EPILEPSY DETECTION USING NEURO FUZZY SYSTEMS

Project Report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Ishu Priya (151424)

Under the Supervision of

Dr. Hemraj Saini

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Epilepsy Detection Using Neuro Fuzzy Systems"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2019 to May 2019 under the supervision of **Dr. Hemraj Saini,**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student's Signature)
Ishu Priya (151424)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor's Signature)
Dr. Hemraj Saini
Associate Professor
Computer Science and Engineering
Dated:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who supported me throughout the entire course of this B. Tech project and without whose guidance and support it would have been quite a difficult task.

I am highly thankful to **Dr. Hemraj Saini, Project Supervisor,** for his valuable supervision in this project, whose constant support and motivation encouraged me to do the project efficiently. I am also grateful to my parents and friends for their support. I gratefully acknowledge each one of them.

Date: 07/05/2019

Ishu Priya (151424)

TABLE OF CONTENT

LIST OF ABBREVIATIONS

- ANN - Artificial Neural Network
- FL – Fuzzy Logic
- NFS – Neuro Fuzzy System
- SLFF – Single Layer Feed Forward
- NNA – Neural Network Architecture
- NFS – Neuro Fuzzy System
- ML – Machine Learning
- SL – Supervised Learning
- UL - Unsupervised Learning
- RL – Reinforcement Learning
- LTLF – Long Term Load Forecasting
- MTLF – Medium Term Load Forecasting
- STLF – Short Term Load Forecasting
- ReLU – Rectified Linear Unit

LIST OF FIGURES

LIST OF GRAPHS

ABSTARCT

Artificial Neural Networks (ANN) refers to the academic and intellectual investigations which make use of mathematical formulations for modeling the operations based on nervous system. These are helpful in a variety of everyday applications. Neural Networks refer to a meaningfully another approach for using computers in the workplace. A neural network is used to represent patterns and relations in data. It does not require explicit coding of the various problems. It contains various interconnected group of several artificial neurons and generates information using a connection – oriented approach for further computation.

Fuzzy logic consists of values ranging between 0s and 1s.

Neuro –Fuzzy hybrid systems combine the advantages of both fuzzy systems as well as neural networks in which fuzzy systems are well explained and understood.

CHAPTER-1

INTRODUCTION

Artificial Neural Networks:

In ANNs, all neurons operate simultaneously that leads to a parallel structure. It is this structure that leads them to perform faster than conventional computers.

Fuzzy logic:

Fuzzy logic is an extension of multivalued logic which is a logical system. It displays output in the form of values between 0 and 1. FL is based on simple if –then rule. FL resembles human reasoning.

Neuro-fuzzy Systems:

Neuro-fuzzy gives an intelligent system which is a hybrid system and is a combination of both NN and FL. Neuro-fuzzy system involves human-like reasoning through the use of fuzzy sets and implicit knowledge of NN.

Basic Building Blocks of ANN

These include:

1. Network Architecture

2. Setting the weights

3. Activation Function

Network Architecture:

Architecture is the configuration of neurons in such a way that neurons form layers and the pattern of interconnection between them. The number of layers is the number of links. These links have certain values called weights and these links are connected to each other. Hidden Layers are the ones having interconnected weights.

Neuron: Neuron refers to the building block of a NN. Several numbers of inputs and bias value are supplied to the neuron. The signal or value gets multiplied by a weight value when it arrives. E.g a neuron having four inputs, consist of 4 values of weight that can then be altered at the time of the NN training.

Connections: It associates a single neuron in one layer to some other neuron in the other layer or a similar layer. An association or connection dependably consists a weigh value related with it. Objective of the training is to refresh that value of the weight to diminish the error or loss.

Input Layer: This refers to the very first layer in the NN. Input values are fed to this layer and are then passed on to the next layers of the NN. No operations are applied on the input signals in this layer and no weights and bias values are associated with it. There are 4 inputs in the given figure below.

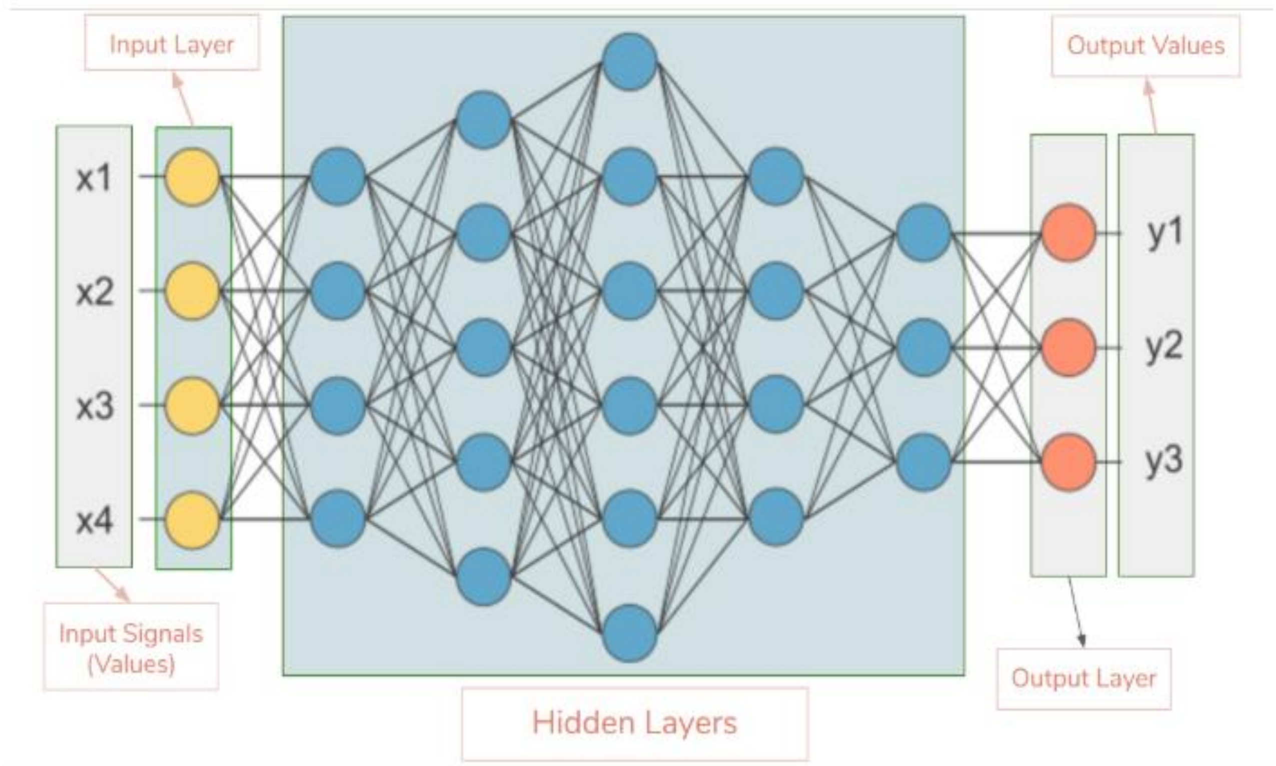

Fig. 1: Neural Architecture

Hidden Layers: Hidden layers have neurons (nodes) which apply different changes to the info or the input data. A single hidden layer is a collection of neurons stacked vertically (Representation). In the system given beneath there are 5 such layers. In this framework, first layer (hidden) has 4 neurons (nodes), second has 5 neurons, third has 6 neurons, fourth has 4 and fifth has 3 neurons. Prop up hidden layer passes on values (qualities) to the output layer. All of the neurons in a hidden layer are related with each and every neuron in the accompanying layer, in this way we have a totally related hidden layers.

Output Layer: This layer is the last layer in the framework and gets commitment from the last hidden layer. With this layer we can get needed number of characteristics and in a perfect range. In this framework we have 3 neurons in the output or final layer and it yields y1, y2, y3.Various types of network Architectures are competitive net, feedback, feed forward, fully interconnected net, etc.

Feed Forward refers to the Architectures in which input units activations are set and are afterwards propagated through the network until the output is determined. Forward propagation is a technique of continuing values form the input to the neural framework and getting an output which is called as the predicted output value. Now and again we imply forward propagation as a inference or derivation. When we feed these input value to the neural framework's first layer, it relinquishes any tasks. The second layer gets values from the primary layer and applies different tasks, for example, multiplications, summations and activations functions and passes this to the accompanying layer. Same system reiterates for resulting layers ultimately we get an output from the last layer.

Single Layer Feed Forward consists of a layer having interconnections which are weighted. The output units are fully connected by input units. In this kind of net, the weights of various output units are not affected by each other.

Multilayer Feed Forward refers to the net in which flow of signals takes place from input units to output units in a forward direction. There can be more than one layer of nodes between input and output units.

Competitive Net is as same as SLFF except that the output nodes usually have negative connections as a result of which there is a competition between the output nodes to represent the current input pattern. The connections may be restricted or completely connected.

Recurrent Net is the one of the simplest ANN in which every unit is input as well as output and all of these are connected to each other. These allow networks to process sequential information that depends on the network state.

2. Setting the weights

The process of training or learning is enabled by assigning certain values called weights. Training a network is the process in which the weights of the connections are modified between the layers of network aiming to achieve the outcome which is expected. When a network is trained, an internal process takes place which is called learning.

3. Activation Function

The output response of a neuron is calculated by Activation Function. Response is obtained by applying activation to the sum of weighted input signal. Same activation functions are used for neurons in the same layer. These functions are either linear or they can be non-linear.

Fuzzy Systems

Fuzzy logic contains similarities and differences with boolean logic that are restricted to 0 and 1 but fuzzy logics allow values between 0s and 1s. It is more similar to human thinking and is basically based on the degrees of truth.

Neural Networks and Fuzzy Systems are important new technologies for industrial and consumer applications. It is advantageous to combine neuro with fuzzy thus creating a hybrid system known as Neuro- Fuzzy System. NFS consists of various logical operators e.g. min, max etc and product operators that are used in conventional neural networks.

Fuzzy Logic (FL) resembles human reasoning.

Fuzzy Logic Systems Architecture

It consists of four major parts:

Fuzzification Module

Knowledge Base

Inference Engine

Defuzzification Module

Fuzzification Module is used to transform the system inputs into fuzzy sets. The input signal is split into five steps which are Large Positive, Medium Positive, small, Medium Negative and Large Negative.

Knowledge Base stores if-then rule.

Inference Engine copies the human reasoning process by implementing fuzzy inference on inputs.

