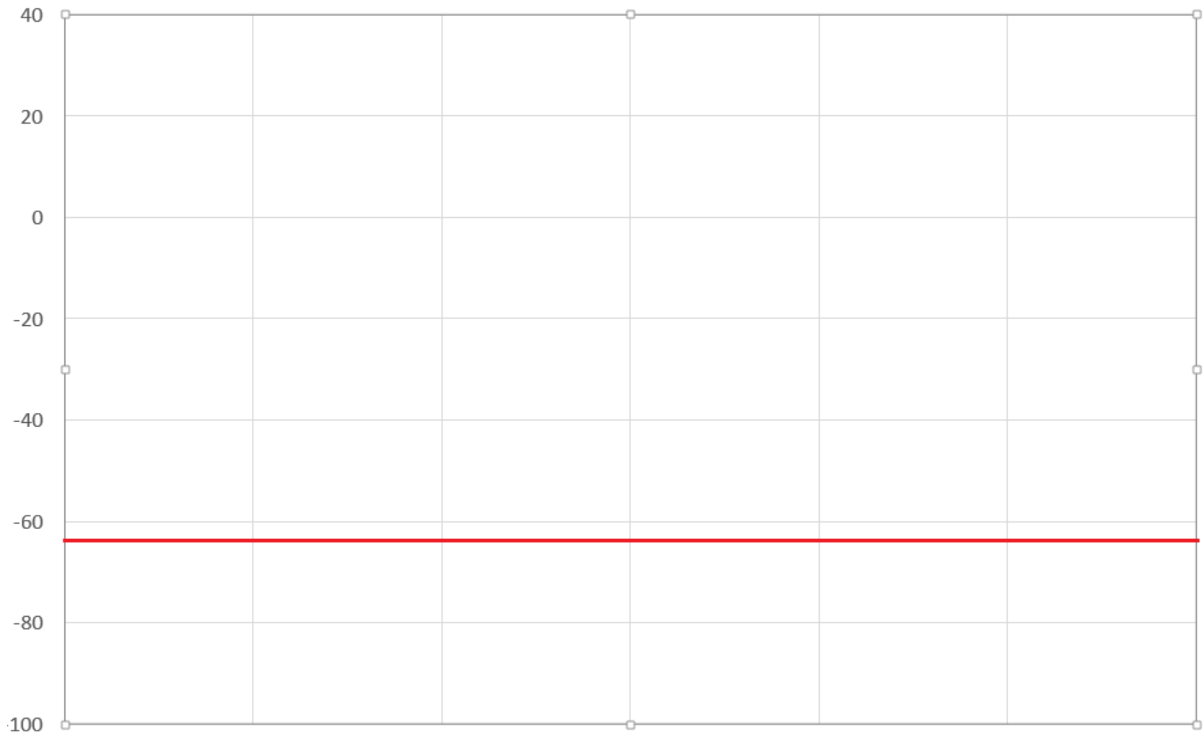
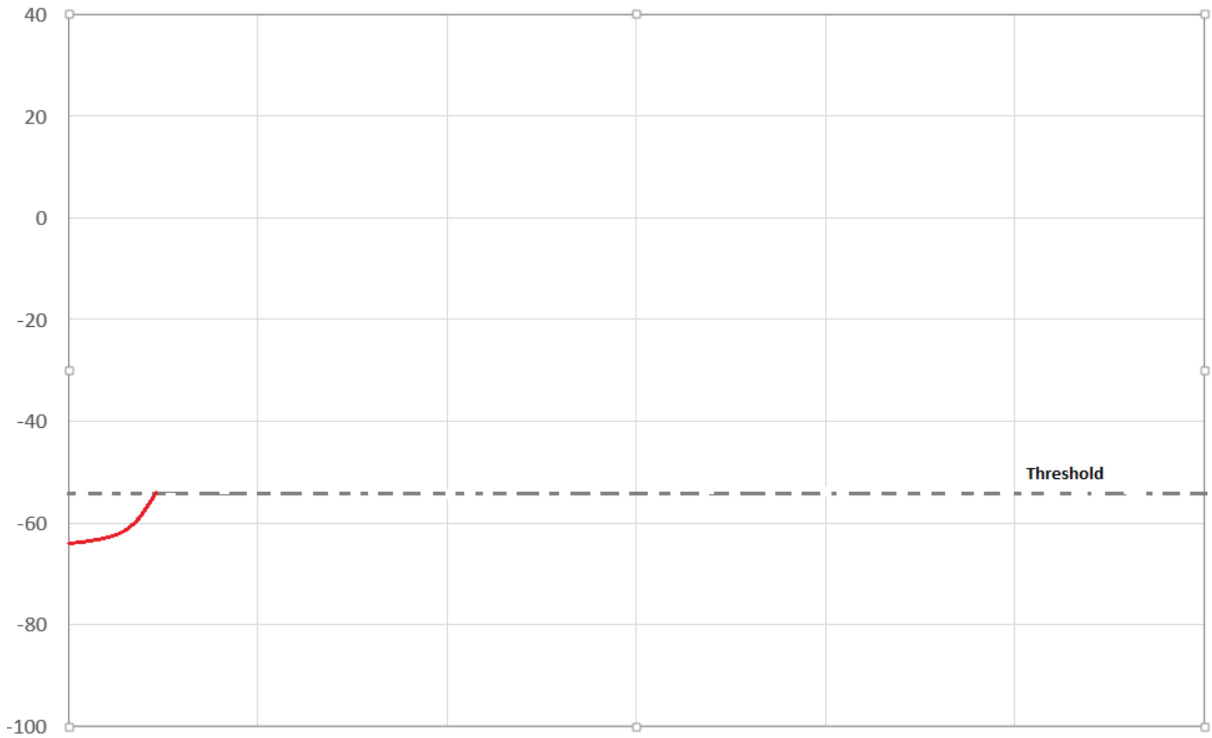


Figure 2. 7 Unidirectional flow of ions from one neuron to another

(Nolte, 1988)



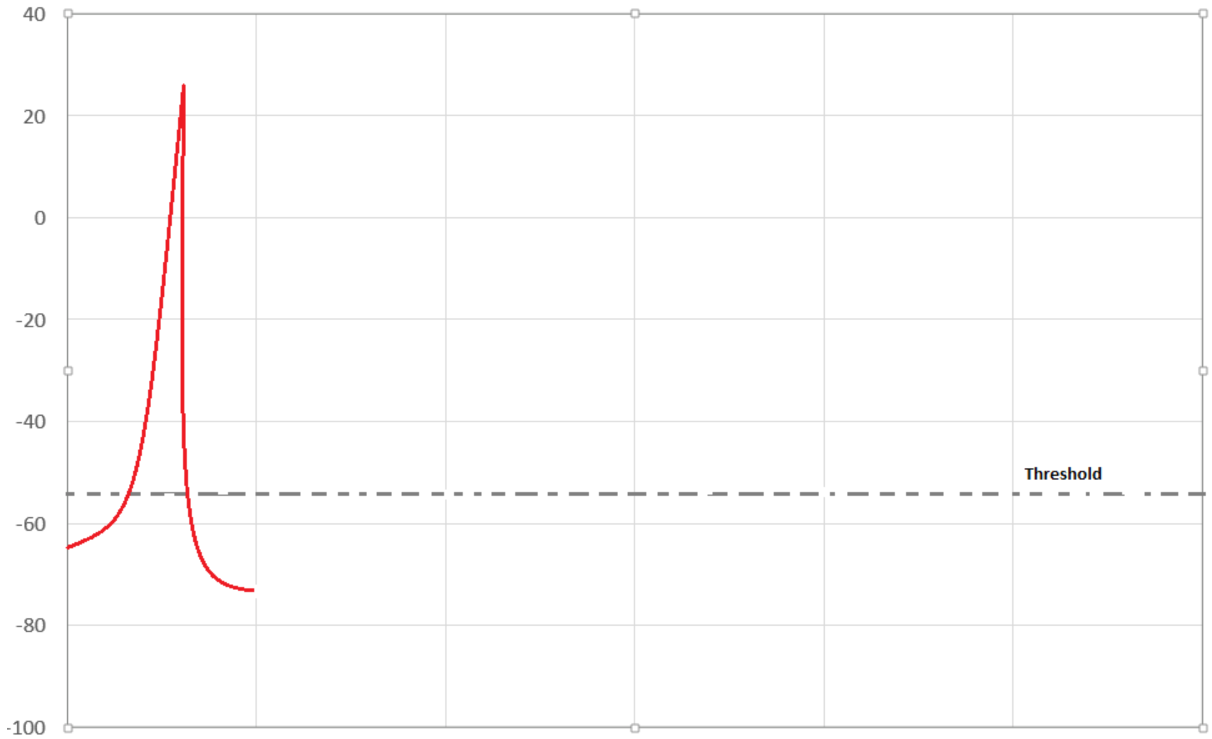
(a)



(b)


40
20
0
-20
-40
-60
-80
-100
Threshold

(c)


40
20
0
-20
-40
-60
-80
-100
Threshold

(d)

(e)

Figure 2. 8 Neuron firing phases

(a) Resting Potential Phase (b) Stimulus Phase (c) Rising Phase (d) Falling Phase (e) Undershoot Phase

2.3 Conclusion

This chapter provides a detailed review of the anatomy of the human brain in relation to BCI. It also outlines the various firing phases of the neurons in the brain which is very important for the study of BCI. In the next chapter, the different BCI applications built on the principal of capturing brain data has been outlined.

Chapter III – BCI techniques and applications

The previous chapter described how the human brain functions and how neurons are fired in response to a stimulus. With that knowledge in mind, BCI scientists have engineered many applications that can transmit signals from the brain to a machine and translate that signal into a meaningful command that can be used to perform a mechanical function. This chapter outlines some of the different types of invasive and non-invasive BCI techniques and their applications and different ways to improve BCI functionality. It also introduces the concept of neuro-feedback that can be used as a technique for improving brain signals as well as a potential treatment for some neuro-psychological disorders like ADHD. This chapter has also outlined some of the common challenges that one needs to address while setting up BCI systems and means to resolve these challenges.

3.1 BCI Types

BCI technologies can be broadly divided into three types namely Invasive, partially invasive and non-invasive.

3.1.1 Invasive and Partially-invasive BCIs

The invasive BCIs are based on detection of single neuron activity by implanting intra-cortical electrodes directly into the grey matter of the brain (Kasabov, Springer Handbook of Bio-/Neuroinformatics , 2014). Invasive BCI types are generally used to help people with serious paralysis. Invasive BCI has also been used in the past to restore mobility to people who has lost a limb. In this case invasive BCI was used to connect to make the connection to robotic arms and legs.

There has also been cases such as the vision BCI project where a blind subject's vision was restored by implanting BCI electrodes in the subject's brain which were connected to an external camera. William Dobell, a private researcher, was the first scientists to ever come up with a working model of a vision BCI. He implanted his prototype, comprised of a single array BCI with 68 electrodes for the first time into a person named Jerry's visual cortex. Jerry had non-congenital blindness (acquired blindness). Dobell was able to produce the sensation of seeing light in Jerry with his implant in 1978.

In 2002, Dobel was able to further extend his research by implanting more sophisticated implants in Jens Neumann to produce better mapping of the visual cortex with BCI and restore a better sensation of light (Anupama.H.S, N.K.Cauvery, & Lingaraju.G.M, 2012).

Figure 3. 1 Jens Neumann, a man with acquired blindness, being interviewed about his vision BCI on CBS's The Early Show

(Anupama.H.S, N.K.Cauvery, & Lingaraju.G.M, 2012)

Invasive BCI techniques provide the best quality of signal intervention as the electrodes reside directly in the grey matter. However there are disadvantages to using invasive BCI techniques.

Firstly, as this method requires surgical intervention and planting electrodes in the brain, the number of human experiments performed are very few. Mostly this method has been tested on lab rats and other animals and human trials have mostly been limited to patients who have already undergone surgeries and have electrodes implanted for medical purposes. Ethical issues are also a major concern when it comes to invasive technique experimentations which in turn has slowed down the progress in this technique.

As the electrodes reside in the brain, they are subject to the brain attempting to reject the foreign material and a build-up of scar tissue which may result in reduction or loss of signal over time.

In partially-invasive techniques electrodes are implanted inside the skull but outside the grey matter which means a less complex procedure. The signal strength is weaker when compared to invasive BCI, but definitely a stronger signal resolution compared to non-invasive BCI. As the electrodes reside outside the brain, scar tissue build-up is usually less for partially-invasive BCI and thus these tends to last longer that invasive BCI.

Partially-invasive BCI technique is commercially used in Electrocorticography (ECoG) which is similar to non-invasive electroencephalography. In ECoG, a thin plastic pad comprised of electrodes is placed just above the cortex of the brain below the dura matter. Eric Leuthardt and Daniel Moran from Washington University in St Louis were the first scientists to perform human trials with ECoG in 2004. Their research eventually resulted an ECoG implanted boy to play Space Invaders (Anupama.H.S, N.K.Cauvery, & Lingaraju.G.M, 2012).

Both invasive and partially-invasive techniques provide better accuracy and less noise as the electrodes are implanted very close to the brain. However due to surgical procedure involved and other ethical concerns, they are not very commonly used techniques in BCI.

3.1.2 Non-invasive BCI

Non-invasive technologies are much more commonly used in modern BCI signal acquisition. They are recorders of brain signals at a macro level rather than a single neuron level. A typical non-invasive BCI technique involves a subject wearing an electrode fitted cap around their head. Special conductive gels are used to improve conductivity between the brain signals and the electrodes. Compared to invasive and partially-invasive techniques, these are less reliable simply because the electrodes are further away from the brain and the skull acts as a barrier to the BCI machine to pick up the brain’s signals. Still, non-invasive techniques are considered to be the safest of all types.

One of the most commonly used non-invasive BCI technologies is EEG. EEG can be used to measure the potential difference of electrodes attached to an EEG cap worn by the participant.

Other non-invasive technologies that are also hot topics of research these days are Functional Magnetic Resonance Imaging (fMRI), Magnetoencephalography (MEG) and Functional Near-Infrared Systems (fNIR).

3.2 BCI Architecture

A typical BCI architecture consists of the following components as can be witnessed in Figure 3.2:

- A participant who is either a volunteer and has submitted themselves to a BCI research program or someone who is physically or psychologically impaired and requires BCI help to perform normal motor functions.

- This participant is connected to a BCI signal acquisition machine such as an EEG, FMRI etc.

- The raw signals from the acquisition machine needs to be processed by a computer to extract meaningful information from the signal. For example, if the BCI research is about collecting the participant's brain data to move a robotic arm, extra signals, such as, perhaps slight movement of the arm, blinking of eyelids needs to be filtered out.

- This processed signal is then passed through classification algorithms. It is at this step the thought is converted into a machine command.

- In the last step, depending on the requirements an output is provided. If the BCI task is a bio feedback, then some sort of a visual feedback is provided to the participant. The participant can then use this feedback to improve their brain activity.

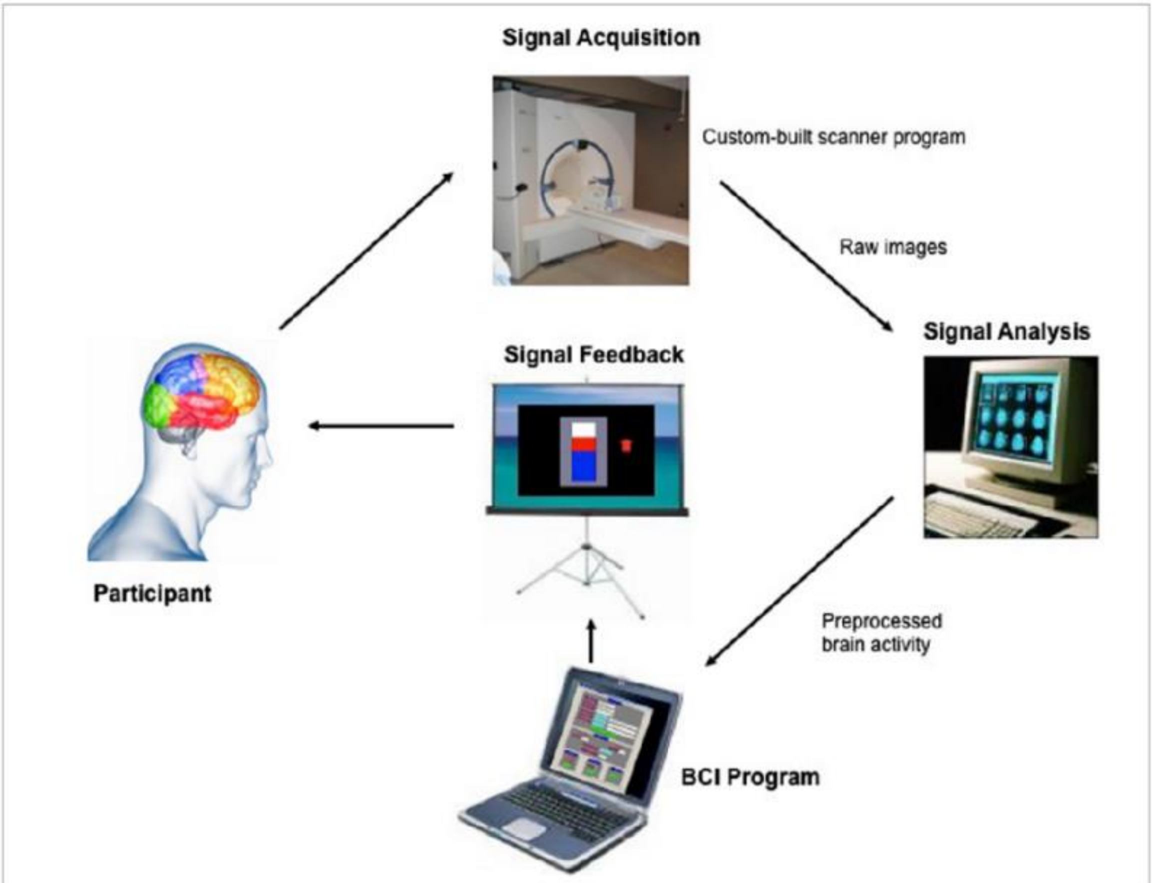


Figure 3. 2 Typical BCI Architecture

3.3 BCI Applications

Some of the BCI applications which are very commonly used these days are listed below.

3.3.1 Electroencephalogram (EEG)

Neurons in the human brain fire at certain frequencies to carry out information from the brain to different parts of the body. An EEG measures this electric current and produces EEG signals. EEG signals consist of waves in the range of 0 – 60 Hz. Different brain activities produces different EEG signals. For example, the EEG signal of a sleeping person will produce signals in the range of less than 4 Hz whereas a brain performing complex mathematical calculations will produce EEG bands of frequency 30 – 50 Hz.

The different EEG frequency bands are listed as below:

Delta Waves – The EEG waves of frequency 0 – 4 Hz are called delta waves. Delta waves are mostly generated by an absence of the firing of neurons and are commonly seen in EEG readings of a person in the state of coma.

Theta Waves – EEG waves falling in the range of 4 – 8 Hz are called theta waves and are generally associated with dreaming or when thinking about old memories. Theta waves can also be produced in the case of an epileptic seizure.

Alpha waves – Ranging from 8 – 13 Hz, this band is associated with a relaxed state of mind.

Sensorimotor rhythms – Also called mu rhythms, these fall in almost the same band of 8 – 12 Hz as that of Alpha waves. However, mu waves and Alpha waves are generated by different portions of the brain and therefore can be easily distinguished from one another.

Beta waves – The EEG band of 13 – 30 Hz is associated with mental alertness, concentration, arousal etc.

Gamma waves – 30 – 50 Hz band in EEG data is associated to complex problem solving.

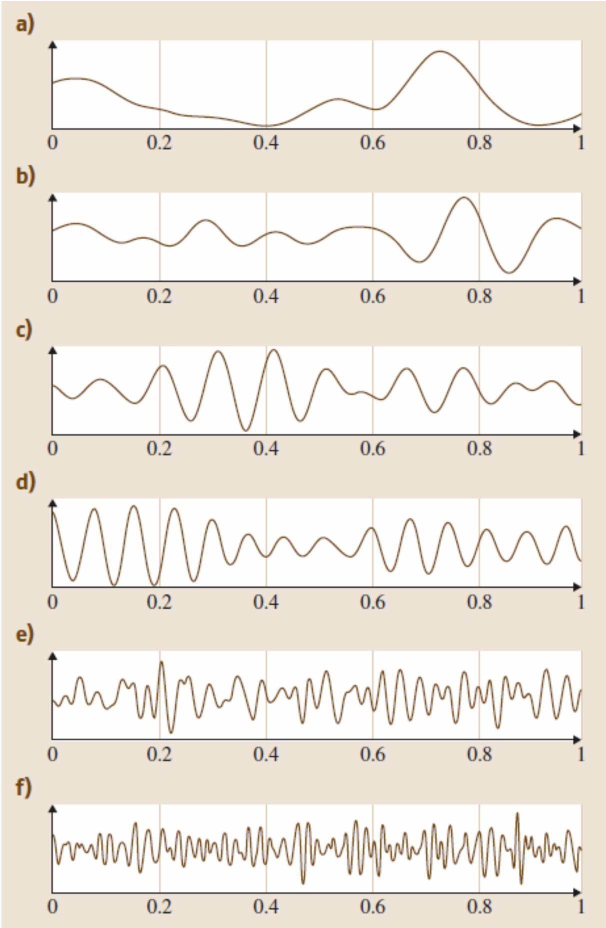


Figure 3. 3 EEG Signals

(a) Delta band (b) Theta band (c) Alpha band (d) mu-rhythm (e) Beta band (f) Gamma band (Kasabov, Springer Handbook of Bio-/Neuroinformatics , 2014)

3.3.2 Functional Magnetic Resonance Imaging (fMRI)

fMRI or functional MRI requires an MRI brain scanner and some specialised sequence of pulses and uses BOLD (**Blood Oxygenation Level Dependent**) in order to map brain activities. This brain mapping technique measures brain activity by monitoring the amount of blood flow in the brain.

Ogawa, Lee, Nayak, and Glynn (1990) were the first to come up with the BOLD measuring technique. They were studying neurons and found that even though neurons are the transmitters of brain signals, they do not have an internal reserve of glucose and oxygen, which are essential for all activities in the human body. They postulated that, in order for the neurons to fire information processing, due to a stimuli being supplied, glucose and oxygen needs to be supplied immediately to that portion of the brain through the bloodstream. This is the basis of fMRI. As soon as neurons need to fire, oxygen and glucose are supplied to the neuron specific to that side of the brain which is going to process the stimuli. As a result, the level of oxyhaemoglobin elicits in the active area of the brain. This allows the MRI scanner to be able to calculate the ratio of oxyhaemoglobin to deoxyhaemoglobin around the active area and this forms the basis of fMRI. The change in level of oxyhaemoglobin is typically found within a radius of 2mm to 3mm around the neural activity.

One major issue with fMRI is the fact that even a slight movement of the human subject inside the magnet of the scanner (even to the order of >0.004 mm) produces distortions and results in blurry images which can lead to incorrect or false conclusions (Ulmer & Jansen, 2013)

Figure 3. 4 FMRI machine

(Hartstein, 2014)

3.3.3 Magnetoencephalogram (MEG)

This is a neuroimaging technique which works by measuring magnetic fields generated by electrical signals in the brain using highly sensitive magnetometers. While EEG signals originate from the intra-cellular electrical activity, MEG signals originate from the inter-cellular postsynaptic currents which flows from the dendrites to the soma of the neurons (Schwartz, Edgar, Gaetz, & Roberts, 2010).

