

EEG CLASSIFICATION USING MACHINE LEARNING TECHNIQUES (SVM AND ELM)

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
[SIGNAL PROCESSING AND DIGITAL DESIGN]

Submitted by:

SAURABH KUMAR

2K17/SPD/13

Under the supervision of

MR. RAJESH BIROK



**DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

JULY, 2019

**DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, SAURABH KUMAR, Roll No. 2K17/SPD/13 student of M.Tech. (Signal processing and Digital Design), hereby declare that the Thesis titled “EEG Classification Using Machine Learning Techniques (SVM and ELM)” which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

(SAURABH KUMAR)

Date: 10 July 2019

**DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Thesis titled “EEG Classification Using Machine Learning Techniques (SVM and ELM)” which is submitted by SAURABH KUMAR, Roll No. 2K17/SPD/13, Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: 10 July 2019

(RAJESH BIROK)

SUPERVISOR

ASSOCIATE PROFESSOR

DEPARTMENT OF ECE, DTU

ABSTRACT

The EEG (Electroencephalogram) signal shows the electrical activity of the brain. They contain useful information about the brain state and their nature is highly random.

Electroencephalography (EEG) is a method to record the electrical activity of the brain. It is typically non-invasive, with the electrodes placed along the scalp. EEG refers to the recording of the brain's spontaneous electrical activity over a period of time, as recorded from multiple electrodes placed on the scalp. Spectral content of EEG is generally focused by diagnostic applications.

In this thesis project, the two EEG signals that were analyzed were obtained by recording the EEG activity that was occurring when a person was moving their arms for the first case and their legs for the second case. A power line rejection notch filter, power spectral density analysis, Principal Component Analysis (PCA) were used to pre-process these EEG signals, and finally, the mathematically based machine learning analysis is done using both Support Vector Machine (SVM) and Extreme Learning Machine (ELM).

Machine learning is used to generate a model that could be used to accurately predict whether a person is moving their arms or their legs by applying the EEGs as inputs to the generated model and reading the output of the model.

The goal of this thesis is to compare the performance of SVM and ELM by using the accuracy of classification that each model produces.

Matlab results compared ELM and SVMs with classification accuracy of both models.

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide **Mr. RAJESH BIROK (Associate Professor, ECE)**, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

At the end I would like to express my sincere thanks to all friends and others who helped me directly and indirectly during this project work.

SAURABH KUMAR

Roll no: 2K17/SPD/13

M.TECH. (SIGNAL PROCESSING AND DIGITAL DESIGN)

Department of Electronics & Communication Engineering

Delhi Technological University

Delhi – 110042

CONTENTS

Candidate's Declaration	i
Certificate	ii
Abstract	iii
Acknowledgement	v
Contents	vi
List of Figures	vii
List of Tables	viii
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 EEG PRE-PROCESSING	6
2.1 Extreme Learning Machine	6
2.2 Single-Hidden Layer Feed-Forward Networks	9
CHAPTER 3 CLASSIFICATION	10
3.1 Machine Learning	10
3.1.1 Support Vector Machine	11
3.1.2 Extreme Learning Machine	17
3.1.2.1 Single-Hidden Layer Feed-Forward Networks	19
CHAPTER 4 NUMERICAL EXPERIMENT	22
CHAPTER 5 CONCLUSION	29
REFERENCES	

List of Figures

Figure 1: The areas of different brain signal processing	2
Figure 2: Electrode Placements over Scalp	3
Figure 3: Compare the rectangular window with hamming window	8
Figure 4: Linear SVM separate two different kinds of data	12
Figure 5: The Soft margin property of SVM	13
Figure 6: Mapping of original input space to the higher dimensional feature space	15
Figure 7: The workflow picture of Extreme Learning Machine	17
Figure 8: Mean of each channel signal in time domain	23
Figure 9: Power density spectrum of Class 1 and Class 2	24

List of Tables

Table i: The classification results of ELM Kernel Function	25
Table ii: The classification results of SVM	25
Table iii: The results of ELM and SVM with different PCA components and c values	27

CHAPTER 1

INTRODUCTION

Brain Computer Interface (BCI) is a way to communicate between the brain and an external device, such as a computer. By using sensors put on the scalp, the external device can catch the brain response signal and do some processing, while the body is doing some movements or the brain is thinking. An application of BCI is intelligent prosthesis, which use the brain signal to control the fake leg or arm movements. In this thesis, the BCI input signal is the EEG signal that results from movement of the legs and hands.

For most BCIs, an electroencephalogram (EEG) is an electric recording of brain activity caused by extremely sensitive currents and the EEG signal, which is considered as the source signal, is sent to a computer to do the signal processing. For the brain signal measurement, the electroencephalogram (EEG) is one of the main methods, which provides a non-invasive measurement without implanting any foreign object inside the body. Depending on different situations, conditions and thought, the brain produces a different EEG signal, which can be captured by the sensors on the scalp. Because the amplitude of the EEG signal is very small, it is difficult to acquire the signal and once it is acquired, it often very noisy and requires filtering. The type of noise that surrounds the signal of interest is in the form of background noise or other signals generated by the brain that are not pertinent to the particular application of interest. This is because of an external disturbance or other mental activity.

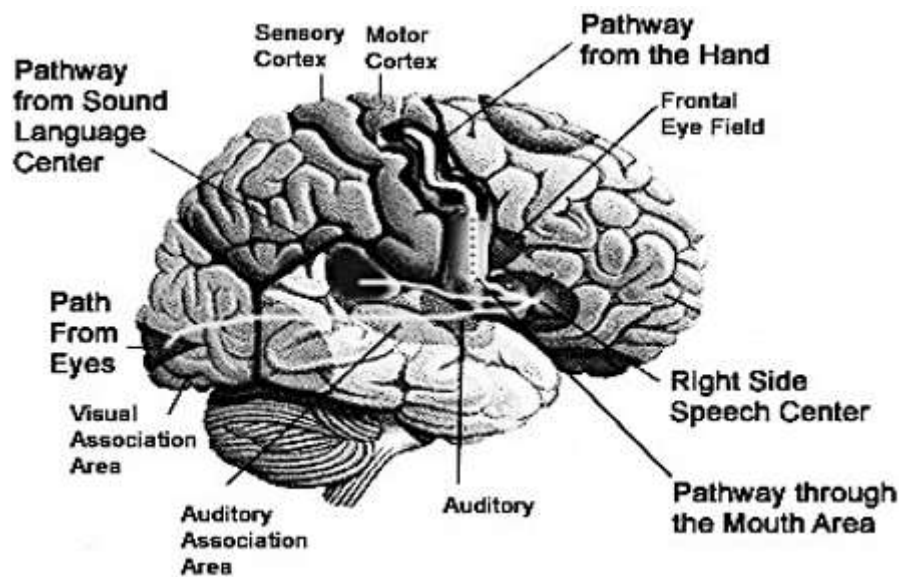


Figure 1: The areas of different brain signal processing

Different activities will cause different parts of the brain to process more intensively or relax more, so by analyzing the origin of a signal, one can gain insight as to what kind of activity is performed. Besides the origin of the signal, the amplification or attenuation of the signals, as well as the frequency of the signal help categorized the type of signal being observed. The frequency bands often used in analyzing EEG signals are defined as Delta (0 Hz – 4 Hz), Theta (4 Hz – 7 Hz), Alpha (8 Hz – 12 Hz), Beta (12 Hz – 30 Hz), Gamma (30 Hz – 100 Hz). The excess or lack of activity can be used to measure the state of a person.