Defuzzification Module converts the fuzzy set obtained by the inference engine to crisp value. Crisp value is binary value whose value is 0 or 1.

In general, Fuzzy Logic gives the values between 0 and 1.It is based on “degrees of truth” and not true or false values.

Neuro- Fuzzy Systems

It is one of the most used forms of hybrid systems and provides various applications in day to day works. NN an FL represent different methodologies in order to deal with uncertainty. Each of these has their own advantages & disadvantages. NN is used to model complex nonlinear relations and are mostly used for classification purposes whereas in FL inputs and outputs are directly defined using fuzzy sets and provide more flexibility in formulating system descriptions. NFS refers to integration of both NN and FL thus, making use of advantages of both the systems.

The NFS consists of the components of FS except that computation is a layered computation of hidden neurons and NN's learning capacity is used for the enhancement of the knowledge of the system. These systems are highly advantageous as they provide both precision as well as learning capability of NNs and still are as easy as fuzzy systems to understand.

A neuro -fuzzy system is basically a 3-layer feed forward neural network in which the first layer represents the various input variables, the middle layer which is a hidden layer, represents fuzzy rules and the third layer represents output variables. Fuzzy sets are encoded as connection weights. In some cases, five layered architecture can also be used where the second and fourth layer represent the fuzzy sets. However, this 5 layered architecture can be transformed into a three-layered one.

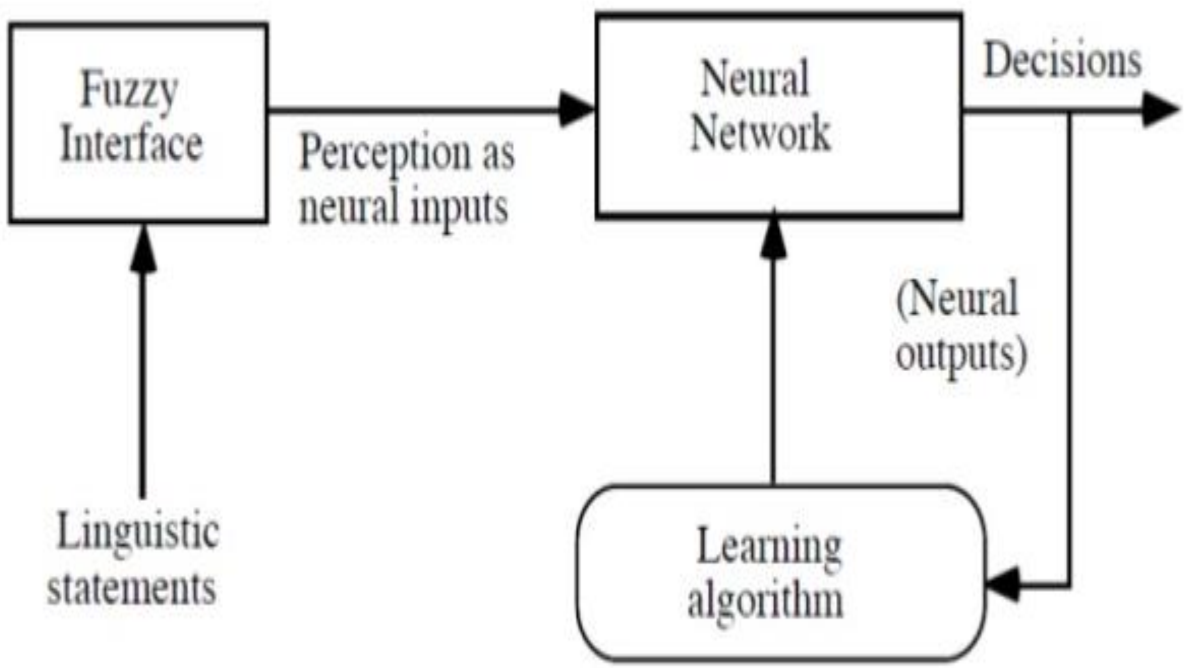

Fig. 2: Structure of a neuro-fuzzy system

1.2 Problem Statement

The problem represents a method for diagnosis of epilepsy in patients having brain related disorders. The purpose is to develop a system which best detects the highest possible order of epilepsy. This is done through collection of large data samples and hence, a diagnosis for the same is prepared using neuro- fuzzy systems.

1.3 Project Objective

The main objectives of this project are:

- To develop a computer aided classification system which can detect epilepsy disorder in patients as per the collected data automatically using the concept of neuro fuzzy systems.
- The benefaction of the given work related to this project is to figure out the best possible classification system making use of the derived features characterizing the epilepsy disorders.

1.4 Methodology

To design the neuro- fuzzy system for epilepsy detection representing enhancement in the signal processing and also machine learning techniques thus making this possible to generate a system to diagnose epilepsy in various patients analyzing the data sets gathered. Once the data is gathered, various algorithms and logics are designed to build the currently required system to detect the epileptic disorder from the given data and then the system so developed is to be made more efficient and accurate to increase the efficiency and reliability.

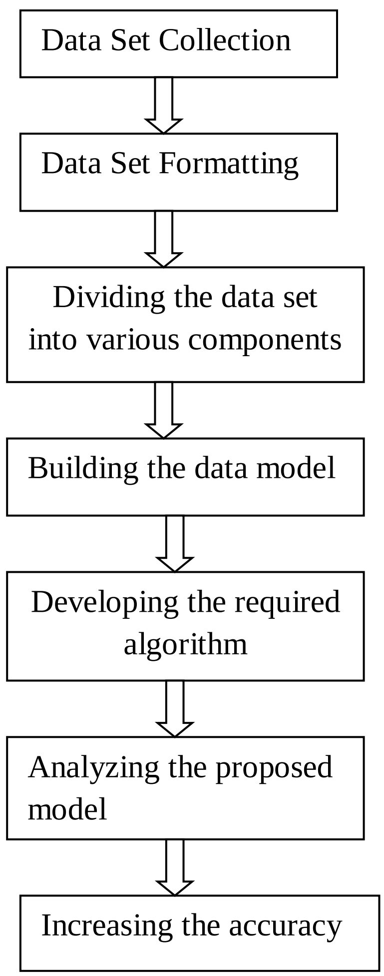

Fig. 3: Methodology for the proposed model

Steps in details:

1. Firstly, the data set needs to be collected for the development of the respective model for epilepsy detection.
2. Then the data set needs to be analyzed and accordingly is to be divided into training data and testing data.
3. Now, the data set is to be normalized.
4. Once, the data set is formatted and normalized, the network architecture is to be designed.
5. An algorithm is developed to bring out the aim of the project into implementation i.e. to detect epilepsy using the neuro fuzzy systems from the acquired data set of various patients.
6. Errors are removed or reduced and the proposed model is analyzed for its effectiveness and efficiency.
7. The accuracy is to be increased in order to attain better performance results.

   The data set consists of 11501 data inputs having (178 input variables and single output variable), out of which 11300 data inputs are taken as training data sets and the rest are considered as testing data sets.

1.5 Organization

The association of this project work is as follows:

- Introduction: This part exhibits the various definitions of the terms related to neural networks, fuzzy logics and hence, neuro- fuzzy systems.
- Literature Survey: This part consists of the related work in the detection of epileptic seizures.
- System Design: This part consists of analysis of the problem, hence developing and designing of the related algorithms to be used for the same.
- Algorithms: This part contains the explanations of the various algorithms used and designed to bring out the project into implementation.
- Test Plan: This section consists of the Data Set, Metrics, and Test Setup etc.
- Results and Performance Analysis: This part consists of the system developed by using the various algorithms and methods and shows the various outputs generated.
- Conclusions: This section depicts the jest of the entire project procedure representing what has been depicted and diagnosed.
- References: This section represents the various sources where the data and research work has been gathered from.

CHAPTER-2

LITERATURE SURVEY

S.No	Author	Year	Title	Remarks	Advantages	Disadvantages
1	**SN Sivanandam**	**2015**	**Introduction to Neural Networks**	**It gives an in-depth and broad overview of the field of neural networks.**	**This gave a complete overview of how the NN works and what it actually refers to.**	**The applications mentioned were not elaborated clearly.**
2	**Sonali. B. Maind**	**2014**	**Basics of Artificial Neural Network**	**It gives an overview of ANN, its working & training**	**All the basics of related to ANN are appropriately covered.**	**Unpredictable operations used.**
3	**K. Vijaya Sri**	**2012**	**Neuro-fuzzy systems and applications**	**It lists principal variants of soft computing.**	**Well defined NN architecture and its many applications.**	**Fuzzy logic not explained much appropriately.**
4	**Aditya Sundar**	**2015**	**Matlab Analysis of EEG signals for diagnosis of epileptic seizures.**	**It gives a method diagnosis of epileptic seizure with brain related disorders in patients.**	**Gave a remarkable diagnosis of ANN related to the main aim of the project**	**The method used for diagnosis is a bit difficult to understand for a layman.**

The above mentioned are the names of a few books that have been referred to gather the knowledge about the respective project work that needs to be done.

Machine Learning (ML) is of three types:

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Supervised Learning:

In supervised learning, well defined data is used while training an algorithm. SL depends on data in which true class of the data is revealed. We can say that there is an external supervisor who has enough knowledge of the environment and he helps to better understand and complete the task. In other words, it uses a labeled or well –defined data set.

Supervised learning is the Data mining assignment of gathering a capacity from named preparing data. The preparing data comprises of a lot of preparing precedents. In supervised adapting, each model is a couple comprising the information objects (normally a vector) and a desired output value or esteem (additionally called the supervisory flag). Supervised learning calculations break down this information and generate a construed work, which can be used for the mapping of new precedents. A standard situation will take into considerations the calculation to effectively decide the class names for concealed occurrences. This requires the taking of calculations to sum up from this preparation information to inconspicuous circumstances in a sensible manner.

Unsupervised Learning:

In unsupervised learning, the machine is provided with a dataset without explicit instructions on what to do with it. The system works by itself to identify the problem. The data set is not well-defined or labeled. The algorithm is just provided with a large amount of data and characteristic of each observation. Unsupervised learning is the preparation of machine utilizing data that is neither characterized nor named and enabling the calculation to follow up on that data without direction. Here the assignment of machine is to gather unsorted data as indicated by similitudes, examples and contrasts with no earlier preparing of information.

In contrast to supervised learning, no instructor is given that implies no preparation will be given to the machine. In this manner machine is limited to locate the shrouded structure in unlabeled information by our-self.