Magnetic fields generated around the neural activities are infinitesimally small (to the order of 10 fT – 1 pT). To get an idea of how small the magnetic field is, if we compare the strength of this field to the earth's magnetic field, it will be smaller than one million times. Hence special technology is used by MEG to measure such small magnetic fields. Magnetic fields generated around the neurons can produce a small electric current in a detection coil and this principal is used in MEG. This detection coil is coupled with a Superconducting Quantum Interference Device (SQUID). This can amplify the small electrical current into a proportional voltage. All coils and sensors in MEG are required to be maintained at a superconducting temperature so that it can detect the infinitesimally small electric field around the neurons. This is achieved by surrounding them in liquid helium and enclosing the entire apparatus in an insulated vacuum flask (Schwartz, Edgar, Gaetz, & Roberts, 2010).

Figure 3. 5 Magnetoencephalography machine

(MEG, 2011)

MEG is extensively used for detecting interictal discharges in the brain of epileptic patients by measuring electromagnetic activities in their brain. Data collected can be used to determine whether the interictal discharges are focal or multifocal and thus help doctors predict whether the patient will benefit from surgery or not (Knowlton, 2006).

MEG is also extensively used during surgery for the treatment of seizures and helps assist with the placement of electrodes by measuring the intracranial activity of the patient's brain.

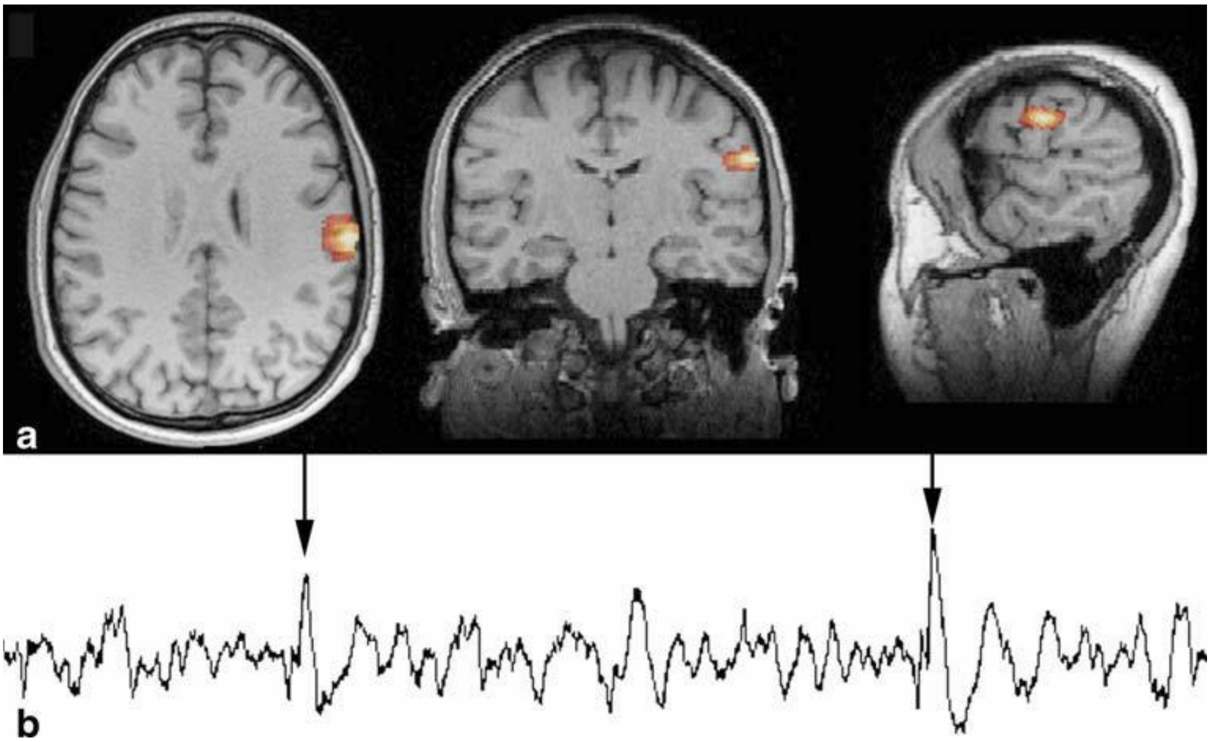


Figure 3. 6 MEG data

a) MEG data of an epileptic patient superimposed on volumetric T1-weighted MP-RAGE gradient echo

b) 10 Sec time activity curve

(Schwartz, Edgar, Gaetz, & Roberts, 2010)

3.3.4 Functional Near-Infrared Sensing (fNIR)

fNIR is a comparatively new technology in the field of BCI. It essentially measures the same signals as fMRI, but it requires less hardware and can be implemented at a much lower cost. Compared to EEG, fNIR is more robust against electrical noise. Requiring a lower cost and less hardware, fNIR presents as a very good choice for

BCI implementation. fNIR works on the principles of optical interaction properties between near-infrared (NIR) lights and oxy-haemoglobin and deoxy-haemoglobin of the tissue. fNIR is a non-invasive BCI technology.

All living tissues have concentrations of oxy-haemoglobin and deoxy-haemoglobin and each of these has their distinctive spectra of NIR light absorption. fNIR exploits this property of light to calculate the concentration of oxy and deoxy-haemoglobin in the brain tissues.

Typically, for a continuous NIR sensing, time modulation is performed on multiple NIR light wavelengths and the intensity of the output is measured. The attenuation changes for each wavelength are measured by the intensity of the output light after the IR has passed through the tissues. A modified Beer-Lambert equation is used to convert the intensity into oxy- and deoxy-haemoglobin concentrations (Roche-Labarbe, et al., 2014).

A great advantage of fNIR is that it can monitor even the slightest fluctuation in the oxygenation of tissues (Roche-Labarbe, et al., 2014).

However, since this is comparatively a new technology, its full possibilities and its limitations are yet to be explored.

Figure 3. 7 fNIR

(BIOPAC, 2008)

3.3.5 Neuro-feedback

The entire topic of whether counselling is therapeutically effective or not is a very controversial one. Countless books have been written on the very topic and many researches has been done to quantify how much a client feels better after being counselled. Through the help of neuro-therapy and neuro-feedback, this is now possible.

Neuro-feedback is a special branch of BCI which allows a subject to self-regulate their brain signals with the aid of some sort of an audio-visual response from a BCI device. In a typical neuro-feedback setup, a subject's brain data is collected by an EEG device. The EEG is connected to a computer running BCI software. The computer provides some sort of visual feedback. The subject uses this feedback to regulate their brain signals. The goal is to achieve a benchmark and the subject can keep repeating the exercise until the goal is reached.

A neuro-therapist can use neuro-feedback as a treatment for treating a patient for psychological disorders. The therapist would select a brain signal band to start with. For example, in a child subject with ADHD, there are very few Beta waves. In such cases, the therapist would concentrate on the 12 – 15 Hz EEG waves. The whole principle lies around the fact that a human brain basically likes challenges. So the subject would try to hit the beta waves node in the therapy and once the target has been hit something exciting will happen, such as moving up to the next level of a game or something happening on the computer screen such as a frog jumping over a fence etc. In some cases there can also be excess of certain brain waves. Again, if we take the example of the same child with ADHD, the child might have an excess of theta and high beta nodes. The therapist would use some kind of an inhibitor to lower those two waves. He might use some sort of a buzzing noise which can act as a voluntary or involuntary reminder to the brain that there is excess of a certain wave and the brain should decrease the amount of such waves. As the brain wants the sound to go away and also wants to go to the next level of the game, the waves would be lowered automatically. Sometimes the therapist might have to resort to relaxation techniques as well.

There has been many cases where neuro-feedback has been implemented successfully to improve brain signals. Two examples of such an implementation has been outlined below.

<u>Improving visual perception through neuro-feedback</u> (Scharnowski et al., 2012) – Perception is the brains capability to learn and understand spontaneous activities within its surroundings which trigger activity in the sensory cortices evoked by stimulus. Based on this principal, this case study tries to prove that training with ongoing spontaneous activity should be sufficient to improve visual perception in a human subject.

16 healthy volunteers participated in the experiment. Training was provided to these participants to learn and control the activation of their local visual cortex by providing them with neuro-feedback. Activation and firing of neurons in the brain of these participants was measured using a 3 T Siemens Allegra MR scanner. Pre-processing of the data was done using Turbo-Brain Voyager and some customised software made for this experiment. Neuro-feedback was provided to these participants by providing them with continuous feedback of their brain activity of these participants. The neuro-feedback comprised a thermometer and a temperature reading of the activity of the visual cortex of the brain. As with any other neuro-feedback experiment, it had a target activation level represented by a dashed line. The dashed line could either be high (the up-regulation condition) or low (the baseline condition).

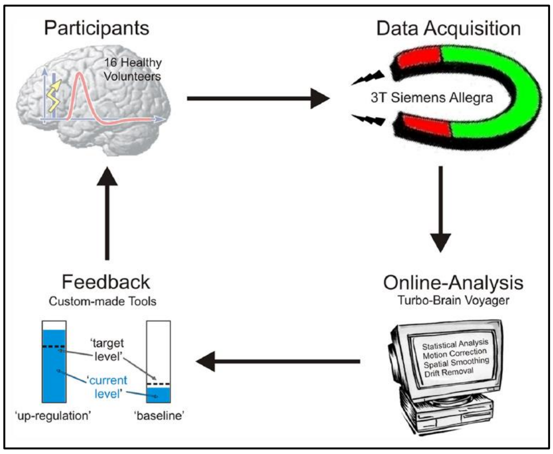


Figure 3. 8 Structure of the experiment for improving visual perception through neuro-feedback

(Scharnowski, Hutton, Josephs, Weiskopf, & Rees, 2012)

During the first session of the neuro-feedback experiment, a very high resolution scan of the visual target was completed (Figure 3.9). A written instruction was given to each participant and they underwent several training sessions spread over several days. Each day the participant underwent of approximate two feedbacks each of which lasted for 8.3 minutes. The neuro-feedback training was comprised of a 38 sec baseline blocks mixed with 38 sec up-regulation blocks. After the training was over the participants were asked to achieve self-regulation without any feedback. The behavioural training was also spread over across several days in separate scanning sessions. To test how much self-regulation training effected their behaviour, they were asked to identify stimuli which could be classified as a near threshold stimulus across the different visual field positions. At the same time, they were also asked to control their brain activity in the target region of the cortex.

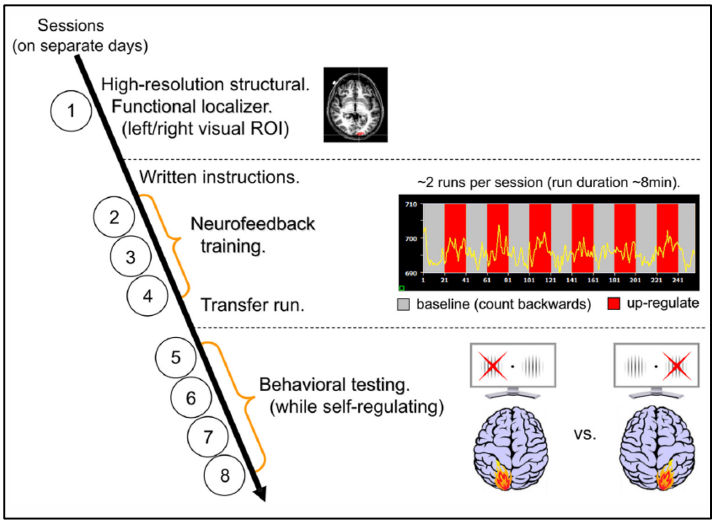


Figure 3. 9 Procedure of the experiment for improving visual perception through neuro-feedback (Scharnowski, Hutton, Josephs, Weiskopf, & Rees, 2012)

This experiment was successfully able to prove the effect of neuro-feedback training on self-regulation. The learners who were trained with neuro-feedback were able to control their visual cortex whereas the non-learners and control group were not able to do so.

Some of the data obtained from the outcome of the experiment is demonstrated in the below graphs (Figure 3.10 A and B). As apparent from the graphs, the visual ROI mean signal change percentile increases more for the next set of training run for subjects who received neuro-feedback training as opposed to the ones who did not receive training.

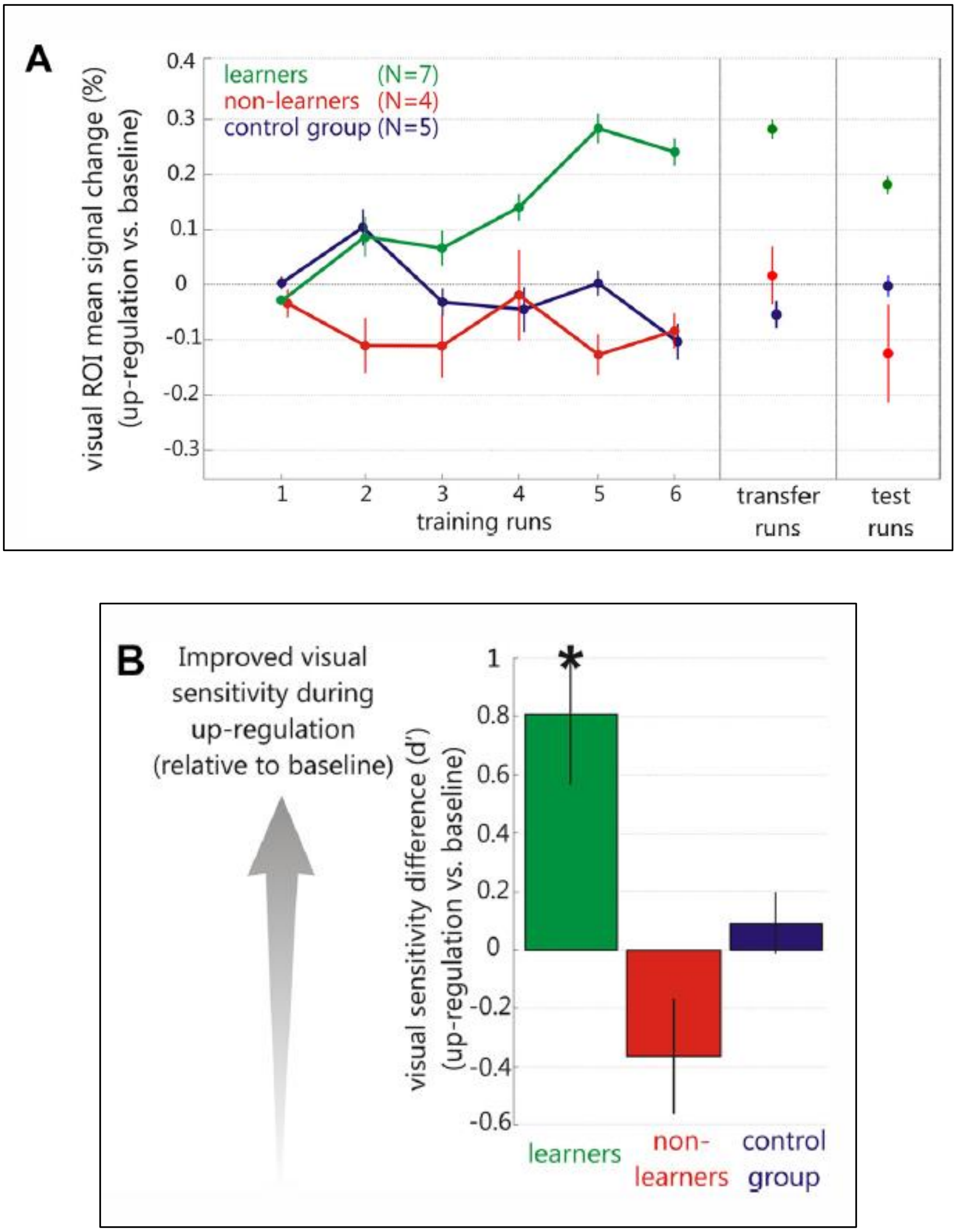


Figure 3. 10 Learner vs. non learner and control group in the experiment for improving visual perception through neuro-feedback

(Scharnowski, Hutton, Josephs, Weiskopf, & Rees, 2012)

Neuro-feedback using functional spectroscopy (Hinds, et al., 2014)

This research was undertaken to demonstrate the concept of providing neuro-feedback in real time by measuring the Blood Oxygenated Level dependent (BOLD) signals of human subjects. Most common methods of neuro-feedback uses signals generated by EEG or fMRI which, although providing signals generated by the entire surface of the brain, lack speed and accuracy. This research used a system called Multivoxel Functional Spectroscopy (MVFS) for measuring BOLD signals in real time. MVFS, simply put, is comprised of the following

- A single voxel magnetic resonance spectroscopy pulse sequence. This was used to acquire a different volume of interest (VOI) for each repetition of the trial and also compensate for subjects head motions.

- A component which would reconstruct BOLD signals. This was used to compute the T2* decay rate which determines the strength of the BOLD signal.

- An estimator for neural activation. This would filter out noise signals. It would also scale the neural activation based on the amount of noise.

- The neuro-feedback system. This would present stimuli for neural activation.

In the trials performed for this experiment, the human subjects were required to perform a repeated task of tapping with a finger while a neuro-feedback stimulus was presented in front of them that displayed the current estimate of their BOLD signals.

To start with, a scan of the subject's brain was undertaken to localise the ROI of the cortex involved when the subject moves their fingers. The tapping task was done in five blocks, each block consisted of sixteen seconds of tapping and then resting for ten seconds. On immediate completion of the scanning task, the scanned data was sent to a different computer for fMRI analysis and analysis was done using FSL within five minutes. A strong ROI was selected from this exercise for further experiment (Figure 3.11).

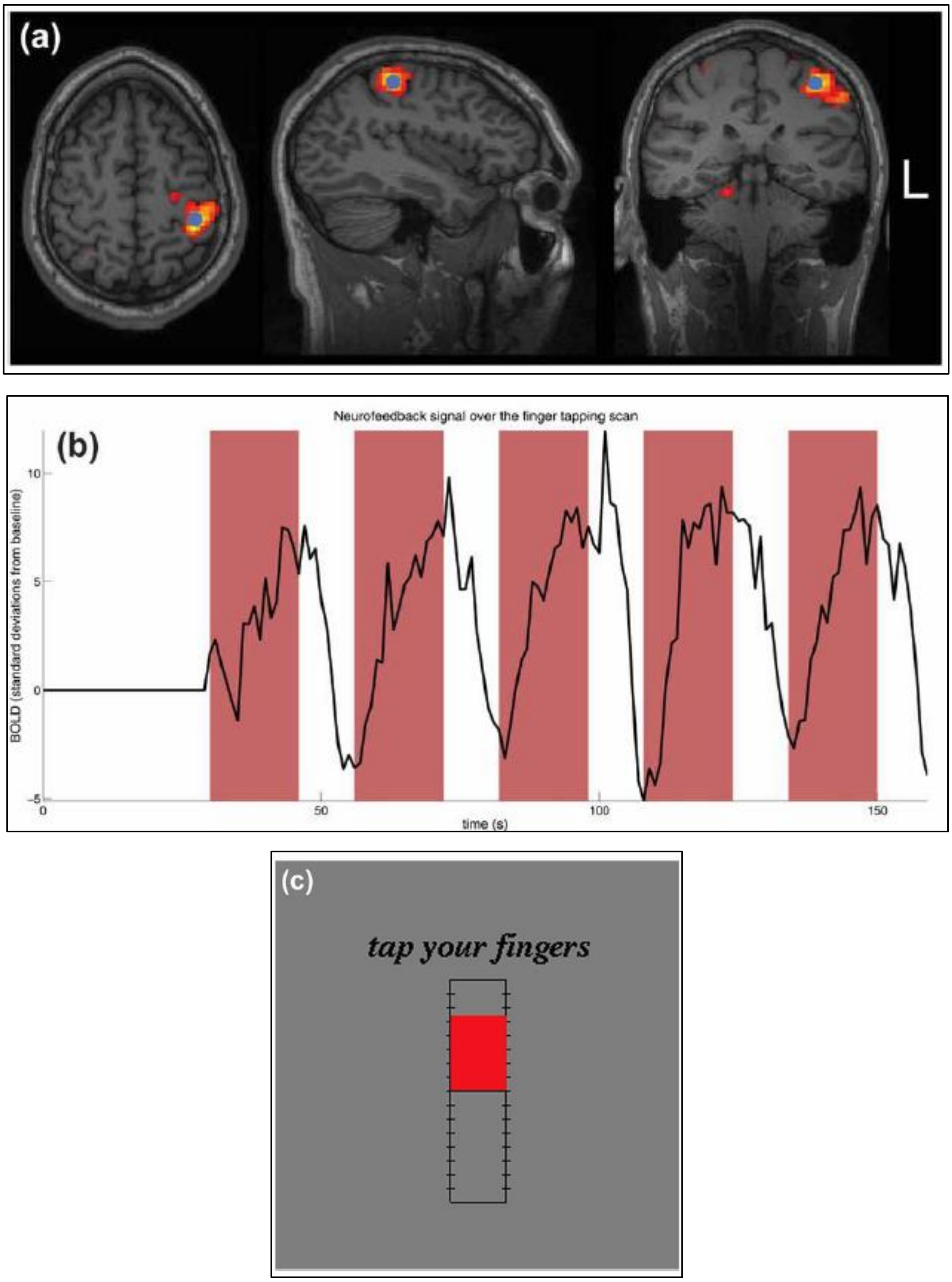


Figure 3. 11 Case Study 2 (a) Region of Interest (b) BOLD signals (c) neuro-feedback

(Hinds, et al., 2014)

After determining the ROI, the MVFS scan was performed while providing neuro-feedback. The subject was asked to perform the same finger tapping task such as in the localizer step. Thirty of such measurements were collected and stored. These measurements were then used for constructing the BOLD component and T2* was determined from that.

The T2* was then sent to the stimulus component. It then computed the standard deviation of the GLM residuals. This was used to provide feedback in the form of a thermometer. The thermometer showed the recent changes in the subject's BOLD signals that had been processed by the entire setup.

3.4 Challenges with BCI

Brain-computer interfacing is presently a field of extensive research. A lot has happened since the 60s when Berger discovered a way to measure electrical brain signals. But BCI has a long path ahead of it before it reaches its goal where commercially viable BCI products are made available to the public.

Some of the known BCI issues are as below:

- ❖ Cumbersome setup – Most of the current BCI solutions require a very long and cumbersome setup. Most of these instruments requires a controlled clinical environment to operate. An assumption is also made that the subject will maintain the proper posture and attention for the optimal functioning of the BCI device. But surprisingly very often are these controlled conditions met.

- ❖ Variability – Most of the current BCI devices lacks consistency in providing the same result every time. This can be caused due to change in the environment, lack of concentration, the subject's personal bias and many other reasons. Without consistency, BCI devices or any device for that matter fails to be a viable solution.

- ❖ Non-invasive vs invasive BCI – Invasive BCI, in theory is supposed to provide better results as the electrodes are closer to the brain as compared to non-invasive BCI, where electrodes are placed outside the brain. Invasive techniques require surgical procedures which can be life threatening if a mistake is made by the surgeon. Not many subjects would be willing to have holes drilled in their skull for human trials. This makes it very hard for researchers to perform invasive experimentation on human.

3.5 Improving BCI functionality

The field of BCI requires research based approach. Many factors determine whether a BCI system will work successfully or not. There are a few steps that one can take in order to improve performance of a BCI application.

- ❖ <u>Improve neuro-imaging sensors</u> – Signal acquisition is the step in which brain data is made available to the BCI researchers via the help of sensors. Improving the sensor to make it more sensitive in picking up brain signals will give better BCI results.

- ❖ <u>Use better classification algorithm</u> – Classification/regression forms the very heart of a BCI application. BCI researchers need to carefully select classification algorithms based on the BCI's requirements. Better classification algorithms will result in better BCI performance (Blankertz & Vidaurre, 2009).

