T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Information Technologies

ARTIFICIAL NEURAL NETWORKS FOR ELECTROENCEPHALOGRAM CLASSIFICATION

Ali Mohsin Lateef ALKHAFAJI

Master of Science

Supervisor

Prof. Dr. Osman Nuri UÇAN

Istanbul, 2021

ARTIFICIAL NEURAL NETWORKS FOR ELECTROENCEPHALOGRAM CLASSIFICATION

By

Ali Mohsin Lateef ALKHAFAJI

Information Technology

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “ARTIFICIAL NEURAL NETWORKS FOR ELECTROENCEPHALOGRAM CLASSIFICATION “prepared and presented by “Ali Mohsin Lateef ALKHAFAJI “ was accepted as Master of Science Thesis in Information Technology.

____

Prof. Dr. Osman Nuri UÇAN

Supervisor

Thesis Defense Jury Members:

Prof. Dr. Osman Nuri UÇAN	School of Engineering and Natural Sciences, Altinbas University	___________________
Asst. Prof. Dr. Muhammad ILYAS	School of Engineering and Natural Sciences, Altinbas University	___________________
Assoc. Prof. Dr. Adil Deniz DURU	Faculty of Sport Sciences, Marmara University	___________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

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

Ali Mohsin Lateef ALKHAFAJI

DEDICATION

First I would like to Allah Almighty for the power of mind , health, strength , guidance knowledge and skills to complete this study .This thesis is wholeheartedly dedicated to my beloved father, who have been my source of inspiration, he tells me that" every success in your life will be best gift for me". To my mother, who have been supporting me with the kind and pure love.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Dr. Osman Nuri UÇAN for all support during my study. It’s great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

ABSTRACT

ARTIFICIAL NEURAL NETWORKS FOR ELECTROENCEPHALOGRAM CLASSIFICATION

ALKHAFAJI, Ali Mohsin Lateef,

M.Sc, Information Technologies, Altinbas University

Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: 06/2021

Pages: 57

Application of Artificial Neural Networks (ANN) in the field of classification of neural network signals may help even non experts interpret the results and arrive to a diagnosis.

in the world of machine learning, artificial neural networks models are used which is inspired from biological neural behavior. They are used to approximate functions that can depend on many inputs which are generally unknown. ANN is a system of interconnected neurons which is capable of exchanging messages among themselves. The associations have loads which are tuned making neural nets versatile to the given information and equipped for learning. An ANN is displayed by three sorts of parameters: They are Architecture which is basically the interconnection design between the layers of neurons, the learning procedure for refreshing loads of the interconnections and the actuation work which changes over a neuron's weighted input information to its produced information enactment. Nowadays Optimization is the most essential part of deep machine learning algorithms. It starts with defining a kind of loss function/cost function and ends with minimizing it.

EEG (electroencephalography) is the technique by which the brain's electrical activities can be recorded. It is widely used in research involving neuroscience, and biomedical engineering which includes sleep analysis, and detection of seizure.

Keywords: EEG Data, ANN, Diagnosis , Biomedical Engineering.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

ML	:	Machine Learning
NLP	:	Natural Language Processing
AI	:	Artificial Intelligence
NB	:	Naïve Bays
SVM	:	Support Victor Machine
LSI	:	Latin Semantic Index
GOM	:	Goals of Matrices

1. INTRODUCTION

1.1 BACKGROUND

Electroencephalogram (EEG) is being used to record the waves of brain and helps to detect the abnormal conditions. Measuring unit of EEG activity is microvolts (mV). A routine recording of EEG usually lasts for an about 20–30 minutes. Recording of EEG is done by putting electrodes on the scalp of the person. Now these electro-activities produced by the brain along the scalp are recorded.

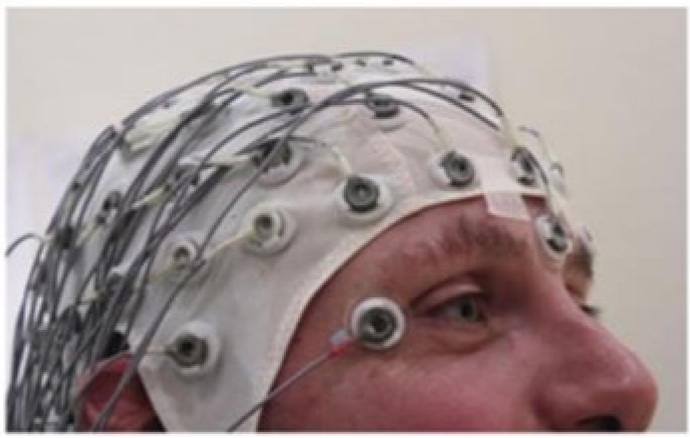

Figure 1.1: A cap used for EEG signals recording

As shown in Fig. 1., An electrode cap is placed on scalp in order to record EEG signals. Use of this cap will make it easier attaching electrodes to the scalp. Precise placement of electrodes can be ensured by these caps. Also, they can help to maintain the sufficient contact of electrodes with the scalp. Most of the time EEG electrode caps with 10-20 electrodes are sufficient for most academic or commercial needs. For more complex imaging studies, caps with 20+ electrodes are also available.

Application of Artificial Neural Networks (ANN) in the field of classification of neural network signals may help even non experts interpret the results and arrive to a diagnosis.

in the world of machine learning, artificial neural networks models are used which is inspired from biological neural behavior. They are used to approximate functions that can depend on many inputs which are generally unknown. ANN is a system of interconnected neurons which is capable of exchanging messages among themselves. The associations have loads which are tuned making neural nets versatile to the given information and equipped for learning. An ANN is displayed by three sorts of parameters: They are Architecture which is basically the interconnection design between the layers of neurons, the learning procedure for refreshing loads of the interconnections and the actuation work which changes over a neuron's weighted input information to its produced information enactment. Nowadays Optimization is the most essential part of deep machine learning algorithms. It starts with defining a kind of loss function/cost function and ends with minimizing it. An artificial neural network learns or trains itself by either supervised or unsupervised learning methodologies. Supervised learning trains the ANN by repeated updating to the weights by supervised algorithms. There are many algorithms under supervised learning method.

1.2 PROBLEM STATEMENT

EEG (electroencephalography) is the technique by which the brain"s electrical activities can be recorded. It is widely used in research involving neuroscience, and biomedical engineering which includes sleep analysis, and detection of seizure [2]. It also has great importance for research in Brain-Computer interfaces, BCI [1]. As it has high temporal resolution, low cost and maintenance, less expertise requirements and is very easy to use. The classification of these signals using deep learning algorithms is crucial to make the use of EEG less reliant on trained professionals. The entire process of automated diagnosis can be subdivided into : pre-processing, feature extraction/selection [3]. The two major challenges in EEG classification are accuracy and robustness. Other factors like added noise can reduce the efficiency of the model. The sensitivity towards noise sometimes makes it less efficient[4]. EEG dataset has real values in a two dimensional matrix of time and channel.

1.3 THESIS CONTRIBUTION

In this review we have briefed the study of the main deep learning algorithms namely literature a) Recurrent Neural Network b) Convolutional Neural Network c) Deep Neural Network d) Deep Belief networks for classifying EEG data. The study of the classification of EEG data has broadly been done into categories like Seizure detection, early prediction Alzheimer Disease, mild cognitive impairment and other categories such as classification of motor imagery, prediction of Brain state, artifact removal for lane change detection, classification of patients with depression, automatic classification of abnormal and normal signals, EEG based emotion recognition, mental workload classification, and also in classification of Sensorimotor Rhythm in Brain-Computer Interfacing. As you can observe the spectrum of applications of EEG is quite diverse in nature and hence it can be made better by deep learning capabilities for better results. In this review, we have done a comparative study among these five major deep learning algorithms which are popularly used with EEG data and have yielded promising results so far. Also, the previous work in each type of algorithm is analyzed separately for various tasks of classification or prediction, by highlighting its key aspects and useful insights. The literature has also been analyzed in terms of results, accuracy, and conclusions. Further, the thesis has also thrown light upon the important aspects such as datasets used, the pre-processing methods such as FFT & PWD, the various architectural models of algorithms proposed, and their features like robustness.

2. NEURAL NETWORKS

2.1 INTRODUCTION

Neural networks are a machine learning technique incorporating principles that have been observed in biological nervous systems. A biological neuron is a cell consisting of a body, multiple smaller protrusions called dendrites and a single long protrusion called an axon. On the surface of the dendrites are found numerous spots called synapses, whereby axons of other neurons are connected to the neuron. Every now and then, the axons fire an electric impulse which affects the permeability of the cell membrane in such a way that the voltage inside the neuron body increases slightly. The more activations come from other neurons, the higher the voltage grows. At the base of the axon is a center which 'measures' the voltage permanently. As long as the voltage value lies below a certain threshold, nothing particular happens. Once this threshold has been exceeded, the neuron fires an action potential, meaning that a wave of high voltage starts propagating towards the end of the axon. This axon can be attached to the dendrites of one or more other neurons, where the process is repeated in an analogous way. This simplified mechanism of function of biological neurons is simulated by artificial neuron networks. Like their biological counterparts, also artificial neurons are units which contain one or more inputs and a single output. Depending on how much non-zero inputs the neuron receives, it either remains inactive, giving a zero output, or generates an action potential, represented by a non-zero output. Multiple neurons can be connected to one another, producing an (artificial) neural network. The following sections will describe the functioning of neural networks in more detail. This chapter draws mainly on Šíma and Neruda (1996), to which book the reader should refer for more information.

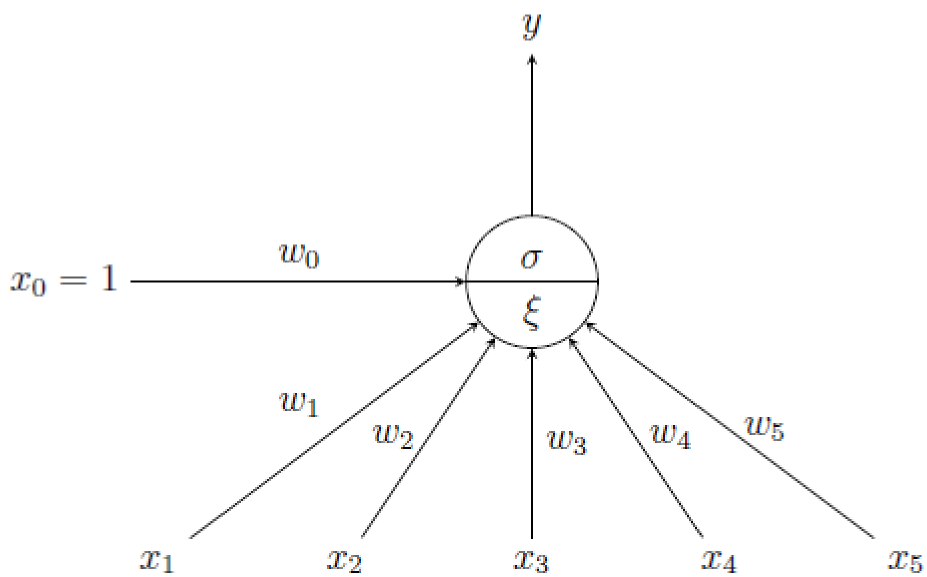

Figure 2.1: An example neuron unit with five inputs and bias

2.2 PERCEPTRON

The basic constituent unit of most neural networks is called a perceptron or simply an (articifical) neuron. It can be viewed as a simplified model of a biological neuron. Much like its biological counterpart, a perceptron is sort of a 'black box' with some fixed number of inputs (sometimes referred to as dendrites) and a single output (also known as the axon).

Each input has a corresponding weight indicating the extent to which the particular connection influences the resulting output.

The rough idea described above can be formalized as follows. Let X = (x1……xn) be the vector of input values and w = (w1……wn) the vector of the corresponding weights. To calculate the output, we need to compute the weighted sum

$$\xi = \mathbf{w}\mathbf{X} = \sum_{i=1}^{n} w_i x_i \qquad (2.1)$$

(sometimes called the inner potential of the neuron) first. The weighted sum is subsequently passed to the activation function _, yielding the output

$$y = \sigma(\xi) = \sigma\left(\sum_{i=1}^{n} w_i x_i\right). \qquad (2.2)$$

The activation function _ may be of various kinds, depending on the task the neuron is designed to perform. In the case of (two-class) classification, threshold activation functions are widely used. A threshold The (two-dimensional) sample vectors are split into two groups, labeled either as 0 (red points) or 1 (blue points). The current weight vector w(k) defines a hyperplane h(k) which splits the vector space into two half-spaces. Samples in one half-space (the 'left' one in the graph) are classified as 0, samples in the other one as 1.

This configuration gives rise to two classification errors: one false negative (x1) and one false positive (x2). The new weight vector w(k+1) is obtained from the old one by adding all the vectors corresponding to the false negative samples (here only x1) and subtracting the false positive vectors (x2 in our case). (In fact, the error vectors are multiplied by the parameter " first;

here ε = 1.) It is the normal vector of a new separating hyperplane h(k+1), which already classifies all the samples correctly.

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq h, \\ 0 & \xi < h. \end{cases} \tag{2.3}$$

activation function has a general form where h is an arbitrary (but fixed) real-valued parameter. Like in a biological neuron, the 'action potential' (i.e. non-zero output) is generated whenever the weighted sum of inputs exceeds a pre-defined threshold value and vice versa.

A variant of an artificial neuron is a neuron with bias. Such a neuron contains an extra formal input x0 whose value is always 1. This input has a corresponding weight w0. When the weighted sum is computed, the product w0x0 = w0 is also included in it, effectively adding w0 to

the weighted sum of the remaining inputs. Note that if w0 = h where h is the threshold mentioned above, the activation function _ can be equivalently defined as

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0, \\ 0 & \xi < 0. \end{cases} \tag{ 2.4}$$

This approach is more convenient for many uses, notably for learning, as will be shown

in a short while. Henceforth, any mention of the term 'neuron' is to be understood as a neuron with bias unless specified otherwise.