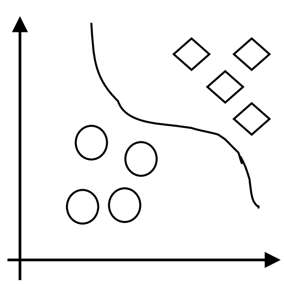

Fig. 4: Supervised Learning

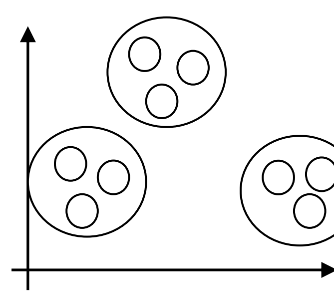

Fig.5: Unsupervised Learning

The above figure depicts the difference between Supervised and Unsupervised Learning.

Reinforcement Learning:

Reinforcement learning is an intermediate between SL and UL.RL aids the machine to learn from its progress. The input itself depends on the action taken. It tries to find the best possible ways to earn the greatest reward. RL works well on small dynamic systems.

Reinforcement learning is a zone of Machine Learning. It is tied in with making appropriate move to augment remunerate in a specific circumstance. It is utilized by different programming and machines to locate the most ideal conduct or way it should take in a particular circumstance. Support taking in contrasts from the supervised learning in a manner that in reinforcement learning the preparation information has the appropriate response key with it so the model is prepared with the right answer itself while in reinforcement learning, there is no answer yet the reinforcement specialist chooses what to do to play out the given assignment. Without preparing dataset, it will undoubtedly gain from its experience.

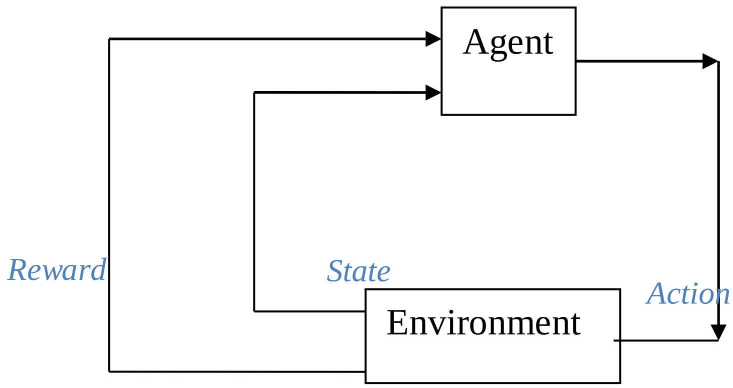

Fig. 6: Reinforcement Learning

Matlab Analysis of EEG signals for diagnosis of epileptic seizures *by Aditya Sundar, 2015*

Aditya Sundar, Texas Instruments Inc., incurred that approximate entropy is a technique which is basically used to quantify the amount of regularity and hence the unpredictability of fluctuations over time-series data in statistics. The accompanying report shows a strategy conclusion of epileptic seizure in patients with mind related disarranges. This is done through gathering of vast examples of EEG information of patients appearing and no seizures and utilizing certain component extraction strategies to help interestingly group the seizure-signals. The information of patients with seizures was removed from ictal area of the cerebrum. The interim in which the patients hinted at no seizure was interictal information. The flag test space utilized comprises of 200 EEG ages, 100 ictal and 100 between ictal. This proposition centers around the usage of paper.

Over the span of this undertaking the wavelet based flag handling procedure was considered and utilized for highlight extraction. Highlights like example entropy, estimated entropy, repeat quantifiers, wavelet vitality, and so forth of the primary flag just as the disintegrated signs were removed. These highlights were then used to group the EEG signals. A wide range of ordering procedures/ calculations was utilized. Their exactnesses were then tried. The arbitrary timberland calculation was found to give the most precise outcomes. Results demonstrate that these extricated could really be utilized for seizure location with great exactness. Scalograms, time-recurrence plots, box-plots, ROC bends were among different instruments used to remove data from the EEG signals. (Sundar, 2015)

Detection of partial seizure: An application of fuzzy rule system for wearable ambulatory systems *by Mohamed Shakir; Aamir Saeed Malik; Nidal Kamel; Uvais Qidwai, 2014*

Electroencephalography (EEG) assumes a wise job, particularly EEG based wellbeing finding of cerebrum issue, just as mind PC interface (BCI) applications. One such research field is identified with epilepsy. The EEG based strategies are not will intended for pre-event acknowledgment plan to identify and foresee halfway seizure for epileptic patients. The framework even turns out to be increasingly muddled if the location framework is to be intended for omnipresent activities, for the distinguishing proof of individuals with seizure inabilities. For this situation, the patients are not limited to the clinical condition in which numerous gadgets are included to the patient remotely while he/she can proceed with every day exercises. This paper shows an order strategy by utilizing Fuzzy Logic System to distinguish, foresee the Partial Seizure from Epileptic information. Here the paper indicates starter consequences of the ordinary state, pre-seizure state and seizure condition of the subject's cerebrum flag information. This can be watched and the calculation with the recognition structure can create forewarning signals for epileptic seizure.

The crude EEG information is taken from the information storehouse of every one of the patient haphazardly; along these lines we have every EEG informational collection from the 24 patients. Here for every one of the patients, the main divert has been chosen in the underlying stage with the information estimation of 10 seconds each. This is done so as to diminish the heap on the implanted processor later on. (Shakir, Malik, Kamel, & Qidwai)

Comparative Analysis of Forecasting Neural Networks in the Application for Epilepsy Detection *by Svetlana Bezobrazova, Vladimir Golovko,2007*

Numerous procedures were utilized so as to distinguish and to anticipate epileptic seizures based on EEG. One of these methodologies for the forecast of the epileptic seizures is the utilization the disorder hypothesis, in particular assurance biggest Lyapunov's example or connection measurement of the scalp EEG signals. This paper exhibits the neural system strategy for epilepsy location. It depends on registering of the biggest Lyapunov's type. This paper additionally portrays investigation of exploratory outcomes where we connected diverse determining neural systems for figuring the biggest Lyapunov 's example.

Location system of epileptic form movement dependent on disarray hypothesis and determining ANN is connected. These genuine EEG signals from medical clinic utilized as starting information for research. They tried diverse EEG signals with epileptic exercises, with moderate wave exercises described not just epilepsy, and furthermore with typical exercises. Thus we have 96.7 % identification inconsistency exercises utilizing MLP arrange. Our investigations demonstrate that Elman's repetitive system is ineffectual at the choice of the given issue. There are no bogus recognitions in EEG signals portrayed just typical movement. Thusly our methodology for epilepsy location is material. (Svetlana Bezobrazova)

Detection of epilepsy disease from EEG signals with artificial neural networks ***by*** *Cansu Ozkan, Seda Dogan, Tgce Kantar, Mehmet Feyzi Aksahin and Aykut Erdamar, 2016*

The analysis of the epilepsy sicknesses are made by doctors with breaking down the electroencephalography (EEG) records. The epilepsy maladies can be resolved with watching the fundamental properties of previously and on-time seizure motions in time and recurrence area. Doctors are assessing the outcomes after some vital scoring on EEG records. Nonetheless, this assessment is specialist, tedious procedures and furthermore may emotional outcomes. Now, to permit identification of epilepsy maladies, a choice emotionally supportive network can give increasingly target results to the doctors for diagnosing. The subject of the investigation is naturally diagnosing the epilepsy maladies on EEG signals. In the proposed examination, investigations of EEG motions in time and recurrence space were done and highlights of maladies were acquired. Subsequently, utilizing fake neural system (ANN) and acquired highlights, a choice emotionally supportive network is acknowledged to analyze the epilepsy. The explicitness and the affectability of the calculation are 94% and 66% separately. (Cansu Ozkan, 2016)

Comparison of Machine Learning Methods for Detection of Epilepsy *by Meenakshi Sood, Vinod Kumar and S. V. Bhooshan, 2014*

Meenakshi Sood and Sunil V Bhooshan incurred that EEG captures the blueprint of the functionality of the brain and represents the physiology of brain. The set of features that was chosen is quite simple but robust for the study of EEG data needed for the required classification problem. It was observed that the accuracy of the best classification was obtained from the six neurons in the hidden layer. The testing and validation accuracy came out to be 100% and the training accuracy was 99.5%. The network hence developed performed efficiently better and a small number of iterations were required to train the neural network. Hence, the performance of the network was evaluated with respect to training, testing and classification efficiencies with different FL.

Also according to the research paper on Comparison of methods of machine learning developed for the detection of epileptic seizures, the classification of the patients includes preprocessing , extraction of various features, selection of topology also the and ANN techniques used for various classifications and comparing the different techniques for the same. The network was designed such that it classifies the patients on the basis of orders of epilepsy using EEG signals and hence an effective model for the respective workflow was maintained. It was finally concluded from the model developed that the motive of validating the predictive model along with a good performance factor, the dataset was divided accordingly in which 70% of the data was used for training the required network, 15% was used for the validation purpose and the remaining 15% was used for testing to assess and analyze the predictive performance of the developed model. Around 20 networks were trained for every sequence in the sets related to training and testing and finally the average was calculated to attain the performance parameters using the five best networks.

Finally, 99.6% of the efficiency came out to be the highest efficiency for the overall classification using MLPNN i.e. Multilayer Perceptron Neural Network and 96.8% was the accuracy acquired by using RBF as the method for machine learning. From the graph attained it was incurred that the accuracy attained for training came out to be better in case of MLPNN but once training was done, the testing accuracy came out to be the same for both the networks although there was a very less variation in the efficiency related to validation hence depicting that MLPNN evolves out to be the best predictive model for the required experiment. Also, from the architecture of the network it was proposed that the number of hidden nodes is not same for the respective methods.

Performance analysis done for respective architectures of the networks involved hidden layers in which the numbers of nodes were varied. Different architectures of the networks were trained varying likely nodes from five to twenty five in the various hidden layers and hence, the respective performance factor was evaluated accordingly.