- ❖ <u>Implement better error correction techniques</u> – BCI systems are highly prone to errors caused by hear beating, blinking of eye, etc. Sensors like EEG are able to pick up these errors. Eliminating or reducing errors will provide better classification and improve overall performance of the BCI application.

- ❖ <u>Use easier brain signals for BCI application</u> (Furtscheller, et al., 2010)– Working with brain signals which are easily picked up by sensors will ensure better functioning of the BCI application.

3.6 Conclusion

In this chapter, the various types of BCI applications such as EEG, fMRI and a review on their functioning has been outlined. Also some of the merits of using non-invasive BCI techniques over BCI techniques and vice versa has been outlined. It has also attempted to prove neuro-feedback as a viable BCI technique for treating neuro-psychological disorders and its use as a method for improving brain signal patterns. And last but not the least, this chapter concludes by outlining some BCI challenges and ways to overcome the same.

So far we have seen that signal acquisition and their classification forms the majority of BCI functionality. In the next chapter, some of the common methods of analysing brain signal patters has been introduced.

Chapter IV – Signal processing and classification methods related to BCI

Brain Computer Interfacing (BCI) is basically the process of converting human thoughts into machine understandable commands where thought processes are transmitted from the brain to the machine without the movement of any peripheral muscles. The field of BCI exploits the fact that neurons in the brain fire at the onset of any thought process which generates an electrical charge. The measurement of these electrical signals and the classifying them into common groups which are in turn associated with an activity is the main idea behind BCI.

Having discussed some of the different types of BCI applications in chapter three, this chapter now focusses on reviewing some of the common signal analysis and classification techniques in BCI. Starting with how the classification technique came about with Fourier's transform algorithm, this chapter introduces some ERP components used in BCI and then illustrates some common machine learning techniques. This chapter then provides introduction to some of the common BCI classification methods.

4.1 Introduction to signal processing

During the traitorous time when Napoleon Bonaparte was out in his quest to dominate the entire civilized world, there lived a brilliant French mathematician, Jean Baptiste Joseph Fourier. Back when he was appointed to the French emperor himself and during his travel to Egypt as his science advisor, Fourier was working on a mathematical concept to find out a way to decompose signals of periodic nature into sums of cosine and sine functions. This work is what we know today as the Fourier series. Although the Fourier series is often applied to heat conduction principals, it is also applied to EEG data collected from the Human Brain. When considering EEG data by applying Fourier's series we can represent the electrodes in the time domain and we can also represent them as a decomposition of sine and cosine functions in the frequency domain of a sinusoidal curve (Freeman & Quiroga, 2013).

This principle is used quite extensively in the field of BCI for analysis of EEG signals with the introduction of the Fourier Transform algorithm by Cooley and Tukey in 1965.

Some of the different types of Fourier transform in the form of equations are discussed below.

4.1.1 The Continuous Fourier Transform

For a function $x(t)$, the Continuous Fourier Transform can be represented as the below equation.

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t}\, dt \tag{4.1}$$

The equation 4.1 represents the continuous Fourier Transform where $e^{-j\omega t} = \cos\omega t - j\sin\omega t$ is complex exponential, $\omega = 2\pi$ is the angular frequency.

The inverse of the above equation also holds true and can be represented in the form of equation 4.2and it gives back the original signal $x(t)$ as an integral of sine and cosine of different frequencies.

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} X(\omega)e^{j\omega t}\, d\omega \tag{4.2}$$

Overall, Fourier Transform can be represented as a relation between $x(t)$ and $e^{-j\omega t}$ which is the complex sinusoidal function as below:

$$X(t) = \langle x(\omega), e^{-j\omega t}\rangle \tag{4.3}$$

4.1.2 The Discrete Fourier Transform (DFT)

Consider a discrete signal $x[n] = 1, \ldots\ldots, N$ derived from $x(t)$ which represents a continuous signal. $x(t)$ is obtained by sampling with a sampling frequency $f_s = \frac{1}{\Delta t}$

The length of the signal is represented as $\mathrm{T} = \mathrm{N} * \Delta t$

Then the Discrete Fourier Transform can be defined as the equation below:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \tag{4.4}$$

In the above equation 4.4, K = 0, 1…N-1, X[k] = $X_R\,[K] + jX_I[K] = |X[K]|e^{j\emptyset}$ which is the Fourier coefficient.

4.1.3 Fast Fourier Transform

The computation of the Discrete Fourier Transform is resource expensive. For every discrete frequency k, the DFT needs to compute the multiplication which has complex exponentials. For a large number of N, the computation can therefore take long time. This is where Fast Fourier Transform comes into picture and dramatically reduces the computational overhead (Freeman & Quiroga, 2013). The basic idea of FFT is to reduce redundancies in the DFT. For example the complex exponentials part of DFT has a periodic nature. So the permutation of n,k can give the same results for number of Ns. FFT takes that into account and thus makes the computation many times faster.

4.2 Event Related Potentials (ERP)

Every stimuli presented to the brain causes a neuron to fire as a response. BCI scientists exploits this phenomenon to study the activation of different portions of the brain.

ERP is the technique in which EEG signals from a brain are recorded while at the same time providing some sort of a stimulus in order to activate particular brain signals. This method can precisely measure EEG signals associated with certain brain thoughts to the nearest millisecond (Landa, Krpoun, Kolarova, & Kasparek, 2014). Through ERP, tiny voltage fluctuations in the brain can be measured.

Some electroencephalographers strongly believe that measurement of event related potentials are as much complex a task as are measurements of EEGs. Therefore simple 16 channel EEGs are not sufficient for the job.

The most common way of extracting ERPs from EEG signals is by taking a multiple number of readings of the same EEG epochs time locked to the same event and then filtering and averaging of these signals (Landa, Krpoun, Kolarova, & Kasparek, 2014).

There are two type of ERPs, the early waves and the cognitive or endogenous waves. The early waves reach a peak within 100-200 milliseconds of receiving the stimulus. They are also referred to as sensory or exogenous because of their dependence on an external stimulus. The late waves or cognitive or endogenous waves are associated with changes in the mental state of the subject while their brain is trying to interpret and understand the meaning of the stimulus.

Further based on the polarity of the ERP (negative or positive voltage), location of the scalp, timing etc, they are further classified into various components as described below.

4.2.1 P300 Waves

The P300 wave is the most commonly studied ERP. P stands for Positive latency and the number 300 is the latency of 300 milliseconds. This is the third positive wave and therefore is also called the P3 waves (Picton, 1992). P300 waves are typically generated in an "oddball" paradigm, when a subject’s brain comes across an occasional "target" stimulus in the midst of regular standard stimuli (Picton, 1992). The subject is presented with regular stimuli and is asked to pay attention and identify a target stimulus and perform a motor function, for example pressing of a button as soon as the target stimulus is presented. Such actions will generate P300 ERP waves (Picton, 1992).

P300 recordings are greatly affected by the position of the electrode on the scalp. In order to successfully record P300 waves at least the three positions on the scalp should be monitored with electrodes. These locations are F_z, C_z and P_z. P300 waves can most accurately be recorded from the midline centro-parietal region of the scalp (Figure 4.1).

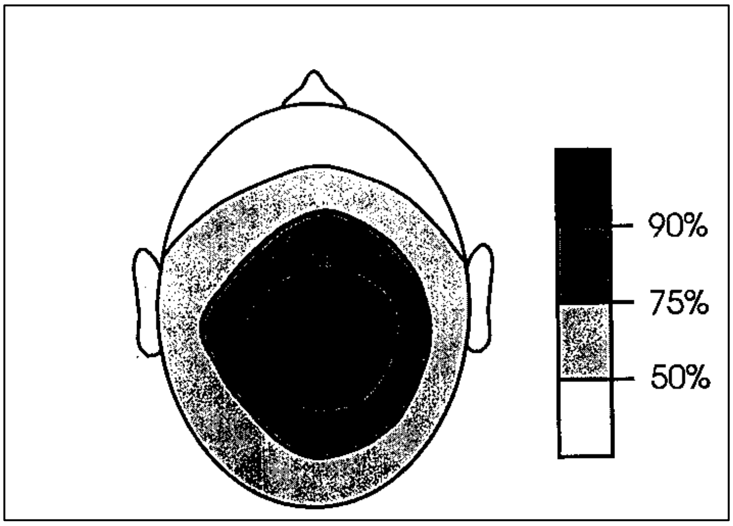


Figure 4. 1 The scalp distribution of P300 wave

(Picton, 1992)

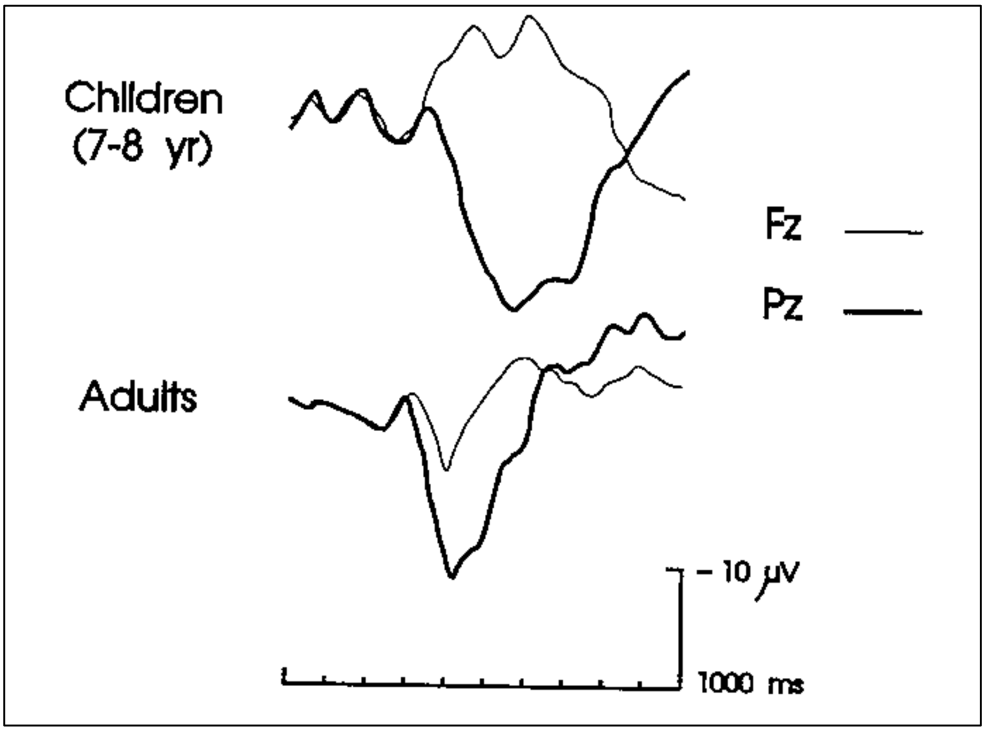


Figure 4. 2 P300 waves in children vs adults

(Picton, 1992)

Figure 4.2 outlines a comparative amplitude difference between P300 waves in adults and children.

P300 waves are usually of amplitude 10 µV. A typical P300 trial involves recording the EEGs of alert subjects with their eyes open for 30 to 100 trials and then averaging them out.

P300 is typically observed as a peak amplitude in relation to the baseline which is the pre-stimulus and a peak latency in relation to the onset of the stimulus. There are certain wiggles which appear superimposed on P300 signals. These can be attenuated by passing them through a low-pass filter at 3-5 Hz (Picton, 1992).

There are certain issues that scientists have to address while recording P300 waves. P300 waves are usually effected a lot by potentials created by eye movements or blinks.

One way of tackling this issue is by eliminating any trials where the subject has blinked their eyes. Typically the subject would have peri-ocular electrodes on the periphery of their eye lids which is connected to an electro-oculogram (EOG). While doing the trial for recording P300 waves, a potential on the EOG is used as a criterion to remove any trials (for example removing all trials where the EOG has recorded a ±µV). This approach has an obvious drawback as it rejects trials thereby reducing the effectiveness of the experiment and which in turn results in loss of data. Also in the worst possible scenarios there is a likelihood that all trials could be rejected because of ocular interference. To overcome this issue subjects could be asked not to blink their eyes. Which brings us to another issue. The subject now has to divide their attention between the P300 generation task and also the task of not blinking their eyes. This can harm the outcome of P300 wave recording as well (Picton, 1992).

The second approach would be to take the EEG reading of each trial and subtract the ocular component of the reading.

There is another issue that scientists come across while recording P300 waves. After the brain receives the stimulus and between the times elapsed before a P300 wave is generated, several other areas of the brain also get activated. Some of these areas generate electrical fields around their area in the scalp. Hence these electrical fields gets intermingled with the measurement of the P300 waves. Therefore a successful recording of P300 waves thus involve removing these disturbances from the P300 recordings.

P300 wave measurements also plays a huge rule in psychology. Psychologists have studied P300 waves in patients and found that the amplitude of the P300 waves varies directly in proportion to the amount of attention the patient pays to the stimuli used to generate the P300 (Picton, 1992). Thus a subject is able to regulate their own P300 wave by paying more attention to the stimuli. For this reason P300 waves are studied for the detection of pathological disorders such as Dementia, depression, schizophrenia etc.

Studies have also shown that P300 waves are stronger when the subject is presented with improbable stimuli. To demonstrate this phenomenon, Duncan-Johnson and Donchin in 1977 performed a series of experiments on subjects listening to randomized sequences of two different tones. They deduced from their experiments that the probability of each tone and the amplitude of the P300 waves were inversely proportional. But this does not hold true for all cases and scientists have also argued of the possibility that the amplitude of P300 waves generated in response to a particular category of stimulus can vary as time progresses. This hypothesis is used by scientists to explain the variance of P300 waves with probability of the stimulus (Figure 4.3).

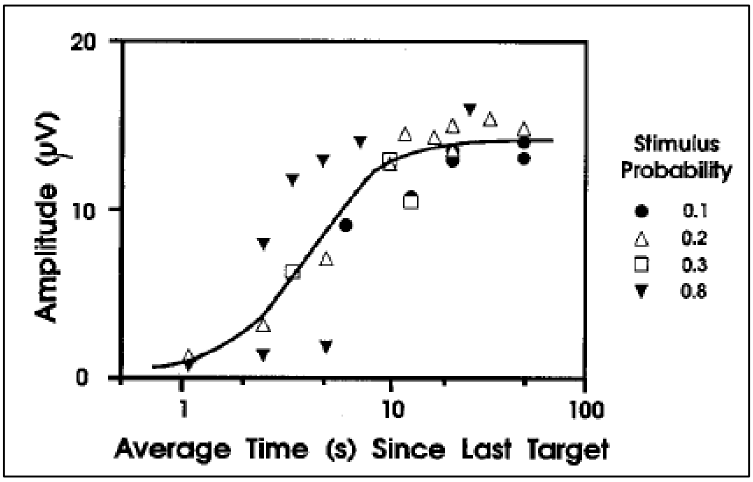


Figure 4. 3 Amplitude of P300 for stimulus probability against time

(Picton, 1992)

Few other factors one also needs to consider while studying P300 waves, are the difficulty of the task in differentiating the target stimulus from a standard set of stimuli, the confidence of the subject in differentiating these targets and the uncertainty of the target stimuli. When the task of discrimination of the stimuli becomes difficult, the amplitude of the P300 wave drops while its latency becomes longer. This can also be a double edged sword as the amplitude also drops if the task is too easy (Figure 4.4). The reason being that if the job is too easy, the mind may wander off to other thoughts not related to the stimuli used for generating the wave.

Where the subject has more confidence in discriminating the stimuli, there is also a decrease in the amplitude of P300 (Picton, 1992).

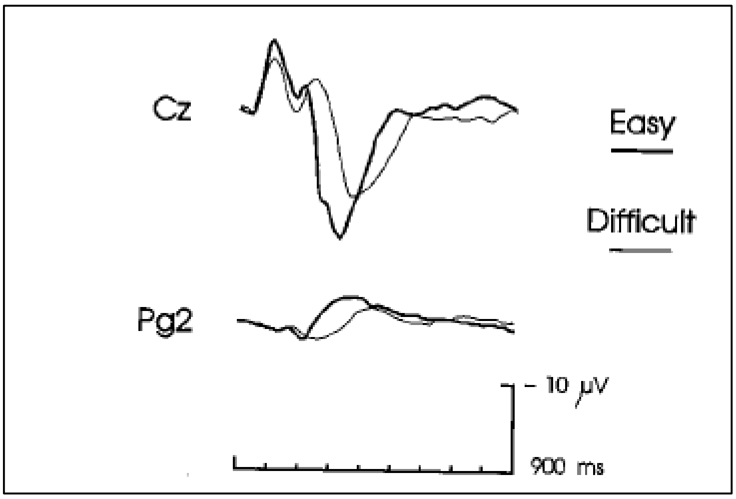


Figure 4. 4 P300 waves of easy v/s difficult tasks

(Picton, 1992)

4.2.2 Contingent Negative Variation (CNV)

CNV is associated with the change in potential of mainly the brain's frontal cortex of the brain where the surface of the brain becomes slightly electro negative of about 20 μV (Lotte, Arnaldi, Lecuyer, Lamarche, & Arnaldil, 2007). There is a small negative deviation in the EEG that occurs during the small interval between the warning stimulus and the imperative stimulus that is associated with the motor activity (Figure 4.5). CNV is an endogenous potential generally associated with cognitive actions such as attention, sensation, stress etc. Averaging different EEG components associated with the time segment of 1 to 1.5 seconds prior to the warning stimulus and about 1 to 2 seconds after the imperative stimulus (Bares, 2011) can provide us with a CNV.

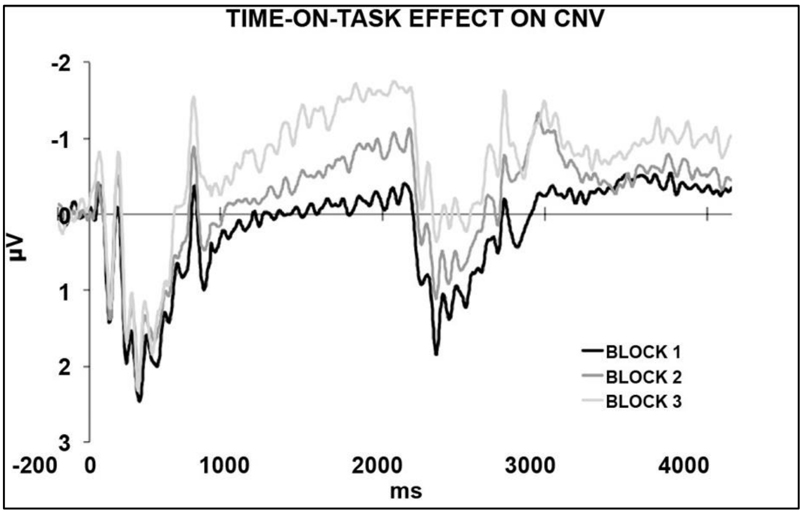


Figure 4. 5 Time on task effect on CNV

(Mento, 2013)

4.2.3 Mismatch Negativity (MMN)

MMN is triggered by the brain when it is trying to violate a rule typically associated with some sort of sequential auditory stimuli. MMNs are believed to be evoked even when the subject is not paying attention (subconsciously listening to some auditory stimulus). For example, MMN will be generated if a person is subjected to a repetitive sequence of short soft tones and suddenly a long loud tone is introduced in the sequence. Thus MNN demonstrates the brains natural capacity to discriminate between sensory learning and perceptual accuracy (Garrido, Kilner, Stephan, & Friston, 2009).

Therefore it is natural then that scientists have studied MMN extensively in healthy subjects with normal cognitive functioning as well as in patients with certain psychological disorders in order to learn about the auditory memory of the brain and attention of the subject (Garrido, Kilner, Stephan, & Friston, 2009).

MMN were first discovered by Näätänen et al., in1978 in a typical oddball paradigm. While studying the ERP, they observed that when a repeating sound (which they called as standard sound) was randomly replaced with a different sound (the deviant sound), the ERP response was elicited. By subtracting the ERP response to the standard tone from the ERP response to the elicited, deviant tone, they observed a negative waveform near the fronto-central scalp, peaking between 100 and 200 milliseconds from the onset of the deviation. Thus the deviant wave minus the standard wave is called the Mismatch Negativity (Figure 4.6).

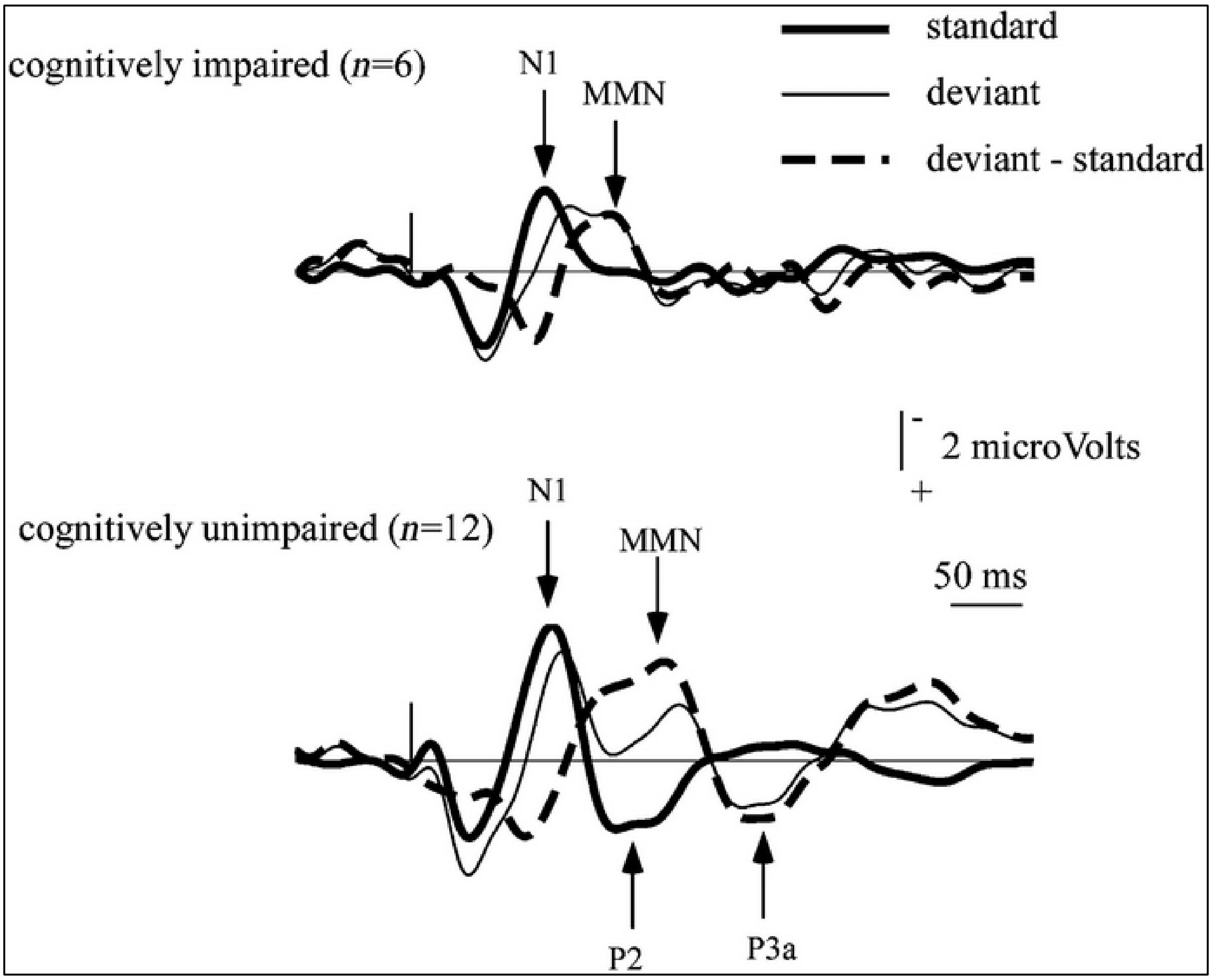


Figure 4. 6 MMN of cognitively impaired vs cognitively unimpaired subjects

(Kullmann, 2011)

4.2.4 Bereitschaftspotential (BP)

This is also called readiness potential (RP) or pre-motor potential and is typically associated with a slow wave generated in anticipation of a motor action. Generally absence or deficit of BP can potentially mean insufficient capacity of the brain to prepare for a motor response.