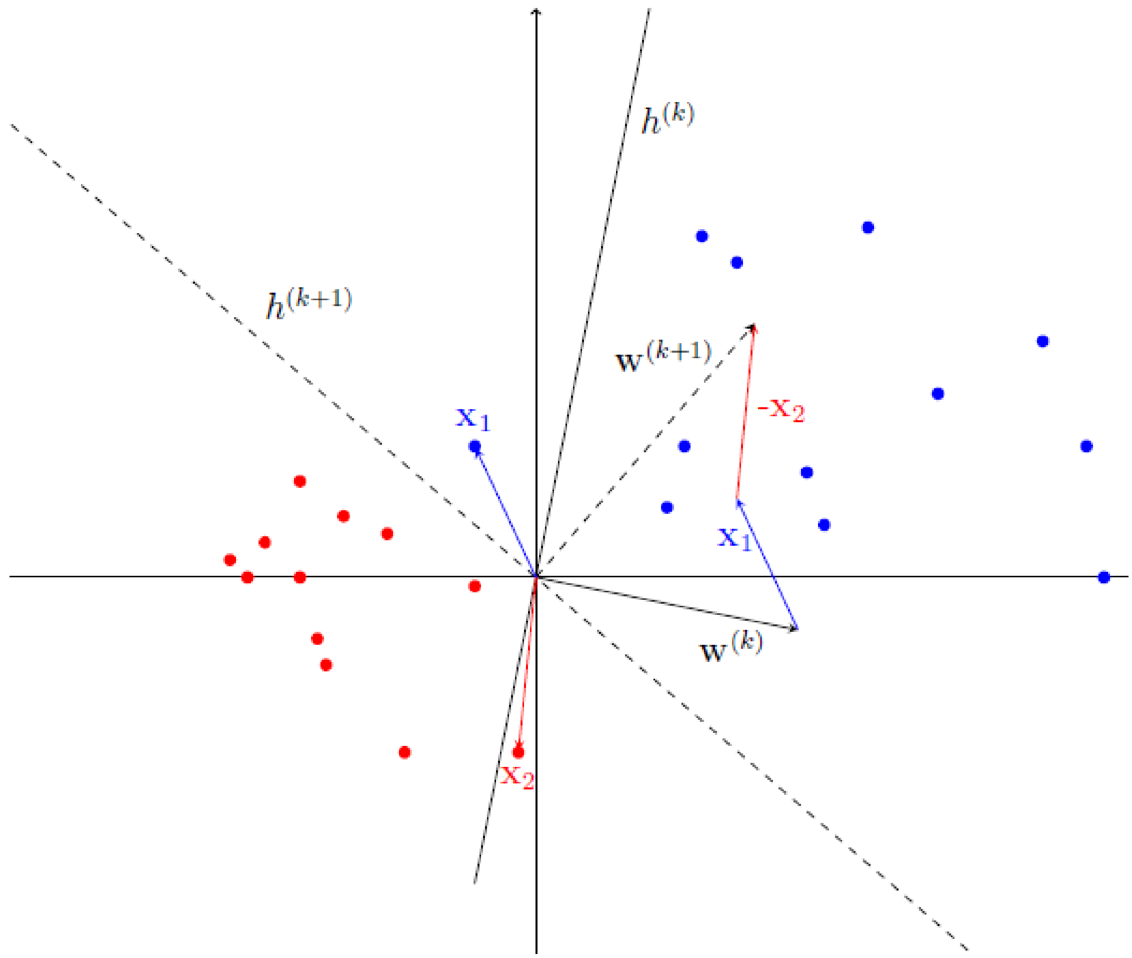

Figure 2.2: Geometric interpretation of perceptron classification and learning.

2.3 PERCEPTRON LEARNING ALGORITHM

Considering what has been said so far, a neuron can be viewed as an nary function from Rn to the set f0; 1g. For each n-dimensional real vector, it returns either 0 or 1, the choice being made as defined in 3.2. Note that _ = w0x0 + …… + wnxn = 0 is an equation of an (n - 1)-dimensional hyperplane which separates the space Rn into two half-spaces. The first half-space contains precisely those vectors X for which £(X) ≥ 0; the other half-space those for which £(X) < 0. This hyperplane is fully defined by the vector w = (w0……wn): the components (w1……wn) constitute the normal vector of the hyperplane, while the terms w0 w1………w0 wn define its intersection points with the corresponding axes. In every subsequent step, the weights are computed as

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \varepsilon \cdot \sum_{k=1}^{p} (d_k - \sigma(\mathbf{w}^{(t-1)} \mathbf{X}_k)) \cdot \mathbf{X}_k, \tag{2.4}$$

It was shown already by Frank Rosenblatt that the algorithm always converges, provided that the sets of examples labeled with 0, resp. 1 are linearly separable. That is to say, if a hyperplane exists such that all examples labeled as 0 are located on one side of it and all examples labeled as 1 on the other, the algorithm guarantees to find it in a finite time.

The condition of linear separability imposes a strong limitation on the data to which the perceptron algorithm can be applied. Not only it fails to learn functions as simple as the logical XOR, but—more importantly—the algorithm is not robust enough to cope with outliers.

A single outlier in the vicinity of the other cluster is enough to make the algorithm run ad infinitum without success. Since real-life data are generally not linearly separable, other techniques must be employed to build a succesful classifier.

2.4 FAST PERCEPTRON TRAINING

The basic perceptron algorithm, as formulated in the previous section, raises a couple of questions which ought to be discussed first in order to boost the algorithm's effectively (most importantly, the rate of convergence).

First, what values should the weights be set to during the initialization? And second, how to choose the optimal value for the parameter ", i.e. learning rate? Should it be constant, or is it better for it to change in each iteration?

One method to tackle these issues was proposed by Gkanogiannis and Kalamboukis (2009). In their approach, the centers of both classes of samples are computed first in the following manner:

$$\mathbf{C}_0 = \frac{1}{|C_0|} \sum_{\mathbf{x_i} \in C_0} \mathbf{x_i} \tag{2.4}$$

$$\mathbf{C}_1 = \frac{1}{|C_1|} \sum_{\mathbf{x_i} \in C_1} \mathbf{x_i} \tag{2.5}$$

The vector w = C1 - C0j is then used as the initial weight vector (w1…..wn). The method to find the initial weight vector is illustrated

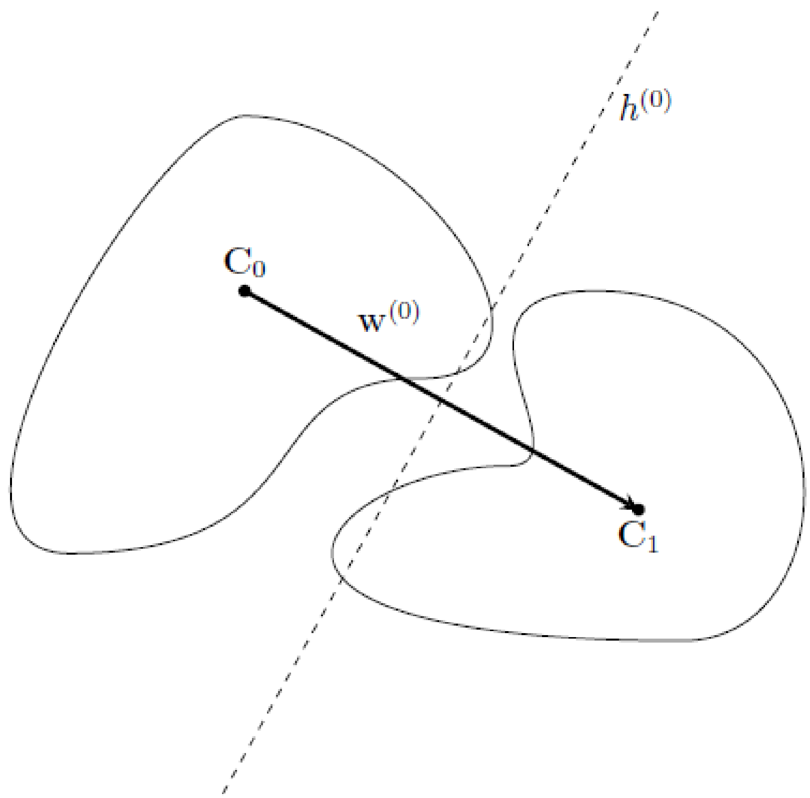

Figure 2.3: Initialization of weights according to

The centres (C0;C1) were computed for both classes of samples. The initial weight vector w(0) was then set to C1-C0. Finally, the bias w0 was set to C0+ C1-C02 which ensures that the initial separating hyperplane h(0) passes halfway between C0 and C1. In this setup, a substantial number of samples is already assigned to the correct class, even though a handful of samples are still classified incorrectly.

In Figure 2.3. As regards the initial bias w0, the authors suggest using the Scut method, originally developed by [12]. Roughly speaking, it means to go through the training samples one by one, shift the separating hyperplane to pass through the particular sample by choosing an appropriate bias value and compute the ‘quality’ of such a configuration (e.g. accuracy or F-measure). The best scoring bias value is then used as the actual bias in the initialization.

Apart from the initialization, the authors also suggest a modified stepwise learning rule for the adaptation of the weights, see Figure 2.4.

In the traditional approach, the output values are computed for each sample and the weights are subsequently updated in such a way that the separating hyperplane is rotated in the direction of the misclassified samples. The extent to which the weights are modified is co-determined by the parameter ", which has to be assessed experimentally. The modified variant is both similar and different. It also determines the incorrectly classified samples of both types—false positives (FP) and false negatives (FN)—using the current weights first. After that, the respective centers FP and FN are computed in the following fashion:

$$\mathrm{FP} = \frac{1}{|FP|} \sum_{x_k \in FP} x_k \tag{2.5}$$

$$\mathrm{FN} = \frac{1}{|FN|} \sum_{x_k \in FN} x_k \tag{2.6}$$

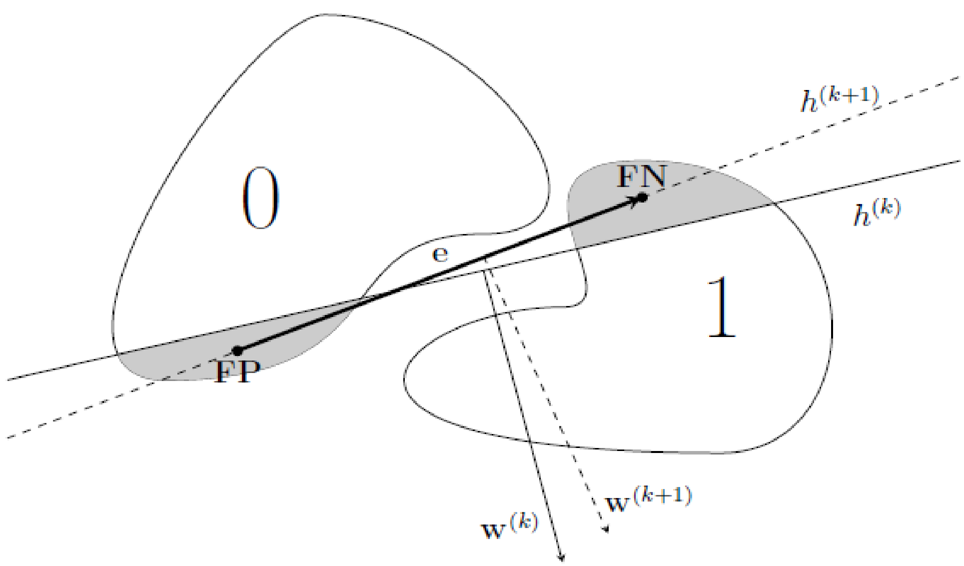

Figure 2.4: Illustration of one step in fast perceptron training

We are trying to find a hyperplane (in this case, line) which separates two classes of examples from each other. The separating line corresponding to the current weight vector w(k) is labeled as h(k). Apparently, a certain number of samples from the class 0 are incorrectly classified as 1 ("false positives" in the left grey area) and contrarily, other samples from the class 1 are misclassified as 0 ("false negatives" in the right grey area). The centers FP, resp. FN are computed for all false positives, resp. false negatives, following which the error vector e can be obtained. Finally, the weights are adapted in such a way that both FP and FN lie on the new separating hyperplane h(k+1). The major benefit of the new approach is that the value of " can be calculated precisely. It is done by adopting the assumption that a hyperplane passing through the points FP and FN will lead to an improvement in the classification rate. Since these points are the centers of the respective misclassified sets, it follows that about half of the previously misclassified samples will now be classified correctly. (Naturally, new erroneously classified samples may emerge instead.)

2.5 MULTI-LAYER FEED FORWARD NETWORKS

In the case of a single neuron unit described above, the output value was directly used as the final result indicating the class assigned to the input vector. An alternative option to make use of the output is to pass it as an input of another neuron instead. In this way it is possible to create a network of interconnected neurons, in certain aspects analogous to biological neural networks. Numerous different topologies have been used in practice for different purposes. One of the most widely used type of neural networks are the so-called multi-layer feedforward networks. Since these have been employed in part-of-speech tagging as well, they are of a particular importance for this thesis. It is therefore worth saying a few words about their structure, computation and mechanism of learning first.