The discrimination ability was used in the results obtained from all the features selected from all three hundred signals. The next experiment investigated the ability to discriminate the various features sets. Finally, it was concluded from all the experiments done so far that the comparison for the same were performed quite successfully and a comparison was done between the respective methods for Machine learning. Both the ANNs that were implemented depicted the effectiveness and efficiency of these methods proposed. (Meenakshi Sood, 2014)

Approximate Entropy-Based Epileptic EEG Detection Using Artificial Neural Networks *by Vairavan Srinivasan, Member, IEEE, Chikkannan Eswaran, Senior Member, IEEE, and Natarajan Sriraam, Member, IEEE, 2007*

The electroencephalogram (EEG) signal assumes a significant job in the finding of epilepsy. The EEG chronicles of the walking recording frameworks create protracted information and the location of the epileptic action requires a tedious examination of the whole length of the EEG information by a specialist. The customary techniques for examination being dreary, many computerized symptomatic frameworks for epilepsy have developed as of late. This paper proposes a neural-arrange based computerized epileptic EEG recognition framework that utilizes inexact entropy (ApEn) as the information highlight. ApEn is a factual parameter that estimates the consistency of the present adequacy estimations of a physiological flag dependent on its past sufficiency esteems. It is realized that the estimation of the ApEn drops strongly amid an epileptic seizure and this reality is utilized in the proposed framework. Two unique kinds of neural systems, specifically, Elman and probabilistic neural systems, are considered in this paper. ApEn is utilized without precedent for the proposed framework for the location of epilepsy utilizing neural systems. It is demonstrated that the general exactness esteems as high as 100% can be accomplished by utilizing the proposed framework. Neurons are called as the building blocks of NN. These are also known as units and sometimes, nodes also. An input is fed to each neuron from various other neurons, then based on the current input, there occurs a change in the internal state which is known as activation. Finally, an output signal is sent to the other neurons and hence the final output is generated.

An ANN exhibits functioning similar to that of a human brain and is a hardware implementation or computer program simulating the similar working to a human brain. Basically, ANN is a technique designed for solving various problems which is achieved by construction of a human brain like working software. (Vairavan Srinivasan, 2007)

A survey on soft computing techniques in epileptic seizure detection *by P. Grace Kanmani Prince , Rani Hemamalini, 2010*

This paper gives a review of various soft computing methods utilized in epileptic seizure recognition from the patient's EEG signal. The soft computing algorithms talked about here are Neural Network, fuzzy logic and probabilistic thinking. The strategy, application and exactness of every technique are investigated. The nerve cells are utilized to impart electrical impulse signals to various pieces of the cerebrum and the other way around. At specific conditions the mind begins terminating the signs up to multiple times more prominent than that of the typical speed. This causes an electrical flood in the mind. This condition is called as a seizure. Whenever rehashed seizure happens then it is called as epileptic seizure. Around 50 million individuals on the planet are influenced because of epilepsy. A portion of the significant foundations for epilepsy are head wounds, cerebrum tumors, lead harming, mal advancement of the mind, hereditary and irresistible sicknesses. In the greater part of the patients the reason of event of epilepsy isn't known. The best indicative instrument for epileptic seizure identification is Electroencephalogram (EEG). At the point when an epileptic seizure happens there is an outstanding change in EEG signals. (P. Grace Kanmani Prince, 2010)

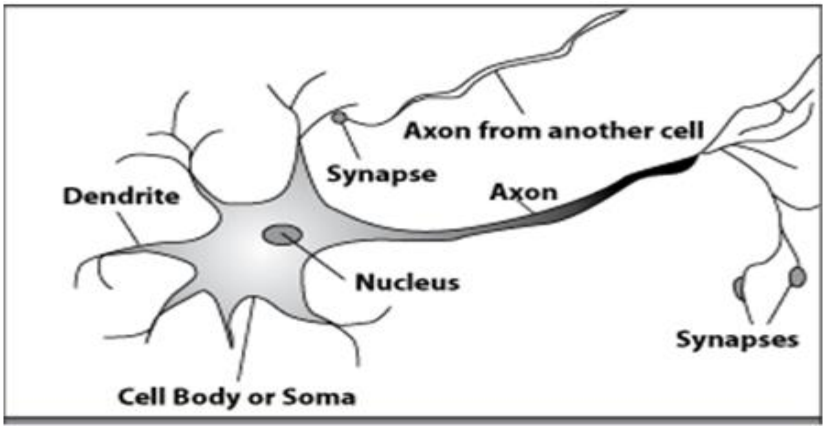

Fig. 7: human brain's structure

Here, the dendrites are used as the input, the cell body acts as the processor, the links to other neurons are creates using the synapse and the final output is generated at the axon. Generally, around 10,000 synapses are used to connect a single neuron to several other neurons. When inputs are received by a neuron from various other neurons, all the inputs are then combined. Once the critical level is exceeded by the inputs, a spike is discharged by the neuron. This spike is basically an electrical pulse which then travels through the entire body to the axon and hence finally, to the neuron(s) connected next to this neuron. The cell body or the dendrites of the next neurons are touched by the axon endings of the previous neuron. Neurotransmitters affect the transmissions of one electrical signal or spike from one neuron to the other one. These neurotransmitters are basically the chemicals that get released from the first neuron and are responsible for binding of the first and the next neuron. This link thus created is called a Synapse. The amount of neurotransmitter available is used to determine the strength of the signal that reaches the succeeding neuron.

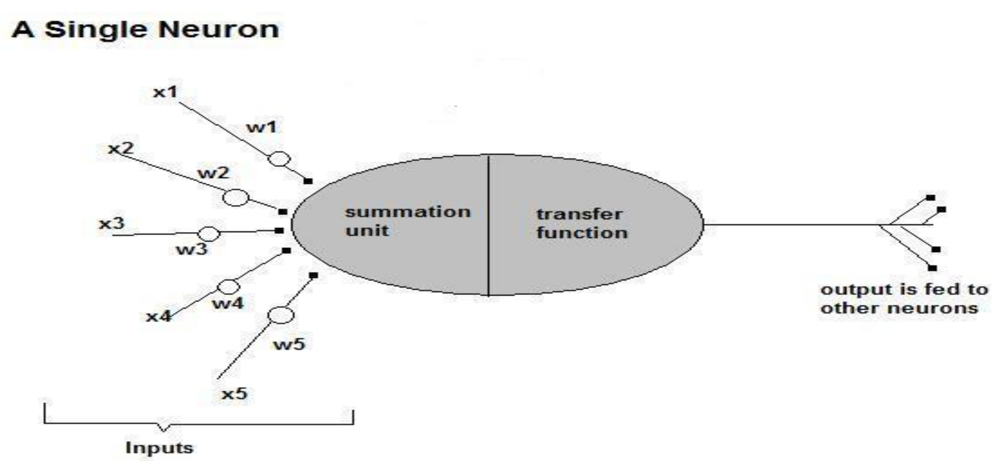

Fig. 8: structure of a single neuron

Therefore, ANN basically comprises of a pool of simple processing units which then communicate with each other by sending or transmitting several signals to each other. These signals are transmitted over a large number of weighted network connections.

Deep learning models can be imagined as a collection of points which are meant to make decisions on the basis of inputs given to the nodes. This kind of network is as same as a human biological nervous system where every node acts like a neuron. These algorithms are a class of ANNs and learn at a rapid rate with improved performance every time.

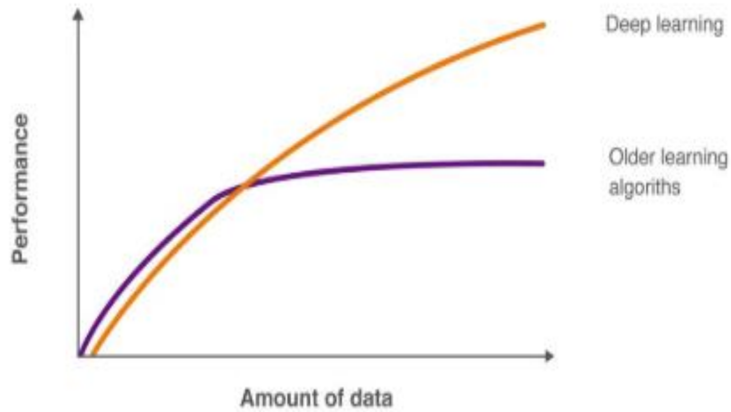

Fig. 9: Comparison of various algorithms and DL

This hunt of important data sources helped to figure out what is as of now thought about the point and how broadly the subject has just been explored till date. One of the extra advantages got from doing the writing audit is that it rapidly uncovered which specialists have composed the most on a specific point and are, in this manner, presumably the specialists on the theme. This survey revealed many new angles that could be further explored and studied. This made the understanding of the neural networks much easier and thus was helpful in acquiring a brief knowledge about the same. It was helpful to audit the sorts of concentrates that past research workers have propelled as a methods for figuring out what methodologies may be of most advantage in further building up the aim. Hence, this gave a rough idea for what needs to be developed and various methodologies that can be used to implement the aim of the project.

CHAPTER-3

SYSTEM DEVELOPMENT

Software:

Python version 3.7.0

Weights:

We know that an NN consists of a large number of neurons which are simple processing elements .These are connected to each other by direct communication links. These links are associated with weights. Therefore, weight is defined as value used by the NN for solving a given problem.

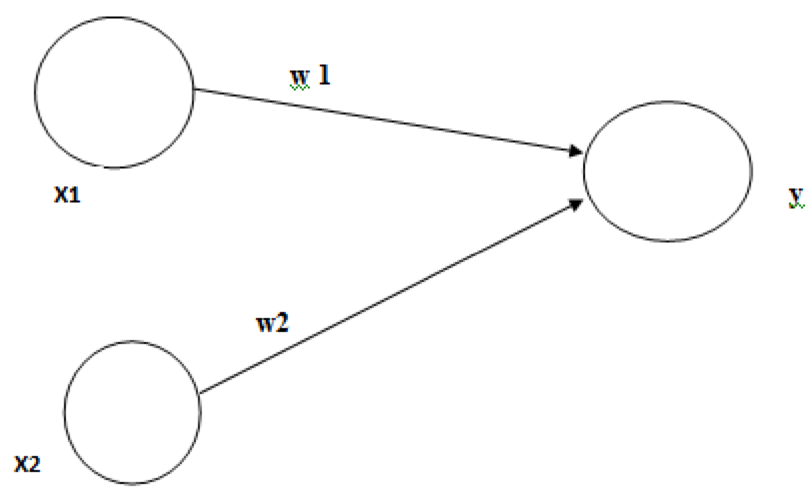

Fig. 10: weighted neuron

The given diagram shows a simple NN. The weights which carry information are denoted by w1 and w2 .w1 and w2 may be fixed or they can take random values. They can also be set to 0 or can be calculated by some methods. Initialization of weights is a cornerstone in NN. The changes in weights indicate the overall performance of the NN. Here,

x1 = Activation of first neuron

x2 = Activation of second neuron

These are the inputs.

y = resultant neuron

w1 = Weight connecting neuron 1 to output

w2 = Weight connecting neuron 2 to output

Net input is given as:

Net = x1w1 + x2w2

In general, it is written as,

$Net\ Input = Net = \sum xiwi$

The output is then calculated by using the activation functions.