BP can be separated into an early BP and a late BP. The early BP is usually associated with the activities of the Supplementary Motor Area (SMA) and is not concerned with the actual body part which is doing the movement. The late BP is generated in the opposite hemisphere of the side of the body which performs the movement in the primary motor cortices (Fumuro, et al., 2010).

Other ERP components which have proved significant to some researchers include: P50 wave, N1 (or N100 wave), P2 (or P200) wave, N2 (or N200 wave), N300, N400, P600 waves etc.

Study of ERPs are highly useful in detecting neurological disorders of the brain. For example, it has been found that patients suffering from Parkinson's disease exhibit less early Bereitschaftspotential (Fumuro, et al., 2010). Further findings by other scientists also indicate that patients suffering from mild Parkinson's produce large amplitudes of P3 waves and can be used for early detection of Parkinson's. ERPs are also associated with Alzheimer's disease. Lai, et al., 2010 suggested that P300 latency is highly prolonged in patients with Alzheimer's exhibiting slightly cognitive dysfunction and thus P300 ERPs can potentially be used for determining the severity of the stage of Alzheimer's disease. Further examples of ERPs association with neurological disorders include prolonged P300 latencies and longer MMN duration in patients suffering from epilepsy, reduced amplitude and longer latency of P300 waves in patients suffering from schizophrenia (Asato, et al., 1999).

4.3 Introduction to machine learning

Computers are not very intelligent. They can be very fast when it comes to computing a mathematical problem, but they cannot think for themselves. How then can we program 'Robot' or 'Artificial Intelligence'? That is when machine learning comes into play. Machine learning uses algorithms to create pattern recognition techniques to learn and predict. A detailed literature review on machine learning is outside the scope of this thesis, therefore only the two methods that are closely related to this study will be briefly outlined, supervised and unsupervised Learning.

4.3.1 Supervised Learning

"The learning task is to obtain a predictive model or an approximation function ˆ*f* able to predict ŷ for future observations of the independent variables x" (Mitchell, 1997).

A typical supervised learning technique consists of training data and novel data. Each data points are also called input vectors, (x_i), are generally paired with a corresponding label (y_i), there are m number of such pairs $(i=1.....m)$.

For example if we are trying to predict whether a person's tumour is malignant or benign, in a supervised learning technique, there would be a training data consisting of input vectors x_i paired with labels y_i and supervised learning would view the input vector points in either side of a hyperplane where y=+1 would be malignant and y=-1 would be benign (Figure 4.7).

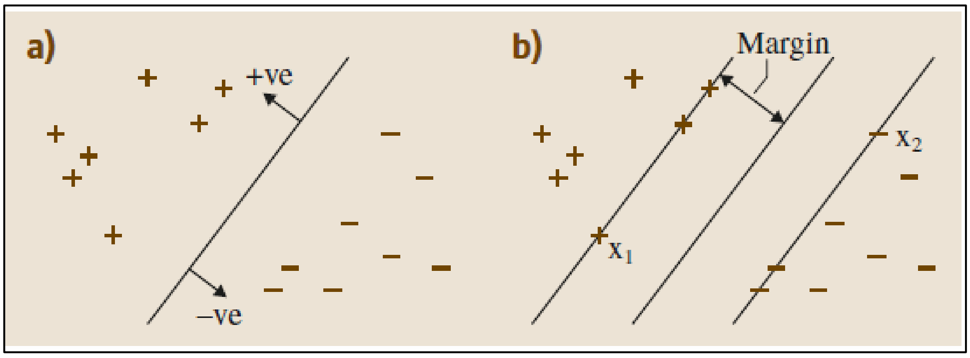


Figure 4. 7 Hyperplane showing benign and malignant tumour

(Kasabov, Springer Handbook of Bio-/Neuroinformatics , 2014)

In Figure 4.7 a) the perpendicular line corresponds to the function $w.x + b = 0$ which separates the points $y=+1$ $(w.x + b>=0)$ on one side and $y=-1$ $(w.x + b<=0)$ on the other side. Figure 4.7 b) depicts the margin which is the area between the separating hyperplane and the hyperplane drawn between the two closest points.

When discussing supervised learning, one has to take into consideration, the generalization error, i.e. the error in prediction of a class when a classifier is applied to a dataset. This generalisation error has a couple of properties:

- Maximizing the margin (the minimum amount of distance between the two closest points of the hyperplane) results in the reduction of the error bound.
- The dimension of space does not dictate the error bound.

If we consider a task for the classification of data points x_i $(i = 1,2,3,\ldots\ldots\ldots m)$, the corresponding labels are $y_i = +1$ and $y_i = -1$ then the decision function can be represented by the below equation:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{sign}(\boldsymbol{w} \,.\, \boldsymbol{x} + \boldsymbol{b}) \tag{4.5}$$

So, from the above equation it can deduced that the learning of data is achieved if $y_i(w \,.\, x_i + b) > 0 \; \nabla \; i$

Therefore a scale can be defined for the two closest points on both sides of the hyperplane by $w \,.\, x + b = 1$ and $w \,.\, x + b = -1$

The hyperplane passing through the above points are called canonical hyperplanes and the region which lies between the canonical hyperplanes is called the margin band.

The above equations can be reduced to the Lagrange Function to demonstrate optimization problem. The Lagrange function is comprised of the constraints m multiplied with the Lagrange multipliers α_i and can be summarised by the below equation:

$$L(w,b) = \frac{1}{2}\,(w\,.\,w) - \sum_{i=1}^{m} \alpha_i\,[y_i(w.\,x_i + \,b) - \,1] \tag{4.6}$$

4.3.2 Unsupervised Learning

In contrast to supervised learning, unsupervised learning on the other hand does not have any labels attached to the input vectors. Classification for unsupervised learning is usually achieved by hierarchical cluster analysis techniques.

4.3.2 (A) Hierarchical Cluster Analysis (HCA)

This is the most commonly used technique for unsupervised learning. The approach of this classification technique is usually either done by divisive methods, where the classifier has to go through the sample data for *m* samples and then keep dividing clusters of datasets into smaller groups or it is done by agglomerative methods where the *m* number of samples are fused to form a large number of clusters. A dendrogram is generally used to represent the unsupervised learning fusion or division methods.

In the implementation of the agglomerative method, single clusters of data *m* corresponding to each data point are considered. For every iteration of the clustering, fusion of the *m* samples that are closest to each other occur. How are the proximity of data points determined? It can be determined by the equation below called the squared distance.

$$D\left(x_i, x_j\right) = \sum_{d=1}^{p} (x_{id} - \,x_{jd}\,)^2 \tag{4.7}$$

In the above equation 4.7, x_{id} is represented by i^{th} sample $(i = 1,2,\ldots..m)$ and the feature index is represented by $d\ (d = 1,2,\ldots..p)$

Another criteria is also considered to determine the closeness. It is represented as the correlation coefficient as shown in the equation below:

$$C(x_i, x_j) = \frac{\sum_d (x_{id} - \bar{x}_i)\left(x_{id} - \bar{x}_j\right)}{\sqrt{\sum_d (x_{id} - \bar{x}_i)^2 \sum_d \left(x_{jd} - \bar{x}_j\right)^2}} \tag{4.8}$$

In the above equation 4.8, $\bar{x}_i = \frac{\sum_d x_{id}}{p}$

From the above equations we can see that standardizing the sample vector to mean zero and unit standard deviation, using the correlation coefficient for clustering, is equivalent to a squared distance measurement method.

There are several methods that can be used for figuring out similarity in order to fuse clusters. Some of the six most commonly used methods are:

- Single linkage
- Complete linkage
- Average linkage
- The centroid method
- The median method
- Ward's method

Giving a description of all of the above methods is out of the scope of this paper. Rather the single linkage method, which is the simplest of all the six methods, is the one which will be focussed on.

Consider a set of five sample data. The initial distance matrix of this set is as below

$$D_1 = \begin{array}{cc} \begin{pmatrix} 0 & 4 & 8 & 9 & 7 \\ 4 & 0 & 6 & 5 & 6 \\ 8 & 6 & 0 & 3 & 8 \\ 9 & 5 & 3 & 0 & 2 \\ 7 & 6 & 8 & 2 & 0 \end{pmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \\ \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} & \end{array}$$

In the above matrix, we can see that the distance difference between samples 4 and 5 is the smallest. This method would put these two samples in the same new cluster.

This in turn will provide a new distance between this new cluster (cluster 45) and the other three samples.

The new distance matrix is now demonstrated by the matrix below:

$$D_2 = \begin{pmatrix} 0 & 4 & 8 & 9 \\ 4 & 0 & 6 & 5 \\ 8 & 6 & 0 & 3 \\ 7 & 5 & 3 & 0 \end{pmatrix}$$

In the above matrix, the last row and the last column is the Cluster 45 distance. The previous step is repeated again to obtain the next cluster and new set of distance matrix. The process is iterated until all sets of distances belong to a cluster.

The steps below outline the iterative steps

Step 1 (1), (2), (3), (4), (5)

Step 2 (1), (2), (3), (45)

Step 3 (1), (2), (345)

Step 4 (1), (2345)

Step 5 (12345)

4.4 BCI classification techniques

BCI broadly uses two techniques for converting raw brain signals into machine understandable commands, classification and regression. Some of the classification techniques along with common classification types and their problems are listed below.

There are many people who have had the misfortune to lose some motor controlling capabilities, such as a complete inability to move their hands or legs. For these people BCI comes to the rescue. BCI can provide the means to make direct communication possible between the brain and a machine without the need for any peripheral muscles. To control a BCI device, the user must generate brain waves and these waves are generally passed through some classification algorithms or regression algorithms to convert the user's thoughts into machine understandable commands, classification being the most commonly used technique.

There are various properties that one needs to consider when determining which classification algorithm to use.

EEG signals are notorious for very poor signal to noise ratio and hence this factor needs to be considered. A typical BCI system usually has feature vectors of high dimensionality (Rakotomamonjy, Guigue, Mallet, & Alvarado, 2005) and features from different EEG channels and various time segments are usually concatenated into a single feature vector. Time information also needs to be included in a BCI, features as variations in EEG signals are time specific. Another feature that one needs to consider is that EEG signals vary at different times and at different sessions and are hence non-stationary. Lastly one needs to keep trial repetitions small as it can be very challenging and frustrating for the subject to undergo.

Some of the most common types of classifiers are as follows:

<u>Generative-discriminative</u>: These types of classifiers generally learn the model for each class and then try to compute the highest likelihood of a feature vector falling into a class and then select the most likely class. An example of this is Bayes quadratic. Discriminative classifiers, such as Support Vector Machines (SVM) on the other hand simply learn a way to discriminate between classes in order to classify a feature vector (Jordan 2002).

<u>Static-dynamic</u>: Where static classifiers fail to classify temporal information as they only classify single feature vectors (e.g. MultiLayer Perceptrons), dynamic classifiers are able to capture temporal data as they can classify multiple sequences of feature vectors (e.g. Hidden Markov Model or HMM) (Rabiner, 1989).

<u>Stable-unstable</u>: Stable classifiers get their name from the fact that small variations in the training data set do not hugely effect their outcome (e.g. Linear Discriminant

Analysis). Unstable classifiers does require the training data set to be non-variant to perform efficiently (e.g. MultiLayer Perceptron or MLP) (Breiman, 1998).

Regularized: Regularized classifiers are usually robust classifiers and provide good performance in classification (Duda, Hart, & Stork, 2001). A Regularized Fisher's LDA (RFLDA) is an example of such classifiers.

4.4.1 Common classifier problems

A classifier, while performing pattern recognition can come across several issues related to the feature properties that it should be able to handle. With regard to BCI, classifiers generally come across two problems, namely the curse-of-dimensionality and the bias-variance trade-off.

4.4.1.1 Curse-of-dimensionality:

With an increase in dimensionality of feature vectors, the amount of data required to classify into classes also increases (Jain, Duin, & Jianchang, 2000). A classifier will give very poor classification results if the size of the training data is small in comparison to the size of the feature vectors. So, in order to get better classification results, it is always advisable to use 10 times more training samples than the dimensionality. However it is not always possible to get so many training samples as it causes a lot of grief to the subject among many other reasons and hence is considered as a curse.

4.4.1.2 The Bias-Variance trade-off:

Classification in simple terms is finding a label y^* corresponding to a feature vector x and then using a mapping f. The classifier learns this mapping from a training dataset T. Considering the mean square error of the best mapping which is unknown, there are three types of classification error (Friedman, 1997).

Noise: Noise is the unwanted disturbance introduced in the signal. Most BCI systems thrive to reduce the noise for a better classification result.

Bias: It is the divergence between the best mapping and the estimated mapping and depends on which method is used to obtain the mapping f (is it linear or quadratic etc).

Variance: Variance is linked to how sensitive the training data set (T) is.

Since very little can be done about the noise, classifiers should have low bias and low variance to perform well. Hence there is generally a trade-off between variance and bias with classifiers. Some classifiers have high bias and low variance while others have high variance and low bias.

4.4.2 Common classifiers

Classification is an intrinsic part of a BCI setup. Therefore some of the most common classifiers used in BCI research are described below.

4.4.2 (A) Linear Discriminant Analysis (LDA)

A simple LDA algorithm works by trying to separate the feature vector out across a hyperplane of classes (Duda, Hart, & Stork, 2001). In a two class scenario (Figure 4.8), the hyperplane is obtained by seeking a line which maximizes the distance between the two classes' means and minimizes the interclass variance (Jain, Duin, & Jianchang, 2000). In scenarios where there are *n* number of classes more than one hyperplane is used and a strategy is undertaken to separate one hyperplane from the rest.

In terms of functionality and efficiency, LDA requires a very low computational overhead as compared to many others. This classifier is also very simple to implement and provides good classification results. For online BCIs, where classification needs to happen in real time, this is a very sought after choice as far as classification algorithms goes.

LDA does have some drawbacks. Since it relies heavily on linearity in data, it gives very poor results for complex non-linear EEG data.

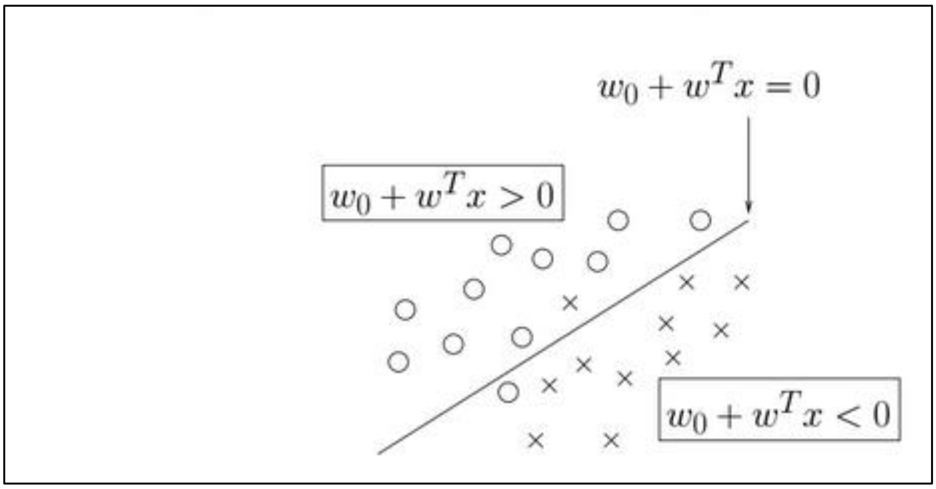


Figure 4. 8 A hyperplane separating two classes, circles and dots

(Lotte, Arnaldi, Lecuyer, Lamarche, & Arnaldil, 2007)

4.4.2 (B) Support Vector Machine (SVM)

Similar to LDA, SVM also uses hyperplanes to separate classes (Figure 4.9). However the hyperplane in an SVM is the one which maximizes the distance between the nearest training points. This distance is referred to as a margin. Maximizing this margin produces better generalization results (Burges. 1998).

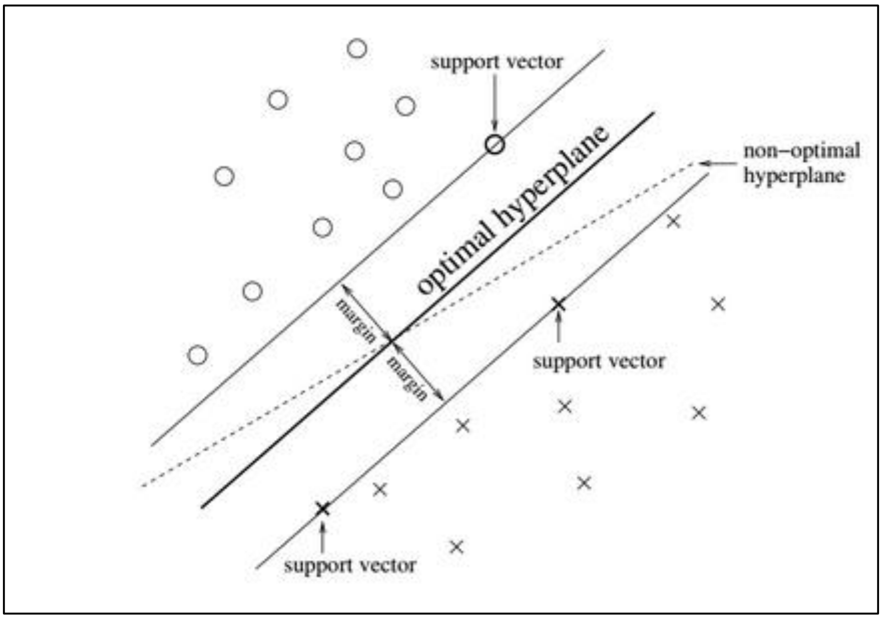


Figure 4. 9 Optimal hyperplane in an SVM

(Lotte, Arnaldi, Lecuyer, Lamarche, & Arnaldil, 2007)

4.4.2 (C) Multilayer Perceptron (MLP)

MLP is a type of neural network classifier. Neural Network classifiers are one of the most commonly used classifiers in BCI research. A typical multilayer perceptron comprises of many number of layers made up of neurons. There is an input layer and an output layer and several in-between layers (Palaniappan, 2006). The setup is such that one layer's input is connected to the previous layers output and the output layer neurons determine which class the input vector would fall into. Adding more layers and more neurons provide better results. A MLP is a very flexible classifier and is able to adapt to any number of classes across continuous functions. This feature is its advantage as well as its drawback. Since it has universal nature of adapting to any class, it is sensitive to over training. Therefore careful thought process needs to go into selection of this classifier.

4.4.2 (D) Bayes Quadratic classifiers

A Bayesian classification uses probability to assign classes to input vectors (Duda, Hart, & Stork, 2001). It uses the "Maximum A Posteriori" rule which determines the likelihood of an input vector falling into a class. As Bayes Quadratic classifiers produce quadratic decision boundaries, it gets the name Bayes Quadratic. This is not a very commonly used classifier but has been used in past researches related to mental task classification (Keirn & Aunon, 1990) and motor imagery.

4.4.2 (E) Hidden Markov Model (HMM)

The Hidden Markov Model (HMM) uses probability in the sequencing of feature vectors and provides automation. These classifiers, very commonly used in speech recognition (Rabiner, 1989) are very useful classifiers for time series classifications such as classification of EEG components which are time.

4.4.2 (F) K nearest neighbour classifiers

During the training phase, the classifier stores the feature vector and its associated labels. During the classification phase, it tries to classify an unlabelled feature vector by calculating its nearest neighbour and using its label for classification. The nearest neighbour is generally obtained using a metric distance such as Euclidian distance

(Fazli, et al., 2009). This classifier is very prone to curse of dimensionality (Freeman & Quiroga, 2013) and hence it is not a very widely used classification technique.

4.5 Conclusion

Signal analysis and signal classification forms the majority of a BCI system. This chapter has brought into light some of the signal analysis techniques common in BCI systems. It has also introduced supervised and unsupervised learning techniques pertaining to machine learning. In the next chapter, a special technique of analysing spatial and temporal brain data has been introduced.

Chapter V – Spiking Neural Networks (SNN)

The Human brain is a very complex structure. Our surroundings are full of phenomenon which have 3 dimensions in space and time and our brain is capable of processing such information. Whenever a human brain comes across an external stimulus such as movement, smell, touch etc. or a combination of several stimuli, a complex pattern is generated in the brain corresponding to these stimuli. Hence, for a BCI researcher it is essential to consider the spatial and temporal nature of the brain's response and develop a way to capture and study Spatio-Temporal Brain Data (STBD). This chapter provides an introduction to SNN which is a classification technique devised for analysing STDB and then introduces NeuCube as an Spiking Neural Network (SNN) framework. NeuCube has also been used in the pilot implementation of the case study described in chapter 6.

5.1 Introduction to Spiking Neural Networks (SNN)

Although several methods of BCI out there collects STBD, this is only partial as STBD is too complex. And then interpreting this complex data and extracting meaningful information from it is even more complex.

In order to bridge this gap the need to come up with a standardised way of mapping this complex brain data collected through different methods surfaced. Figure 5.1 shows brain mapping in Talairach Atlas. Talairach and Tournoux in 1988 (Talairach & Tournoux, 1988), came up with the idea of mapping the brain activities into 3D coplanar coordinate system which would represent the spatio-temporal nature of the brain data. A software was made available online (www.tarairach.org) to calculate the Talairach coordinates (x,y,z) for a given point of a brain image.

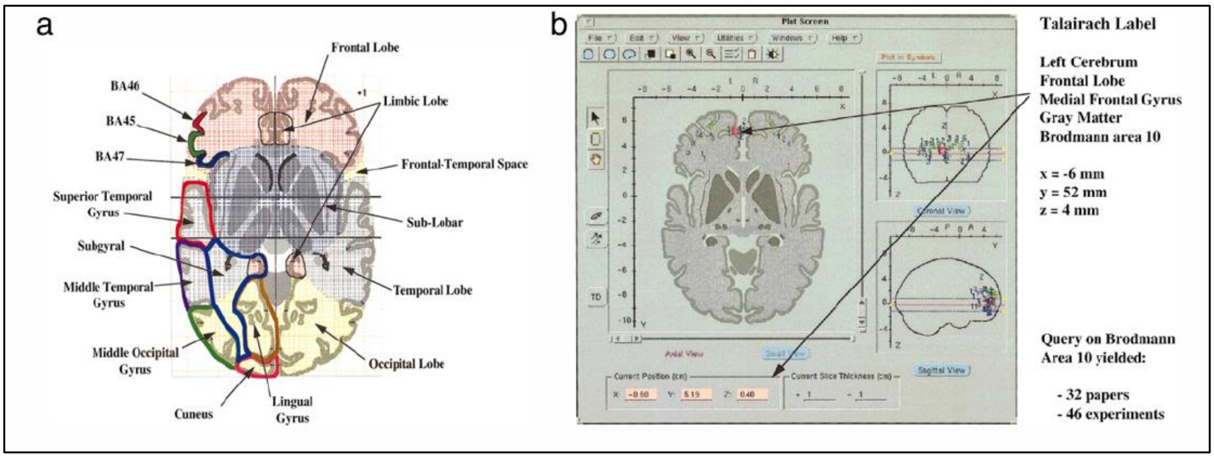


Figure 5. 1 Brain mapping

(a) Talairach stereotaxic brain atlas (b) Talairach Daemon software

(Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014)

Talairach coordinates was good for analysis of a single brain data but not enough to be considered as a standardised method of mapping brain data.