If the brain of an awake adult is experiencing large amount of delta activities or theta activities, it means the person has some neurological diseases. A person's state of alertness can be predicted by looking at the amplitude of alpha rhythms, which increase when the body is relaxed and the eyes are closed, otherwise it attenuates. Beta rhythms relate with muscle movements. The gamma waves relate to certain muscle functions and perceptions [1].

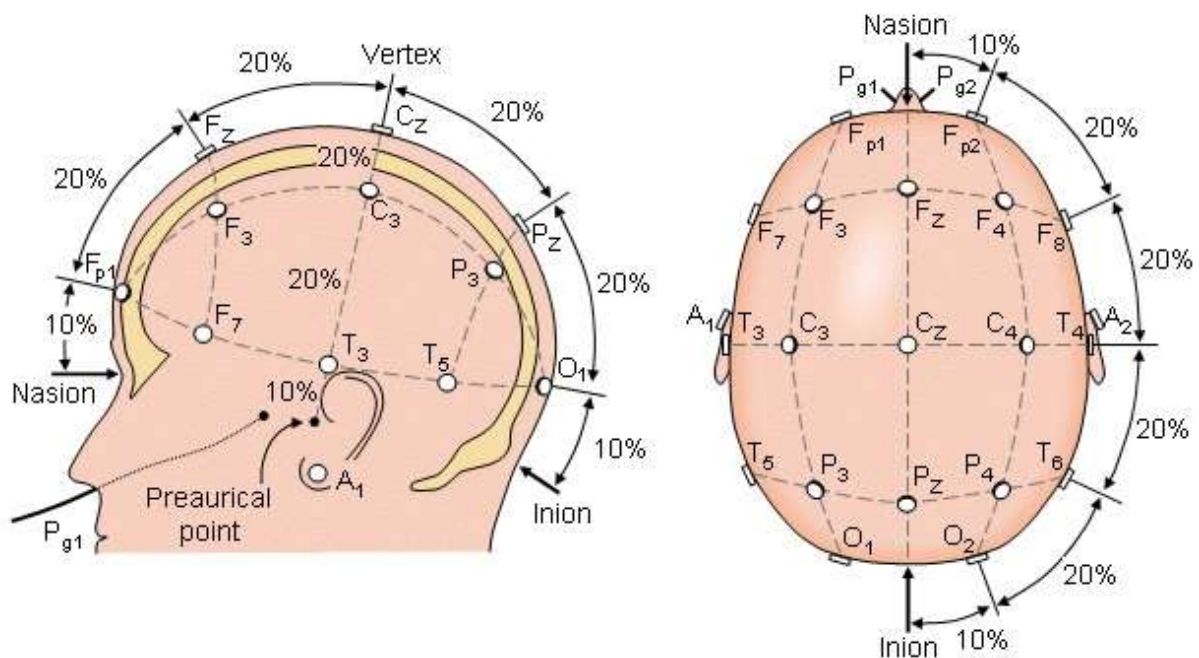


Figure 2: Electrode Placements over Scalp

Machine learning plays a major role in BCI systems. In this thesis, it is used to learn the patterns in EEG activity for cases when a person is moving their arms and when they're moving their legs. There are two main applications of machine learning: regression and classification. The regression analysis is used for forecasting and prediction. The regression analysis is a process for estimating the relationships and the conditional expectation among the dependent variable and one or more independent variables, which includes many techniques for modeling and analyzing several variables. The estimation target is the regression function which is a function of the independent variables. [2].

In regression, the outputs of the system are continuous rather than discrete. On the other hand, classification is the process of identifying which set of discrete categories certain data belongs to. Machine learning can generate a model, which sets boundaries to classify data as belonging to one class or another. This is achieved by first extracting features from the training dataset and then progressively updating the model to increase the accuracy with which it predicts the class of training data. The category of machine learning that contains a label for each of the

trials in a dataset is called supervised learning. This data and the labels are used to generate a mathematical model that can be used to predict which class other EEG data belongs to. This is done by feeding features from a trial into the model and looking at the predications that the model makes. If the model works well, it will correctly predict which class the data belongs to. To build this model, training data is required to progressively help shape the parameters in the model so that it can be trained to make data driven predictions and perform accurate signal classification on future data.

The two types of data in machine learning are training data and testing data. Training data is used to build the model by providing some feedback to the system that indicates which class the data belongs to. This allows the model to constantly change its parameters to allow the training data to be classified correctly. The ultimate goal of machine learning does not lie in the accuracy of prediction for the training data, but instead what is most desired is to have the model be general enough that the model can correctly classify future data that the model has not seen before. To get a measure of how good the model will perform, testing data is used to test the predications that the model produces and check the accuracy of the model. This testing data is data that the model has not seen before and therefore this accuracy is a measure of how accurate it is in the general case (not just for data that it used to generate the model).

In this thesis, there are two machine learning algorithms used: Support Vector Machine (SVM) and Extreme Machine Learning (ELM). The models that were generated in this thesis are specifically catered to learning the difference between the patterns in the two classes of EEGs (hands and legs). More details about machine learning are described in Chapter 3.

The dataset provided for this thesis is large and not every single point of data is needed to describe the characteristics or pattern of the signal. The system attempts to classify the data as being arm movement related or leg movement related using whatever data is provided to it, but for better processing efficiency, the number of data points that are processed can be reduced by extracting certain patterns from the data and then use this as the input to the learning algorithm. This provides a more efficient way of processing the signal, since instead

of processing a huge set of data, the system only has to process a few of patterns/features. The process of extracting these patterns is called feature extraction and is discussed in Chapter 2.

On this thesis, the purpose is to compare two machine learning models performances: Support Vector Machine (SVM) and Extreme Machine Learning (ELM) by measuring the accuracy of the predictions that each model makes based on the analysis and feature extraction of the electroencephalogram (EEG) signals for the movement of either the legs or arms moving. The procedure followed in this project can be summarized as following: run the data through filters, use PCA to extract features of the filtered data set, run the features through a machine learning algorithm (SVM and ELM) to generate a model, test the model and record accuracies of the learning algorithm using different machine learning functions, parameters and models.

In Chapter 2, the thesis describes the pre-processes, which include the following steps: transform the dataset from time domain to frequency domain (power density spectrum) and feature extraction.

In Chapter 3, the thesis describes the mathematical details of ELM and SVM.

Thesis details, results and the conclusion can be found in Chapter 4 and Chapter 5.