Activation Functions:

The output response of a neuron is calculated by using the activation function. The response is obtained by applying an activation to the sum of weighted input signal. Same activation functions are used for neurons of same layer. These functions may be linear or non linear.

Identity Function:

This is given as,

F(x) = x; for all x.

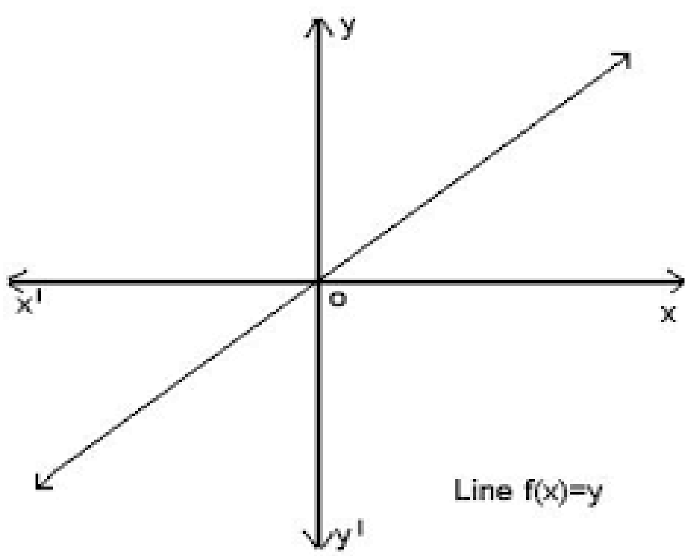

Fig. 11: Identity Function

Binary Step Function:

This is given as,

$f(x) = 1; if\ x \geq 0$

$f(x) = 0; if\ x < 0$

Binary step function is mostly used to calculate output from net input. This function is also known as **Heaviside function** or **Threshold function.**

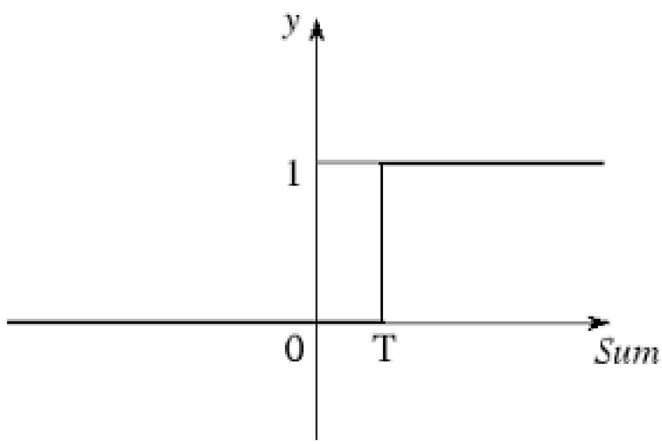

Fig. 12: Binary Step Function

Here T implies theta.

A binary function basically implies whether the neural network is firing or not.

It displays whether the result is 1 or 0.

Sigmoidal Functions:

These are generally S-shaped curves. Commonly used functions are hyperbolic and logistic functions. These are used in radial basis function networks and back propagation networks. These are of two major types:

Binary Sigmoidal Functions

Bipolar Sigmoidal Functions

Binary Sigmoidal Functions:

It is also known as logistic function. Its value lies between 0 and 1.

$$f(x) = \frac{1}{1 + \mathrm{e}^{-\sigma x}}$$

Where sigma is the parameter of steepness.

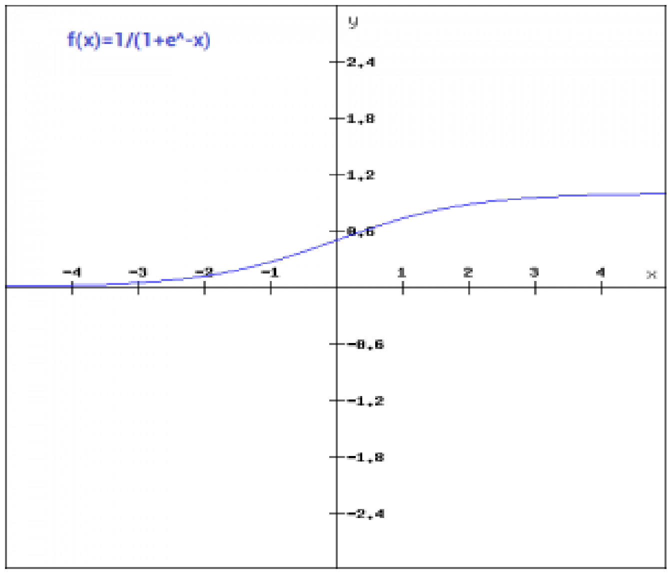

Fig. 13: Binary Sigmoidal Function

Bipolar Sigmoidal Functions:

The range lies between +1 and -1.This function is a kind of hyperbolic function which is tangent. It is given by the equation,

$$b(x) = 2f(x) - 1$$

Where f(x) is the binary sigmoid function.

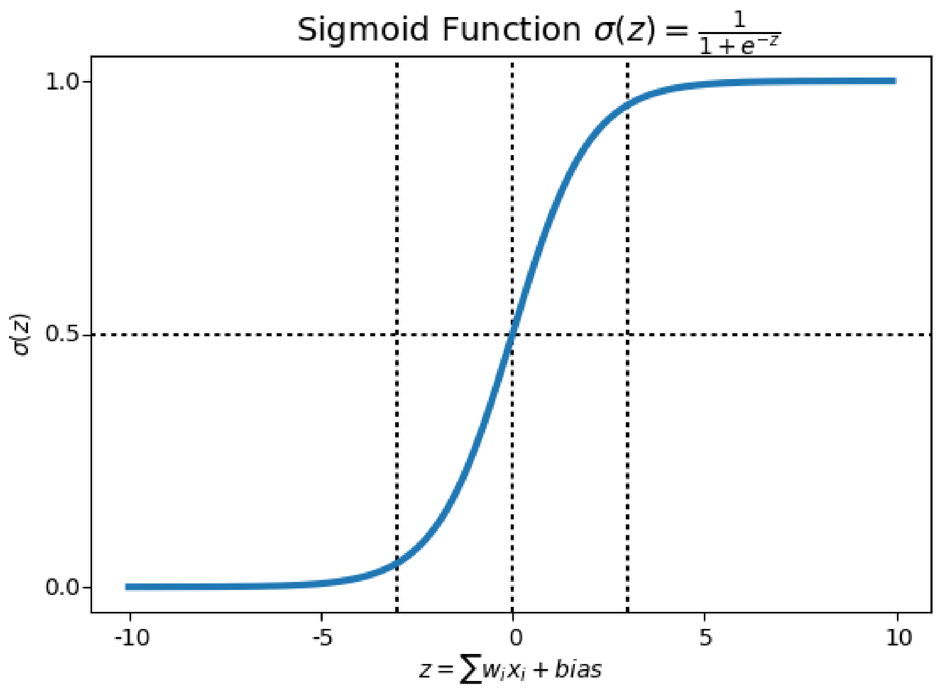

Fig. 14: Bipolar sigmoid function

ReLU Function: Rectified Linear Unit activation function is one of the most widely used activation functions. It is used in deep learning and in almost all NNs. The function is:

$f(x) = \max(0, x)$

This function improves NNs by hastening the training.

It is one of the most widely used activation functions. It is widely used in deep learning or convolutional NN.

ReLU activation function is a half rectified function from the bottom.

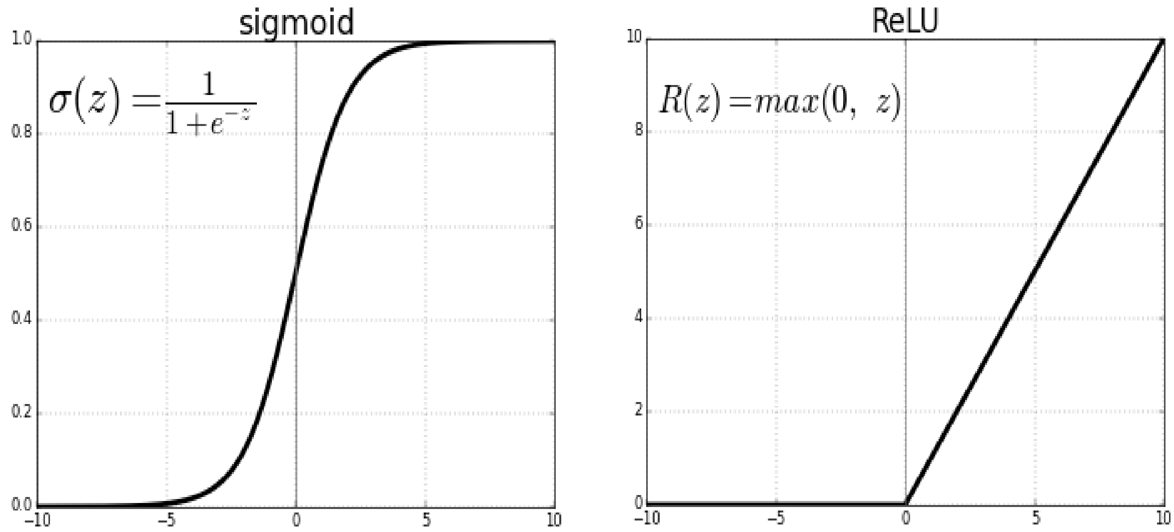

Fig. 15: ReLU function graph

Gradient Descent:

It is an optimization algorithm. Gradient descent is an order one iterative algorithm used to find the minima of a function. Steps corresponding to the negative of a gradient are taken to minimize or find the local minima for a function using gradient descent of the current point function. Whereas if the steps are taken corresponding to the positive descent, the value of the function increases and a local maximum can be obtained. This is called gradient ascent.

Gradient descent is also called steepest descent.