Further development in stereotaxic mapping was made when Montreal Neurological Institute’s (MNI) coordinates were introduced. This was based on data obtained by averaging MRI data obtained for several individuals rather than one person e.g. MNI152, MNI305 (Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014).

The next development in mapping of brain stereotaxic coordinates was made with the release of several other standard brain maps such as ICBM452, ICBMChinese56, ICBM AD for Alzheimers, ICBM MS for multiple sclerosis by Montreal Neurological Institute (MNI).

Now days MNI coordinates have become the standardised method of mapping brain activities. MNI coordinates have the flexibility of being able to be converted into Talairach coordinates and also into Brodmann Areas and vice versa.

The above methods of mapping brain data and many more have given rise to a new method of machine learning STBD called the Spiking Neural Network model. This model learns and interprets STBD as trains of spikes. In SNN, both spatial and temporal brain data provide encoding of the locations of synapses and neurons and the time in which a spike occurred within a given neuron (Figure 5.2).

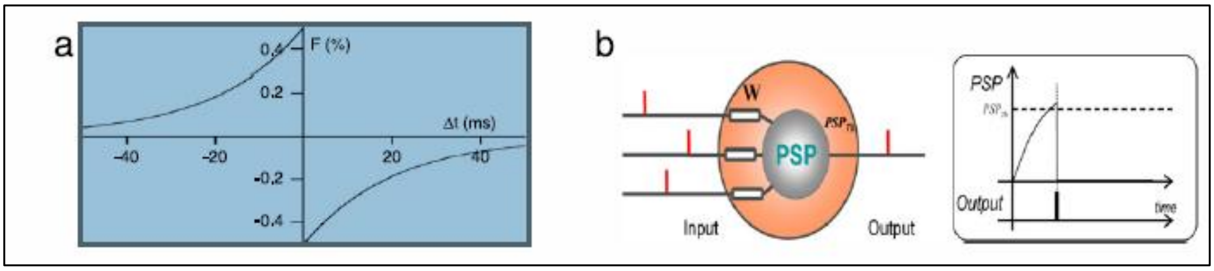


Figure 5. 2 (a) Synaptic change in a STDP learning neuron (b) Rank-order learning neuron

(Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014)

5.2. The Leaky Integrate and Fire Model (LIFM)

The LIFM is considered as one of the most popular methods of SNN. The principal of LIFM is for every input neural spike corresponding to time *t*, there is a slight increase in the membrane potential. This multiplied by the synaptic efficacy, the membrane potential reaches a threshold; θ. This results in emission of an output spike after which the membrane potential is set back to zero. Between the reset and the spike there can be some leakage denoted by π. Figure 5.4 outlines the LIFM model.

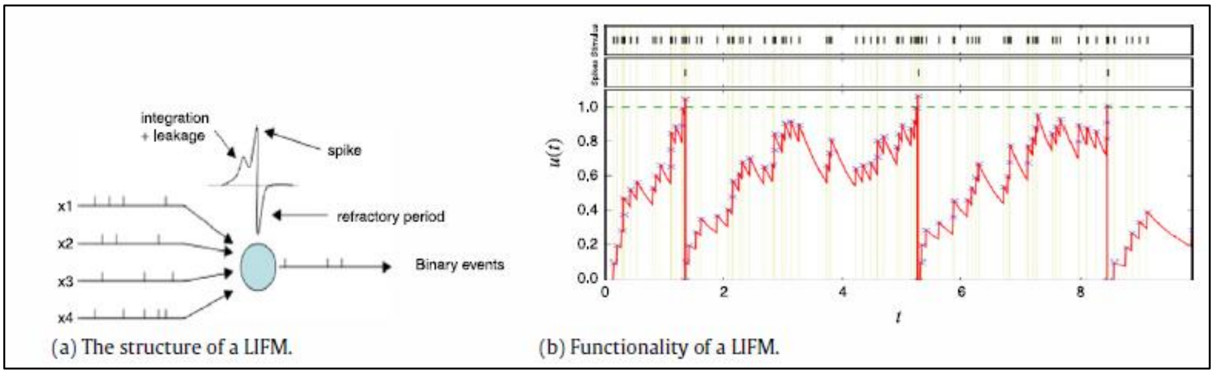


Figure 5. 3 The Leaky Integrate and Fire Model of a spiking neuron

(Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014)

‘The integrate and fire neuron’ model goes back to 1907 when Lapicque proposed an electrical model of the neuron. This electrical model consisted of an electrical circuit comprised of a resistor and a capacitor in parallel, for the purpose of representing the leakage and capacitance of the membrane. To demonstrate ‘the integrate and fire’ property of the neuron, the capacitor is provided with an electric charge in order for it to reach a threshold potential or spike. Once the threshold is reached, the potential is reset. Lapicque was able to calculate the spiking rate of the neuron based on this model.

In LIFM, the membrane potential of a neuron determines the state at which a given neuron is at a given time. The membrane of a neuron can receive excitatory or inhibitory potentials from synaptic inputs arriving from other neurons through the association of synapses. The weights of these inputs are determined by the strength of their respective synapses. Based on their weights, they are either classified as injected current or a change in the conductance of the membrane. The LIFM does not take into account the spatial structure of the neurons in relation to the dendrites.

A neuron is called ‘leaky’ because the summation of the membrane potential decays over time. This is the only drawback to an otherwise perfect model.

The model can be described mathematically with the equation below:

$$C_m \frac{dv(t)}{dt} = I_{leak}(t) + I_s(t) + I_{inj}(t) \tag{5.1}$$

In the equation 5.1 C_m = membrane capacitance, $I_{leak}(t)$ = current leakage over time, $I_s(t)$ = current produced by the synaptic input of neurons and $I_{inj}(t)$ = injected current of the neuron by an intracellular electrode.

The leak current is described by the equation below.

$$I_{leak}(t) = -\frac{C_m}{\tau_m}[v(t) - V_0] \tag{5.2}$$

In the above equation 5.2, V_0 = Resting Potential, τ_m = Passive membrane time constant = $R_m C_m$ which are resistance membrane and capacitor membrane constants.

As described above, a spike is generated when the membrane reaches a threshold V_{th} and then it resets itself toV_{reset}. This spike can be mathematically represented as below.

$$I_{spike}(t) = C_m \left[\frac{dv(t)}{dt}\right]^{-1}_{v=V_{th}} (V_{reset} - V_{th})\delta[v(t) - V_{th}] \tag{5.3}$$

In the above equation 5.3, $\delta(f)$ is also called the Dirac delta function.

5.3 Spike Timing Dependent Plasticity

Another SNN method is the Spike Timing Dependent Plasticity (STDP). This method uses Hebbian plasticity which is comprised of Long Term Potentials (LTP) and Long Term Depressions (LTD). In this method, the synapses efficacy is determined in relation to the pre-synaptic spike and post-synaptic potential. The connection weight between two neurons increases if the difference between the post-synaptic potential and the pre-synaptic spike is negative. Otherwise the connection weight decreases. The advantage of using the STDP method of learning is that the connected neurons can learn the consecutive temporal associations of neurons by learning from the data.

5.4 Rank-order Learning Rule

Another SNN learning rule is the rank-order learning rule (Delorme, Gautrais, van Rullen, & Thorpe, 1999). This learning rule assigns ranks to the input spikes among the spike trains. Based on the sequence of order of the incoming spike into the synapses, it assigns ranks to the spikes, thus assigning priority to the incoming neuronal spikes.

The formula for calculating the post synaptic potential of a neuronal weight ‘i‘ in a time ‘t ‘ is

$$PSP(i,t) = \sum mod^{order(j)}\ w_{j,i} \tag{5.4}$$

In the above equation 5.5, *mod* is the modulation factor. It can take a value between 0 and 1; *j* represents the index of the incoming spike; *i* and $w_{j,i}$ is the synaptic weight; $order(j)$ equals to the rank of the spike. So for the first spike $order(j)$ equals zero and increases with the increase in the order of the spike.

When training a BCI system using rank-order learning, the calculation of the connection weights are decided based on the order in which the incoming spikes arrive.

$$\Delta w_{j,i}(t) = mod^{order(j,i(t))} \tag{5.5}$$

There are several advantages to using this learning rule.

- It is fast
- One-pass learning is possible as it uses ranking to establish priority
- It is asynchronous

5.5 SNN Classifiers

Using STDP and several other learning rules, BCI scientists have come up with the concept of evolving Spiking Neural Networks (eSNN) for spatio-temporal pattern recognition. Some of these models are deSNN (Kasabov, Dhoble, Nuntalid, & Indiveri, 2013), SPAN (Mohemmed, Schliebs, Matsuda, & Kasabov, 2012, 2013), reservoir eSNN (Schliebs, Hamed, & Kasabov, 2011; Schliebs, Kasabov, & Defoin-Platel, 2010; Schliebs, Mohemmed, & Kasabov, 2011; Schliebs, Nuntalid, & Kasabov, 2010) etc.

5.5.1 eSNN learning

In evolving Spiking Neural Network learning techniques, the classifier attempts to create an output neuron with a particular class label (Schliebs & Kasabov, 2013). For every input sample presented to the network, there occurs a corresponding spiking train. As a result a certain output neuron is fired. In some cases there is no activation of output neurons. In that case the result of the classification problem is not determined. However if emission of a spike happens as a result of the firing of output neurons, then the neuron with the earliest spiking time is determined. The label of this neuron becomes the result of the classification problem.

The algorithm basically works on the principal of building a repository of trained output neurons during the training phase. The training algorithm of eSNN is outlined as below:

Require: m_l, s_l, c_l for a class label $l \in L$
1: initialise neuron repository $R_l = \{\}$
2: **for all** samples $X^{(i)}$ belonging to class l **do**
3: $w_j^{(i)} \leftarrow (m_l)^{order(j)}$, $\forall_j$ | j pre-synaptic neuron of i
4: $u_{max}^{(i)} \leftarrow \sum_j w_j^{(i)} \ (m_l)^{order(j)}$
5: $\vartheta^{(i)} \leftarrow c_l u_{max}$
6: **if** $\min(d(w^{(i)},w^{(k)})) < s_l$, $w^{(k)} \in R_l$ **then**
7: $w^{(k)} \leftarrow$ merge $w^{(i)}$ and $w^{(k)}$ according to Equation 5.6
8: $\vartheta^{(k)} \leftarrow$ merge $\vartheta^{(i)}$ and $\vartheta^{(k)}$ according to Equation 5.7
9: **else**
10: $R_l \leftarrow R_l \cup \{w^{(i)}\}$
11: **end if**
12: **end for**

$$w_j^{(k)} \leftarrow \frac{w_j^{(i)}+Nw_j^{(k)}}{1+N}, \ \forall_j \mid j \text{ pre-synaptic neuron of } i \qquad (5.6)$$

$$\vartheta^{(k)} \leftarrow \frac{\vartheta^{(i)} + N\vartheta^{(k)}}{1 + N} \tag{5.7}$$

5.5.2 deSNN learning

Dynamic eSSN (deSNN) (Schliebs & Kasabov, 2013) uses a combination of the rank-order learning rule and the STD learning rule.

Consider a set of training samples, for each sample, an output neuron is calculated and labelled to show that it belongs to a particular class. In deSNN the synaptic weight gets an initial value based on the rank order rule while making the assumption that the first incoming spike is more important than the others. After that thew STDP learning rule is then applied to these weights for better learning. During this process, the Euclidean distance is calculated between this neuron and all other existing neurons as a validation process for the sample.

The formula for calculating the connection weight between the output neuron '*i*' for a given time '*t*' and another neuron '*j*' is :

$$\omega_{j,t}\,(t) = \alpha\,.mod^{order_{j,i}} + \sum_{k=1}^{t} e_j\,(k).D \tag{5.8}$$

In the above Equation 5.7, α is the determination factor of the rank order part in the connection weight calculation and $mod^{order_{j,i}}$ is the modulation factor of the order of i^{th} to j^{th} neuron of the deSNN classifier.

5.6 NeuCube – An SNN framework

NeuCube is an SNN architecture developed at KEDRI which employs the fundamentals of dynamic evolving SNN (deSNN).

The architecture of NeuCube is as below:

- ❖ Encoding module – This module performs the function of converting input vector data into a time sequence of spikes. This function enables the capturing of the time and spatial nature of the STBD. This module makes use of encoding techniques like Address Event Representation (AER), Population Coding, Ben's Spike Algorithm (Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014).

  <u>Address Event Representation</u> (AER) is a method of encoding spikes where a threshold is maintained by recording the value of the same input variable consecutively and measuring their differences over time. For this encoding method to work, the input data should be streaming.

  <u>Ben's Spike Algorithm</u> (BSA) is another type of method that is used for transforming EEG data into trains of spikes. In this method the input data is transformed into series of spikes by mapping the series in the SNN reservoir of neurons.

- ❖ SNNr module – NeuCube consists of a 3D SNN reservoir of leaky integrate and fire model (LIFM) neurons. The encoded sequence of spikes are entered into this reservoir. Each spike in the reservoir has a unique 3D spatial coordinate. The unsupervised method of learning such as STDP is applied to the reservoir in order to create connections between neurons based on their temporal associations. As soon as a pattern of input neurons is entered in the reservoir, the SNNr can generate trajectories of spiking activities based on its learning.

- ❖ Classification module – Once the unsupervised STBD model of learning is applied to the SSNr (training), the SSNr is again propagated with the same input data used for training one pattern at a time in order to train the output classifier to identify the patterns in a predefined format. Evolving SNN classifiers (eSNN) such as Dynamic eSNN (deSNN) (Kasabov, 2007) are used by Neucube to provide it with fast learning capabilities.

During a typical classification of STBD, the below steps are followed in NeuCube:

1. Collect STBD EEG/fMRI data.
2. Encode input vectors into spike sequences using AER (Address Event Representation) method. In AER, a spike for a neuron will only be encoded (can be a negative or a positive spike), if there is a difference between two consecutive input vectors and it is above a defined threshold.
3. Apply learning rules to learn from the spatial connection of the neurons.
4. Learn about new classes from the spatial temporal connections and evolve.
5. Store the learning knowledge for future classification and pattern matching.

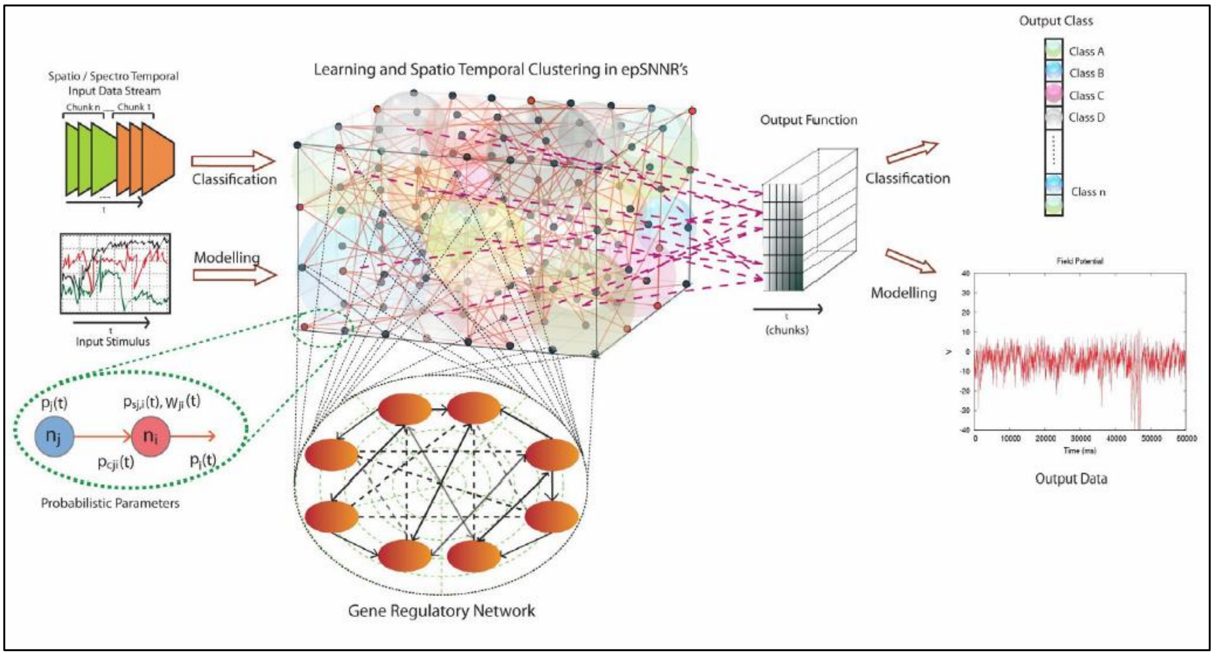


Figure 5. 4 Block diagram of NeuCube architecture

(Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014)

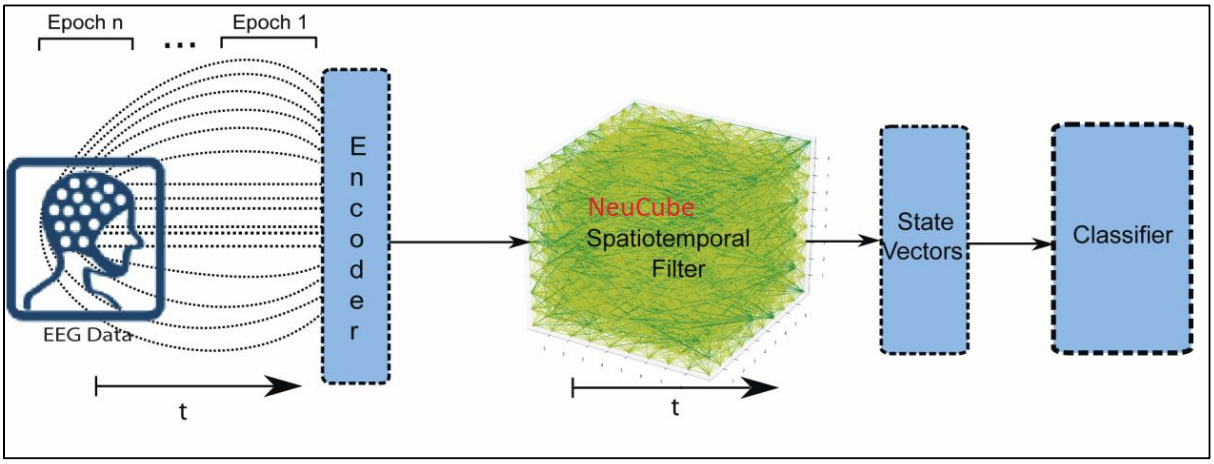


Figure 5. 5 Diagrammatic representation of NeuCube as a classifier

(Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014)

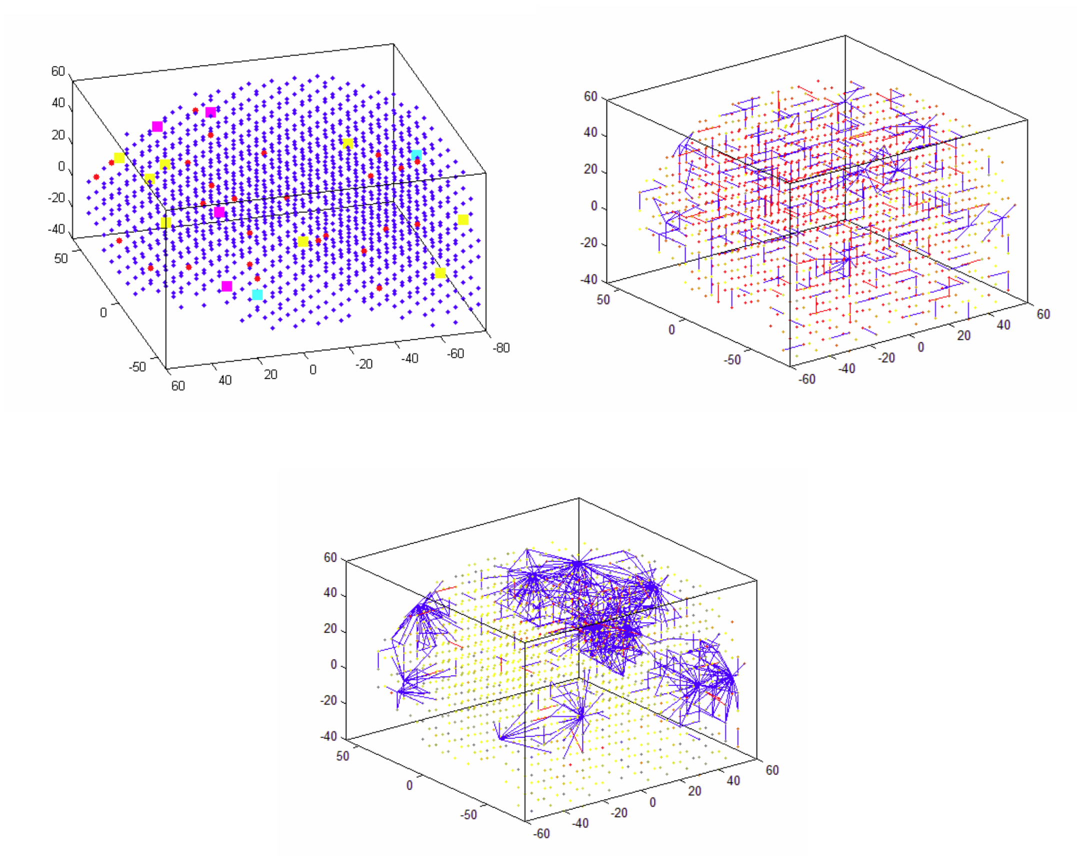


Figure 5. 6 NeuCube Classification outputs

(a) Spikes (b) Connectivity before training (c) Connectivity after training

5.7 Conclusion

This chapter has emphasised the fact that human brain signals are spatial and temporal in nature and it is very essential to take this factor into account during classification. It has then introduced SNN as a special technique of classifying spatio-temporal brain data and NeuCube as an SNN architecture. In the next chapter, a pilot gaming environment developed as a part of this thesis has been introduced where the NeuCube framework has been used to engineer a tool to provide neuro-feedback in order to improve memory of a subject.

Chapter VI – A methodology for BCI with neuro-feedback and a pilot implementation

This thesis introduces a new method of providing neuro-feedback to subjects in the form of a gaming environment. This chapter describes this gaming environment and how it has been connected to the NeuCube architecture to achieve classification.

6.1 Introduction

Back in the 60s scientists were doing a lot of experiments around brain activities. Hans Berger had by now established the fact that human brain activity can be associated to certain patterns of electrical waves the basis of which formed his EEG invention. Another scientist, Joe Camiya, was the first to coin the term neuro-feedback. He performed several experiments with an EEG machine and the relation and the impact of the subject's state of mind on these signals. Through one of his experiments he provided real time feedback to the subject based on their alpha wave readings. He provided audio stimuli to these subjects and asked them to enhance or inhibit their alpha wave activity. The results of his experiments were published in *Psychology Today* in 1968.

This thesis is an attempt to prove that neuro-feedback is a valid BCI technique. A detailed literature review on neuro-feedback and several case studies has already been presented in Chapter 4. This chapter is devoted to the research undertaken, the various tools used and the research methodology.

6.2 The methodology

Neuro-feedback is a technique through which a subject is provided with real-time feedback of their brain activity with the aid of a computer and a signal acquisition technology. This gives the subjects a chance to self-enhance their brain activity. The subjects can actually see the brain signals. For example, they are asked to inhibit their EEG alpha waves and achieving this will result in some sort of reward, such as completion of a game level or a point. The process of this self-regulation of brain activity by the subjects would initially result in the subject being able to control their brain signals.