CHAPTER 2

EEG PRE-PROCESSING

The datasets used in this thesis is the EEG activity signals, which are collected over 15 channels and are recorded at a sampling frequency of 512 Hz when the hands or foot of participants were moving. These signals are band pass filtered from 0.5 Hz to 100 Hz because the useful parts of the collected brain response signals are lower than 100Hz. There is a very strong noise at 50 Hz because the data used in this project is from Germany, which will affect the result. Therefore, there are four-tap notch filter in the system, which cleans the power noise as much as possible. The data used in the project is after the filtration and contains only the relevant portions from the data.

2.1. Frequency Transformation

The data is transformed from time domain to frequency domain so that the latent periodic components can show up and be identified. It is also easier to see the data differences in frequency domain and figure out which frequency bands of the brain signal are active, when the body is moving the arms or legs.

In this thesis, the time domain EEG signals are transformed into the frequency domain using spectrum density estimation to obtain the power density spectrum. The power spectrum shows the variation of power across different frequencies, which can also be used to show the energy distribution. In statistical signal process, spectrum density estimation is a tool to estimate the spectral density of a random signal from a sequence of time samples of the signal, which is a

way to describe the frequency features and content of spectrum density. There two main kinds of SDE: parametric SDE and non-parametric SDE [3].

For non-parametric spectrum estimation, no special model is assumed to generate the frequency response data. In this case, it can use Fourier methods to analyze the power in the frequency domain. The parametric spectrum estimation uses a predefined model to represent the data in the frequency domain. These models require calculating certain parameters to fit the model to the data.

In this thesis, Welch's method which is a non-parametric spectrum estimation method was used to transform the time domain signal into one in the frequency domain. Welch's method is based on the standard density spectrum estimation and one of its features is that in exchange for reducing the frequency resolution to the desire of the user, it can also be used to reduce noise. In this method, the signal data segments are split up into L data segments, which have M points, and is overlapped by D points.

If $D = M/2$ then overlap is 50% and if $D = 0$ then the overlap is 0%.

Each of these data segments can then be multiplied with a window function and the periodogram of the product is calculated for the individual segments and the average of these periodograms is the result of Welch's method. The equations for the process that was just described are shown below.

$$\tilde{P}_{xx}^{(i)} = \frac{1}{MU} \left| \sum_{n=0}^{M-1} x_i(n)w(n)e^{-j2\pi fn} \right|^2, i = 0,1, \dots, L-1 \quad 2.1$$

$$U = \frac{1}{M} \sum_{n=0}^{M-1} w^2(n) \quad 2.2$$

$$P_{xx}^W(f) = \frac{1}{L} \sum_{i=0}^{L-1} \bar{P}_{xx}^{(i)}(f) \quad 2.3$$

The data used in this thesis is cut into no more than 8 segments with 50% overlap and the modified periodogram is computed for each segment using a Hamming window. Compared to the rectangular window, the magnitude of the side lobes for the hamming window is much less than that of the rectangular, which is desirable, but the main lobe of the hamming is also wider, which is undesirable.

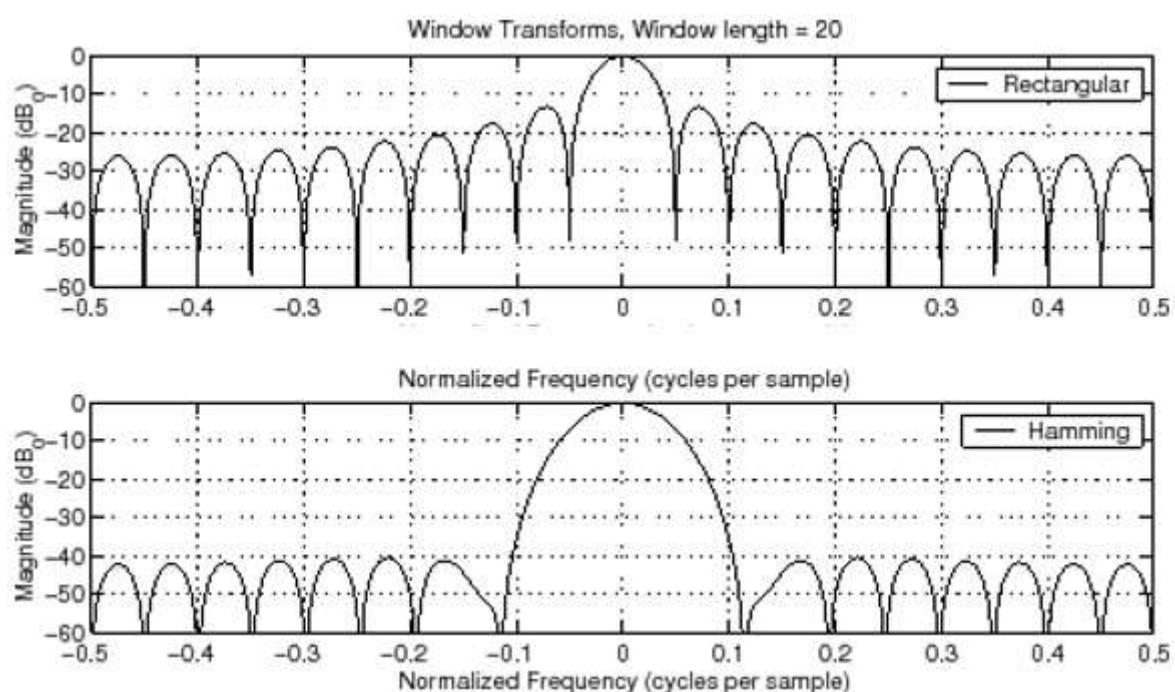


Figure 3: Compare the rectangular window with hamming window

2.2. Feature Extraction

A feature is a noticeable characteristic, which can describe the data. As mentioned in Chapter 1 Introduction, there is a large set of data provided in this thesis. The thesis used feature extraction to get the features in order to process data more efficiently, which can be very useful to describe the pattern or trend, since they provide a more compact way of describing the pattern.

Principal Component Analysis (PCA) is a very effective algebraic feature extraction method to deal with the information in the sample, compression and extraction is based on the variables covariance matrix. The core idea of PCA is using the characteristics of less data to describe samples in order to reduce the dimension of the feature space. PCA reduce the dimensions of the data sets, while keeping the largest contribution to the variance of characteristics of the data set, by retaining lower-order principal component and ignoring higher-order principal component. In this way, the most important aspect of the data is retained by lower-order components.

To build up the PCA model, the first step is to get the mean of each channel and subtract the mean from every data dimensions in order to set the means of every dimension data to zero. Then, the covariance matrix is calculated using the zero mean data. After this, the eigenvalues and eigenvectors of the covariance matrix are found, the eigenvectors are sorted by the eigenvalues from highest to lowest, which results in data from most significance to lowest significance. Lastly, components of lowest significance can be ignored without much loss in information and multiply the original data with the eigenvector matrix with to get the new data.