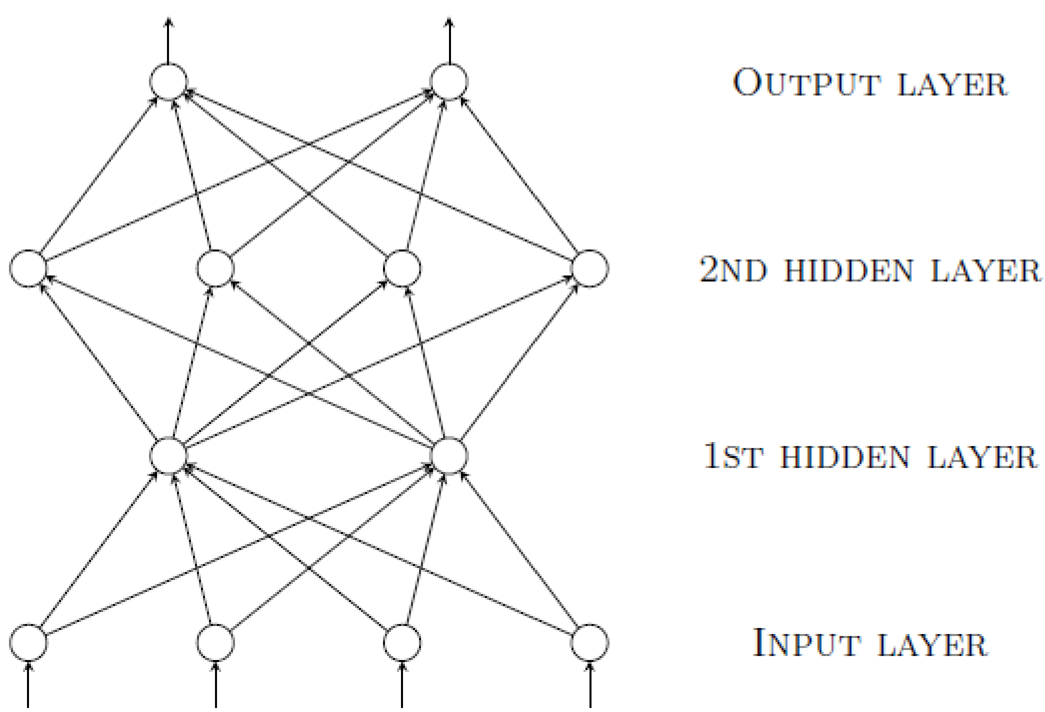

Figure 2.5: An example MLP network with two hidden layers

As the name hints, multi-layer feedforward networks, also known as multi-layer perceptrons (MLPs) consist of a certain number of neurons which are split into several disjoint sets, so called layers. Every MLP contains an input layer, to which the input vector is fed, and an output layer, which produces the vector of output values. In addition, the network may contain one or more hidden layers, located between the input and the output layer. Each neuron is connected to all the neurons in the adjacent layers: the outputs of the neurons from the previous layer serve as inputs and the output is passed to input of the neurons in the next layer.

As with a single perceptron unit, various activation functions can be used. For reasons that will be shown shortly, it is required that it be differentiable. As the threshold function cannot be used (it is discontinuous in 0), other functions of a similar shape are often employed instead.

These functions behave in an analogous way to the threshold function in that they return a value close to the maximum (1 in this case) in for sufficiently large inputs; similarly, a value close to the minimum (0, resp. -1) is returned for inputs low enough. Between these two extremes, the functions grow continuously, their growth rate being determined by the parameter _ (in the logistic sigmoid), resp. b (in the hyperbolic tangent).

2.6 BACK-PROPAGATION LEARNING ALGORITHM

Let us consider an arbitrary multilayer perceptron network. Let N be the set of all the neurons it consists of, IN N the set of the neurons in the input layer and OUT _ N the set of the neurons in the output layer. The individual neurons are indexed by numbers from 1 to jNj. The weight of the connection from a neuron i to a neuron j is denoted by wji, the set of all the weights by W. The notation i! refers to the set of all the neurons to which a connection from i exists, and contrarily, i is the set of all the neurons which are connected to the input of i.

Once all the output vectors yk have been obtained, they need to be compared to the desired output vectors dk to determine how well the network does and in what way the weights should be modified to improve the performance.

The more training samples generated incorrect outputs, the higher the value of the squared error function is, and vice versa. If the network produces correct outputs for all the samples, the squared error is zero. The problem of finding the best weights for a network is therefore equivalent to the problem of finding the global minimum of the function E. One way to find the global minimum is to compute the gradient $\nabla E(\mathbf{w}) = (\partial w_{ji})_{w_{ji} \in W}$ in the point w, corresponding to the current vector of all the weights wji in the network. The gradient is a vector which determines the direction of the fastest growth of the error function in the specified point, as well as the steepness of the slope in the respective direction. If the gradient is subtracted from the current weight vector, we obtain a new vector which is likely to lie 'below' the original one in terms of the value of E. Therefore, the vector w is updated in the t-th iteration (t = 1; 2….)

3. BACKGROUND AND RELATED WORKS

3.1 DEEP LEARNING

The most important period and first began of Deep Learning using Convolutional Neural Network by Yann LeCun & Yoshua Bengio in 1995 by take their steps from MLP and make an improvement by reducing both of computation task as well as number of measurements. From that point forward, a paper was distribute [36] it shows better example acknowledgment framework by dropping random variable.

Then E. Hinton publish a paper that introduced Restricted Boltzmann Machines (RBM) [37] which is used for Filter, classification label and non-label data, and reducing the data dimensionality which helps to make simple computational.

In 2012 Alex Krizhevsky and his team used convolutional neural network named AlexNet which wins ImageNet LSVRC-2012, 2010 (Large Scale Visual Recognition Competition) by using train set of images .more than one million RGB label images with high resolution in order to classify them into 1000 categorize.

DL algorithms had improved and be used to solve may task (image segmentation, detection, style transfer, analogies, image classification), this improvement is due to the existence of huge number of data (big data) as well as the hardware with high level which capable to deals with high computational task. Finally, in 2017 the research uses a deep neural net and the error rates were 3%.

3.1.1 Deep Learning Overview

The primary meaning of DL is a Neural Network with many shrouded layers, so "profound" here it alludes to the profound of layers which is in excess of 2 concealed layers. It gives programmed highlight extraction by decided the properties of information which can be utilized as a pointer to mark of information precisely, where each layer removes highlights from yield of past layer This resembled an upset PC vision task, conversely with shallow system which required hand intended for include extraction and high experience on picture preparing field, then again change from input pictures to vectors prompted lose a great deal of fascinating data.

There are 7 principle applications that for the most part have been polished in DL [38]

3.1.2 Deep Learning Architectures

Restricted Boltzman Machines (RBMs):

It stochastic NN, consist of two type of layer (visible layer, hidden layer), where every neuron in noticeable layer is completely associated with every neuron in concealed layer, vice versa with neuron of hidden layer, beside that each neuron had to make stochastic determination in order to transmit the signal between the two layer or not, and there is not any connection between the neuron on the same layer so it restricted with connection, where each input will be part of the network which don't have output layer so it generative model.

RBMS consider as unsupervised ML and energy base model use to determine the probability of distribution with consideration input data or minimize the energy.

the Visible layer (input layer) passes the input to the Hidden layer by using the weights (w_{ij}) and the product operation will be on the Hidden layer as well as applying activatin function, and update the weight by the equation :

$$\frac{\partial \log p(v^o)}{\partial w_{ij}} = (vi^o hj^o) - (vi^\infty hj^\infty) \qquad 3.1$$

Where i is visible unite, j is hidden unite, w_{ij} is the weight between the layers, $(v\ i^o hj^o), (v\ i^\infty hj^\infty)$ are the correlations when (i,j) are in minimum and maximum layers.

The use of RBM is to reduce the dimension, filtering, classification, feature extraction but the drawback it cannot obtain the partition function to the same resolution and difficulte to train [40].

Deep Belief Networks (DBNs)

It stack of RBM, each layer in DBN represent two performance visible layer considered to the layer after it, and hidden layer considered to the layer beforeit. DBN Hinton applies two-phase to solve the un accuracy of the network due to the deep of layers[41]:

Phase1: Layer-Wise

By train Input data on RBM and obtain the parameters the output will be as input to

Next RBM, this form will be a DBN which contains several numbers of RBMs and the parameter which used for feature extractions. The train in phase 1 is unsupervised learning.

Phase2: Fine-Turning

A convenient classifier will be placed at the end of DBN by using Contrastive Divergence CD algorithm and the feature extraction to build FFNN, which can be BP or use DBN Softmax classifier.

The main use of DBNs is used for feature extraction, image generated, clustering, recognition, etc...

Aautoencoder (AE):

Also named Autoassociator, is unsupervised learning and FFNN which consist of three main components (encoder, code, decoder), after the data is shrink in encoder also called **latent-space representation** that present the code that transmit to decoder which is rebuild the input data using this code transmit to code.

Aautoencoder (AE) can be used for:

a) Reduce data dimensionally.
b) Data compression
c) Reform data corruption
d) Avoid local minimum by a pre-train deep network

Convolutional Neural network (CNN or ConvNet):

The main drawback of earlier neural network were as follow:

a) Large number trainable parameter which were difficult to deal with even in GPU, especially when deals with high-dimensional input (RGB).
b) Insufficient or no-invariance to shifting or scaling or other type of distortion.
c) Ignore topology of input data, where there were no any relation between the pixels.
d) With CNN which considered as revolution in computer vision with mathematical operation, main negative aspects that have been resolved using CNN are:
e) Invariance using scale, shift, etc.…
f) Solving overfitting by limiting the numbers of weights using parameter sharing
g) Make use of spatial proximity of the pixels, by make connection between the neurons that are similar to neighboring pixels.

It classifies directly from picture elements with minimum treatment and best performance using small parameter because of parameter sharing and lower connections. And in many types of research CNN show high accuracy and achievement on image classification Due to automatic feature extraction with lower image pre-processing as well as parameter sharing and Sparsity of connection.

3.2 THE MAIN OPERATION OF CONVOLUTION

Edge detection

in order to specify the method to find the vertical or horizontal edges in images by using a filter (kernel).

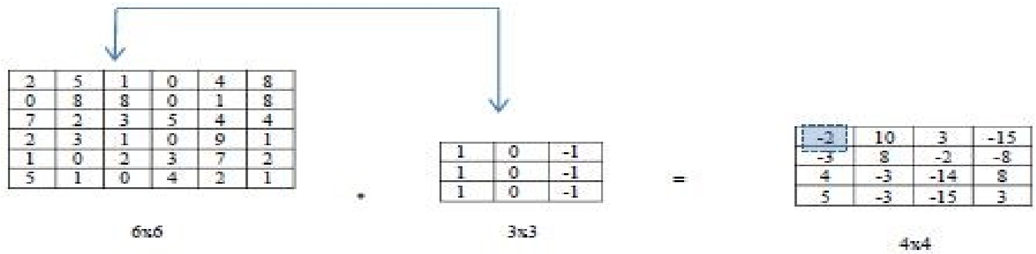

Figure 3.1: Edge detection using convolution operator

As shown in **Figure 3.1** construct a 3*3 kernel (also known as filter or vertical detector) and take the 6*6 image and make the convolution operation (*) by applying element product then summation, the first element on the top right corner will be:

2*1+5*0+1*-1+0*1+8*0+8*-1+7*1+2*0+3*-1 = -2 By shifting the blue box one step to the wright and do the same convolution operation we obtain the second element on the 4*4 matrix and do this to all elements.

Padding

The used of padding is to avoid the following factors:

a) Images shrink when applying convolution operator in every time by using edge detection.
b) Throwing away a lot of information near the edge of the images.

To specify the dimension of output matrix we used the following equation:

$$(n+2p-f+1)x(n+2p-f+1) \qquad 3.2$$

Where: n: input image dimension, p: padding, f: filter.

In order to verify padding to be use or not we use:

- **Valid convolution**: no padding and the output dimension will be

$$(n-f+1)x(n-f+1) \qquad 3.3$$

- **Same convolution**: output size is equal to input size

$$p=\frac{f-1}{2} \qquad 3.4$$

Stride

The number of steps that shifts the kernel. So the output dimension will be

$$\left(\frac{n+2p-f}{s}+1\right)x\ (\left(\frac{n+2p-f}{s}+1\right) \qquad 3.5$$

Where: n: input image dimension, p: padding, f: filter.

The main cons of the stride that are big stride means less extend in output dimension that leads to small output size to the next layer with small computation and memory allocated.

Types of layers in CNN

Convolutional layer:

The core of CNN is Conv layer, it represents a mathematical part of the network by apply element-wise and addition operations using (height, width, depth) of the input image (respective field).

Main hyper-parameter of Cov.layer:

a) Size and number of filter.
b) Padding
c) Stride
d) Activation function (linear, non-linear)

The purpose of earlier conv.layer is to detect low level feature such as edge, line, etc., where the deeper conv.layer used for high level feature detection such as shape.

The depth refers to RGB or number of channel and Filter(s) (kernel) is a square matrix consist of weights that slide on each region and the output is the feature(s) map which will be as input to next layer**.**

By taking image matrix, and kernel matrix, and apply element-wise operation as shown in **Figure 12**, where we have a simple image with 2-dimension (height, width) with 6x6 dimension, and filters (height, width) with 3x3 dimension, after apply convolution (*) operation for both of them we have one features map with output dimension is 4x4 volume:

(n-f+1) x (n-f+1) = (6-3+1) x (6-3+1) = 4x4 this output will be the input to next layer. So in conv layer, the size of the input image decreases from 6x6x1 to 4x4x1 and by applying this operation many times may loss much image information so we used padding.

Filter types:

a) Sobel filter which can apply edge-detection
b) Filter to blur the image

c) Filter to sharp the image

d) Schoss filter for v – edge detection

Non –linearity layer (activation layer):

Received the stack of feature map of Conv layer and apply non-linearity operation by add bias and apply activation function that used in CNN like (Softmax, ReLU, ELU, sigmoid)

$$y^{[1]} = w^{[1]}a^{[0]} + b^{[1]} \qquad 3.6$$

Pooling Layer:

Third convolution layers which the output of non-linearity layer and be as input to pooling layer which used to reduce the size and the computation operation by reducing the size of height and width without any changing to the size of the number of channels. So its fixed function without parameters to learn.

Two types of pooling layer (avg-pooling, max-pooling) but max-pooling is widely used by apply the max pool we slide the window of 2x2 regions and take the max value on each part and place it to the new matrix, while average pooling takes the average of the window of 2x2 regions.

Fully Connected:

the final layer of CNN and it responsible of classification task by take the output matrix (feature map) (SxSxn) from previous layer and convert it into vector (Sx1) so each pixel in matrix represent the neuron value on vector which is fully connected to the next hidden layer and with the previous layer, then apply activation function like Softmax for multiclass classification to indicate the probability for each class, or using sigmoid for binary classification to predict to which class the images are belong.