This research is an attempt to enable memory enhancement in a subject through the process of neuro-feedback. It also attempts to prove that neuro-feedback can not only be effective but also cost effective.

With that aim in mind, this research can be divided into three components.

1. Cost effective solution to signal acquisition.
2. Classification.
3. Neuro-feedback.

The entire setup of the research is outlined in the Figure 6.1. NUN (Neurofeedback using NeuCube) is the software module which sits between these three components to achieve the goal of the experiment. Using EEG, brain data is collected as part of the experiment. This data is processed by Neucube running on a computer. NUN is used to analyse that data and Neucube's output to provide a Neuro-feedback to the subject. The three modules of the experiment is described in details in the following sections.

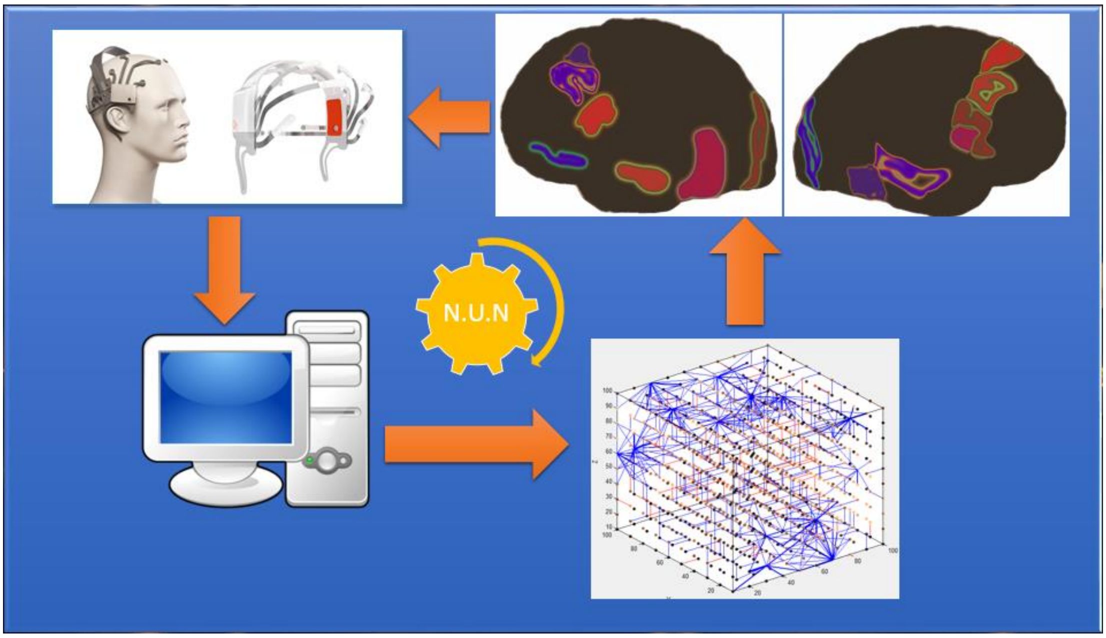


Figure 6. 1 Experiment setup

6.2.1 Signal acquisition

In the conventional BCI setups, bulky EEG machines are usually used to gather the subject's brain signals. These machines are very effective in picking up brain signals simply because they are better equipped for that function. These machines usually have 32 channels which provide more and better electrode positioning around the scalp and can gather more accurate data. However these machines are very expensive. Additionally countless hours are required for the setup of the experiments. Use of special conductive gels are also required to improve conductivity. The bulky nature of the machines can also be daunting for many subjects who may already be suffering from psychological disorders.

In this research a cost effective EEG machine called Emotiv Epoc is used for the purpose of collecting the EEG signals. This is a 14 channel (F3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4 locations) + two base channel (P3 and P4 locations) EEG device specially developed for the gaming industry as demonstrated in Figure 6.6. It is very light and fits easily on the head of a subject. Its sporty look can be quite inviting for some subjects. It does not require sticky conductive gels to be used. It comes with Bluetooth allowing easy wireless connectivity.

Figure 6. 2 Emotiv Epoc EEG Device

(EMOTIV, 2014)

As a part of this research project a customized software was developed called NUN (Neuro-feedback Using NeuCube) to connect to the Emotiv EEG device and gather a second worth of data for each class and provide a neuro-feedback. The experiment has four classes and hence 4 csv files are created by the software, one file for each class respectively. This software was developed in the .Net C# language.

6.2.2 Learning and classification

Most BCI applications achieve their goal through classification. Classifications usually use pattern recognition techniques to identify and group brain signals.

There are various methods and types of classification techniques that are currently used by BCI researchers. One technique which is of particular interest to this research is Spiking Neural Networks.

The human brain can be considered a spatio-temporal neuro-processor. Most classification techniques do not consider this dynamic spatial and temporal nature of the brain. SNN is able to collect this spatio-temporal brain data (STBD) because it treats the action of the firing of each neuron through synaptic connections as binary data which translates to a spike (Kasabov, NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, 2014). There is also a time factor associated to the firing of spikes. SNN tries to learn this information. Chapter 5 has introduced SNN and NeuCube as an SNN architecture. For the purpose of this research a special module has been incorporated into the NeuCube Architecture called NUN. This module allows the user to select the directory where the Emotiv NUN software will put the csv data collected from the Emotiv EEG device. This module has a training module and a live module. The training module will use the data file for training the NeuCube architecture and then save the reservoir. If the live module is used, then the reservoir saved in the training step will be used for classification.

NUN has four classes that needs to be classified (Figure 6.3).

Class 1 – When the left green ball turns red.

Class 2 – When the top green ball turns red.

Class 3 – When the right green ball turns red.

Class 4 – When the bottom green ball turns red.

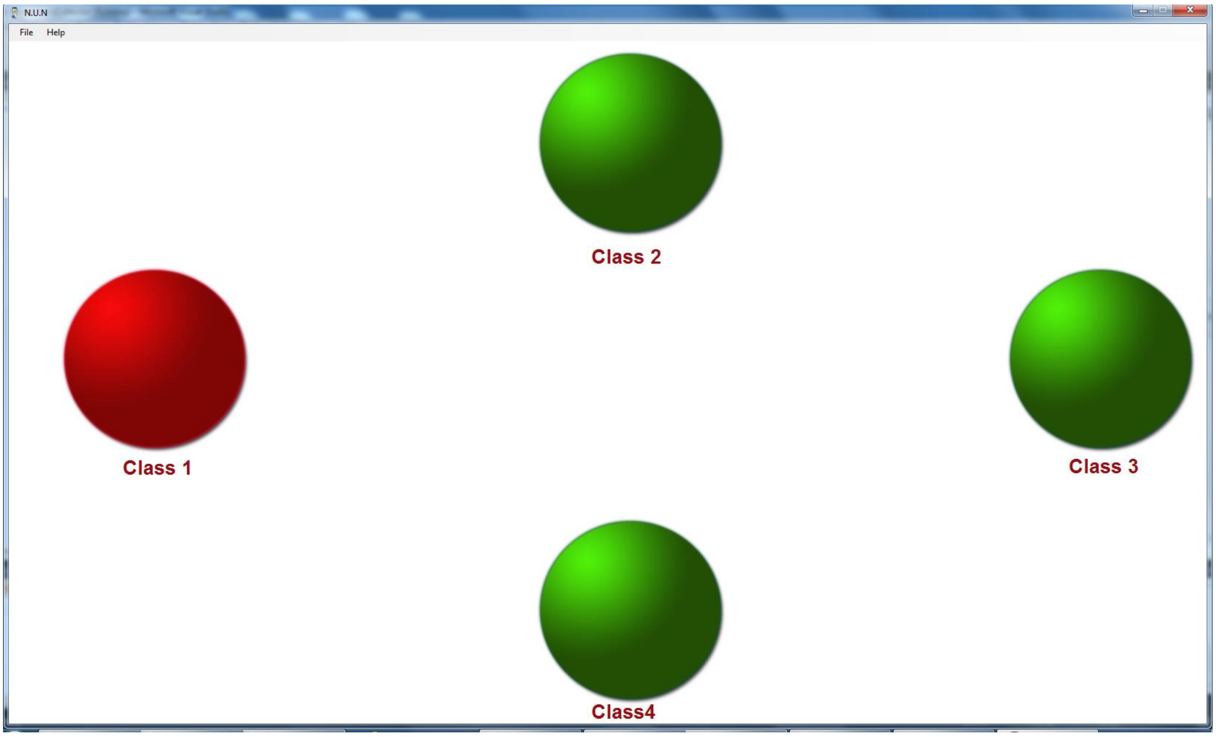


Figure 6. 3 Classes in NUN

6.2.3 Neuro-feedback

The NUN software developed for this research is also used to provide neuro-feedback to the user. This has been developed in the form of a gaming environment. The software has two modules, training and live. The training module is developed to collect brain data to train the cube. The live module is actually where the game is played by a user to receive neuro-feedback. Neuro-feedback is provided in the form of a score. The game involves four coloured balls. The colour of the ball changes every two seconds in a predefined sequence. During the training phase, the user's memory data is stored into the NeuCube reservoir corresponding to every changing colour of the ball.

Then, during the live phase, the user is supposed to remember the sequence in which the balls change colour and for every correct recall a positive point is awarded. The user needs to keep playing the game until a perfect score is achieved and in doing so their memory improves.

6.3 A pilot implementation of the proposed methodology

This research used the author as the subject but it is in theory transferrable to other players of the game as well. Below is the methodology of the research.

6.3.1 Setup

The subject of this research wore the Emotiv Epoc EEG device headgear properly. Connection to the electrodes was checked properly against the Emotiv SDK (the software development kit shipped with Emotiv) to check for any loose connections between the scalp and the electrodes. As shown in Figure 6.4 all the electrodes were required to be green before th experiment could be performed.

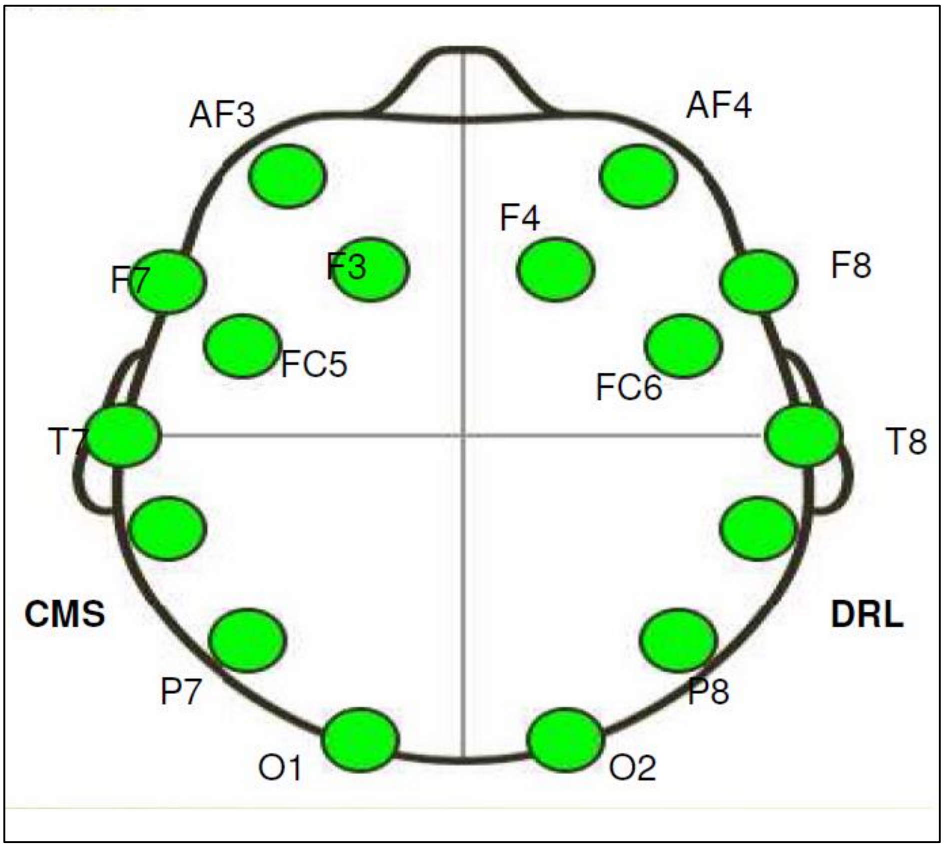


Figure 6. 4 Emotiv Electrode connectivity

6.3.2 Training

The next step was training the NeuCube. This was done through the NUN training module. The GUI of the training module of the game is shown in Figure 6.5. The user was provided with a visual stimulus through the NUN gaming environment. The graphical user interface (GUI) of the game consisted of four green coloured balls corresponding to four classes. Every twelve seconds, one of the green balls would turn red. It stayed red for ten seconds and the player was asked to think about the location of the ball which had just turned red, i.e. left, right, top or bottom. During this time one second of the brain data was collected using Emotiv. After ten seconds of staying red it would turn back into its initial green colour. The player was then warned to get ready for the next ball to turn red and once it did was again asked to think of the location of the ball which had just turned red. The brain data would again be collected. This process was continued until all four balls had turned red and then turned green again. This was repeated 20 times. For each step the data collected was in a separate csv file, which would amount to one second of EEG data.

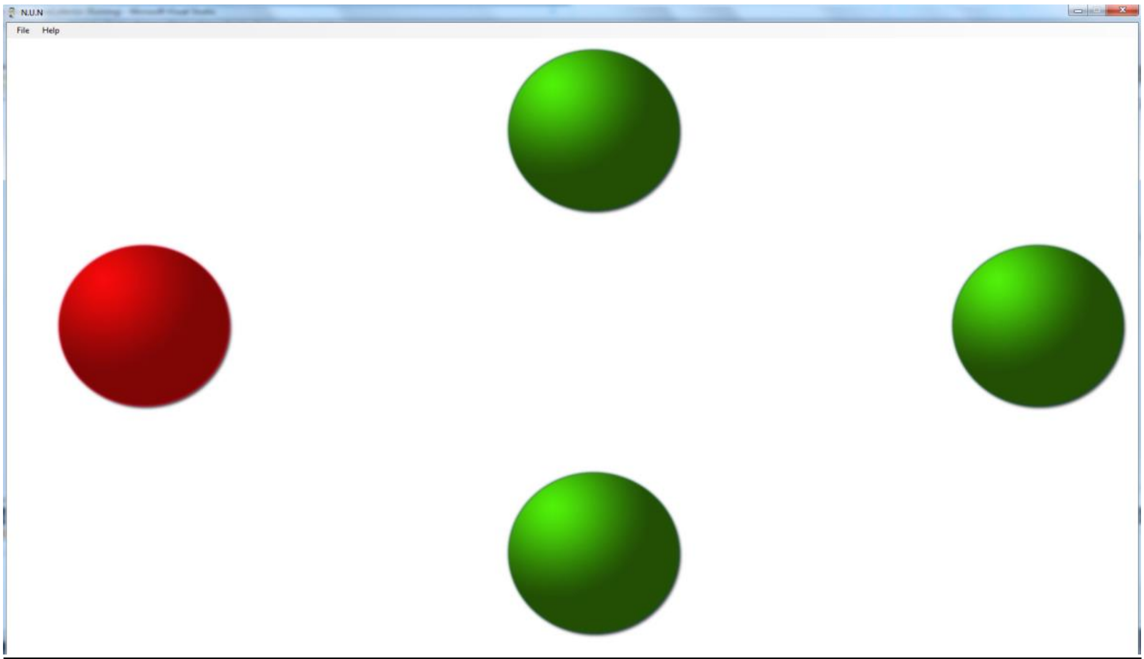


Figure 6. 5 NUN training module

Once the training data was collected, NeuCube's NUN module (Figure 6.6) was used to point to the directory where the data from the training was placed. NUN converted the twenty csv files into overlapping 3D matrices and stored it in a .mat format. NUN would then pass this .mat file to NeuCube to train the reservoir (refer to Appendix A for code).

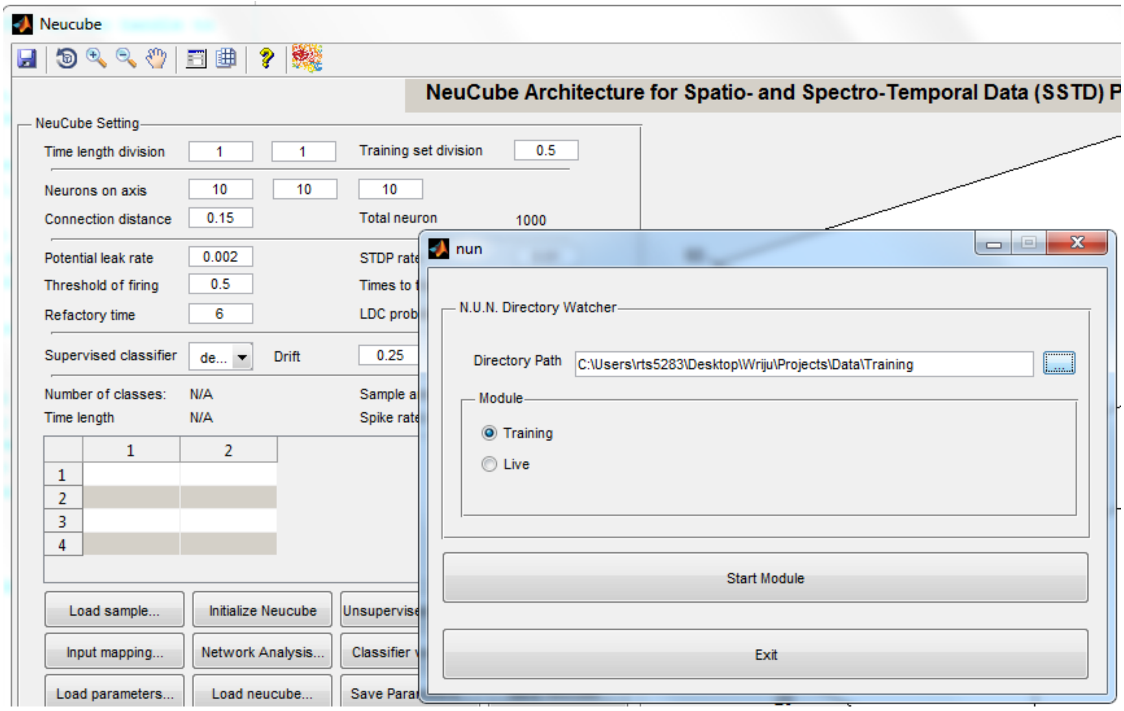


Figure 6. 6 NUN module GUI in NeuCube

The entire training step can be summarised in the below chart:

- Step 1 • Initial setup - Setup Emotiv and check connections, Initialise NUN.
- Step 2 • Start training iteration
- Step 3 • Collect brain data corresponding to the predefined 4 classes by providing visual stimuli in the form of a green ball turning red.
- Step 4 • Store the brain data in folder.
- Step 5 • Go to the next iteration. Repeat 20 times.
- Step 6 • Finish training.
- Step 7 • Use NUN module of NeuCube to train on the data collected.
- Step 8 • Save cube for future classification.

6.3.3 Live play and neuro-feedback

After the cube was trained with the training data, the game was then used as a tool to enhance memory. The user was provided with the same visual stimulus as that of the training, only this time, the player was asked to remember the sequence in which the balls turn red. The same ball would turn red in the same sequence and stay red for ten seconds. The player was given a warning and was asked to concentrate on the location of the ball about to turn red for ten seconds. After all four balls had run their course, the corresponding brain data was stored in csv file in a separate folder.

Using NeuCube's NUN module, this folder was monitored and once the training file was placed in that folder, it would pick up the csv file, convert it into a matlab file (.mat) and then run the classification algorithm of NeuCube to classify the four classes against the NeuCube model saved earlier during the training phase.

Based on the percentage of the classification of the four classes, user was given a score.

The entire gameplay logic can be summarised in the below chart:

- Step 1 — Initial setup - Setup Emotiv and check connections, Initialise NUN.
- Step 2 — Start game.
- Step 3 — Collect brain data corresponding to the predefined 4 classes by providing visual stimuli in the form of a green ball turning red.
- Step 4 — Store the brain data in folder.
- Step 5 — Wait for text file to arrive from NUN module of Neucube with classification results.
- Step 6 — Use NUN module of NeuCube to classify and create text file.
- Step 7 — Read text file and provide neuro-feedback.

6.4 Results

The experiment was performed on the author using Emotiv as the data acquisition method and NUN as the classification and neuro-feedback method.

Several trials were undertaken in the training phase and the live phase of the game. Some results were promising while the others were not.

The reason behind the failure of some of the experiments to provide sound classification results can be outlined as below:

- ❖ Noise: The experiment was performed in KEDRI lab which is not a sound proof lab. And involuntary auditory stimuli would have affected the outcome of the experiment.

- ❖ Change in condition: Since the subject was also the person conducting the experiment, undivided attention on the part of the subject was not always possible and hence there would have been difference between the conditions of the training and live experiments.

- ❖ EEG device wear and tear: The EEG device that was used to collect brain data for the experiment was an old one and did not always collect the data properly.

- ❖ Memory impairment: The subject forgot the next change in the sequence and hence the class did not classify correctly.

For more promising trials NUN was able to classify the four classes corresponding to left, top, right and bottom green balls turning red correctly.

The results of the experiment has been outlined below:

6.4.1 Experiment result 1

Action	Corresponding class number	Classified correctly	Score
Left green ball turned red	1	No	0
Top green ball turned red	2	No	0
Right green ball turned red	3	No	0
Bottom green ball turned red	4	No	0

Table 6.1: The result of experiment performed where none of the 4 classes got classified correctly

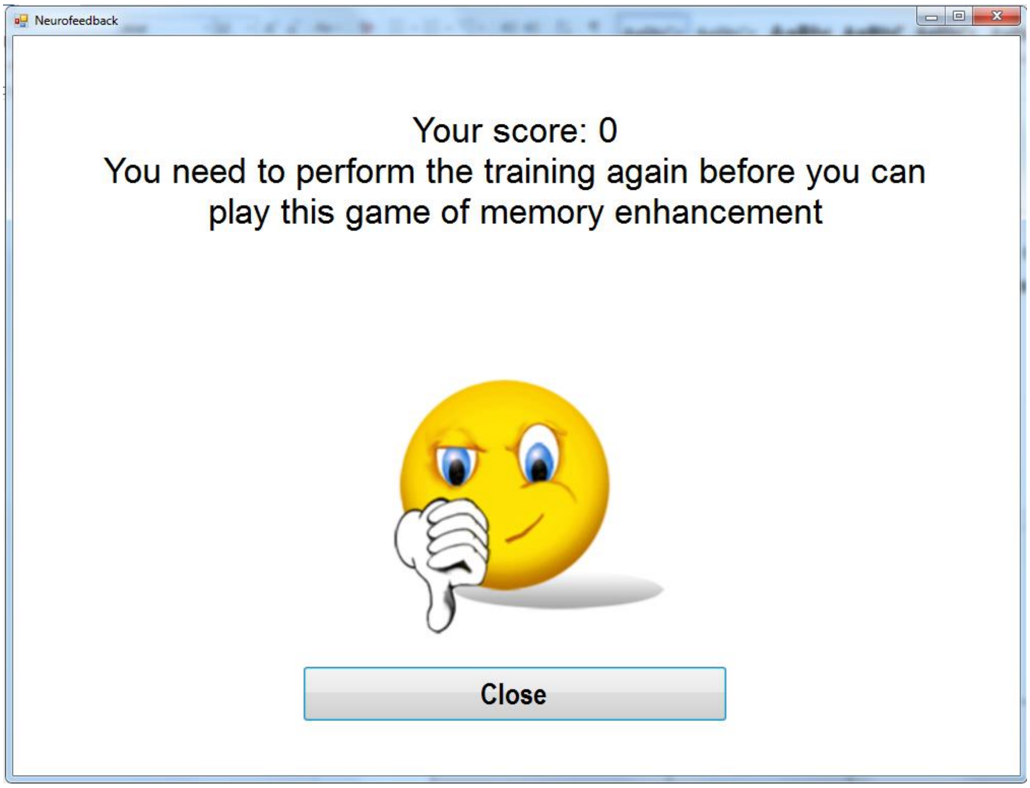


Figure 6. 7 Neuro-feedback provided by NUN for a 0 score

The above results shows that none of the classes were classified correctly. Hence NUN suggests that the subject needs to re-train before they can start improving their memory through neuro-feedback as demonstrated in Figure 6.7.