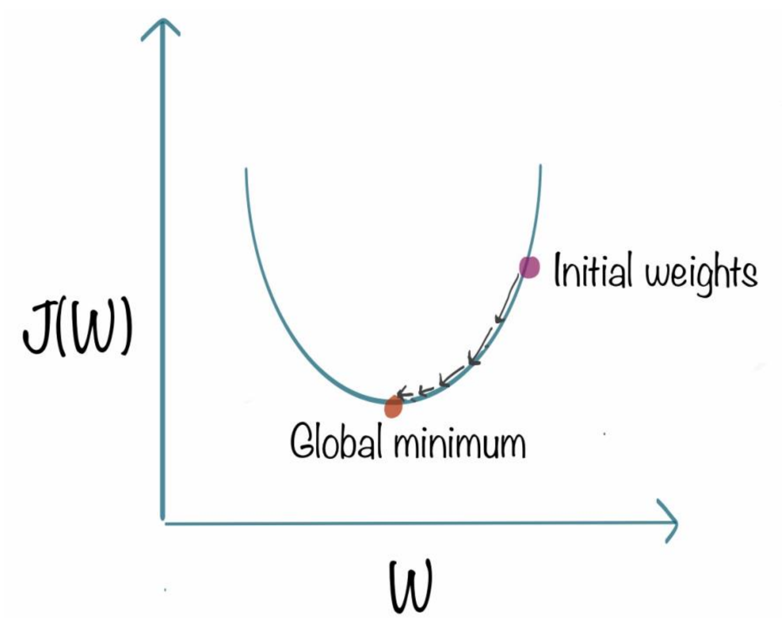

Fig. 16: Gradient Descent

When the parameters cannot be determine analytically, gradient descent is the best option to be used and can be used as an optimization algorithm. The main is aim is to continuously trying various coefficient values, estimate their cost and hence select or predict certain new values for these coefficients such that the final output returns a minimum cost result.

For non-differentiable capacities, slope strategies are not well characterized. These strategies are commonly slower than angle plummet. Another option for non-differentiable capacities is to smooth the capacity, or bound the capacity by a smooth capacity. In this methodology, the smooth issue is illuminated with the expectation that the appropriate response is near the response for the non-smooth issue (every so often, this can be made thorough).

Learning Rate: When we train neural systems we for the most part use Gradient Descent to streamline the weights or loads. In every iterations back-engendering is used to figure the derivative for the loss function concerning each and every weight and then subtract it from that particular weight. Learning rate decides how rapidly or how gradually you need to refresh your weight (parameter) values. Learning rate ought to be sufficiently high with the goal that this won't take ages to unite or converge, and it ought to be low so that it discovers the nearby minima.

Procedure to calculate the Gradient Descent:

The methodology begins off with introductory qualities for the coefficient or coefficients for the capacity. These could be 0.0 or a little irregular esteem.

The value of coefficient is 0.

The coefficient cost is calculated by fitting these in the function and hence determining the cost.

Thus, the value of cost becomes = f (coefficient)

Or

$cost = evaluate\big(f(coefficient)\big)$

The cost derivative is then estimated. The derivative is a part of mathematical calculus and signifies to the slope of the given function at a given point. The slope is required to be known in order to know the direction or sign which reflects in which the movement of the coefficients values thus generating a low value cost on further iterations.

$delta = derivative(cost)$

Now from this, the direction of the derivative is well indicated and hence the coefficient values can now be updated. Learning rate parameter (alpha) is needed to depict the control how much we can update the values of the coefficients.

$$coefficient = coefficient - (\propto * \Delta)$$

This process is to be repeated until and unless the coefficients cost value (cost) reaches to 0.0 or becomes approximately zero for efficient and accurate results.

Hence, the gradient descent technique for optimization is quite simple and efficient.

Batch Gradient Descent

The objective of all supervised learning algorithms is basically to best predict a target function which maps the input data onto the output variables. This portrays all characterization and regression issues.

Certain learning algorithms include coefficients which characterize the algorithms which estimate for the respective target function. Diverse calculations have distinctive portrayals and diverse coefficients, however huge numbers of them require a procedure of streamlining to locate the arrangement of coefficients that outcome in the best gauge of the objective capacity.

Normal instances of calculations with coefficients that can be streamlined utilizing angle plunge are Linear Regression and Logistic Regression.

The cost function includes assessing the coefficients in the AI model by ascertaining an expectation for the model for each training instance (occurrence) in the dataset and contrasting the predictions with the actual output values and computing an aggregate or normal error.

From the cost function a derivative can be determined for every coefficient with the goal that it very well may be refreshed utilizing precisely the update condition portrayed previously.

The cost is determined for an AI calculation over the whole training dataset for every iteration of the gradient descent algorithm. Single iteration of the calculation is called one batch and this type of gradient descent is alluded to as a batch gradient descent.

Batch gradient descent is the most widely recognized type of gradient descent depicted in AI.

Stochastic Gradient Descent

Gradient descent can be moderate to keep running on vast datasets.

Since one iteration of the Gradient descent the calculation requires an expectation for each occasion in the training dataset, it can take quite a while when you have a huge number of cases.

In circumstances when you have a lot of information, you can utilize a variety of gradient descent called stochastic gradient descent.

In this variety, the gradient descent technique depicted above is run yet the update to the coefficients is performed for each training case, as opposed to toward the finish of the group of instances.

Net Input Calculation using Matrix Multiplication Method

Provided the weights are given in the form of matrix, the net input for the output is calculated by the dot product of the input vectors.

$x\ (=x_{1,\ldots\ldots,}x_{n})$ and w (= jth column of the weight vector matrix).

$$Y_{inj} = x_i w_j$$

$$Y_{inj} = \sum x_i w_j$$

Therefore, we can use matrix multiplication method to calculate the net input.

Bias:

It represents a weight on a link having a unit value for activation. On increasing the value of bias the net input to the unit also increases.

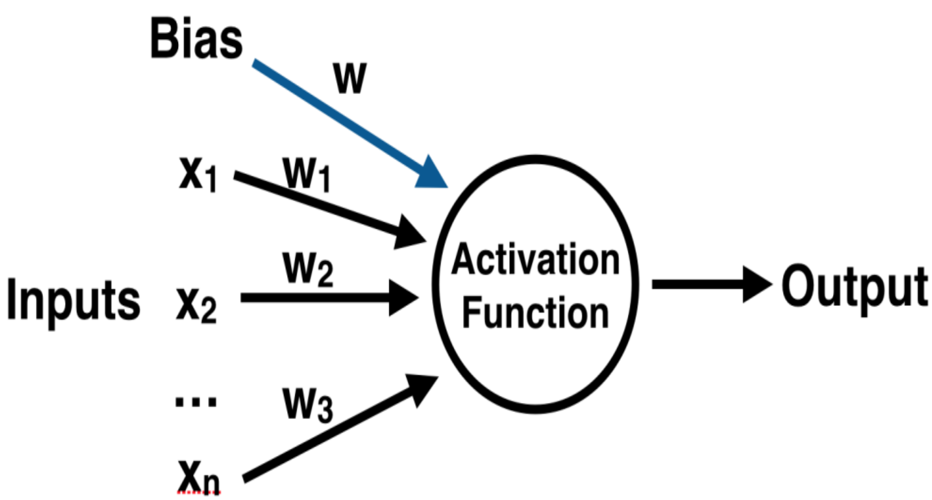

Fig. 17: A simple neural network with Bias included

Performance of the NN is improved by bias. Bias should also be initialized either to zero or any specific value just as weights are initialized.

With bias net input is calculated as,

$$Net = b + \sum_{i=1}^{n} xiwi$$

Here,

Net – net input

b – Bias

x_i – Input from neuron i

wi – Weight of neuron i to the output neuron

Therefore, the activation function is obtained as,

$$f(net) = +1; if\ net \geq 0$$

$$f(net) = -1; if\ net < 0$$

Threshold: It is used to calculate the activations of the NN. The output may be calculated based on the value of threshold. Activation function is based on the value of this threshold only.

Threshold is denoted by θ that is theta. System responses are calculated due to thresholds.

Process Identification and Control:

It is an important generic application domain of NNs. The process which is to be controlled needs to be understood before the problem and issues of control need to be tackled. Hence, we need to first address the identification of the process.

Fault Diagnosis:

It consists majorly of three phases which are:

- Detection
- Identification
- Isolation

Fault Detection:

In this phase, fault appearance is identified. And the moment of detection is then identified.

Fault Isolation:

Kind of appearance of the fault, its place and time are determined in this phase. Detection of fault is followed by this phase.

Fault Identification:

In this phase, fault size and the variability of this fault with time are determined. Isolation of the fault is followed by this phase.

Various concepts such as monitoring, Supervision and Protection are considered as the phases of fault diagnosis.

Epilepsy:

The disorder of epilepsy is chronic and it is characterized by unprovoked seizures. These seizures may be the result of some brain injury or some family tendency. The word itself does not give any idea about the severity or magnitude of the seizures. It is here that the concept of neuro fuzzy comes into action and we are able to detect the epilepsy with a large accuracy value.

Seizures may be of more than one type and not only this people who are suffering from epilepsy may show some other symptoms of brain disorders. In such situations, we can define their condition to be a particular epilepsy syndrome.

There are certain electrical events which occur in the brain. These events are responsible for the symptoms of epilepsy. So basically epilepsy is a central nervous system disorder. Symptoms of epilepsy may include:

Stiffness in arms or hands

Temporary confusion

Sudden fear

a rising feeling in the stomach

Seizures are roughly classified into various types such as :

Tonic- clonic seizures

Complex partial Seizures

Simple partial Seizures

These have a range of severity which varies from individual to individual. Anti-seizure medications are used to cure epilepsy.

High fever, nightmares, sleep disorders and panic attacks are not the symptoms of epilepsy.

The development of the required model involves the following steps and algorithms:

1. Initializing the network
2. Forward propagation
3. Back propagation
4. Training the network
5. Predicting the output
6. Predicting the accuracy and efficiency

1. Initializing the network

It involves the creation of the network for the training purpose. In this, each neuron is assigned a weight and these weights are adjusted. We have a single weight corresponding to every input link. Additional weights are assigned to bias. Additional properties for a neuron need to be stored during the training, hence, a dictionary is used for representing each neuron and store the respective properties by several names such as 'weights' for representing the weights.

The initial step of the learning is to begin from some place: the underlying speculation. Like in hereditary calculations and advancement hypothesis, neural networks can begin from anyplace. Along these lines an arbitrary initialisation of the model is a typical practice. The objective behind is that from wherever we begin, on the off chance that we are perseverant enough and through an iterative learning process, we can achieve the pseudo-perfect model.

A network is divided into layers. The hidden layer is represented by the first layer and is succeeded by output layer.

The network weights are initialized weights to small random numbers. The random numbers are chosen in the range from 0 to 1.

A function called as initializenetwork() is used to create a new NN for training. It basically takes 3 inputs: no. of inputs, no. of neurons present in the hidden layer and the no. of outputs.

2. Forward propagation:

The output is calculated by sending the input signal via each layer until the output layer displays the output. This is called Forward propagation.