Batch normalization layer

Using BN (internal covariate shift) is to reduce the variety of distribution in the input layer during the training process using two parameter:

a) Shifting: by subtract mean.

b) Scale: by divide the batch standard deviation

c) Where the output will be between [0, 1] or [-1, 1]. That speedup the training process as well as reduce the overfitting, and increase the learning rate, finally it make the training more stability.

Dropout layer

It used to ignore or dropout neurons and their connection form hidden layers with probability of p, to reduce overfitting and avoid co-adapation feature detection, when notice some neurons detects the same feature.

So, it decrease the connections between previous and next layer,

Types of convolutional neural network architectures:

LetNet-5

The first CNN architecture was introduced in 1995 by Yann LeCun et. al [42]; the main goal was to use Gradient-Based learning algorithms in order to classify pattern recognition with low dimension and less computational. The focus was to build automatic handwritten binary classification using MNIST dataset which consist of 50.000 train images and 10,000 as validation and test images, with (28x28) as resolution of gray.

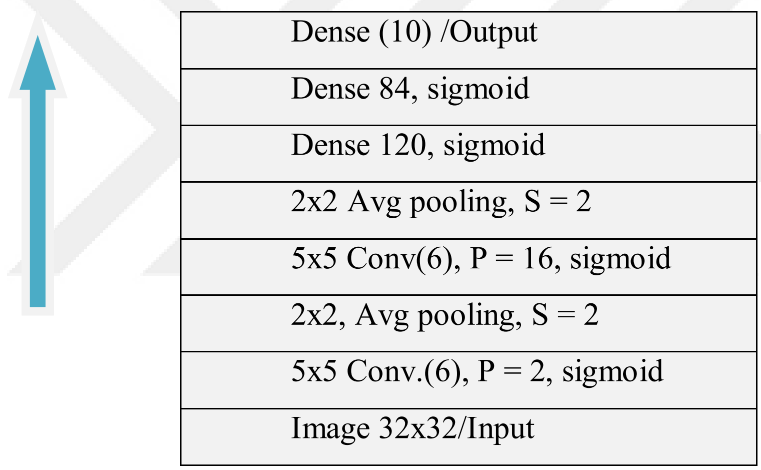

Figure 3.2: LetNet

By building a network with eight layer as shown in **3.3** , the main idea of this network is minimize the number of high and width when we go deeper, while the number of the channel (feature map) is maximizing. Besides that, after every pooling layer there is sigmoid non-linearity layer. By using weight sharing technique allow to minimizing the number of trainable parameters, finally the total connection in LetNet-5 was 340,908 and 60,000 trainable parameter.

AlexNet

A paper was published in 2012 by Alex Krizhevsky et. al [43]. To classify 1.2 Million images with high resolution into one thousand different classes using in ImageNet dataset, the result was 37.5% error rate in Top-1, 17.0% in top-5. The total was 60 Million trainable parameter and 650 thousand neurons. The main aspect was using the following parameter:

a) "Dropout-Regularization" so as to minimize the overfitting.

b) Maxpooling instead of Avergpooling layer.

c) Increase the number of the convolutional layer with ReLU as activation function, so that will increase the non-linearity through the deep of the network and solving the vanishing gradients.

d) Using a small learning rate (0.01) and large image size (224x224x3).

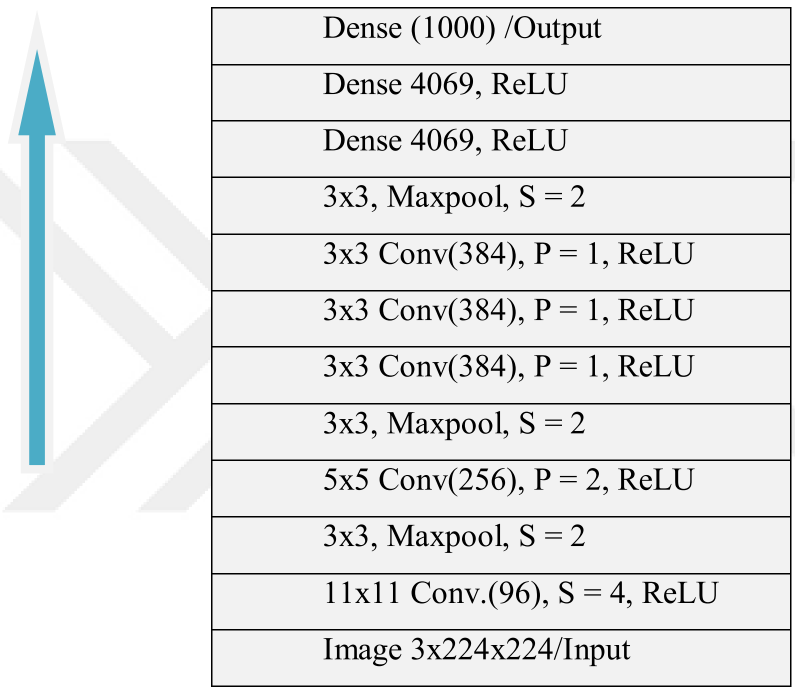

Figure 3.3: AlexNet

All dataset was collected by using crowdsourcing in amazon mechanical Turk. By building a network containing five convolution layer, three Max pool layers, and three fully connected layers with 1000 neurons using softmax as activation function, consists of 12 layers.

This network could solving the overfitting problem using two methods:

a) Data augmentation by a label- preserving transformation.

b) Dropout layer after each dense layer: which assign zero to the output of hidden layers with 0.5 probabilities.

VGG-16:

Both Karen Simonyan and Andrew publish paper [44] which introduce a stack of convolution layers that considered as a new architecture of convolution neural network which try to simplified the network using small receptive field in Conv layers (3x3) with stride equal to one, where all Max-pooling size with 2x2 and the stride equal to 2. This concept of the block is now used in many DL architecture, where each block contained a stack of (Conv.layers, Max pool layer) follow by three Fully-connected layers as shown in **Figure 3.4.**

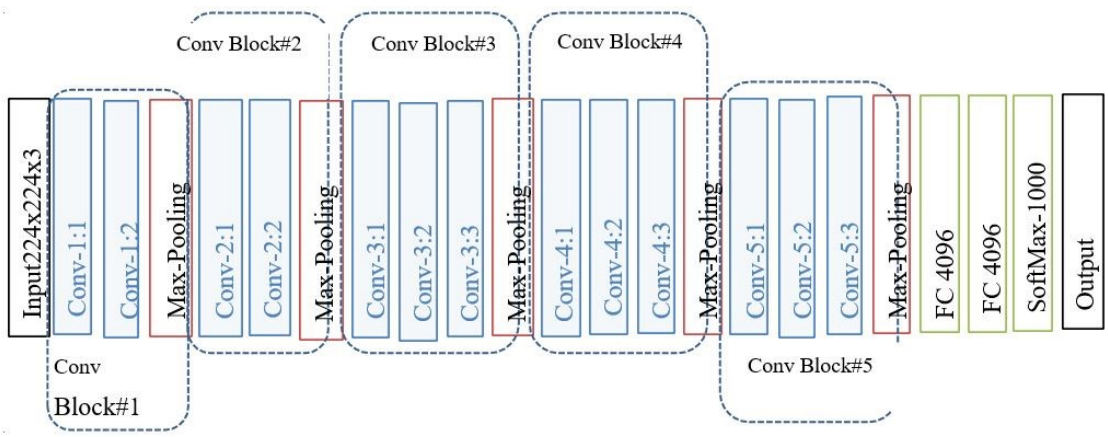

Figure 3.4: VGG-16 Architecture

VGG-16 Architecture The aim of this network was contributing to the competition of ImageNet 2014; the result was on top-1 val.error 23.7%, top-5 val.erro 6.8%. They indicate that using deep layers will increase the performance and accuracy of the network.

The name of VGG-16 refers to the number of layers which is 16 (13 Conv-Layers, 3 FC-Layers), it has 128 million parameters to train and the main strength of this network is to double the value of the channel in every block while the number of the height and width is decreased.

Inception/GoogleNet:

A paper was introduced to contribute with the competition of the ImageNet on ILSVRC 2014 and achieve 93.3% top-accuracy, the main idea was instead of choosing specific convolution layer with specific filter size, Inception network could concatenated more than convolution layer and max-pooling layer with different filter size then stack them in one block which named inception block as shown in **Figure 3.5**, That will reduce the numbers of the parameters as well as computational task.

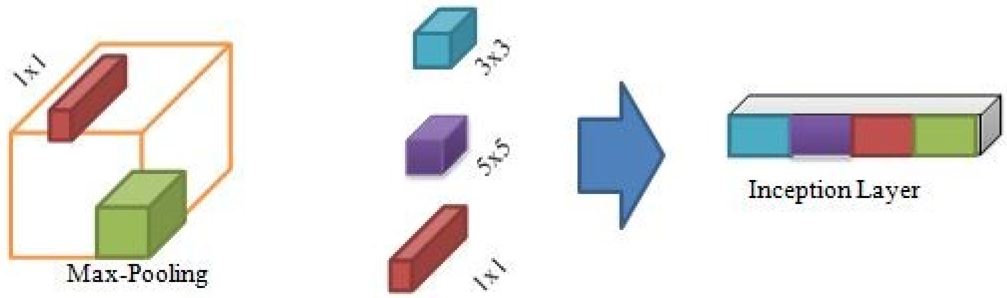

Figure 3.5 : Single Inception Block

The inception network was build using different numbers of Convolution layer (1x1), and nine of inception blocks, then center and cropping the input image to 224x224x3.

ReLu was used in hidden layer, while softmax in output layer to specify the probability of the output for each class.

The optimizer that used in inception network was (sgd) stochastic gradient descent equal to 0.9 momentums, and minimizing the learning rate to 0.04 at every eight epochs. Another aspect was used in inception network is using Average Pooling layer after the last convolution layer instead of a fully connected layer which reduces the number of parameters.

By increasing both of depth (number of layers) and width (number of units), show more performance in the accuracy of network.

This method was inspired by two main concepts:

c) Network in Network which introduced by Lin et al.[45]

d) The familiar sentence which had taken from the film “we need to go deeper”.

The main challenge in the increase of depth was the increase in computational cost problem which named “bottle neck”.

In order to solve bottle neck conv.net (1x1) was the solution by making it between the layers as shows in Figure 3.6.

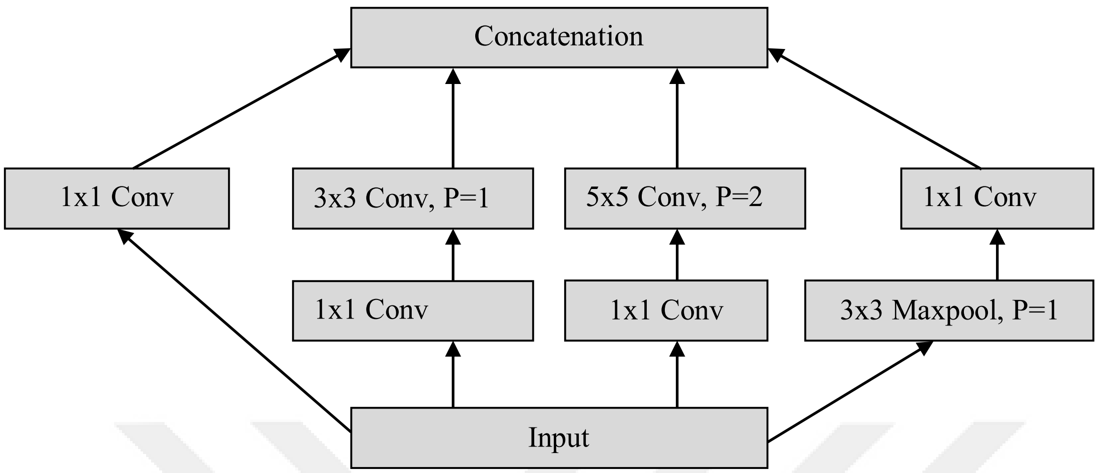

Figure3.6: Inception module

To make a compare between computational cost with and without Conv.net (1x1) as follow:

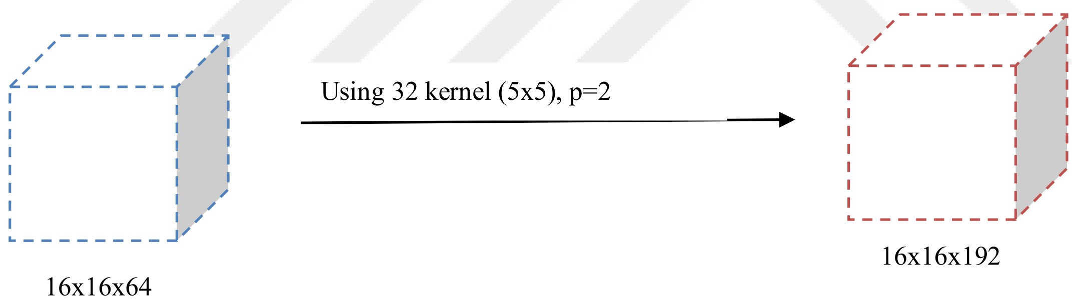

Figure 3.7 :Before using Conv.layer (1x1)

Computational cost in Figure 3.7 = $(N_h x N_w x N_{ch}) \times (F_h x F_w x F_{ch})$

$= 16*16*64*5x5x32$

$= 13,107,200$

Where:

N_h: height of input.

N_w: width of input.

N_{ch}: number of input channel.

F_{ch}: numbers of filters.

F_h: height of filter.

F_w: width of filter.