6.4.2 Experiment result 2

Action	Corresponding class number	Classified correctly	Score
Left green ball turned red	1	Yes	10
Top green ball turned red	2	No	0
Right green ball turned red	3	No	0
Bottom green ball turned red	4	No	0

Table 6.2: The result of experiment performed where only one of the 4 classes got classified correctly

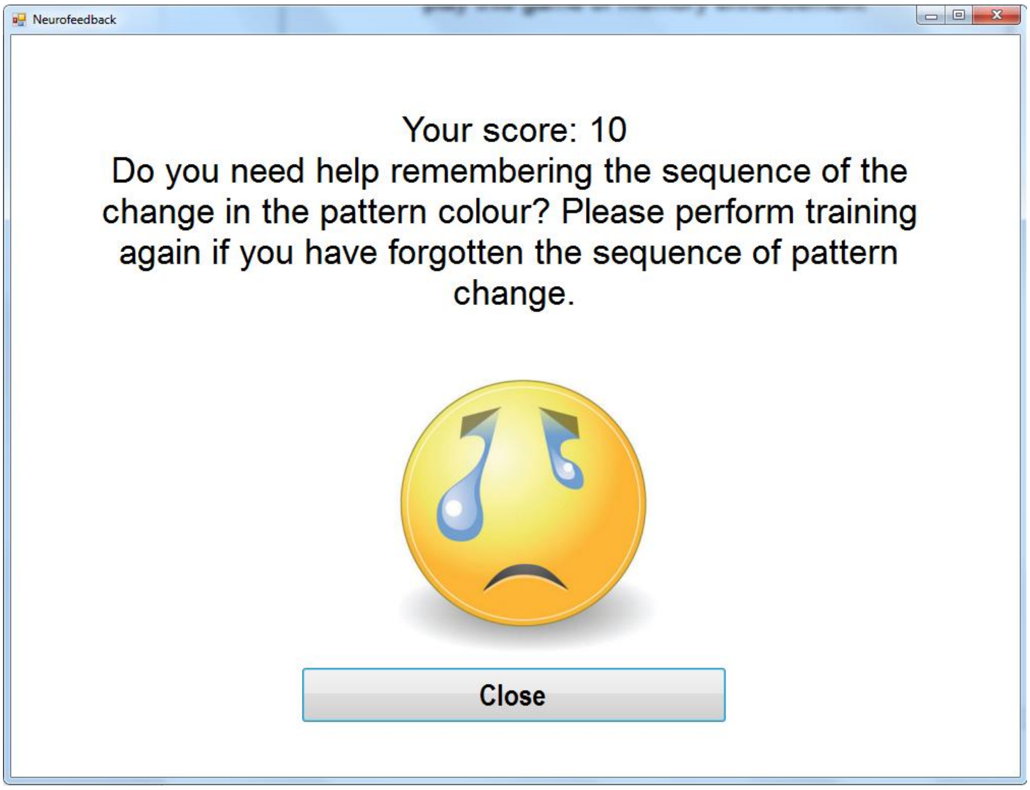


Figure 6. 8 Neuro-feedback provided by NUN for a score of 10

In the above set of results, only class 1 corresponding to the action of the left ball turning red has been classified correctly by NUN. Hence the subject scored a 10. The neuro-feedback suggests the subject to perform training again to improve score as shown in Figure 6.8.

6.4.3 Experiment result 3

Action	Corresponding class number	Classified correctly	Score
Left green ball turned red	1	Yes	10
Top green ball turned red	2	No	0
Right green ball turned red	3	Yes	10
Bottom green ball turned red	4	No	0

Table 6.3: The result of experiment performed where 2 out of the 4 classes got classified correctly

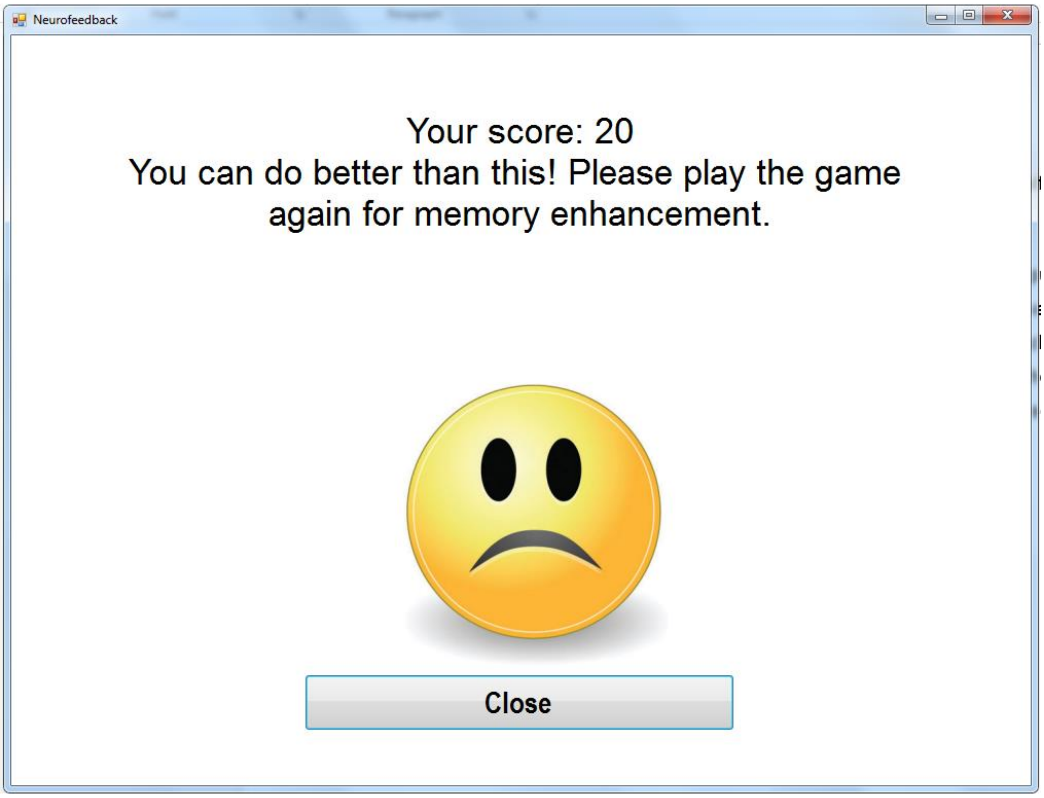


Figure 6. 9 Neuro-feedback provided by NUN for a score of 20

In this experiment, NUN classified two of the classes correctly, i.e. the left ball turning red and the right ball turning red. Hence the score of 20. The neuro-feedback provided encourages the subject to play the game again to improve their score.

6.4.4 Experiment result 4

Action	Corresponding class number	Classified correctly	Score
Left green ball turned red	1	Yes	10
Top green ball turned red	2	Yes	10
Right green ball turned red	3	Yes	10
Bottom green ball turned red	4	No	0

Table 6.1: The result of experiment performed where 3 out of the 4 classes got classified correctly

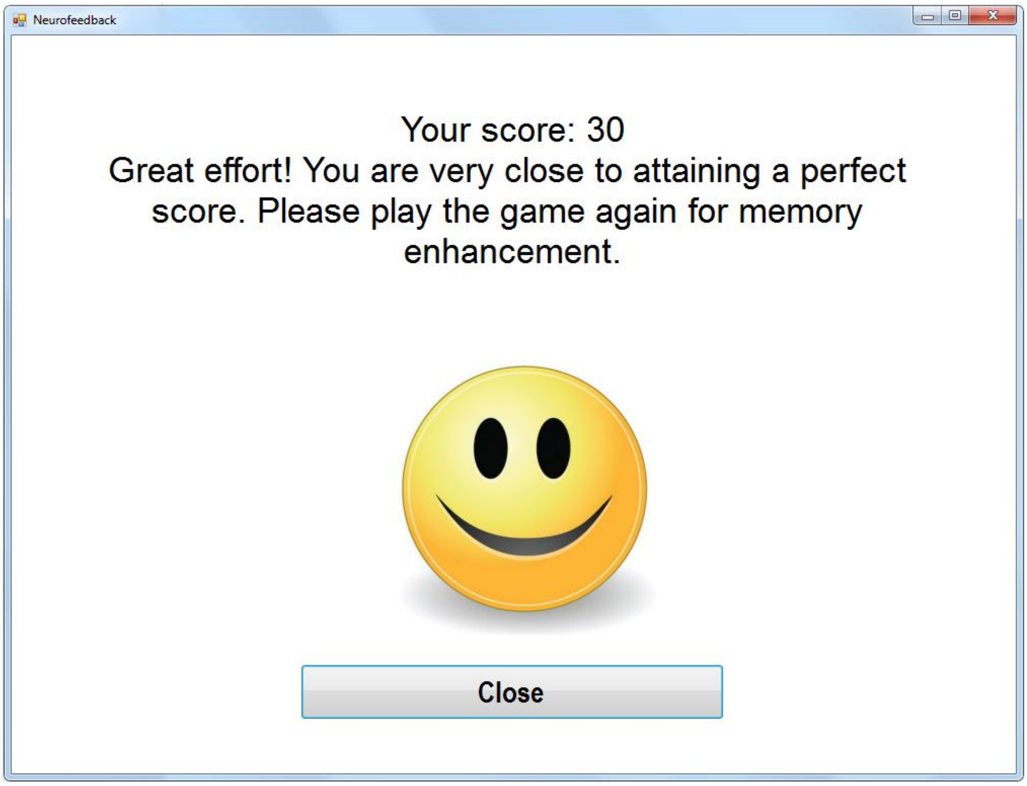


Figure 6. 10 Neuro-feedback provided by NUN for a score of 30

In this experiment, the subject is very close to attaining a perfect score. Therefore a neuro-feedback suggesting the subject to play the game again in order to get a perfect score is provided by NUN as shown in Figure 6.10.

6.4.5 Experiment result 5

Action	Corresponding class number	Classified correctly	Score
Left green ball turned red	1	Yes	10
Top green ball turned red	2	Yes	10
Right green ball turned red	3	Yes	10
Bottom green ball turned red	4	Yes	10

Table 6.1: The result of experiment performed where all of the 4 classes got classified correctly

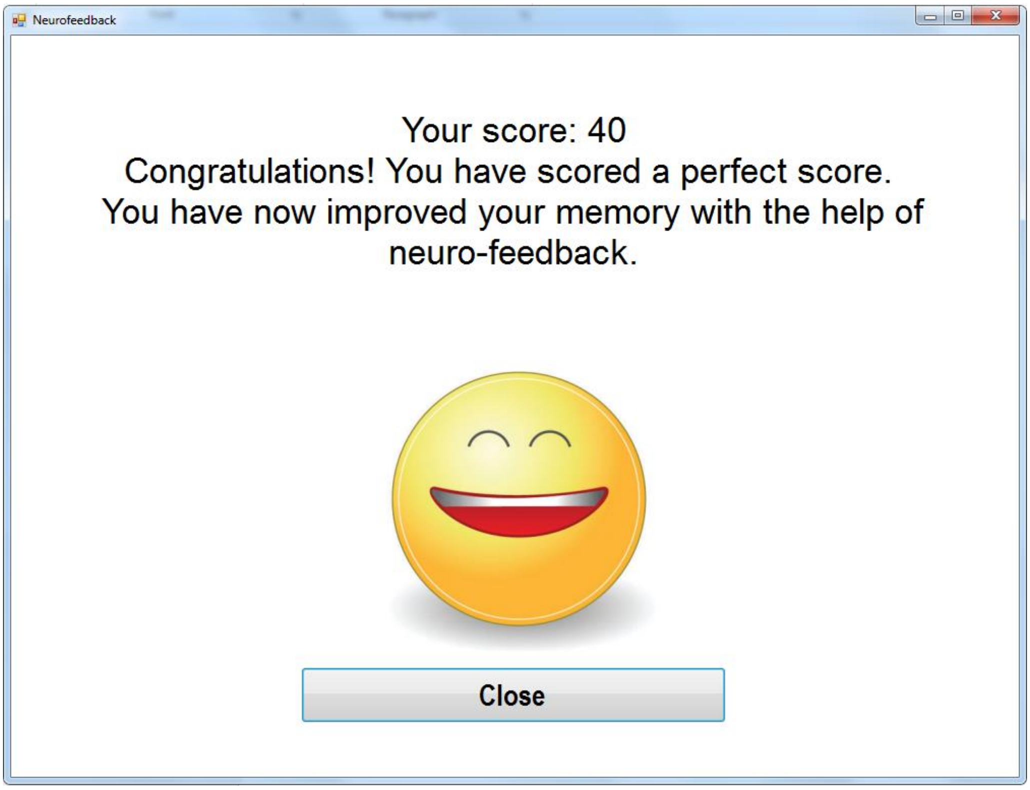


Figure 6. 11 Neuro-feedback provided by NUN for a score of 40

This experiment has resulted in all the classes being classified correctly. Therefore a very positive neuro-feedback is provided to the subject while giving a perfect score as demonstrated in Figure 6.11.

6.5 Neucube settings

The below parameters were used in the Neucube software for training it with the brain data of the subject.

Parameter Name	Parameter Value
Time division	1
Length division	1
Training set division	0.5

Neurons on axis (x,y,z)	10,10,10
Connection distance	0.15
Potential leak rate	0.002
STDP rate	0.01
Threshold of firing	0.5
Times to train	2
Refactory time	6
LDC probability	0.0
Supervised classifier	deSNN
Drift	0.25
Mod	0.4

6.6 Limitations of the case study

- This study makes use of a 14 channel EEG device for signal acquisition. Although such EEG devices can be used for the gaming industry, it lacks the dexterity to capture complex emotionally related signals, and hence can provide inaccurate results when used for providing neuro-feedback.
- More trials were required to be undertaken with different subjects. As only one person's brain signals were used in the experiment, proving the concept of memory enhancement has also been limited.
- The gaming environment built lacked complexity and randomness.

6.7 Conclusion

This chapter has demonstrated a pilot implementation of a case study to provide neuro-feedback to a subject who wants to improve their memory. The neuro-feedback is provided via a simple gaming environment and the subject is allowed to keep playing the game until they gets a perfect score of 40 and thereby improve their memory. Some of the results of the experiments performed have also been outlined in this chapter. This chapter has also outlined some of the limitations of the study and provided suggestions for future studies which could not be included as part of this study because of time constraints.

Chapter VII – Conclusion and future work in BCI

BCI is quite an innovative and emerging field which has only recently started capturing the wider scientific community's attention. Through the field of BCI complex human brain signals can be classified into simple meaningful machine understandable commands. The possibilities of such a technology are endless. Although the origin of the field itself dates back many years, when technology of that time limited the scope of the scientists, now current technology is slowly catching up with the fullest potential of the field. But the road to unlocking the fullest potential of the field is still a long one.

7.1 Conclusion

This research has been structured in such a way that full weight is given to previous and current researches in the field while building on some of their ideas and pointing to a viable future for the field of BCI.

While the thesis started with a brief introduction to the field of BCI and raised some valid research questions that have led to the structuring of this thesis, over the course of the thesis, attempt has been made to answer some of these research questions.

There are several ways of collecting and measuring brain signals of a subject. The decision to whether use an invasive or a non-invasive method of analysing brain signals is entirely dependent on the requirement of the process and its tolerance of accuracy. Many cases have demonstrated the use of invasive methods have been more effective over non-invasive methods and vice versa.

The selection of the classification algorithms to be used in the BCI procedure also contributes a lot to the success or failure of the BCI implementation. Some of the merits and demerits of common classification algorithms has been outlined in this thesis. As human brain data is spatial and temporal in nature, this property of human brain data must be considered while classifying them in order to be used in a BCI application. Spiking Neural Network architecture has been proposed as a solution to this problem. SNN uses pattern recognition techniques to identify spikes related to a response to a stimuli.

The thesis also suggested neuro-feedback as a potential solution for treating psychological disorders like ADHD in small children. The technique of neuro-feedback allows a neuro-therapist to help a subject self-regulate their brain patterns by providing positive feedbacks when a certain level of brain waves are achieved by the subject. Although several case studies has been presented in this thesis to provide scenarios where neuro-feedback has been successfully implemented and has proved effective in helping the subject with their respective psychological disorders, it is noteworthy to mention that a lot more research is required in this particular field.

In order to demonstrate the feasibility of neuro-feedback technique a pilot neuro-feedback software has been implemented as a part of this thesis. This software has made use of the Neucube architecture developed at KEDRI for pattern recognition and classification of brain signals. This software can be used to train a test subject with a certain pattern change and depending on how much they can remember from this training, is provided with a neuro-feedback in terms of a smiley face and a message on the computer screen. Because of the scope of this thesis being small and to avoid rejection of ethics approval, decision was made to carry out the test experiment against the author's brain signals only. Successful test results were obtained after many repetitions which has been outlined in the previous chapter. It has been pointed out in the limitations of this software that the implementation has been simplified and will require more complexities to be added for future versions. The suggested improvements have also been brought about in chapter 6.

7.2 Future of BCI

When one considers it, BCI is a direct connection between the human brain and a machine. No traditional input peripherals such as a keyboard or a mouse are required. A person can actually think about a task and that can be converted into a machine understandable command. The applications of such technology are limitless. It can be applied to numerous fields such as medicine, the military, surveillance, robotics, AI to name only a few. Although the field of BCI has progressed quite a lot in the last few decades, a lot of more research will be required in order to fully unlock the real potentials of the field.

Some possible future applications of BCI have been listed below:

- One day we may be able to establish a neural link between a human brain and a machine. As in the movie, Avatar where humans are able to establish a connection to drive an alien humanoid, perhaps one day we are able to do the very same thing. Japanese engineers have already been able to build a “surrogate anthropomorphic robot” which a user can link to through a BCI interface. The robot can even send back feedback in terms of the temperature that the robot is experiencing back to the user (Fernando, et al., 2012).

- Total self-sufficiency for physically challenged people could be achieved one day through BCI where the user would be able to move with the help of prosthetic limbs operated by their thoughts.

- BCI has the potential to revolutionise the entertainment industry. Instead of playing a game a person could actually be part of it. We already have oculus rift devices which are head mount virtual reality devices that can provide virtual reality based on user location and head movements. Very soon we will be able to extend these to combine with BCI technology to play games by thinking about it.

- It is said that the best ideas come to us in our dreams. But most of the time we forget it when we wake up. One day we would be able to decode human dreams with the help of BCI and unlock the fullest potential of human brain.

- There is so much that we are yet to understand about the working of the brain. BCI could be the key to obtaining that knowledge.

Appendix A – Code of NUN module in Neucube

The code below has been written in Matlab to be able to do the following.

1. Training mode – To monitor a directory and sub directory for csv files containing one second of the user's brain data. The csv files are put in the folders as a result of the NUN's interaction with the Emotiv device. Once the code below receives all the files, it converts them into a 3D matrix and creates a Matlab 3D matrix of all csv files in all the sub directories combined. Please be advised, the sub directory structure represents the number of trials. So if there are 20 trials, this code will iterate through 20 folders and create a 3D matrix of the data from those folders. Once the matrix is created, it is passed back to Neucube which can use it for learning and creating a Neucube Matlab learning file. This file is used for classifying the data. It also attaches a label to the matrix, depending on the number of trials.
2. Live mode – This takes two arguments as its parameter. One argument for the folder in which the data that needs to be classified resides. The second argument takes the full file name of the training output file that has been created as a result of Step 1. This code also creates a matrix which is passed back to Neucube along with the training output file and initiates classification of this matrix through Neucube. The Neucube classifies this file and produces a text file in the directory which specifies the class name.

nun.m

```
function varargout = nun(varargin)
%
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @nun_OpeningFcn, ...
                   'gui_OutputFcn',  @nun_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end




% --- Executes just before nun is made visible.
function nun_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
handles.FileName = '';

% Update handles structure
guidata(hObject, handles);
set(handles.radioTraining,'Value',1);
setappdata(0,'NUNModule','training');





% --- Outputs from this function are returned to the command line.
function varargout = nun_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;




function txtDirPath_Callback(hObject, eventdata, handles)

function txtDirPath_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end


% --- Executes on button press in btnStart.
function btnStart_Callback(hObject, eventdata, handles)


if(get(handles.radioTraining,'Value') == 1)
    module = 'training';
else
    module = 'live';
end
%msgbox(strcat('Module name:',module));

DirName = strcat(get(handles.txtDirPath,'String'),'\');
FileName = ConvertCSVToMatrix(DirName,module);
setappdata(0,'FileNm',FileName);
setappdata(0,'DirNm',DirName);
close();


% --- Executes on button press in btnStop.
function btnStop_Callback(hObject, eventdata, handles)

close();


% --- Executes on button press in btnBrowseDir.
function btnBrowseDir_Callback(hObject, eventdata, handles)

directoryname = uigetdir;
set(handles.txtDirPath,'string',directoryname);



function txtNeucubeFile_Callback(hObject, eventdata, handles)
```

```
% --- Executes during object creation, after setting all properties.
function txtNeucubeFile_CreateFcn(hObject, eventdata, handles)

if          ispc          &&          isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in btn_BrowseNeucube.
function btn_BrowseNeucube_Callback(hObject, eventdata, handles)

[filename, pathname] = uigetfile('*.mat', 'Pick the saved neucube file');
set(handles.txtNeucubeFile,'string',strcat(pathname,filename));
setappdata(0,'SavedNCFileNm',filename);
setappdata(0,'SavedNCpath',pathname);

% --- Executes when selected object is changed in RadioGrp.
function RadioGrp_SelectionChangeFcn(hObject, eventdata, handles)

if(get(handles.radioLive,'Value') == 1)
   set( handles.txtNeucubeFile,'Visible','on');
   set( handles.btn_BrowseNeucube,'Visible','on');
   setappdata(0,'NUNModule','live');
else
   set( handles.txtNeucubeFile,'Visible','off');
   set( handles.btn_BrowseNeucube,'Visible','off');
   setappdata(0,'NUNModule','training');
end
```

convertCSVToMatrix.m

```
function [ MatFileName ] = ConvertCSVToMatrix( directory_name,module )

    %directory_name                                                  =
'C:\Users\rts5283\Desktop\Wriju\Projects\Data\Training\';
    %module = 'training';
```

```
    blankArray = [];
    matrix(:,:,1) = blankArray;
    eeg_data(:,:,1)  = blankArray;
    TotalClasses = 4;

    class_label = [];




    %training module
    if(strcmp(module,'training')==1)
        Progress = waitbar(0,'Creating training file...');
        samplecounter = 1;
        while samplecounter <= 20
            folderName                                                    =
strcat(directory_name,'Trial',num2str(samplecounter),'\');

            %msgbox(strcat('Folder',folderName));

            matrix = CreateNeucubeMatrix(folderName);

            eeg_data = cat(3,eeg_data,matrix);
            samplecounter = samplecounter+1;
            PercentageComplete = (samplecounter/20)*100;

waitbar(samplecounter/20,Progress,strcat(num2str(PercentageComplete),'     %
complete'))
        end
        close(Progress);

        %Create second matrix

        CellNo = 1;
        for i=1:TotalClasses
           samplecounter = 1;
           while samplecounter <= 20
               class_label(1,CellNo) = i;
               samplecounter = samplecounter +1;
               CellNo = CellNo +1;
```

```
            end

        end

         %save matrix
        datestring = datestr(clock,30);
        MatFileName = strcat('training',datestring,'.mat');
        save(strcat(directory_name,MatFileName),'eeg_data','class_label');

    else
        eeg_data = CreateNeucubeMatrix(directory_name);
        datestring = datestr(clock,30);
        MatFileName = strcat('Live',datestring,'.mat');
        save(strcat(directory_name,MatFileName),'eeg_data');

    end
end
```

CreateNeucubeMatrix.m

```
function [matrix] = CreateNeucubeMatrix( directory_name )
%directory_name                                                           =
'C:\Users\rts5283\Desktop\Wriju\Projects\Data\Training\Trial1\';
pastHeader = 0;
TotalArrayString = '';
TotalNoOfFiles = 1;
files = dir(directory_name);

fileIndex = find(~[files.isdir]);

for i = 1:length(fileIndex)

fileName = files(fileIndex(i)).name;
FullFileName = strcat(directory_name,fileName);
[pathstr,name,extension] = fileparts(FullFileName);

if (extension == '.csv')
```

```
    fileId = fopen(FullFileName);

    line = fgets(fileId);
    linecounter = 1;
    Array = [];

    while ischar(line)
        updatedString = regexprep(line, ',', ' ');
        if(pastHeader == 1 && ~isempty(updatedString))

            TempCells = [str2num(updatedString)];

            if(isempty(Array))
                Array = TempCells;
            else
                 Array = [Array;TempCells];
            end



           linecounter = linecounter+1;
        end

        line = fgets(fileId);
        pastHeader = 1;
    end
    disp(TotalArrayString)
    fclose(fileId);
    matrix(:,:,TotalNoOfFiles) = [Array];
    TotalNoOfFiles = TotalNoOfFiles + 1;
end
end

end
```

Appendix B – Code of NUN providing neuro-feedbacks

This is the code written in C# language with a GUI. The GUI has four balls which change colour based on a predefined sequence. This code has a training module and a live module. The training module, when selected, goes through the colour change sequence twenty times and executes the code to collect the brain data for that instance and puts it in the directory for the Neucube NUN module to pick up.