CHAPTER 3

CLASSIFICATION

3.1. Machine Learning

Machine learning is an algorithm operation that builds up a model with the training data and uses the model to process some test data. Machine learning is used in many applications. Supervised learning and classification are used in this project. To supervise the learning process, the data provided to the system must have the predicted class labels for each set of features. The feature space for the training data is divided into two classes by using the labels and the algorithm learns the general rules to build up the model. Then this model is used to classify the testing data and to calculate the accuracy of the system.

There are two kinds of machine learning used in this project: the first is Support Vector Machine (SVM), and the second is Extreme Machine Learning (ELM).

3.1.1. Support Vector Machine

Support Vector Machine (SVM) is a traditional algorithm operation to separate two or more kinds of data. It relies on a mathematically defined function that represents a division, known as a hyper plane in higher dimensions, between the two classes of data. Data lying on one side of this hyper plane is classified as belonging to one class, while data on the other side belongs to the other class.

By applying SVM, a classification model in the form of linear function $f(x) = wx+b$ can be generated, which can separate data to the classes.

The location of the hyper plane is defined as $f(x) = 0 = wx+b$.

The example in Figure 4 shows some blue circles on the left side, which represent data belonging to Class 1, and some red circles on the right side, which represent data belonging to Class 2. The blue circles are on the side of the hyper plane, which corresponds to the case when $f(x) > 0$ and the red circles are on the side of the hyper plane, which correspond to the case where $f(x) < 0$.

By looking at the location of the data with respect to the hyper plane, the class that the data belongs to can be predicated by the model.

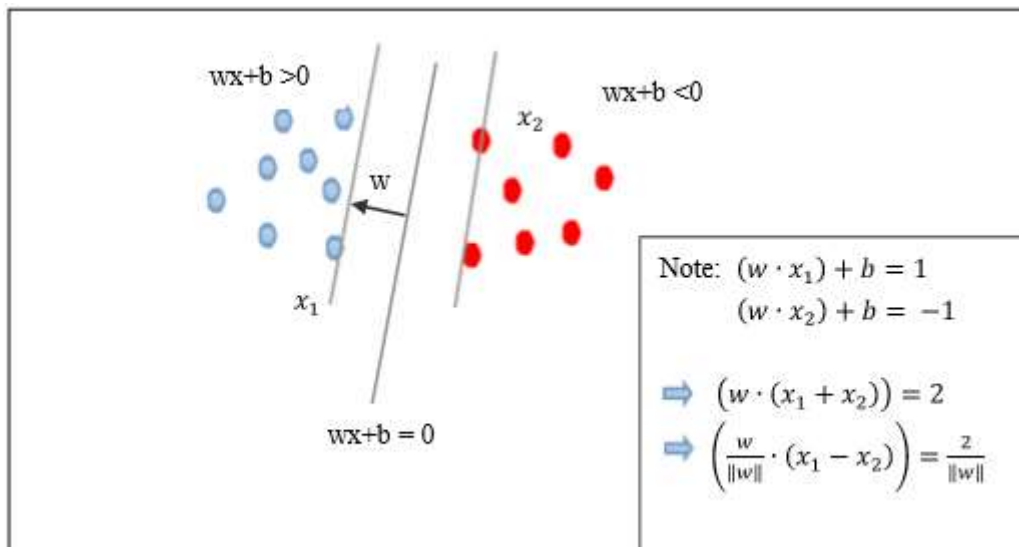


Figure 4: Linear SVM separate two different kinds of data

From Figure 4, there are two parallel lines beside the line at $wx+b = 0$. The line is used to separate between two classes (red circles vs blue circles). The wider the area between the red circles and blue circles, the easier it is to separate the circles and therefore, the easier it is to separate between the two classes. Putting any line in between the two classes will work for the example shown in the figure, but since the goal is to make the model work for data not belonging to the training dataset, it is best to provide the widest margin between the two classes.

The best condition is when the maximum distance between two hyper-planes at the edges occurs. The distance is $\frac{2}{|w|}$, so max value of $|w|$ is needed to minimize the distance. When the w is fixed, the three lines are also fixed. As the norm of w uses the square root, instead of using $|w|$, $\frac{|w|^2}{2}$ will be used instead. Therefore, it can formulate a quadratic optimization problem and solve for w and b :

$$\min = \frac{\|w\|^2}{2} \quad 3.1$$

$$\text{subject to : } y_i(wx_i + b) \geq 1, \forall i \quad 3.2$$

$$\text{Primal classification equation: } f(x) = \text{sign}(w^T x + b) \quad 3.3$$

Where the goal is to find w and b such that $\Phi(w) = 1/2w^T w$ is minimized and for all $\{(x_i, y_i)\}: y_i(wx_i + b) \geq 1$. Quadratic optimization problems are a famous class of mathematical programming problems and many algorithms exist for solving them [3].

Quadratic programming is a special type of mathematical optimization problem, which minimize or maximize a quadratic function of several variables subject to linear constraints on the variables.

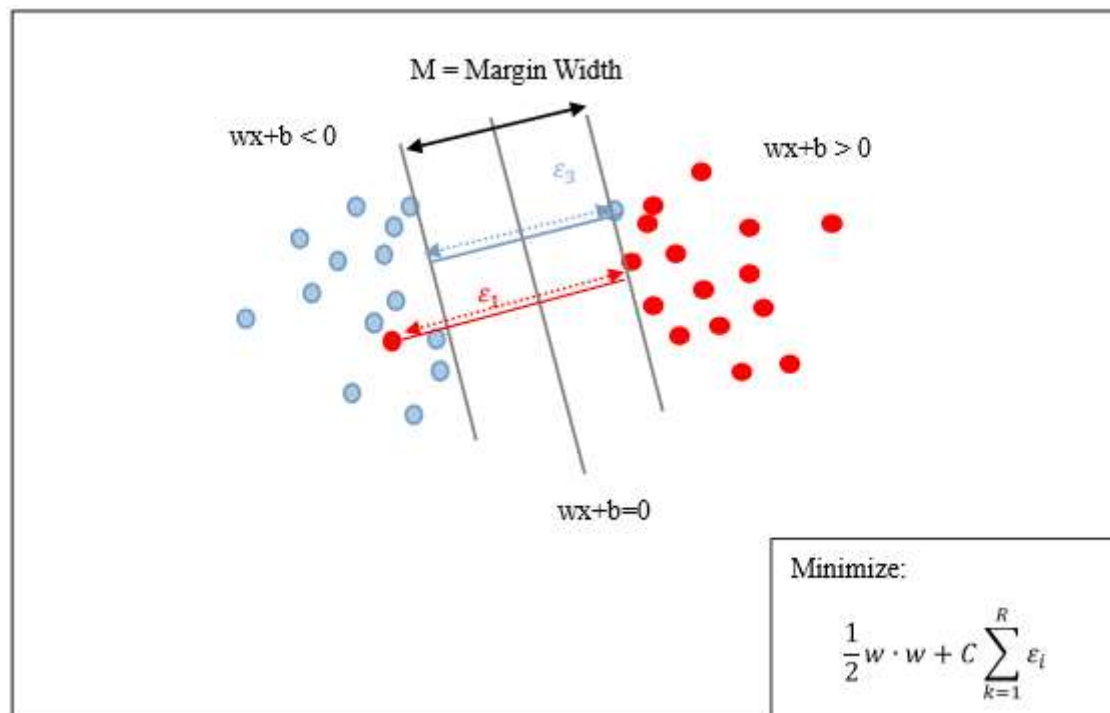


Figure 5: The Soft margin property of SVM

There will be times where there is some data that cannot be separated. From Figure 4, one or two blue circles show up on the side of red circles, which cannot generate a clear boundary to

separate the circles by color. The solution to this problem is using a concept known as soft margin.