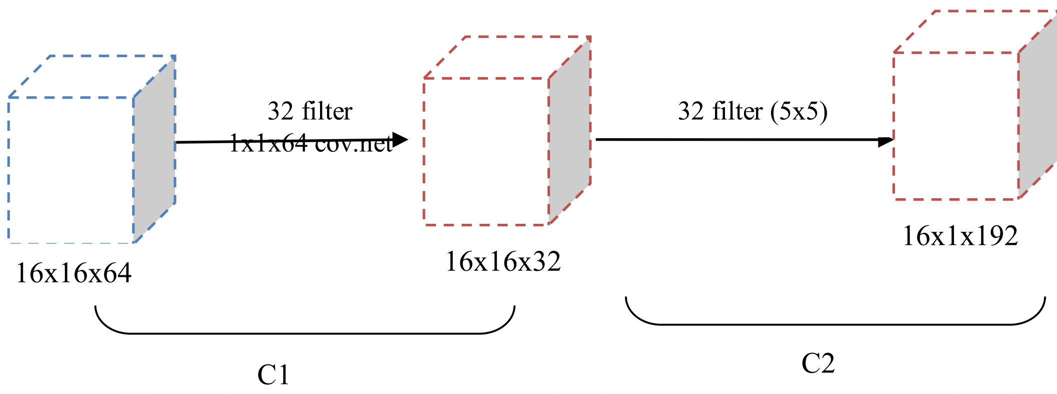

Figure 3.8 After using Conv.layer (1x1)

Numbers of parameters in Figure 3.8 = C1 + C2

16*16*64*1*1*64 + 16*16*32*32*5*5 = 7.602.176 which is much are smaller than computational cost in Figure 3.9 .

3.3 LONG SHORT-TERM MEMORY NETWORKS

Even though RNNs are widely used for sequential data, they have cer- tain limitations. For instance, RNNs can remember the past, but they cannot keep track of long-term dependencies. This problem appears due to the Vanishing Gradient and Exploding Gradient that makes the training of RNN difficult in several ways. Gradient problems are described in more detail in [31]. However, this issue is overcome by Long Short-term Memory Networks (LSTMs, [32]).

LSTMs are a kind of RNNs that is able to learn long-term depen- dencies. This advantage allows the network to remember important information for long periods as well as forget the less important one. This is why this approach is currently very successful in sequence pre- diction tasks, such as translation, speech recognition, or even music composition. Compared with traditional RNNs, the LSTM networks have a different structure of repeating cells that contains four inter- acting layers.

Figure 3.10 shows the basic LSTM cell. However, in reality, there are many of them in various versions. The important parts of the cell is a *cell state* represented as c and *hidden state* represented as h. Input e stands for the input at particular timestamp t. As we can see, the state of the cell can pass through the cell unchanged.

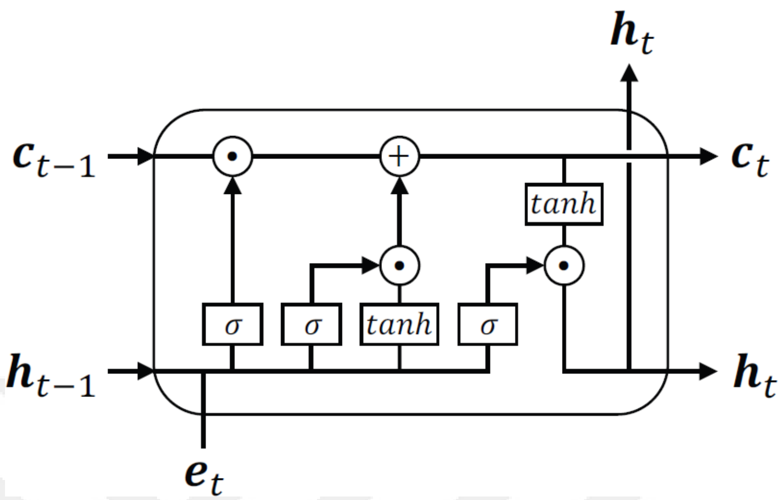

Figure 3.9: LSTM cell of the recurrent neural network [33]

Another important parts of the cell are *gates* that manipulate with the cell memory. *Gates* are:

Forget gate – responsible for removing information from the cell state. It applies a *sigmoid* function σ for an input z and a hidden state h. Output of the *sigmoid* function is the vector with values from 0 to 1.

Input gate – responsible for the addition of information to the cell state.

Output gate – responsible for what we are going to output from the cell. This decision is based on the current *cell state*.

LSTM networks also have some drawbacks. They take longer to train and require more memory during training. The downside is the requirement for many training data; otherwise, there is a possibility of overfitting. Those problems arise mainly because LSTM networks have more parameters comparing with other neural network architectures.

3.4 LITRATURE REVIEW

High-Density Electroencephalography Signals:

In [31], N. Mammone et al. discussed that AD, being the most common form of dementia and estimated to account for more than half of the cases. They pointed out that diagnosis needs to be improved for better treatment of the patients. Discussed on the demerits of the standard 10-20 EEG, though has good temporal resolution but poor spatial resolution due to very large average inter-sensor distance, by using 256 channels HD- EEG, the demerit of spatial resolution of EEG can be overcome. HD-EEG can also be used for remote uses. The potential of HD-EEG is highly unexplored. The recordings were pre-processed, filtered, and later on reviewed by EEG experts.

In the study [32], D. V Moretti found the relation between increased alpha3/alpha2 power ratio with AD, and other MCI subjects show an increase in the theta-frequency power. By combining the above features along with various cerebral images revealed a complex relation between EEG cerebral rhythms, structural and functional brain modifications.

Li G et al. in [33] surveyed the different deep learning techniques used by various researchers for the analysis of EEG data along with the problem they were trying to solve. It was observed that researchers prefer to use the type frequency spectrum instead of raw data as the EEG data. It listed the advantages of the Adam optimizer which is widely used by many deep learning models. The survey papers of EEG analysis mostly included clinical applications such as AD and Epileptic Seizure.

C. Ieracitano et al., in [34] proposed a Machine Learning system that extracted some statistical coefficients (mean, SD, kurtosis) from EEG‟s 5 main conventional sub-bands (alpha-1, beta, theta, alpha-2, delta).Then these features were fed into different classifiers like SVM, Logistic Regression(LR) and different deep learning algorithms like Auto Encoder, Multi-Layer Perceptron(MLP), to classify whether a person has AD. Abnormalities in brain activity can be monitored using EEG recording with a special attention to following sub- bands: delta band which ranges from 0.5 to 4Hz, theta band ranging from 4 to 8 Hz, alpha-1 band spanning from 8 to 10 Hz, alpha-2 band which ranges from 10 to 13 Hz, and beta band which spans from 13 to 32 Hz. EEG may show peculiarities (slowing down of rhythm, loss of complication and loss of connection between channels) caused because of AD. Single hidden layer MLP outperformed all other classifying systems with classifying accuracy of rate up to 95.7%.

G. Fiscon et al., in [35] stress the prominence of EEG signals in identifying the onset of AD in the form of MCI. The sub-bands (alpha, beta, theta, gamma, and delta) have different activities in AD persons as compared to a healthy subject. The alpha and beta bands have decreased activity, while delta and gamma bands have increased activity suggesting slowdown of EEG signals. The author used Fast Fourier transform for feature extraction of EEG so that they can be divided into their sub-bands after which classification methods are applied based on t supervised learning. The author also made use of classification tools such as Data Mining Big (DMB) and Weka tools.

Specific Deep Learning models for Neurological diseases

M. A. Naderi et al. in [3] propose a 3-step model for detecting epileptic seizures. The model uses the Welch method of power-spectrum density estimation technique to extract features. This method splits the data into multiple segments. Then it calculates an average value of their period grams. They take into account the fact that for accurate classification of signals an accurate method for pre-processing is required. The RNN model is successful in classifying epileptic signals from normal EEG signals with an accuracy of 100%.

In [36], K. Tsiouris et al. introduced LSTM networks, which expand the application of deep learning algorithms. After a preliminary analysis, a LSTM Network consisting of 2 layers was used for prediction of seizures. The model used the CHB-MIT EEG dataset. It is an open database. The network was successful in predicting all the 185 seizures. It meant a high rate of seizure sensitivity and also significantly low false prediction rate. The results of the proposed method provide a significant enhancement predicting seizure as compared to normal machine learning methodologies..

M. Nguyen et al. in [37] proposed an interesting and practical system that included the fact that not every Medical Dataset is complete. There has been significant research in predicting AD progression in [31], [32], and [33]. Deep Learning Methods require large and feature-complete data to avoid overfitting. This also helps the model being generalized. But practically many reasons lead to missing values in the dataset. Thus, this paper proposes an RNN model, with a single hidden layer of 256 nodes, initially to interpolate and extrapolate the missing data. The main focus was to perform multi-step prediction and model progression of AD for upto 7 years into the future. The system used the ADNI dataset, and was successful with a classification accuracy of 86%.

Author F. C. Morabito et al, in [38] proposed to classify a subject into MCI, AD, HC based on the EEG signals recorded. MCI is an early onset of Alzheimer‟s that may later turn into dementia. Identifying this onset of MCI is important as AD although being incurable we can still slow down the progress. They made use of a deep convolutional network consisting of 2 hidden layers to classify Alzheimer‟s Disease (AD) and Healthy Control (HC).82% accuracy was obtained in classifying them in three subjects (MCI, AD, HC).The classification accuracy was upped to 85% when classifying two-way scheme (MCI/AD vs HC).

T. T. Chowdhury et al. in [39] propose a 1-D convolutional network to detect seizure activity of EEG signals. Conventionally to detect seizure we were required to manually select the features and classify them individually. Now with the help of simple 1-D CNN architecture feature selection can be automated. The author implemented CNN architecture having a single convolution layer succeeded by a down-sampling(max-pool) layer . Next comes the fully connected layer with 23 units, and another with 2 units that classified the subject having seizure or non-seizure. Accuracy was obtained to be up to 99%.This architecture provided better accuracy than the one in [28].

In recent years, Deep belief networks emerged as promising Deep learning models used for the classification of EEG data. In many of the literatures, researchers have implemented its various models. For instance, In [40] Al- kaysi proposed different DBN implementation models such as conventional single stream, a multi stream with one DBN at the top to combine the results of all the underlying DBNs, and the multi stream (or multichannel) for classifying multichannel data. This study was aimed at classifying the EEG to majorly differentiate between active and sham tDCS(Transcranial direct current stimulation). For each session, 62 electrodes were placed as per 10/20 international system. For pre-processing, Fourier Transform is used to convert and represent the signal in power spectral density in five frequency bands. A detailed study of these models revealed that multi- stream DBN outperformed the traditional classifiers like SVM, LDA, and ELM classifiers with an error rate of 23.5% and also the other two DBN implementations.

In study [41], D. Kim et al. have pointed that there being no effective treatments for AD, the only method for limiting the cognitive decline is by early diagnosis and providing appropriate care management. The symptoms of AD, being often confused with aging acts as a hindrance towards the early diagnosis. [15] Also states the merits of using EEG for the early diagnosis of AD due to its advantages enabling frequent check-ups.

The following studies reflect on other applications of EEG data mostly focused on Brain-Computer Interface.

The Brain Computer Interfacing has attracted many researchers in the past two decades[30]. In [30], they have elaborated about the importance of BCI and its application. The collaboration of BCI with automobiles is now going to revolutionize the automation we see around us. EEG plays an important role here. The classification of EEG motor imagery is a crucial step in BCI wherein the dynamics of the brain of the motor imagery are measured using EEG. From the past few decades many scientists are tirelessly working to enhance the classification results using Deep learning. Glimpses of some of these research are explained below:

BCI using RBM was studied in [1]. Generally, In EEG the algorithms which are applied on data are discriminative. This is because very few trials are there for each subject. In this study adaptation and a pre- trained RBM was compared. But the aim was to analyze the generalization of the pre-trained RBM when new data is fed to the model.

A novel (RBM) architecture was proposed by Jianhua Zhang [2] classifying EEG data of mental workload (MWL) using MAD-DBN. The RBM layers are pre trained using Mean Absolute Difference, and DBN is fine tuned. Till now the optimization of the model was not taken into consideration. In this study [2], they focused on optimizing the RBM model and to choose appropriate EEG channels. They compared the random channel selection with the proposed one and showed that the proposed method performed well. For identifying the optimal RBM structure they showed comparative results of trial-and-error & entropy-based pruning methods [2]. MAD-DBM achieved the highest accuracy in all cases having an average accuracy of 94% and also DBN has proven the best among other classifiers such as , LDA, NB (Naïve Bayes) etc. In this way in all these years the research arena of DBN for classification of EEG data has been studied and analyzed meticulously and many novel and efficient models have been proposed along with their improving accuracy.

S. Patnaik et al. in [4] proposes a method for extracting features from EEG sub-band. This method includes a wavelet-transformation technique which is followed by classification using RNN. The method aims to discriminate the EEG features that correspond to different states of mind. It takes into account the effect of noise in EEG signal collection and uses a coherence analysis technique which reduces the effect of noise. It uses the BCI dataset that includes 5 subjects performing 4 different tasks. It performs the tasks by learning the topographic patterns of EEG sub-bands. The system uses an Elman"s RNN model with four hidden layers. The model is successful in achieving an accuracy of 90% for 2 tasks, followed by 82% for 3 tasks, and 77% for all the 4 tasks.

M. Moinnereau et al. in [42] focused on Movement Related Cortical Potentials (MRCP). It aims to assist the driver by detecting lane change intentions. It also considers the fact that the irrelevant artifacts need to be filtered out by already developed algorithms. The experiment was carried out on 5 subjects and data was recorded using a 64-channel. Then, the data was pre-processed using 3 artifact removal algorithms. The RNN model, when tested 4 seconds before the lane changing, offered a classification accuracy of 82.88%, 82.76%, 82.69% using the 3 techniques, which was 25%-30% more than SVM. This accuracy further increased to about 86%-88% when the window size considered was 2 seconds.

In [43], Jianhua Wang made use of convolutional neural networks to classify motor imagery-based movement (left or right hand). BCI is used to decipher brain activity to control external devices. BCI makes use of EEG signals to classify certain tasks. The EEG signals recorded and features are extracted. Features are extracted using methods like principal components analysis (PCA), common spatial patterns (CSP), or independent components analysis(ICA). In this paper, they made use of CNN to classify Motor Imagery EEG signals. Average accuracy for CNN based classification was 84% better than most others.