This technique is used to generate predictions during the training and needs to be corrected.

It takes place in three parts:

- Activation of neuron

The very first step is to depict the activation of a neuron for the given input. The input can be a training dataset row for a hidden layer or for an output layer, output from every single neuron of hidden layer.

- Transfer of neuron

After the activation of the neuron, the activation of the neuron needs to be transferred in order to check the output.
For this purpose we can use various transfer functions such as sigmoid activation function etc.

The sigmoid function is also called a logistic function as it looks like an S shape.
It takes any input value and displays an output value between 0 and 1 on the S- curve.
From this function, the derivative can also be calculated easily which is used to detect the back propagation error.
The sigmoid function is calculated as:

$$output = \frac{1}{1 + e^{-a}}$$

Here a implies activation.
Where e refers to Euler's number

In forward propagation of an input, the output for each neuron is calculated in each layer. Every output from a layer becomes the inputs of the neurons for the next layer.
The input gives the underlying data that at that point engenders to the concealed units at each layer lastly produce the yield or output.
Forward propagation involves the inputs to be traversed along the neural network and hence generating the output at the output layers by performing various calculations using the weights and the transmission or activation functions.

The system architecture involves deciding the width, depth, and activation functions utilized on each layer. Depth represents the number of hidden layers. Width represents the number of units (nodes) on each hidden layer since we control neither input layer nor output layer measurements. There are many arrangements of activation functions such as Rectified Linear Unit, Sigmoid, Hyperbolic tangent, and so forth. Research has demonstrated that more profound systems or deeper networks outflank systems with progressively hidden units. Hence, it's in every case better and won't damage to prepare a deeper network.

Following is given the structure of a FEED FORWARD NEURO FUZZY SYSTEM:

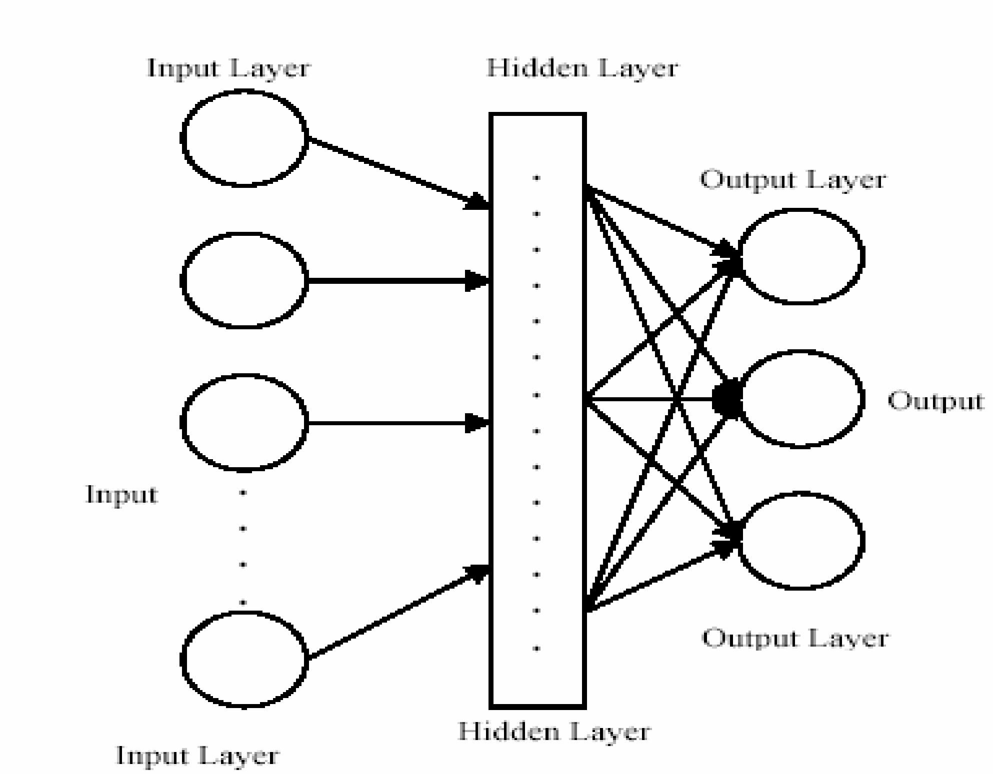

Fig. 18: Feed forward neural network

(i) Calculating the Loss Function:

Now, the actual for the randomly initialized NN is known to us at this stage. And on the contrary, we also have the desired output for the NN that is desired to be learnt by the NN. Loss Function is used to generalize any sort of problem. Essentially it is a performance metric on how well the NN figures out how to achieve its objective of producing outputs as close as conceivable to the desired values.

The lost function is basically given as:

$$Loss = desired\ output - actual\ output$$

When the network undershoots, a positive value is returned by this network i.e. when predicted value < desired value and this network returns a negative when there is overshooting in the network i.e. when predicted value > desired value. For the loss function to depict an absolute error value regarding the performance no matter if the loss function undershoots or overshoots, the loss function is given as follows:

$$Loss = absolute\ value(desired\ value - actual\ value)$$

(ii) Differentiation Calculation

Clearly we can utilize optimization technique that alters the inner loads called internal weights of neural systems so as to limit the all the loss function that we recently characterized. These systems can incorporate genetic algorithms or brute force techniques or even a greedy search.

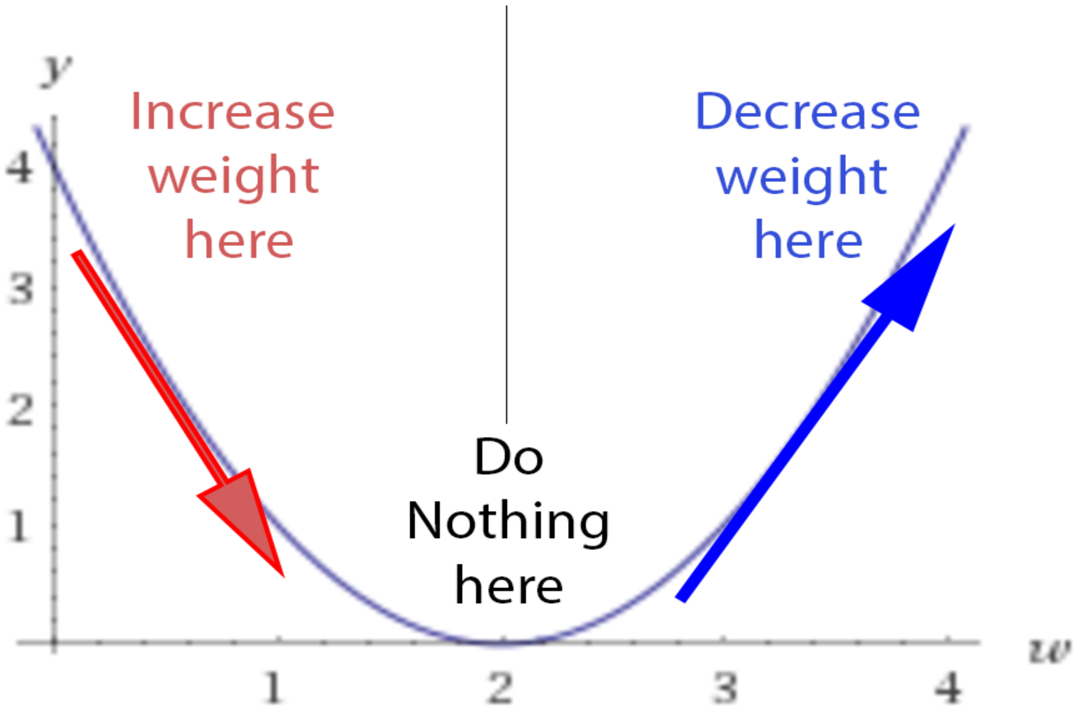

Fig 19: Differentiation Calculation

On the off chance that we initialize arbitrarily the system, we are putting any irregular point on this curve. Thus, according to the learning procedure:

- First the derivative is checked
- If the derivative comes out to be positive, which means that on increasing the weights, the error will increase hence we need to decrease the weights
- If the derivative returns out to be negative, meaning that on increasing the weights, the error will decrease hence we need to increase the weights
- When the derivative becomes zero, nothing is to be done as this is the stable point and hence the outcome lies here at this point.

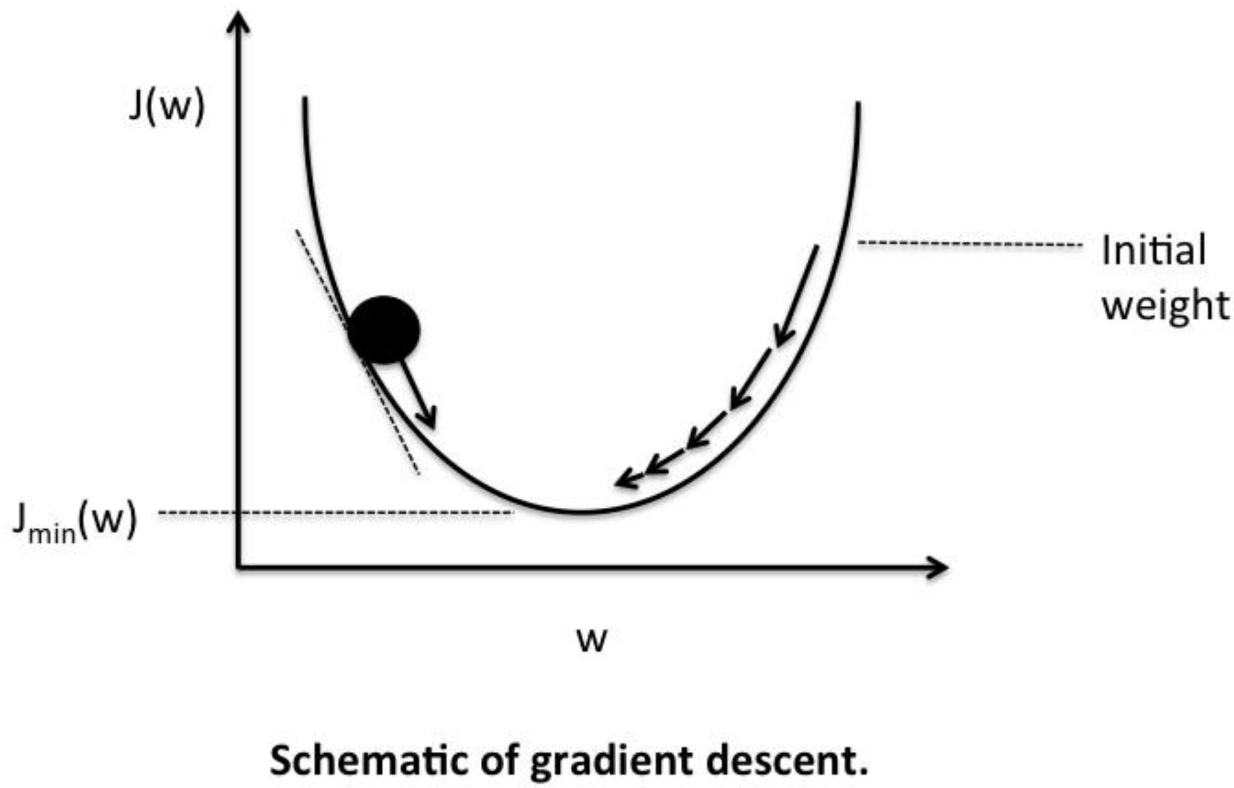

Fig. 20: Gradient Descent

3**. Back- propagation**

It is a supervised learning method from the field of Artificial Neural Networks used for multilayer fee- forward networks. Feed- forward neural networks work on processing of information for at least one neural cell, known as neuron. An input signal to a neuron is given through the dendrites, through this passes the signals to the body of cell. This signal is carried out for synapses by the axon, where synapses refer to the inter-connections of a particular cell's axon to dendrites of another cell.