According to the study done by Corinna Cortes and Vladimir N.Vapnik, they found the soft margin in 1995 [5]. In the situation described, the soft margin method can choose a hyper plane that keeps the maximum distance and splits the data as cleanly as possible. In this method, there are non-negative slack variables ε_i , which checks the measurement degree of the misclassified data x_i .

Now, using the soft margin, the equation changes to

$$\text{minimize: } \frac{1}{2}w \cdot w + C \sum_{k=1}^R \varepsilon_i \quad 3.4$$

$$\text{subject to : } y_i(wx_i + b) \geq 1 - \varepsilon_i \geq 0, \forall i \quad 3.5$$

For dual problem [5]:

$$\text{max: } Q(\alpha) = \sum \alpha_i - 1/2 \sum \sum \alpha_i \alpha_j y_i y_j x_i^T x_j \quad 3.6$$

$$\text{Subject to: } \sum \alpha_i y_i = 0 \quad \alpha_i \geq 0 \text{ for all } \alpha_i \quad 3.7$$

$$\text{classifying function : } f(x) = \sum \alpha_i y_i x_i^T a_j + b \quad 3.8$$

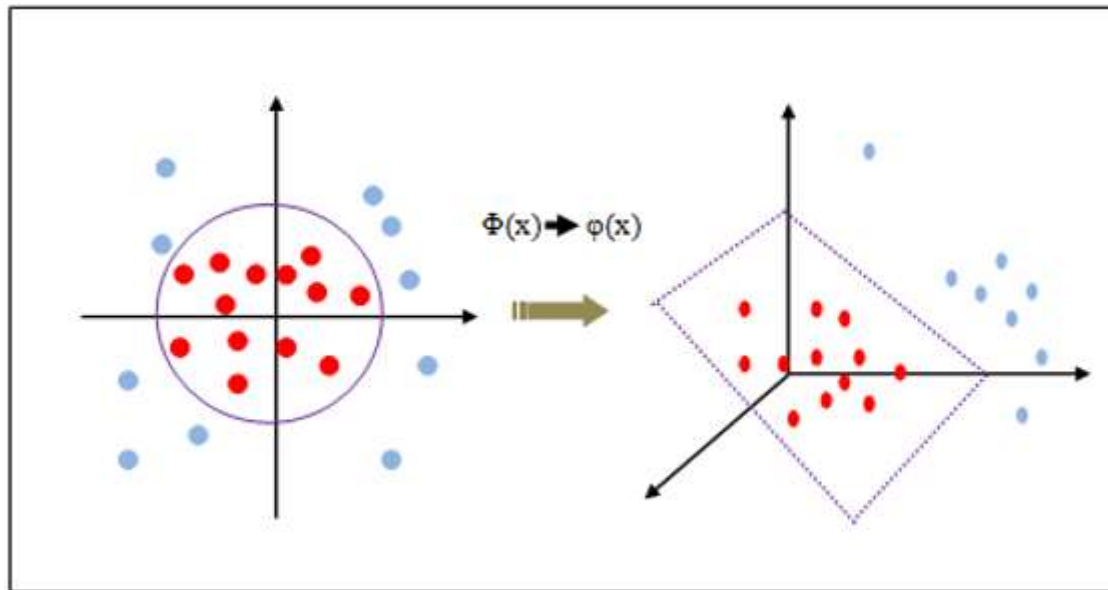


Figure 6: Mapping of original input space to the higher dimensional feature space

Since the data is not always linearly separable in the initial feature space, there are occasions where increasing the dimensionality of the feature space may actually make the separation of classes much easier. To perform this process, the kernel function is used to transform the initial feature space into a higher order feature space, which can make the data be linearly separated by a hyper-plane and achieve the maximum margin between the two classes [6].

As shown in Figure 5, the initial data is transformed from $\Phi(x)$ to $\varphi(x)$ so that the dot product becomes:

$$k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \quad 3.9$$

And the equation (3.1) changes to:

$$\text{Learning} : \min \|W\|^2 + \sum_i^N \max(0, 1 - y_i f(x_i)) \quad 3.10$$

$$\text{classifying function} : f(x) = \text{sign}(w^T \varphi(x) + b) \quad 3.11$$

Instead of $x_i^T x_j$, which shows in Eq.6, the non-linear SVM mathematical classification uses $K(x_i, x_j)$ and for dual classifier in transformed feature space:

$$\max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k k(x_j, x_k) \quad 3.12$$

$$\text{Subject to : } 0 \leq \alpha_i \leq C \text{ for } \forall i \text{ and } \sum_i \alpha_i y_i = 0 \quad 3.13$$

And the classifying function is shown as following:

$$\text{Classifying function : } f(x) = \sum \alpha_i y_i k(x_i, y_i) + b \quad 3.14$$

3.1.2. Extreme Learning Machine

Huang Guang-Bin presented Extreme Learning Machine in 2004 [7]. At first, ELM was designed for the single hidden layer feed-forward neural networks (SLFNs). Then, it is developed and extended to the generalized single hidden layer feed-forward networks [8] [9] [10] [11].

The idea of ELM is to work for the SLFNs, and extends to the multi-hidden layer feed-forward neural networks. The ELM theory assumes that the hidden nodes or the neurons can be generated randomly as shown in Figure 7. The hidden nodes parameters are independent from the training data set. The ELM theories state that the randomness may be the real method for how the brain learns.