The live mode collects the brain data for one run of the colour change sequence and puts the csv file in the folder for the Neucube NUN live module to pick up and classify.

On classification, a score and a neuro-feedback is provided to the subject

DataCollector.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;


namespace EmotivDataCollector
{
    public partial class DataCollector : Form
    {
        int Timerseconds = 0;
        int TotalIterations = 0;
        int score = 0;
        int TotalTrial = 0;
        int ClassNumber = 0;
        string EEGLoggerFilePath = "";
```

```
public void ReadConfig()
{
    try
    {
        string[] LinesArr = System.IO.File.ReadAllLines(@"config.txt");
        int positionOfEqualTo = 0;
        foreach (string line in LinesArr)
        {
            if (line.Contains("EEG_Logger_Path"))
            {
                positionOfEqualTo = line.LastIndexOf("=");
                EEGLoggerFilePath = line.Substring(positionOfEqualTo + 1);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error in reading config file" + ex.ToString());
        StopGameEvent();
    }
}


public DataCollector()
{
    InitializeComponent();
}

private void DataCollector_Load(object sender, EventArgs e)
{
    StopGame.Enabled = false;
    InitialiseBalls();
}

private void Exit_Click(object sender, EventArgs e)
{
    this.Close();
}

private void StopGame_Click(object sender, EventArgs e)
```

```
        {
            DialogResult res = MessageBox.Show("Are you sure you want to stop the game?",
"Abort", MessageBoxButtons.OKCancel);
            if (res == System.Windows.Forms.DialogResult.OK)
            {
                StopGameEvent();
            }

        }

        public void InitialiseBalls()
        {
            txtTimerInterval.Enabled = false;
            LeftGreen.Location = new Point(62, 328);
            LeftRed.Location = new Point(62, 328);
            TopGreen.Location = new Point(714, 27);
            TopRed.Location = new Point(714, 27);
            RightGreen.Location = new Point(1358, 328);
            RightRed.Location = new Point(1358, 328);
            BottomGreen.Location = new Point(714, 678);
            BottomRed.Location = new Point(714, 678);
            EnableBalls(false, false, false, false, false, false, false, false);
            GameOverLogo.Location = new Point(421, 525);
            GameOverLogo.Size = new System.Drawing.Size(872, 302);
            GameOverLogo.Visible = false;
        }

        public void StartGame()
        {
            ReadConfig();
            //MessageBox.Show("EEG file path: " + EEGLoggerFilePath);
            Lblneuro-feedback.Visible = false;
            PicBigBrain.Visible = false;
            NewGame.Enabled = false;
            TrainNeucube.Enabled = false;
            StopGame.Enabled = true;
            this.BackgroundImage = null;
            txtTimerInterval.Visible = true;
            GameOverLogo.Visible = false;
            Timerseconds = 0;
            TotalIterations = 1;
```

```
            GameTimer.Start();
            EnableBalls(true, false, true, false, true, false, true, false);
        }

        private void NewGame_Click(object sender, EventArgs e)
        {
            StartGame();
        }

        public void CollectData()
        {

        }

        private void GameTimer_Tick(object sender, EventArgs e)
        {

            if (Timerseconds <= 5)
            {
                txtTimerInterval.Visible = true;
                txtTimerInterval.Text = "Get ready to move your hand in the direction
of the Ball which will turn red next. You should remember it from your training..";
            }

            if (Timerseconds > 5 && Timerseconds < 10)
            {
                txtTimerInterval.Text = "Get ready to move your hand in " + (10 -
Timerseconds).ToString() + " secs...";
            }

            if (Timerseconds >= 10 && Timerseconds <= 12)
            {
                txtTimerInterval.Visible = false;
                if (TotalIterations == 1)
                {
                    EnableBalls(false, true, true, false, true, false, true, false);
                }

                if (TotalIterations == 2)
                {
                    EnableBalls(true, false, false, true, true, false, true, false);
```

```
                }

                if (TotalIterations == 3)
                {
                    EnableBalls(true, false, true, false, false, true, true, false);
                }

                if (TotalIterations == 4)
                {
                    EnableBalls(true, false, true, false, true, false, false, true);
                }


            }

            if(Timerseconds>12)
            {
                EnableBalls(true, false, true, false, true, false, true, false);
                Timerseconds = 0;
                TotalIterations = TotalIterations + 1;
            }

            if (TotalIterations > 4)
            {
                StopGameEvent();
            }

            Timerseconds = Timerseconds + 1;
        }

        public void StopGameEvent()
        {
            Lblneuro-feedback.Visible = true;
            txtTimerInterval.Visible = true;
            score = 30;
            txtTimerInterval.Text = "Thank you for playing" + Environment.NewLine + "
Your Score: " + score.ToString() ;
            GameTimer.Stop();
            TrainingTimer.Stop();
            InitialiseBalls();
            NewGame.Enabled = true;
```

```
            TrainNeucube.Enabled = true;
            StopGame.Enabled = false;
            GameOverLogo.Visible = true;
            Timerseconds = 0;

        }

        public void EnableBalls(bool LG, bool LR, bool TG, bool TR, bool RG, bool RR,
bool BG, bool BR)
        {
            //Make left green ball visible
            if (LG)
                LeftGreen.Visible = true;
            else
                LeftGreen.Visible = false;


            //Make Left red ball visible
            if (LR)
                LeftRed.Visible = true;
            else
                LeftRed.Visible = false;


            //Make Top green visible
            if (TG)
                TopGreen.Visible = true;
            else
                TopGreen.Visible = false;


            //Make Top red visible
            if (TR)
                TopRed.Visible = true;
            else
                TopRed.Visible = false;


            //Make right green visible
            if (RG)
                RightGreen.Visible = true;
```

```
            else
                RightGreen.Visible = false;



            //Make right red visible
            if (RR)
                RightRed.Visible = true;
            else
                RightRed.Visible = false;



            //Make Bottom green visible
            if (BG)
                BottomGreen.Visible = true;
            else
                BottomGreen.Visible = false;



            //Make Bottom red visible
            if (BR)
                BottomRed.Visible = true;
            else
                BottomRed.Visible = false;



        }

        private void TrainNeucube_Click(object sender, EventArgs e)
        {
            DialogResult res = MessageBox.Show("Are you sure you want to train Neucube
now?", "Train Newucube", MessageBoxButtons.OKCancel);
            if (res == System.Windows.Forms.DialogResult.OK)
            {
                Lblneuro-feedback.Visible = false;
                PicBigBrain.Visible = false;
                NewGame.Enabled = false;
                TrainNeucube.Enabled = false;
                StopGame.Enabled = true;
                this.BackgroundImage = null;
                txtTimerInterval.Visible = true;
                InitialiseBalls();
```

```
                EnableBalls(true, false, true, false, true, false, true, false);
                ClassNumber = 1;
                TotalTrial = 1;
                TrainingTimer.Start();
            }
        }



        private void TrainingTimer_Tick(object sender, EventArgs e)
        {

            if (Timerseconds <= 5)
            {
                txtTimerInterval.Visible = true;
                txtTimerInterval.Text = "Get ready to train Neucube with trial " +
TotalTrial.ToString() + " / 5 for Class " + ClassNumber.ToString();
            }


            if (Timerseconds > 5 && Timerseconds < 10)
            {
                txtTimerInterval.Visible = true;
                if (ClassNumber == 1)
                    txtTimerInterval.Text = "The left Ball will turn red next.Get ready
to move your hand in the left direction in " + (10 - Timerseconds).ToString() + "
secs...";

                if (ClassNumber == 2)
                    txtTimerInterval.Text = "The top Ball will turn red next.Get ready
to move your hand in the upward direction in " + (10 - Timerseconds).ToString() + "
secs...";

                if (ClassNumber == 3)
                    txtTimerInterval.Text = "The right Ball will turn red next.Get ready
to move your hand in the right direction in " + (10 - Timerseconds).ToString() + "
secs...";

                if (ClassNumber == 4)
                    txtTimerInterval.Text = "The bottom Ball will turn red next.Get
ready to move your hand in the downward direction in " + (10 - Timerseconds).ToString()
+ " secs...";
            }
```

```
    if (Timerseconds >= 10 && Timerseconds <= 12)
    {
        txtTimerInterval.Visible = false;
        if (ClassNumber == 1)
            EnableBalls(false, true, true, false, true, false, true, false);

        if (ClassNumber == 2)
            EnableBalls(true, false, false, true, true, false, true, false);

        if (ClassNumber == 3)
            EnableBalls(true, false, true, false, false, true, true, false);

        if (ClassNumber == 4)
            EnableBalls(true, false, true, false, true, false, false, true);

    }


    if (Timerseconds > 12)
    {
        EnableBalls(true, false, true, false, true, false, true, false);
        Timerseconds = 0;
        TotalTrial = TotalTrial + 1;
    }

    if (TotalTrial > 5 && ClassNumber < 4)
    {
        TotalTrial = 1;
        ClassNumber = ClassNumber + 1;
    }

    if (TotalTrial > 5 && ClassNumber >= 4)
    {
        stopTraining();
    }
    Timerseconds = Timerseconds + 1;
}
```

```
        public void stopTraining()
        {
            TrainingTimer.Stop();
            EnableBalls(false, false, false, false, false, false, false, false);
            txtTimerInterval.Visible = true;
            txtTimerInterval.Text = "Training Complete.";

        }

        private void MnuAboutNUN_Click(object sender, EventArgs e)
        {
            AboutNUN aboutform = new AboutNUN();
            aboutform.Show();
        }
    }
}
```

References

Andrieu, C., Freitas, N. D., Doucet, A., & Jordon, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning, 50*(1-2), 5-43.

Anupama.H.S, N.K.Cauvery, & Lingaraju.G.M. (2012). Brain Computer Interface and its types - A Study. *International journal of advances in engineering and technology, 3*(2), 739.

Asimov, I. (1965). *The human brain :its capabilities and functions /Isaac Asimov ; illustrated by Anthony Ravielli.* London: Nelson.

Badcock, N. A., Lissa, P. d., Mousikou, P., Thie, J., Mahajan, Y., & McArthur, G. (2013, February 19). Validation of the Emotiv EPOC EEG gaming system for measuring research quality auditory ERPs.

Bernaciková, M. M. (2012). *Nervous control of muscular contraction*. Retrieved from http://www.fsps.muni.cz/~tvodicka/data/reader/book-4/05.html

BIOPAC. (2008). Retrieved from FNIR FUNCTIONAL NEAR INFRARED OPTICAL BRAIN IMAGING SYSTEM - FNIR100B: http://www.biopac.com/fNIR100A-Optical-Brain-Imaging-System

Blankertz, B., & Vidaurre, C. (2009). Towards a cure for BCI illiteracy: machine learning based co-adaptive learning. *Eighteenth Annual Computational Neuroscience Meeting: CNS 2009*, (p. 85). Berlin.

Braingamereview. (2015). Retrieved from http://www.braingamereview.com/neuroplasticity-brain-plasticity/

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. Homogeneous. *Biological Cybernetics, 95*(1), 19.

Carter, R. (2009). *The brain book: An Illustrated Guide to its structure, function and disorders.* Dorling Kindersley Publishing, Inc.

Carter, R. (2009). *The brain book: An Illustrated Guide to its structure, function and disorders.* London: Dorling Kindersley Publishing Inc.

Chapin, Ted, R.-C., & Corporation., L. A. (2014). Neurotherapy and neurofeedback :brain-based treatment for psychological and behavioral problems /Theodore J. Chapin and Lori A. Russell-Chapin.

Danny Plass-Oude Bos, H. G. (2012). Improving BCI performance after classification. *Proceedings of the 14th ACM international conference on Multimodal interaction* (pp. 587-594). Santa Monica, California, USA: ACM.

Delorme, A., Gautrais, J., van Rullen, R., & Thorpe, S. (1999). SpikeNET: A simulator for modeling large networks of integrate and fire neurons. *Neurocomputing, 26–27*, 989-996.

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. *Neural Networks (IJCNN), The 2012 International Joint Conference on*, (pp. 1-7).

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (second ed.).

Duvinage, M., Castermans, T., Petieau, M., Hoellinger, T., Cheron, G., & Dutoit, T. (2013, June 25). Performance of the Emotiv Epoc headset for P300-based applications.

EMOTIV. (2014). Retrieved from http://emotiv.com/

Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K.-R., & Grozea, C. (2009). Subject-independent mental state classification in single trials. *Neural Networks, 22*(9), 1305-1312.

Fedorenko, E., Fillmore, P., Smith, K., Bonilha, L., & Fridriksson, J. (2015). The superior precentral gyrus of the insula does not appear to be functionally specialized for articulation. *Journal of Neurophysiology, 113*(7), 2376-2382.

Fenner, J. (2013, November). Retrieved from SIDS Crib Death Linked to Brainstem Abnormalities, Says Study: http://guardianlv.com/wp-content/uploads/2013/11/Lateral-view-of-the-brainstem.jpg

Fernando, C. L., Furukawa, M., Kurogi, T., Hirota, K., Kamuro, S., Sato, K., . . . Tachi, S. (2012, January). *TELESAR V: TELExistence surrogate.* Retrieved from ResearchGate: http://www.researchgate.net/publication/254463414_TELESAR_V_TELExistence_surrogate_anthropomorphic_robot

Freeman, W. J., & Quiroga, R. Q. (2013). *Imaging Brain Function With EEG - Advanced Temporal and Spatial Analysis of Electroencephalographic Signals.*

Friedman, J. (1997). On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery, 1*(1), 55-77.

Fumuro, T., Matsuhashi, M., Mitsueda, T., Inouchi, M., Matsumoto, R., Kawamata, J., . . . Ikeda, A. (2010). P36-15 Improved bereitschaftspotential (BP) in Parkinson's disease (PD) by means of neuro-feedback (NFB) training of slow cortical potentials (SCPs). *Clinical Neurophysiology, 121, Supplement 1*, S321.

Furtscheller, G., Allison, B. Z., Brunner, C., Bauernfeind, G., Solis-Escalante, T., Scherer, R., . . . Birbaumer, N. (2010). The Hybrid BCI. *Frontiers in Neuroscience, 4*, 42.

Garrido, M. I., Kilner, J. M., Stephan, K. E., & Friston, K. J. (2009). The mismatch negativity: A review of underlying mechanisms. *Clinical Neurophysiology, 120*(3), 453-463.

Geller, B. (2007). Neurofeedback for ADHD. *Journal Watch. Psychiatry*.

Hartstein, G. (2014, April 28). *Some important science on disorders of consciousness – part 2*. Retrieved from https://formerf1doc.wordpress.com/category/interesting-medical-stuff/

Heinrich, H., Gevensleben, H., & Strehl, U. (2007). Annotation: Neurofeedback – train your brain to train behaviour. *Journal of Child Psychology and Psychiatry, 48*(1), 3–16.

Hinds, O., Wighton, P., Tisdall, M. D., Hess, A., Breiter, H., & Kouwe, A. v. (2014). Neurofeedback Using Functional Spectroscopy. *International Journal of Imaging Systems and Technology, 24*(2), 138–148.

Jain, A. K., Duin, R. P., & Jianchang, M. (2000). Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22*(1), 4-37.

Kasabov, N. (2009). Integrative connectionist learning systems inspired by nature: current models, future trends and challenges. *Natural Computing, 8*(2), 199-218.

Kasabov, N. (2014). In *Springer Handbook of Bio-/Neuroinformatics* . Springer.

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

Knowlton, R. C. (2006). Magnetic source imaging versus intracranial electroencephalogram in epilepsy surgery: A prospective study. *Annals of neurology, 59*(5), 835 - 842.

Kotchoubey, B., Schneider, D., Uhlmann, C., Schleichert, H., & Birbaumer, N. (1997). Beyond habituation: long-term repetition effects on visual event-related potentials in epileptic patients. *Electroencephalography and Clinical Neurophysiology, 103*(4), 450-456.

Kullmann, D. M. (2011, May 30). *The mismatch negativity: an index of cognitive decline in neuropsychiatric and neurological diseases and in ageing*. Retrieved from http://brain.oxfordjournals.org/content/134/12/3435

Landa, L., Krpoun, Z., Kolarova, M., & Kasparek, T. (2014). *EVENT-RELATED POTENTIALS AND THEIR APPLICATIONS.* Neuroscientia o.s.

Leung, C. S., Lee, M., & Chan, J. H. (2009). Neural Information Processing :16th International Conference, ICONIP 2009, Bangkok, Thailand, December 1-5, 2009, Proceedings.Part II /Chi Sing Leung, Minho Lee, Jonathan H. Chan (Eds.). Bangkok.

Li, X. (2014). *Functional magnetic resonance imaging processing.*

Lotte, F., Arnaldi, M. C., Lecuyer, A., Lamarche, F., & Arnaldil, B. (2007). A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces. *Journal of neural engineering, 4*(2).

MEG. (2011). Retrieved from http://www.supraconductivite.fr/en/index.php?p=applications-medical-meg

Mento, G. (2013, December 12). *The passive CNV: carving out the contribution of task-related processes to expectancy*. Retrieved from frontiers in Human Neuroscience: http://journal.frontiersin.org/article/10.3389/fnhum.2013.00827/full#B9

Millett, D. (2001). Hans Berger: From Psychic Energy to the EEG. *Perspectives in Biology and Medicine, 44*(4), 522-542.

Nieuwenhuys, R., Voogd, J., & Huijzen, C. v. (2008). The Human Central Nervous system. New York.

Nolte, J. (1988). *The human brain : an introduction to its functional anatomy.*

nyest.hu. (2015, January 11). *What happens in our brains when we learn language?* Retrieved from http://www.nyest.hu/media/mi-tortenik-az-agyunkban-amikor-nyelvet-tanulunk.png?large

Palaniappan, R. (2006). Utilizing Gamma Band to Improve Mental Task Based Brain-Computer Interface Design. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 14*(3), 299-303.

Picton, T. W. (1992). The P300 Wave of Human Event-Related Potential. *Journal of Clinical Neurophysiology, 9*(4), 456-479.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Rakotomamonjy, A., Guigue, V., Mallet, G., & Alvarado, V. (2005). Ensemble of SVMs for Improving Brain Computer Interface P300 Speller Performances. In W. Duch, J. Kacprzyk, E. Oja, & S. Zadrożny, *Artificial Neural Networks: Biological Inspirations – ICANN 2005* (pp. 45-50). Springer Berlin Heidelberg.

Roche-Labarbe, N., Fenoglio, A., Radhakrishnan, H., Kocienski-Filip, M., Carp, S. A., Dubb, J., . . . Franceschini, M. A. (2014). Somatosensory evoked changes in cerebral oxygen consumption measured non-invasively in premature neonates. *NeuroImage, 85*, 279-286.

Scharnowski, F., Hutton, C., Josephs, O., Weiskopf, N., & Rees, G. (2012). Improving Visual Perception through Neurofeedback. *The Journal of neuroscience : the official journal of the Society for Neuroscience, 32*(49), 17830.

Schliebs, S., & Kasabov, N. (2013). Evolving spiking neural network - a survey. *Evolving Systems, 4*(2), 87-98.

Schwartz, E. S., Edgar, J. C., Gaetz, W. C., & Roberts, T. P. (2010). Magnetoencephalography. *Pediatric Radiology, 40*(1), 50-58.

Serendip Studio. (2015). Retrieved from http://serendip.brynmawr.edu/exchange/brains/definitions/def-cerebellum

Smolinski, D. T., Milanova, P. M., & Hassanien, P. A.-E. (2008). *Applications of Computational Intelligence in Biology.* Berlin: Springer.

Spinal Injury Network. (2009). Retrieved from http://www.spinal-injury.net/spinal-injury-network/images/spine.jpg

Stufflebeam, R. (2008). *Neurons, Synapses, Action Potentials, and Neurotransmission*. Retrieved from http://www.mind.ilstu.edu/curriculum/neurons_intro/neurons_intro.php

Talairach, J., & Tournoux, P. (1988). *Co-planar stereotaxic atlas of the human brain: 3-dimensional proportional system—an approach to cerebral imaging.* New York: Thieme Medical Publishers.

Tan, D., & Anton, N. (2010). *Brain-computer interfacesapplying our minds to human-computer interaction.* Springer.

Tanaka, K., Matsunaga, K., & Wang, H. O. (2005). Electroencephalogram-Based Control of an Electric Wheelchair. *Robotics, IEEE Transactions on, 21*(4), 762-766.

Taylor, D., Saywell, N., Scott, N., Chen, Y., Kasabov, N., Hu, J., . . . Tu, E. (2014). Feasibility of NeuCube SNN architecture for detecting motor execution and motor intention for use in BCIapplications. *Neural Networks (IJCNN), 2014 International Joint Conference on*, (pp. 3221-3225).

Townsend, G., Graimann, B., & Pfurtscheller, G. (2004). Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 12*(2), 258-265.

Ulmer, S., & Jansen, O. (2013). fMRI :basics and clinical applications. Springer Berlin Heidelberg. doi:10.1007/978-3-642-34342-1

Vidal, J. J. (1977). Real-time detection of brain events in EEG. *Proceedings of the IEEE, 65*(5), 633-641. doi:10.1109/proc.1977.10542

Voos, A., & Pelphrey, K. (2013). Functional Magnetic Resonance Imaging. *Journal of Cognition and Development, 14*(1), 1-9.

Wolpaw, J. R. (2007). Brain–computer interfaces as new brain output pathways. *The journal of physiology, 579*(Pt 3), 613–619.