P. Sandheep et al. in [45] used EEG signals to classify if a person has depression. Depression is a mental illness that affects a person‟s mind, thoughts, and feelings. The author used deep CNN to train the model using EEG signals on a total of 60 subjects out of which half of them were normal and the other half depressed. Making use of Deep learning gave them an edge as it returned „state of the art‟ accuracy and didn‟t need manual pre- processing or feature extraction. Their architecture consisted of 5 layers: the first convolutional layer succeeded by a single down-sampling(max-pool) layer and then accompanied by classification layer made of three FC layers.The accuracy obtained in [29] was upped by the author by increasing the number of subjects

Yıldırım Ö. et al. in [46] proposed to use a 1-D CNN for instant identification of ordinary or non-ordinary EEG signals and doesn‟t need feature extraction. EEG signals can be tabulated according to their frequentness, magnitude, and magnitude on the scalp. First we need to identify which Class the EEG signals will belong to (Normal or Abnormal). Then using the abnormal signals we can determine the type of brain disease by dividing the signals to sub-groups. The 1-D CNN model was constructed of 23 layers which consisted of 1-D convolution, Max-pooling,batch normalization,Dropout, and dense layers. They achieved 79.34% accuracy in classification.

H. Shao et al. in [47] proposed a modern method to adjust the convolutional kernel of the CNN to adapt to the input of EEG signals by first convolving time-domain information and then convolves the spatial information in the second convolutional layer. Emotion analysis and recognition can be used to identify certain mental illnesses. It consisted of 7 layers – convolutional layer succeeded by max-pooling layer followed by repeating the same structure again followed by a penultimate fully connected layer consisting of 512 nodes and lastly output layer with 3 nodes. The input was 62 channels EEG signal which gave the most accuracy during classification.

An elaborative comparative study between the time domain and frequency domain input was done in [21]. The FDBN (Frequential - DBN) was introduced with three RBMs and one output layer. It has a median accuracy of 82%. The study discussed the FFT (Fast Fourier transform) and WPD (Wavelet packet decomposition). They concluded FFT is better than WPD. These are the two pre-processing methods to convert the EEG time series data into the frequency domain. The RBM is pre-trained on the Frequency domain input. They also compared FDBN with many other models. FDBN outperformed all other models with a classification accuracy of 84% and also proved to be robust. Another study dedicated to BCI and motor imagery classification was presented in [22], Zhongliang

Yu proposed novel multi-class classification method SVD2DM. They used SVD and DBM. Artefact removal is an important step of pre-processing. This was achieved using SVD. In comparison to other models like CSP (Common spatial patterns) & FBCSP (filter bank common spatial patterns) SVD2BM achieved an accuracy of 80.4%. Studies also highlighted the fact that parameters of the model affect the classification performances of SVD2BM.

Study of MRI

The following papers discuss the use of MRI, PET Images in the detection and diagnosis of neurological diseases.

In [48], R. Cui et al. proposes a combined system which comprises MLP and RNN. This study focuses on Longitudinal Analysis, which is a long term analysis of subjects (people), and data (ADNI Database) is collected from various time points rather than just a single instance of time. Thus, relatively longitudinal samples are highly sensitive to early pathological changes. This model uses an 11-layer MLP for single time feature learning and RNN that uses the BGRU (Bidirectional Gated Recurrent Units) for longitudinal feature learning. The model was successful in achieving an accuracy of 89.69%.

In [49], D. Cheng et al. proposes a model involving a hybrid network consisting of CNN and RNN. Positrons Emission Tomography, abbreviated as PET, is a nuclear functional imaging technique. It aids the diagnosis of diseases by observing metabolic processes. The model uses the 3-D images generated by the PET technology, collected from ADNI database. These images are then decomposed into a sequence of 2-D image slices. The 2- D CNN (5 Layered) is modelled to learn the intra-slice characteristics and the BGRUs are used to model the interslice characteristics. These are then integrated for overall classification predictions. The model is claimed to save computational costs by the combined use of RNN and CNN. The model achieves an accuracy of 91.19%.

S. Tabarestani et al. in [50] proposed an implementation of RNN for prediction of Mini-Mental State Examination scores and multiclass multimodal classification process which involves 3 stages of AD"s progression namely, Cognitively Normal, followed by Mild Cognitive Impairment and Alzheimer"s Disease. The proposed technique exploits RNN"s capability to observe temporal dependencies in longitudinal data. This is then used to predict the progression of AD over multiple future time instances. The model uses the ADNI dataset that involves 34 features some of which are Cerebrospinal Fluid, MRI and Cognitive Test Scores. The model predicts the stages with an accuracy of 87-88% over the period of 2 years.

In the study [51], P. Forouzan Nejad et al. pointed out that the count of AD subjects in developed countries is expected to increase over 13 million. Most of these cases can be diagnosed, but by that time the patient might have lost a significant part of their brain function. DNN has the power to model complex non-linear relationships. A study of various research papers and the machine learning or deep learning methodologies used by them are listed out. They proposed a DNN model to which the inputs MRI and PET features were given. By MRI, PET and NTS (Neuropsychological Test Scores) of RAVLT, ECogT & MoCA, achieved an accuracy of 93.4% for diagnosing AD from CN.

In [52], S. O. Orimaye et al. discussed that MCI is typically diagnosed through various neuropsychological exams with questions and images. Tests like MMSE and MoCA are screening tools to assess different cognitive abilities. Using the Dementia Bank dataset"s "Cookie-Theft" picture, the patients" verbal utterances were recorded and transcripts were generated in text format. The model D2NNLM achieved an accuracy of 87.5%. They proposed a D2NNLM model based on the study and success of the DNNLM model. The inputs given to the proposed model was 6-gram 1-skip-trigrams, which reduced the perplexity of the model down to 1.6 and an accuracy of 87.5%. The drawback was that the data set is limited to 1 image, and to insert additional images into the dataset will be a very tedious task for the physicians.

Table 3.1 showcases the use of all the Deep Learning models elaborated above, with a description of the dataset and neural network architecture used to solve a particular problem. The last column depicts the various output parameters that conclude the effectiveness of the algorithm used.

Table 3.1 Comparative Analysis of Deep Computational Models for classification of EEG and MRI data

Ref. No	Number of Subjects	Input	Architecture	Problem	Output Parameters
[31]	12 subjects	22 Ag/AgCl electrodes placed according to the internationally accepted 10/20 electrode system 13 channels	One hidden layer and one output layer.	Restricted Boltzmann Machines in	Median accuracy of 82%
[32]	6 Subjects	17 channels out of which 15 are ranked in descending order and only a few best channels are selected for each subject	Input layer has 17 nodes. Hidden layers are 5. Hidden nodes are 4-40. nodes in the output layer is 4-7	MWL classification using RBM	96.1%
[33]	N/A	100 (128-Channel) EEG signals of 23.6 secs period.	Elman RNN: Includes an input layer, followed by a hidden layer, then a context layer, and output layer.	Detection of Epileptic Seizure.	Accuracy: 100%
[34]	7 Subjects 4 tasks: Letter, Math, Rotation, Counting.	EEG Recording from 6 Electrodes	Elman RNN: 4 Layers	Prediction of Brain State	Accuracy: 2 Tasks: 90% 3 Tasks: 81% 4 Tasks: 81%
[40]	23	980 Hour EEG recording organized in 24 cases and a total of 185 seizures. (18 channels per case.)	RNN: 2 LSTM layers of 32 and 128 memory units 2 LSTM layers of 128 Memory Layers Each	Prediction of Epileptic Seizures.	Sensitivity*: 99.28% - 99.84% FPR*: 0.2-0.11

[41]	1500	7 Recordings per subject. (1 per month up to 6 months) ADNI Dataset	RNN: 1 Hidden Layer (size: 256)	Prediction of Alzheimer‛‛s Disease.	AUC*: 86%

[42]	119 Dementia subjects (63 AD & 56 MCI)	228 Entries (19 channels of which 12 features were extracted.)	6 Layer CNN 2 Hidden ConvNet Layers Input layer: 3 features ConvNet 1: 57 nodes Max-Pool Layer: 19 nodes ConvNet 2: 10 nodes Classification NN: (10,7,2)	To identify if a subject has Alzheimer, MCI or is HC	82% accuracy for MCI vs AD vs HC classification 85% accuracy for MCI/AD vs HC classification
[43]	5 data-sets labeled A-E A,B:5 healthy volunteers C, D, E:5 seizure patients	Bonn university seizure database Signal was obtained by 23 electrodes placed on 5 patient and 100 records were obtained	4 layer 1-D CNN consisting of Conv1D layer: Kernel size=173, stride =1 Max-pool layer Fully connected: 23 nodes Fully connected: 2 nodes	Seizure and Non-seizure EEG signals detection	The accuracy obtained up to 99.4%
[44]	12 subjects	62 electrodes placed according to the Internationally accepted 10/20 electrode system	DBN: single-stream DBN of two or more hidden layers multiple DBNs, four hidden layers multi-stream DBN, four hidden layers	patients suffering from depression classifying into active tDCS and sham tDCS patients	Accuracy depends on selection channels.
[45]	20 subjects CN (10) AD (10)	EEG (4 – 30Hz) 32 channels * 3 RP (theta, alpha and beta bands)	DNN:	Early Detection of AD	[Threshold 50%]
			1 Hidden layer		11/20 50.3%
			2 Hidden layers		12/20 49.1%
			3 Hidden layers		14/20 48.4%
			4 Hidden layers		15/20 59.4%
[46]	5	64 Channel EEG Recording	RNN: Reservoir Layer Readout Layer. and	Artifact Removal for Lane Change Detection.	Accuracy: For window size = 2 sec 86%-88%. For window size = 4 sec 82%.

[47]	3(two male, Average age 25)	EEG recordings from 3 electrodes (C3, C4, Cz) according to the internationally accepted 10-20 electrode system	7-layer CNN Input Layer: 23x32x6 Convnet Layer: 20x20x16 Max-pool Layer: 16x16x32 Convnet Layer: 12x12x32 Max-pool Layer: 8x8x32 Fully Connected: 10 Output Layer: 2	To classify Motor Imagery Task	Average accuracy of 84%
[48]	60 subjects (30 Normal & 30 Depressed)	6000 records with 3000 from normal And 3000 from depressed patients.	5-layer CNN ConvNet 1: [1,5] kernel with stride size 2 Max-Pool Layer: [1,2] nodes FC layer:: 1024 nodes FC layer: 32 nodes FC layer: 2 nodes	To identify if a subject has depression	Accuracy for classification depression in the right brain hemisphere was 99.31% Accuracy for classification depression in the left brain hemisphere was 96.3%
[49]	N/A	TUH EEG CORPUS DATABASE EEG . Signals from 24 to 36 channels were gathered.	23 layer 1-D CNN consisting of Conv1D layer of varying stride, kernel size, and activation ReLu Dropout: rate=0.2 Max-pool: stride = 2 Batch normalization Softmax: Kernel size =2	Classification of EEG signals into Abnormal and normal signals.	Error rate: 20.66% for classification. Accuracy: 79.34%
[50]	15 subjects	SJTU Emotion EEG dataset EEG recordings from 62 channels	7-layer CNN Convnet layer: Kernel size=1*m Max-pool layer: 2*2 Convnet layer: Kernel size=n*1 Max-pooling layer: 2*2 FC layer: 512 nodes Output layer: 3 nodes	EEG based emotion recognition	Accuracy depended on the number of EEG channels with 62 EEG channels used, it gave them the highest accuracy for classification
[51]	9 subjects 5 sessions of per subject	Recorded using three channels (C3, Cz, and C4)	3 hidden layers,+ 1output layer	Motor Imagery Classification based on Restricted Boltzmann Machines	An Average accuracy of 84%

[52]	9 subjects	recorded by 22 electrodes	three hidden layers, with units are 500, 500, and 2000 respectively.	Multi-class Motor Imagery Classification by SVD2BM	after the K-fold cross-validation 80.4% with 6.5 Std deviation
[53]	428 198 (AD) 229 (NC)	MR Images.	MLP: 11 Layers RNN: 2 Hidden BGRU Layers	Prediction of Alzheimer‟s Disease.	Accuracy: 89.7%
[54]	339 93 (AD) 143(MCI) 100 (NC)	3D PET Images (193*153*163)	RNN: 2 BGRU Layers CNN: 12 Layers	Prediction of Alzheimer‟s Disease.	AUC*: AD and NC: 95.28% MCI and NC: 83.90%
[55]	1458 341(CN) 255(EMCI) 529(LMCI) 333 (AD)	MRI, PET, Cognitive Tests, Genetic, Demographic, CSF (Cerebrospinal Fluid).	N/A	Prediction of Alzheimer‟s Disease	Accuracy: Initial– 88% After 1 Year – 87% After 2 Years – 88%
[56]	896 CN - 248 AD - 159 EMCI - 296 LMCI - 193	MRI PET MRI + PET MRI + PET + NTS	DNN 3 Hidden layers	Early Detection of MCI	82.2% 88.9% 89.6% 96.8% (only for Classifying AD and CN)
[57]	N/A	38 transcripts (MCI (210)) (CN (236)) gram=6, skip-gram=1-skip-trigram	Deep-Deep Neural Network Language Model	Early Detection of MCI	87.5% (perplexity = 1.6)

*Sensitivity: Ratio of cases that were predicted as positive to actual no. of positive cases.
*AUC (Area Under Curve): Provides a measure of model‟s quality, irrespective of classification thresholds.
*FPR (False Prediction Rate): Defined as the number of false alarms generated in 1 hour.

Evaluation Of Related Research

Taking references from various research papers and the table above, we were able to differentiate between the ease of using EEG and MRI/PET images. The following table summarizes the differences between EEG and MRI.