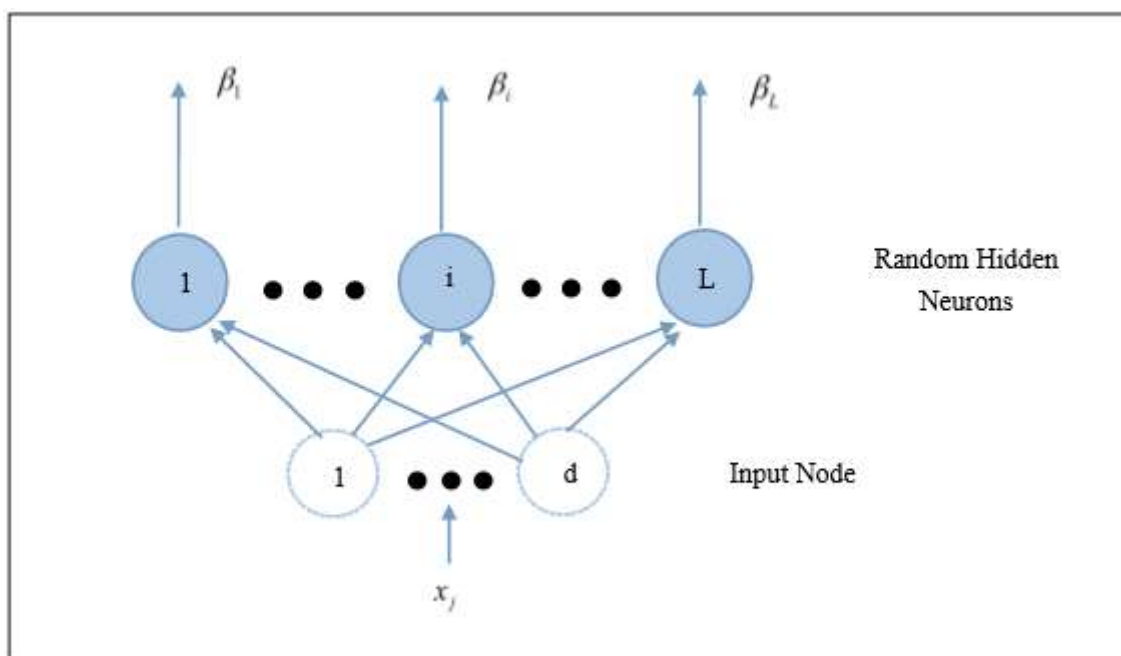


Figure 7: The workflow picture of Extreme Learning Machine

In order to describe the ELM model further, the following parameters need to be defined:

- ❖ N : the total number of the training data
- ❖ \tilde{N} : the total number of hidden nodes
- ❖ n, m : the dimensions of the input layer and output layer

- ❖ $(x_j + t_j), j = 1, 2, \dots, N$: the testing data

$$x_j = (x_{j1}, x_{j2}, \dots, x_{jn})^T \in R^n, t_j = (t_{j1}, t_{j2}, \dots, t_{jm})^T \in R^m$$

Put the output vectors together to get the output matrix:

$$T = \begin{bmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N1} & t_{N2} & \dots & t_{Nm} \end{bmatrix}$$

- ❖ $o_j, j = 1, 2, \dots, N$: the particular output vector according to the t_j
- ❖ $W = (w_{ij})_{\tilde{N} \times n}$: the related weight matrix of input layer and hidden layer

$$w_i = (w_{i1}, w_{i2}, \dots, w_{in})^T$$

- ❖ $b = (b_1, b_2, \dots, b_{\tilde{N}})^T$: the offset vector, b_i is the threshold of the i^{th} hidden node

- ❖ $\beta = (\beta_{ii})_{\tilde{N} \times m}$: the related weight matrix of output layer and hidden layer

$$\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})^T$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{\tilde{N}1} & \beta_{\tilde{N}2} & \dots & \beta_{\tilde{N}m} \end{bmatrix}$$

- ❖ $g(x)$: active function

3.1.2.1. Single-Hidden Layer Feed-Forward Network (SLFNs)

Mathematically, the standard SLFNs model is the following:

$$\sum_{i=1}^{\bar{N}} g(w_i \cdot x_i + b_i) \beta_i = t_j, \quad j = 1, 2, \dots, N \quad 3.15$$

In order to have zero error and be closer to the N samples,

$$\sum_{j=1}^N \|o_j - t_j\| = 0 \quad 3.16$$

So the Eq. 3.10 change to

$$\sum_{i=1}^{\bar{N}} g(w_i \cdot x_i + b_i) \beta_i = o_j, \quad j = 1, 2, \dots, N \quad 3.17$$

Use the matrix to describe

$$H\beta = T \quad 3.18$$

H is the hidden layer output matrix of neural network. $H = H(W, b) = (h_{ij})_{N \times \bar{N}}$ And

$$h_{ij} = g(w_j \cdot x_i + b_j).$$

When the number of the hidden units is odd and $\tilde{N} = N$, and the matrix is revertible.

The number of hidden-layer units is much smaller than the number of training data, which means $\tilde{N} \ll N$. To solve the difficulties, a new effective learning machine appears. At first, the values of parameters 'W' and 'b' can be given randomly and used to calculate the matrix 'H' (although keep changing the values of 'W' and 'b' does not result in any gain change.) 'β' is the only parameter is used as a decision factor.

When 'W' and 'b' is fixed, Eq. 3.19 is used to solve for the least square solution $\hat{\beta}$ of the linear system.

$$\|H\hat{\beta} - T\| = \min_{\beta} \|H\beta - T\| \quad 3.19$$

$$\hat{\beta} = HT \quad 3.20$$

On the other hand, $\hat{\beta}$ is also the minimum norm, which means $\hat{\beta}$ is the minimum norm of all the least square solutions. For the feed-forward neural networks, the magnitude of $\hat{\beta}$ is very important: $\hat{\beta}$ is smaller, while the generalization ability of the system is better.

The Extreme Learning Machine build up process:

Given the training data set $R = \{(x_i, t_i)\}_{i=1}^N$, where $x_i \in R^m$, $t_i \in R^m$, the activation function $g(x)$ and the number of hidden units \tilde{N} .

Step 1: Assign input weights w_i and threshold b_i , $i = 1, 2, \dots, \tilde{N}$.

Step 2: Calculate the hidden layer output matrix of neural 'H'.

Step 3: Calculate the related weight matrix of output layer and hidden layer $\beta = HT$.

CHAPTER 4

NUMERICAL EXPERIMENT

The data used in the thesis are EEG activity signals, which download from the website: <http://bnci-horizon-202.eu> [9]. The data includes the training data, the testing data, trials and predicted class labels. Trials are used to separate the data for each sample and the predicted class labels assist the training and the calculation of the accuracy for the classification processes. The training data and the testing data had 15 channels and were recorded using a sampling frequency of 512 Hz, when the hands and foot are moving.

There are 160 samples: 100 training sample data and 60 testing sample data, where each one has over 11000 points. Because the frequency of the signals is lower than 100 Hz, which include the main useful information, the EEG signals are band-pass filtered from 0.5Hz to 100Hz. There is also a four-tap 50 Hz notch filter in the system to clean the power noise. The data used in the thesis only contains the relevant portions.

After loading the data to MATLAB, the mean of each class of 15 channels in time domain is plotted, which are shown in the following Figure 8. Then the data is transformed from time domain to power density spectrum through the non-parametric model Welch, which has been introduced in Chapter 2. In MATLAB, the function ‘pwelch’ is used to perform the operation and the result is showed as Figure 9.

The last step of the data pre-process is to let the data run through PCA to get the features extracted.