Table 3.2 : Comparison between MRI and EEG

MRI, PET Images	EEG
Costly as brain scans need advanced Machinery	Based on EEG, hence less costly.
Does not give instantaneous information about brain activities.	Gives instantaneous information about brain activities.
Involvement of a bunch of specialized doctors such as radiologist, neurologist	Involvement of mainly just a neurologist.
MRI, PET and CT scans are not portable due to heavy machinery.	EEG being a light-weight as compared to brain scans can be portable.
Brain scans have better spatial (space) resolution.	EEG recording have better temporal (time) Resolution.
PET scans make use of radioactive tracers that may have an allergic reaction	EEG doesn't make use of any tracers at all and hence has no allergic reaction.
Brain scans are not preferred for pregnant women because the radiation may have effects on developing features.	There are no effects of EEG on pregnant women and hence are usually safe.
It usually takes a week for results from different brain scans.	Results vary from 48 hours to a week depending on the cause of the EEG test.

The outcome of this comparative study between MRI and EEG is that MRI and PET scans are quite expensive, are not portable and hence EEG can be preferred for use over MRIs. Thus, we have also implemented Deep Learning Algorithms using EEG Seizure Dataset.

4. COMPUTATIONAL ANALYSIS OF DEEP LEARNING MODELS FOR EEG DATA CLASSIFICATION

After reviewing all the papers of different deep learning strategies, we performed the computational analysis of three specific different deep learning strategies viz. RNN & CNN for the classification of EEG data. In this module, we tried to implement the basic model of these three most prominent deep learning algorithms. The implementations of the basic model include inter and intra-modular comparison of the three models based on their performance and their architectures. The dataset used in the basic implementation of the models is explained below.

4.1 DATASETS USED

The EEG seizure dataset from Kaggle.com is used for the said work which is pre-processed EEG seizure data. The dataset consists of the EEG data of 500 subjects. For each subject we have 4097 data points that last over

23.5 seconds. The dataset is well shuffled. All the data points are divided into 23 chunks. For each chunk we have 178 data points for 1 second. These are the EEG recording at different points of time. There are a total of 11,500 rows and 179 columns. The last column i.e. 179th column contains the label y (which can hold the values: 1, 2, 3, 4, and 5) which is the response variable. There are 178 explanatory variables X1 to X178. Here, y is a categorical variable which we have converted it into binary i.e. 0 or 1 label. We have considered the y=1 as seizer and all other variables as non-seizure.

4.2 RECURRENT NEURAL NETWORK (RNN)

Recurrent Neural Networks are emerging as very powerful deep learning strategies. They are considered suitable for the time-series data. We implemented the RNN using LSTM, for classifying EEG data. EEG data that we used was a shuffled form of the original time series data. But still, RNNs performed well, although not the best. We used LSTM models on the dataset by varying the parameters. For ex, the number of hidden layers, nodes, and different activation functions. The train to test ratio was kept at 80% to 20%. We initially modelled the architecture with 3 layers. A single input LSTM layer followed by a hidden dense layer and an output layer. This model gave us the best accuracy of 83.32 %A further increase in the number of nodes in the dense layer actually reduced the accuracy down to 81.22 %. We then found that the LSTM model, where we used a model with 4 layers in which first is input layer, one is hidden LSTM layer and one hidden dense, and an output layer with activation function for the first three layers being tanh, gave an accuracy of 80.3%. The same model with ReLU as the activation function gave an accuracy of 70.10 %. The comparison of each case is depicted in the table below.

Table 4.1: Experimental Results of RNN LSTM classifier

	RNN-LSTM (Model-1)	RNN-LSTM (Model-2)	RNN-LSTM (Model-3)	RNN-LSTM (Model-4)
Layers	3	3	4	4
Layer	LSTM: 128	LSTM: 128	LSTM: 128	LSTM: 128
Description	nodes	nodes	nodes	nodes
	Dense : 32	Dense : 128	LSTM: 128	LSTM: 128
	nodes	nodes	nodes	nodes
	Output : 2 nodes	Output : 2 nodes	Dense:32 nodes	Dense:32 nodes
			Output : 2 nodes	Output : 2 nodes
Activation	Tanh	Tanh	Tanh	ReLU
Accuracy %	83.32 %	81.22 %	81.30 %	70.10 %

4.3 CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is favored with image related data and also works well with the same. But it is said that CNN does not perform well on time series-based data. Recent studies suggested that using 1-DCNN can provide better results on time-series data.

4.3.1 Feature Extraction for Classification

The proposed system consists of 8 layers, divided into five categories:

The convolutional layer, The Max-Pool Layer, Batch Normalization Layer, the Flattening, and the dense layer. The first layer contains a filter used for detection of low-level features and a kernel of size 8 which can also be known as height. In the first layer if we use a single filter then the network will learn a single feature and it might not be sufficient. Hence we will increase the filters to 8. Increasing filters to 8 will train 8 different features in the first layer itself.

Max Pool or the pooling layer is often used after a convolution layer to reduce the dimensions of the output which in turn reduces complications and creates a summary of all important features and also prevents overfitting of the data. In our example, we chose a size of 2.

Another convolutional layer having 16 filters and having a kernel of size 11 is added after batch normalization. The second convolutional layer assists the model to learn more complex features. The results and analysis of the basic implementation of these algorithms are discussed in the following subsections.

4.3.2 Experimentation of 1-D CNN Classifier

We have proposed a 1-DCNN to classify our EEG data. The performance of CNN is very much better than other models. We studied the execution of the 1-Dimensional CNN on the same dataset. We experimented by changing different hyper parameters. For the first 1-D CNN (Model-1) we used 5 layers with the ReLU activation function. First one is the input layer with 8 filters, second is the max-pooling layer. Its pool size was kept 2, the next layer is added to flatten the data, and the second last layer is the hidden layer with 1000 nodes followed by an output layer with 2 nodes since we are doing binary classification. In the second model 1-D CNN (Model-2) we used 7 layers, by doing slight changes. We added a layer after the max-pooling layer with 16 filters and one layer of max pooling after this new layer. This addition enhanced the accuracy of the model to 97.08%. We observed that the model performed differently for each case. The highest accuracy obtained is 98.5%. The comparison depicted in table 4.3.

Table 4.2 : Experimentation Results of 1-D CNN Classifier

	1-D CNN (Model-1)	1-D CNN (Model-2)	1-D CNN (Model-3)
No of Layers	5	10	15
Layer Description	Conv1D (no. Of filters=8) Max-Pool (stride=2, pool_size=2) Flatten Dense(nodes=1000) Dense(nodes=2)	Conv1D (no. Of filters=8) Max-Pool (stride=2, pool_size=2) Conv1D (no. Of filters=16) Max-Pool(stride=2, pool_size=2) Flatten Dense(nodes=1000) Dense(nodes=2)	Conv1D (no. Of filters=8) Max-Pool (stride=2, pool_size=2) Batch Normalization Conv1D (no. Of filters=16) Max-Pool (stride=2, pool_size=2) Flatten Dense(nodes=1000) Dense(nodes=2)
Activation	ReLU	ReLU	ReLU
Accuracy	94.6	96.03	97.2

4.4 PERFORMANCE OF 1-D CNN (MODEL-3) WITH DIFFERENT ACTIVATION FUNCTIONS

4.4.1 Performance Measures

As mentioned above the performance of 1-D CNN gives the best results on our EEG seizure dataset. The 1-D CNN (Model-3) got an accuracy of 97.2%. The implementation of the model is done using TensorFlow and Keras. The performance is evaluated using the evaluate() method which is provided in TensorFlow. The method evaluates the model and has parameters like x which corresponds to the input data, y which is the target data, verbose, batch size, etc. It returns two important values, first the loss value and second metrics value for the model in test mode. The computations are done in batches, and its value is defined as the batch size. The model is implemented using activation functions like sigmoid, tanh, and ReLU. There are many loss functions like Hinge, Kullback-Leibler, Mean absolute error or L1 loss, and Binary cross-entropy. Here, we are doing binary classification; we are using log loss or Binary cross-entropy. The following formula from [34] is used to calculate the binary cross-entropy.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} \quad (y_i . log(p(y_i)) + (1 - y_i). log(1 - p(y_i)) \tag{4.1}$$

Where, y is the label or ground truth & p(y) is the predicted probability.

4.4.2 Implementation Details & Initialization of parameters

To implement our 1-D CNN (Model-3) consists of eight layers with a single input layer, six hidden layers, and finally an output layer. The implementation was done in python 3. We also used various python libraries like Keras and TensorFlow. Many other libraries like pandas and NumPy were used to pre-process the data derived from our .csv file. Methods like dense, flatten, Dropout, Sequential, Batch Normalization etc. from Keras. To build CNN, methods like Conv1D and MaxPooling1D were also used. For the input layer, the number of filters is kept 8 and for the subsequent layer, they are increased to 16. In the MaxPooling layer, the pool size is kept as 2. A flattening layer is added to flatten the input data without disturbing the batch size. The pre-final layer and final layer are added to the model with 1000 and 2 nodes respectively. Due to its best performance among other activation functions, for the input and hidden layer ReLU was used and for the final output layer sigmoid was used for binary output. While configuring the model for training, the Adam optimizer is used along with binary loss function. Eventually, the model is fitted and evaluated.

4.4.3 Results and Discussion

Referring to table 4.1, the 1-D CNN architecture is implemented. The numbers of layers, and nodes were different in all three models. They produced different accuracies. It has also been studied with various activation functions and gave different results.

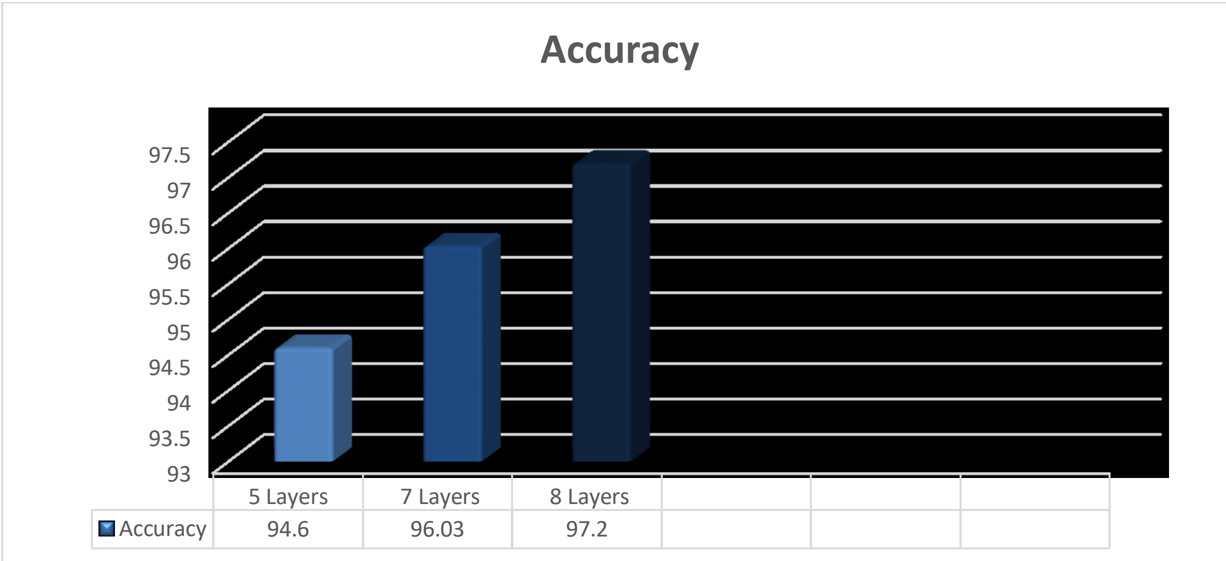

Figure 4.1: Comparison of accuracies of 1-D CNN models with different layers

5. CONCLUSION

The EEG signals have a high potential of uncovering many neurological disorders in an early stage. We have shown why it is better than MRI. Being cheap and efficient, it can be used for many researches even with a lower budget as compared to MRI. Many Deep Learning algorithms have been applied on EEG signals in order to classify various neurological disorders and brain interface applications. In this paper we have done an extensive literature survey about application of different neural networks on EEG signals and how using these neural networks one was able to classify several neurological diseases like Seizure, AD and Depression. The preprocessing step greatly affects the performance of the model. The general preprocessing methods used are Independent Component Analysis (ICA) followed by discrete wavelet transform (DWT),FFT, Z-score normalization, Continuous wavelet transform (CWT) & discrete Fourier transform (DFT). The studies revealed that the choice of the model depends on the type of input data used. Applying CNN on time-series EEG resulted in accuracies ranging from 79% to 85%. One of the papers gave an accuracy of 99% due to use of Ten- fold cross validation and increased number of records. Using 1-D CNN improved the accuracy to 99% with a decrease in computation time and resources, making it an appropriate choice for classifying EEG signals. Our study also revealed that RNN-LSTM models are perfect for time-series data and work significantly well on EEG data also. The accuracies for Neurological Disease prediction varied from 83.9% to 99%, whereas in classification tasks related to various activities performed by the brain accuracies varied from 81% to 90%. Various hybrid models including CNN+RNN for image dataset and MLP+RNN were also tested by many authors. This hybrid modeling remains a field to be explored and holds a great potential to outperform other single models. Further studies in Deep belief Networks showed that these are emerging as good classifiers as compared to the conventional classifiers. Many architecture models of Restricted Boltzmann Machine gave an accuracy ranging from 80% to 96%. A model which implemented for the mental workload classification attains an accuracy of 96.1%. One study also suggests that the model gives better accuracy on frequency domain input rather than time domain. The selection of channels played a vital role in the execution of the DBN model. The DBN models are claimed to be robust but they need to be optimized. DNN or MLP, performs well when the dataset is in a tabular and numeric format. The success of MLP ranges from 75% with EEG classification using relative powers of EEG sub-bands to 96.8% by using MRI and PET imagery along with NTS values. But procuring the later data is not cost efficient. In the later part of study the Computational analysis was done using three specific different deep learning strategies viz. RNN, MLP & CNN to classify EEG data. As observed from the Experimental Results and Literature Survey, 1d-CNN model performed better than RNN and MLP on EEG Time Series. At the raw level, EEG signals are a time-series based data. CNN, in general, performs better with image related data. Hence, we made use of 1-D CNN which applies well to the time-series data acquired by sensors. By applying 1-D CNN model we had a classification accuracy of 98.5% thus suggesting that 1-D CNN is applicable to EEG data which is scarce, low-cost and real-time implementation is required. Deep learning on EEG classification has to be explored more. Our study is aimed at providing a detailed literature survey for other researchers who are willing to contribute to this field.

REFERENCES

[1].Baltatzis V, Bintsi K-M, Apostolidis G K and Hadjileontiadis L J 2017 Bullying incidences identification within an immersive environment using HD EEG-based analysis: a swarm decomposition and deep learning approach Sci. Rep. 7 17292 Crossref Google Scholar

[2].Abbas W and Khan N A 2018 DeepMI : deep learning for multiclass motor imagery classification 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC) pp 219–22

[3].Tabar Y R and Halici U 2017 A novel deep learning approach for classification of EEG motor imagery signals J. Neural Eng. 14 016003 IOPscience Google Scholar

[4].Pereira A et al 2018 Cross-subject EEG event-related potential classification for brain–computer interfaces using residual networks Preprint (HAL-id:hal-01878227) Google Scholar

[5].Vrbancic G and Podgorelec V 2018 Automatic classification of motor impairment neural disorders from EEG signals using deep convolutional neural networks Elektron. Elektrotech. 24 3–7 Crossref Google Scholar

[6].Tang Z, Li C and Sun S 2017 Single-trial EEG classification of motor imagery using deep convolutional neural networks Optik 130 11–8 Crossref Google Scholar

[7].Dose H, Møller J S, Iversen H K and Puthusserypady S 2018 An end-to-end deep learning approach to MI-EEG signal classification for BCIs Expert Syst. Appl. 114 532–42 Crossref Google Scholar

[8].Wang Z, Zhang Z, Gong X, Sun Y and Wang H 2018 Short time Fourier transformation and deep neural networks for motor imagery brain computer interface recognition Concurrency and Computation: Practice and Experience 30 e4413 Crossref Google Scholar

[9].By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=4567 9374

[10]. Ntoniades A, Spyrou L, Took C C and Sanei S 2016 Deep learning for epileptic intracranial EEG data 2016 IEEE 26th Int. Workshop Machine Learning Signal Processing pp 1–6 Crossref Google Scholar

[11]. Salama E S, El-khoribi R A, Shoman M E and Shalaby M A W 2018 EEG-based emotion recognition using 3D convolutional neural networks Int. J. Adv. Comput. Sci. Appl. 9 329–37 Crossref Google Scholar

[12]. Ajayi, O.E., Ajayi, E.A., Akintomide, O.A., Adebayo, R.A., Ogunyemi, S.A., Oyedeji, A.T.,

Balogun, M.O.Ambulatory blood pressure profile and left ventricular geometry in Nigerian hypertensives(2011) Journal of Cardiovascular Disease Research, 2 (3), pp. 164-171. DOI: 10.4103/0975-3583.85263

[13]. Qiao R, Qing C, Zhang T, Xing X and Xu X 2017 A novel deep- learning based framework for multi-subject emotion recognition ICCSS 2017—2017 Int. Conf. Information, Cybernetics and Computational Social Systems pp 181–5

[14]. Gait-pattern E, Goh S K, Abbass H A, Member S and Tan K C 2018 Spatio—spectral representation learning for classification IEEE Trans. Neural Syst. Rehabil. Eng. 26 1858–67 Crossref Google Scholar

[15]. Kobler R J and Scherer R 2016 Restricted boltzmann machines in sensory motor rhythm brain–computer interfacing: a study on inter-subject transfer and co- adaptation 2016 IEEE Int. Conf. Syst. Man, Cybernetics SMC 2016—Conf. Proc. pp 469–74

[16]. Zheng W L and Lu B L 2015 Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks IEEE Trans. Auton. Ment. Dev. 7 162–75 Crossref Google Scholar

[17]. Li F et al 2017 Deep models for engagement assessment with scarce label information IEEE Trans. Hum.-Mach. Syst. 47 598–605 Crossref

[18]. Huang J, Xu X, Zhang T and Chen L 2017 Emotion classification using deep neural networks and emotional patches 2017 IEEE Int. Conf. Bioinformatics Biomedicine pp 958–62

[19]. Mohamed Saleem TS, Jain A, Tarani P, Ravi V, Gauthaman K. "Aliskiren: A Novel, Orally Active Renin Inhibitor." Systematic Reviews in Pharmacy 1.1 (2010), 93- 98. Print. doi:10.4103/0975-8453.59518

[20]. https://www.researchgate.net/figure/Architecture-of-a- Deep-Belief-Networks-DBNs-and-b-Restricted-Boltzmann- Machines_fig1_332671752 [accessed 9 Feb, 2020]

[21]. Zhao Y 2014 Deep learning in the EEG diagnosis of Alzheimer's disease 2014 Asian Conf. Computer Visison pp 1–15

[22]. Kulasingham J P, Vibujithan V and Mieee A C D S 2016 Deep belief networks and stacked autoencoders for the P300 guilty knowledge test 2016 IEEE EMBS Conf. Biomedical Engineering Sciences pp 127–32 Crossref Google Scholar

[23]. Bablani A, Edla D R and Kuppili V 2018 Deceit identification test on EEG Data using deep belief network 2018 9th Int. Conf. Computing Communication Networking Technologies pp 1–6

[24]. Hajinoroozi M, Mao Z and Huang Y 2015 Prediction of driver's drowsy and alert states from EEG signals with deep learning 2015 IEEE 6th Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing CAMSAP 2015 pp 493–6 Crossref Google Scholar

[25]. Li X, Song D, Zhang P, Yu G, Hou Y and Hu B 2016 Emotion recognition from multi-channel EEG data through convolutional recurrent neural network 2016 IEEE Int. Conf. Bioinformatics Biomedicine pp 352–9.

[26]. Bashivan, Pouya, et al. "Learning representations from EEG with deep recurrent-convolutional neural networks." arXiv preprint arXiv:1511.06448 (2015).

[27]. Bresch E, Großekathöfer U and Garcia-molina G 2018 Recurrent deep neural networks for real-time sleep stage classification from single channel EEG Front. Comput. Neurosci. 12 85

[28]. Dong H, Supratak A, Pan W, Wu C, Matthews P M and Guo Y 2018 Mixed neural network approach for temporal sleep stage classification IEEE Trans. Neural Syst. Rehabil. Eng. 26 324–33 Crossref Google Scholar

[29]. Hussein R, Palangi H, Ward R K and Wang Z J 2019 Deep neural network architecture for robust detection of epileptic seizures using EEG signals Clin. Neurophysiol. 130 25–37.

[30]. Kalaiselvi, C., & Nasira, G. M. (2014, February). A new approach for diagnosis of diabetes and prediction of cancer using ANFIS. In 2014 World Congress on Computing and Communication Technologies (pp. 188-190). IEEE.

[31]. Reinmar J. Kobler and Reinhold Scherer Institute of Neural Engineering Graz University of Technology Stremayrgasse 16, Graz, Austria "Restricted Boltzmann Machines in Sensory Motor Rhythm Brain- Computer Interfacing: A Study on Inter-Subject Transfer and Co-Adaptation," 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 | October 9-12, 2016 • Budapest, Hungary

[32]. Jianhua Zhang, Sunan Li, "A deep learning scheme for mental workload classification based on restricted Boltzmann machines," Springer-Verlag London Ltd. 2017

[33]. M. A. Naderi and H. Mahdavi-Nasab, "Analysis and classification of EEG signals using spectral analysis and recurrent neural networks," 2010 17th Iranian Conference of Biomedical Engineering (ICBME), Isfahan, 2010, pp. 1-4.

[34]. S. Patnaik, L. Moharkar and A. Chaudhari, "Deep RNN learning for EEG based functional brain state inference," 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, 2017, pp. 1-6.

[35]. N. Mammone et al., "Brain Network Analysis of Compressive Sensed High-Density EEG Signals in AD and MCI Subjects," in IEEE Transactions on Industrial Informatics, vol. 15, no. 1, pp. 527-536, Jan. 2019.

[36]. D. V Moretti, "Theta and alpha EEG frequency interplay in subjects with mild cognitive impairment: evidence from EEG, MRI, and SPECT brain modifications," Front. Aging Neuro., vol.7, p.31, 2015.

[37]. Li G, Lee CH, Jung JJ, Youn YC, Camacho D. Deep learning for EEG data analytics: A survey.*Concurrency Computat Pract Exper*. 2019;e5199.

[38]. C. Ieracitano, N. Mammone, A. Bramanti, S. Marino, A. Hussain and F. C. Morabito, "A Time-Frequency based Machine Learning System for Brain States Classification via EEG Signal Processing," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8.

[39]. G. Fiscon et al., "Alzheimer's disease patients classification through EEG signals processing," 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Orlando, FL, 2014, pp. 105-112.

[40]. K. Tsiouris, V. Pezoulas, M. Zervakis, S. Konitsiotis, D. Koutsouris and D. Fotiadis, "A Long Short-Term Memory deep learning network for the prediction of epileptic seizures using EEG signals," *Computers in Biology and Medicine*, vol. 99, pp. 24-37, 2018. Available: 10.1016/j.compbiomed.2018.05.019

[41]. M. Nguyen, N. Sun, D. C. Alexander, J. Feng and B. T. T. Yeo, "Modeling Alzheimer‟s disease progression using deep recurrent neural networks," 2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI), Singapore, 2018, pp. 1-4.

[42]. F. C. Morabito et al., "Deep convolutional neural networks for classification of mild cognitive impaired and Alzheimer's disease patients from scalp EEG recordings," 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, 2016, pp. 1-6.

[43]. T. T. Chowdhury, A. Hossain, S. A. Fattah and C. Shahnaz, "Seizure and Non-Seizure EEG Signals Detection Using 1-D Convolutional Neural Network Architecture of Deep Learning Algorithm," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-4.

[44]. Alaa M. Al-kaysi, Ahmed Al-Ani, and Tjeerd W. Boonstra, "A Multichannel Deep Belief Network for the Classification of EEG Data," Springer International Publishing Switzerland 2015

[45]. D. Kim and K. Kim, "Detection of Early Stage Alzheimer‟s Disease using EEG Relative Power with Deep Neural Network," 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, 2018, pp. 352-355.

[46]. M. Moinnereau, S. Karimian-Azari, T. Sakuma, H. Boutani, L. Gheorghe and T. H. Falk, "EEG Artifact Removal for Improved Automated Lane Change Detection while Driving," 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 2018, pp. 1076-1080.

[47]. Jianhua Wang , 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA) , 2019

[48]. P. Sandheep, S. Vineeth, M. Poulose and D. P. Subha, "Performance analysis of deep learning CNN in classification of depression EEG signals," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 1339-1344.

[49]. Yıldırım, Ö., Baloglu, U.B. & Acharya, U.R. “A deep convolutional neural network model for automated identification of abnormal EEG signals,” *Neural Comput&Applic* (2018)

[50]. H. Shao, J. Wang, Y. Wang, Y. Yao and J. Liu, "EEG-Based Emotion Recognition with Deep Convolution Neural Network," 2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS), Dali, China, 2019, pp. 1225-1229.

[51]. Na Lu*, Tengfei Li, Xiaodong Ren, Hongyu Miao, “A Deep Learning Scheme for Motor Imagery Classification based on Restricted Boltzmann Machines” 2016 IEEE.

[52]. ZhongliangYu, JinchunSong School of Mechanical Engineering and Automation Northeastern University Shenyang, China, “Multi-class Motor Imagery Classification by Singular Value Decomposition and Deep Boltzmann Machine,” 2017 IEEE

[53]. R. Cui, M. Liu and G. Li, "Longitudinal analysis for Alzheimer's disease diagnosis using RNN," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, 2018, pp. 1398-1401.

[54]. D. Cheng and M. Liu, "Combining convolutional and recurrent neural networks for Alzheimer's disease diagnosis using PET images," 2017 IEEE International Conference on Imaging Systems and Techniques (IST), Beijing, 2017, pp. 1-5.

[55]. S. Tabarestani et al., “Longitudinal Prediction Modeling of Alzheimer Disease using Recurrent Neural Networks,” 2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), Chicago, IL, USA, 2019, pp. 1-4.

[56]. P. Forouzannezhad, A. Abbaspour, C. Li, M. Cabrerizo and M. Adjouadi, “A Deep Neural

Network Approach for Early Diagnosis of Mild Cognitive Impairment Using Multiple Features," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, 2018, pp. 1341- 1346.

[57]. S. O. Orimaye, J. S.-M. Wong, and J. S. G. Fernandez, "Deep neural network language models for predicting mild cognitive impairment," in BAI@ IJCAI, 2016, pp. 14–20.

[58]. llah, Ihsan, Muhammad Hussain, and HatimAboalsamh, "An automated system for epilepsy detection using EEG brain signals based on deep learning approach" Expert Systems with Applications, 107 (2018): 61-71.

[59]. Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2018). "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," Computers in Biology and Medicine, 100, 270–278.

[60]. Y. Han and H. Bin, "Brain-Computer Interfaces Using Sensorimotor Rhythms: Current State and Future Perspectives," Biomedical Engineering, IEEE Transactions on, vol. 61, pp. 1425-1435, 2014.

[61]. D. Zhang and D. Shen, "Multi-Modal Multi-Task Learning for Joint Prediction of Multiple Regression and Classification Variables in Alzheimer‟s Disease," NeuroImage, 2012.

[62]. E. Moradi et al., "Machine learning framework for early MRI-based Alzheimer‟s conversion prediction in MCI subjects," NeuroImage, 2015.

[63]. X. Zhang et al., "Bayesian model reveals latent atrophy factors with dissociable cognitive trajectories in Alzheimer‟s disease," Proc Natl Acad Sci USA, 2016.