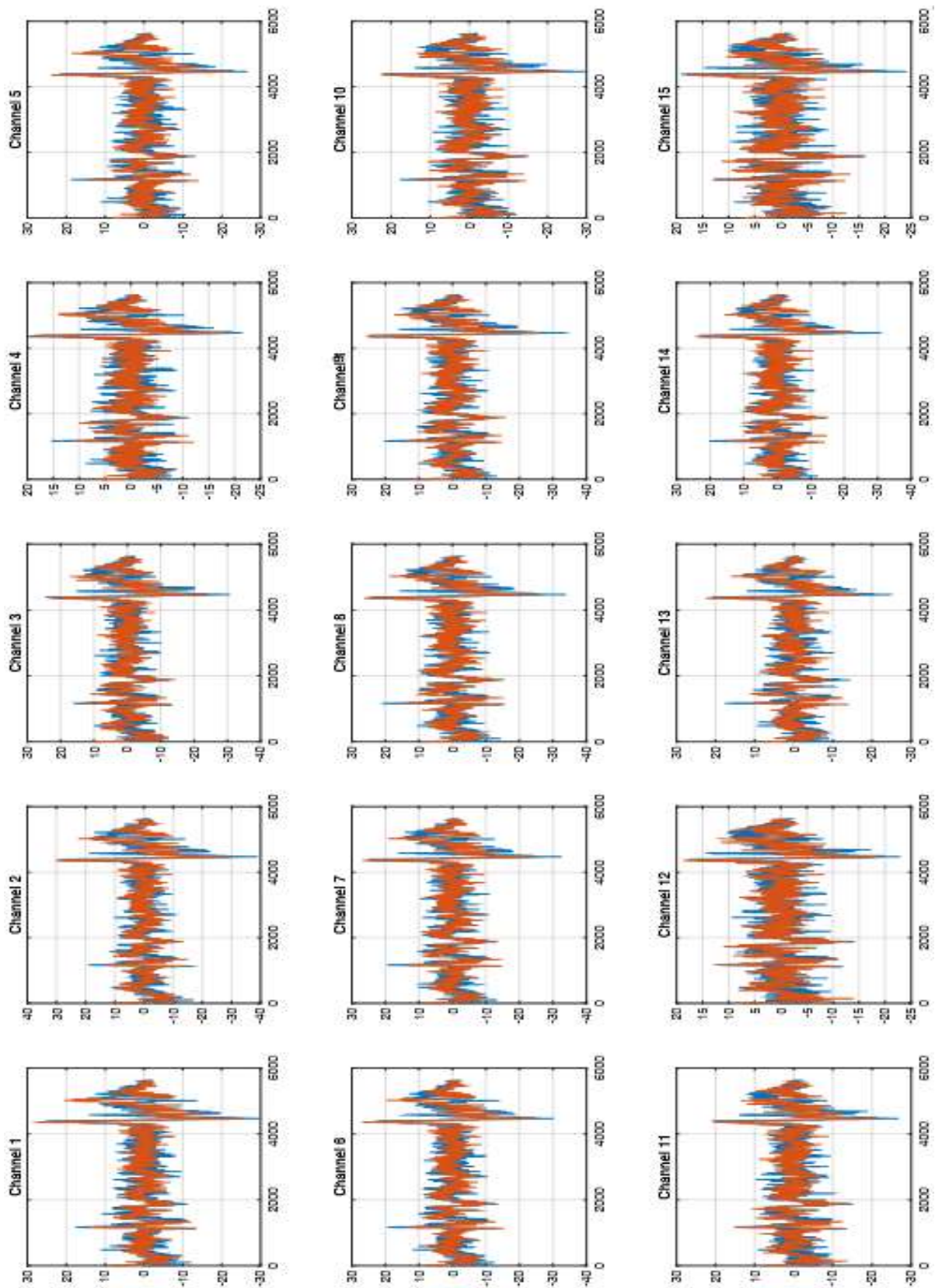


Figure 8: mean of each channel signal in time domain

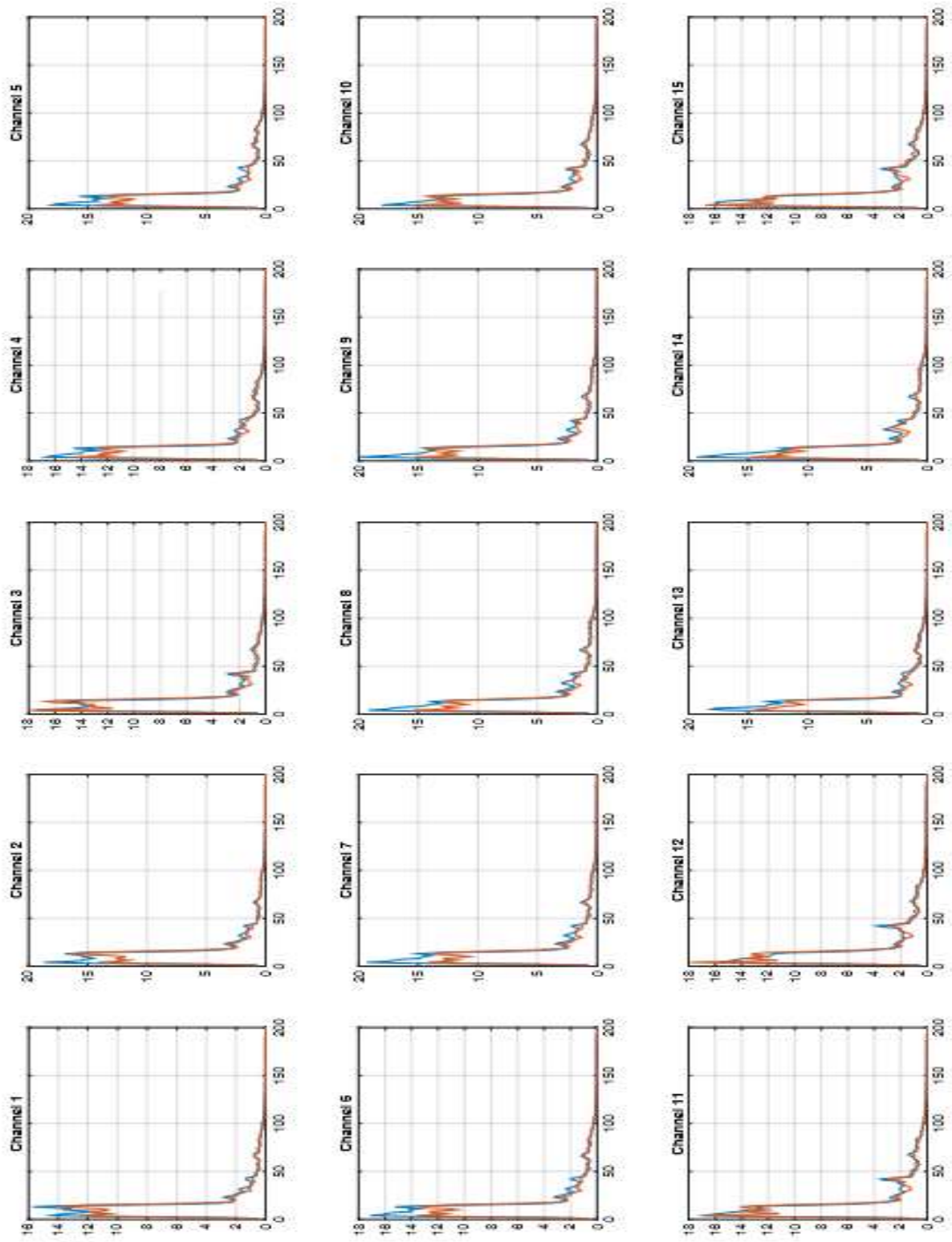


Figure 9: Power density spectrum of Class 1 and Class 2

This model was built by Prof. Huang Guang-Bin study team, which sets up the theories and models of ELM [11].

For SVM model, it is build up by the MATLAB functions: svmtrain.

For Table i and Table ii, they are the classification results of ELM and SVM kernel function with all features, which are pass the PCA method. There are two kind modes of kernel function doing the comparison. One is Linear Kernel Function, and the other is Polynomial.

Table i: The classification results of ELM Kernel Function

MODE OF KERNEL FUNCTION	TRAINING ACCURACY	TESTING ACCURACY
LINEAR	100%	68.33% , C = 0.001
POLYNOMIAL	100%	60% , C = 0.001

Table ii: The classification results of SVM

MODE OF KERNEL FUNCTION	TRAINING ACCURACY	TESTING ACCURACY
LINEAR	100%	68.33% , C = 0.001
POLYNOMIAL	100%	68.33% , C = 0.001

According to the tables, it is very clear that for the linear kernel mode, the performance of ELM and SVM are same. However, for polynomial, there is a noticeable difference between them, where the ELM performs better.

After comparing two methods with total data, the projects also need to find out how the method performances change when the method conditions and parameters are changed. No matter which model is used, SVM or ELM, there is a very important parameter C , which is a way to control over fitting. Higher values of C indicate that the model sets a harder margin on the classification, while smaller values of C indicate that the model sets a softer margin. This project used K-fold Cross-validation to get the suitable C values.

The purpose of using Cross-validation is to get reliable and stable models. The initial training data is divided into K groups, one separated group is kept aside to use as testing data, the other groups are used to do the training; repeat the cross-validation for K times and each group has been verified one time, then pick up the C value, which has the highest testing accuracy. The advantage of this method is that it repeats using randomly generated groups to do the training and validation and the accuracy of each model is evaluated after each run. The method used in this project is 5-fold cross-validation.

As theories described in Chapter 2, there is very important parameter, which can affect the classifying result directly, is the number of PCA components used in the classification process. The columns of the PCA method output matrix has already sorted from low-order to high order by how much important information of EEG signal is included. This project used the MATLAB function 'pca' to build up the PCA method directly and picked up 1500, 3000, 4500, 6000, 7500, and 9000 as the numbers of PCA components.

Table iii: The results of ELM and SVM with different PCA components and c values.

PCA components	ELM WITH LINEAR KERNEL		SVM WITH LINEAR KERNEL	
	C VALUE	TESTING ACCURACY	C VALUE	TESTING ACCURACY
1500	0.0534	46.67%	0.0027	45%
3000	0.0132	53.33%	0.0010	56.67%
4500	0.0100	60%	0.0010	65%
6000	0.0010	68.33%	0.0010	68.33%
7500	0.0010	68.33%	0.0010	68.33%
9000	0.0010	68.33%	0.0010	68.33%
PCA components	ELM WITH POLYNOMIAL KERNEL		SVM WITH POLYNOMIAL KERNEL	
	C VALUE	TESTING ACCURACY	C VALUE	TESTING ACCURACY
1500	0.0534	46.67%	0.0010	45%
3000	0.0132	53.33%	0.0010	56.67%
4500	0.0100	58.33%	0.0010	65%
6000	0.0010	60%	0.0010	68.33%
7500	0.0010	60%	0.0010	68.33%
9000	0.0010	60%	0.0010	68.33%

From Table iii, it can see that as the number of PCA components used in the process of classification, the testing accuracy keeps increasing until reach the highest testing accuracy, which are shown in Table 1 and Table 2. From Table 3, it looks like that 6000 features are enough to show the most characteristic of the differences between Class 1 and Class 2. According to the table, when the find the best $C = 0.001$, C value is fixed by the system and do not change again. It can see from table that SVM find the best C value much faster than ELM and for polynomial kernel method, the testing accuracy of SVM is higher than the testing accuracy of ELM.

CHAPTER 5

CONCLUSION

The testing of the EEG signal classification of hand and feet movements is successful by using SVM and ELM. It is clear that compared with ELM, SVM has more stable performance and better accuracy on polynomial kernel method.

From the last section, the result is not perfect. The following are possible reasons:

- The amount of the training data is not enough to generate a classification model. For each class, there are only 100 samples. As the brain signal is not a simple and clear signal, only 50 samples cannot show the entire characteristic. Therefore, the accuracy, universality and applicability of the classification model are affected.
- The training data does not have universality so that the model does not work well with the test data. Training data is always limited, but this particular data is unlimited and very changeable. As the limitation of the training data, the model does not work well on the test data in this specific case.
- The dimensions of the classification are too high. During the processes of the model building, the feature space transforms to higher dimensions. As dimensions increase, the universality and applicability of the model will also decrease.

REFERENCES

- [1] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil, "Brain Computer Interfaces, a Review".
- [2] Jason Brownlee, "A Tour of Machine Learning Algorithms", on November 25, 2013 <http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
- [3] W. J. G. v. d. Wassenberg, "Signal analysis techniques for evoked and event-related potentials," University of Groningen, Groningen, 2008.
- [4] Corinna Cortes and Vladimir Vapnik, "Support Vector Networks", 1995 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
- [5] Carlos Guestrin, "SVMs, Duality and the Kernel Trick", Carnegie Mellon University, <http://www.cs.cmu.edu/~guestrin/Class/10701-S07/Slides/kernels.pdf>
- [6] N. Cristianini, Support Vector Machines and other kernel-based learning methods, Cambridge: Cambridge University Press, 2000.
- [7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks," 2004 International Joint Conference on Neural Networks (IJCNN'2004), (Budapest, Hungary), July 25-29, 2004.

[8] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, 2012. (This paper shows that ELM generally outperforms SVM/LS-SVM in various kinds of cases.)

[9] Two class motor imagery (002-2014), <http://bnci-horizon-2020.eu/database/data-sets>

[10] ELM Official Website: <http://www.ntu.edu.sg/home/egbhuang/>

[11] G.-B. Huang, "kernels," Cognitive Computation, vol. 6, pp. 376-390, 2014. (Also briefing the differences and relationships between different methods such as SVM, LS-SVM, RVFL, Quick Net, Rosenblatt's Perceptron, etc.)

[12] "Support Vector Machines (SVM)," MATLAB, [Online].

Available: <http://www.mathworks.com/help/stats/support-vector-machines-svm.html>.

[Accessed 05 May 2019].

[13] Batal, "PCA," [Online]. Available: <http://people.cs.pitt.edu/~iyad/PCA.pdf>

[Accessed 30 April 2019].

[14] Stroke Family: The Sensory Trigger Method: Making New Pathways for Speech after Stroke or Brain Injury.

[15] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in Extreme Learning Machines: A Review" Neural Networks, vol. 61, no. 1, pp. 32-48, 2015.