The principle on which Back propagation works is that it models a given function by modifying the internal weights taken on the input signals in order to produce the required output signal.

A supervised learning model is used to train the system in which the internal state is modified by presenting an error between the output depicted and the actual output expected.

Back propagation requires a network architecture having at least one layer such that each layer is connected fully to the next one. A network structure consists of three layers basically which are input, hidden and output layer.

The objective of back propagation is to figure the partial derivatives ∂C/∂w and ∂C/∂b of the cost function C as for any weight w or bias (b) in the system. For back propagation to work we have to make two principle suppositions about the type of the cost function. Before expressing those suspicions, however, it's valuable to have a precedent cost function as a primary concern.

Back propagation algorithm can be used for many purposes such as classification as well as regression.

In case of classification, when there is a neuron for each class value in the output layer, the results come out to be the best results.

For instance, consider a binary network for classification. A and B are the two class values. The expected output need to be changed to binary vectors having single column for every value of the classes. E.g. [1, 0] and [0, 1] for class values A & B.

This is known as One Hot Encoding.

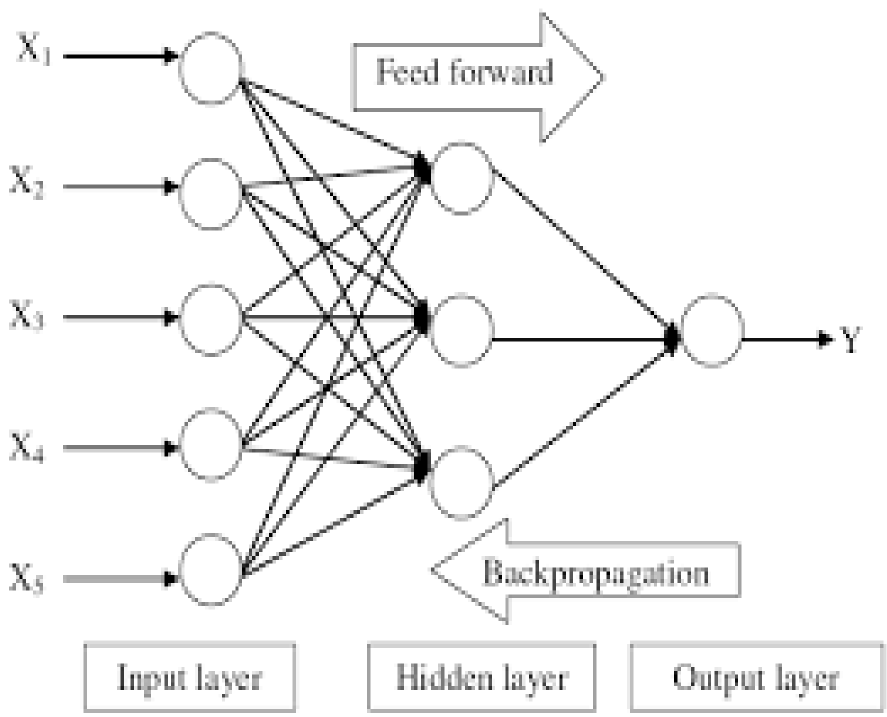

Fig. 21: Back propagation in a neural network

Back propagation Error:

The back propagation algorithm is named as per the ways as the training of the weights is done. Error is calculated as the difference between the expected output value and the output generated by propagating forward from the network. Once the errors are calculated, the errors are then back propagated via network to hidden layer from the output, adjusting the respective weights.

The back propagation error includes:

- Transfer derivative

In this, Slope is calculated from the value of the output of a neuron.

The derivative is calculated as:
$d = output(1.0 - output)$

- Error propagation

The error is calculated to back propagate the error input signal through the network.

The error is given as:

$e = (expected\ value - output\ value) * transferderivative(output\ value)$

Where the transferderivative function is used to calculate the slope of the output value of the neuron.

But in case of a hidden layer, the error is calculated using the weights for each neuron of the output layer. Hence, a weighted error is calculated.

The error signal in the hidden layer is given as:

$e = (w * e) * transferderivative(output\ value)$

e implies error and w implies weight

where weight refers to the weight of the neuron connected to the present (current) neuron

and error refers to the error signal of a neuron from the output layer.

Starting from the output and moving in the back direction, the network layers are iterated reversely.

The figure given below depicts the general working of a neural network which includes feed forward propagation and back propagation.

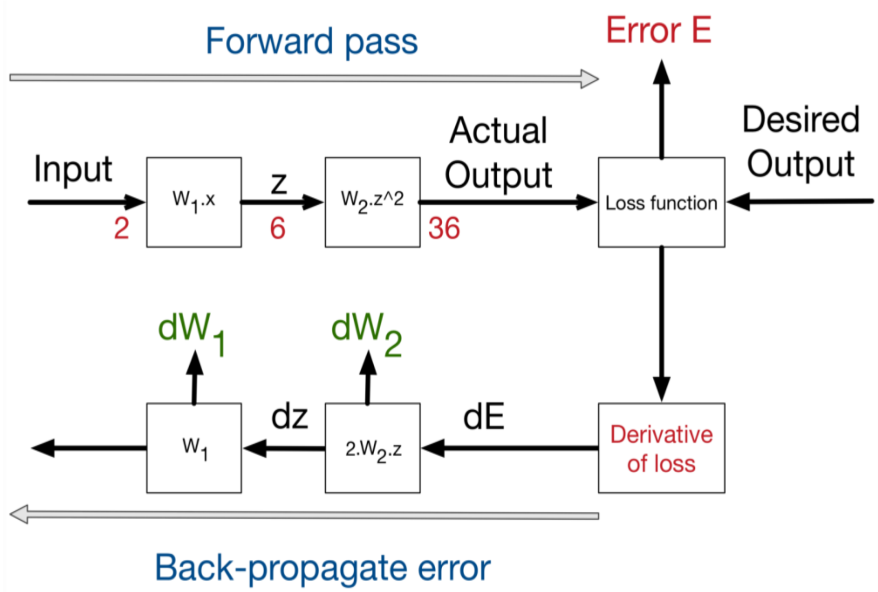

Fig. 22: general working of NN

(iii) Updating of weights

As we exhibited before, the derivative is only the rate of which the error changes generally to the weight changes.

Generally, the delta rule is used to update the weights.

$New\ wt = old\ wt - rate\ of\ derivative * learning\ rate$

The learning rate is presented as a constant (normally exceptionally little), so as to drive the weight to be updated in all respects easily and gradually (to maintain a strategic distance from huge advances and tumultuous conduct).

This equation can be validated as follows:

- If the derivative rate comes out to be positive, this means that increasing the weight will result in increase of the error, hence the updated weight needs to be smaller.
- If the derivative rate results in negative, thus meaning that increasing the weight will result in decreasing of the error, hence the new weight should be increased.
- If the derivative rate is zero, this means that a stable minimum is achieved and hence no updating or degradation of the weights is required. This means that a stable state has been reached.

This updating of weights requires a number of iterations to be done for the network designed to learn. In NN, the gradient descent, after each iteration, forcibly updates the weights towards minimum global loss function.

The delta rule is used as a kind of mutation operator whereas the loss function behaves like a fitness function used for minimization purpose. The number of iterations required is dependent on the strength of the learning rate applied. It also depends on several other factors such as the number of layers used, complexity of the non- linear functions, optimization methods used, and random initialization of the network and also on the training set quality.

3. Training the network

Scholastic gradient descent is used to train the network.

This includes various iterations of revealing a training dataset in/to the network and propagating forwards the input values for each data row, propagating the errors backwards and also updating of the network weights.

This is done in 2 parts:

4.1 update the weights
4.2 train the network

Updating the weights:

After the calculation of the errors for every neuron in the network using the method of back propagation, the weights can be updated.

The updating of the weights of the networks is done as:

$wt = wt + (\propto * \, e * input\ value)$

Where $\propto$, learningrate refers to a parameter which must be specified,

Error, e is the back propagation error calculated,

Input value is the input value due to which the error was caused.

The learning rate is used to control the change that must be applied to the weight for the correction of the error. Small rates of learning are more preferred as they cause slower learning as compared to the large no. of iterations used for training.

This is done because the likelihood of the network gets increased for finding or analyzing an efficient set of weights over all the layers instead of the fastest weights set that cause the error to minimize.

Training the network:

The updating of the network involves a fixed no. of the epochs to be looped and updating of the network in each row within the epoch in the dataset used for training.

Since, for every training pattern the updates are made hence this is called as Online Learning.

This is called as Batch Learning when before updating weights; the errors accumulate across an epoch.

The expected or estimated no. of the outputs is used to change the class values of training data to One Hot Encoding.

5. Predicting the output

It is quite easy to make predictions once a network is trained.

The output values estimate the probability of an arrangement belonging to every output class.

A crisp class prediction can be done by making selection of the class value having a larger probability, which is also known as arg max function.

6. Predicting the accuracy

Accuracy: It is defined as how close the measured value is to the known or standard value.

Once the output for the network structure is predicted, it is important to predict and analyze the accuracy of the network thus designed for reliability and getting more accurate outputs and results.

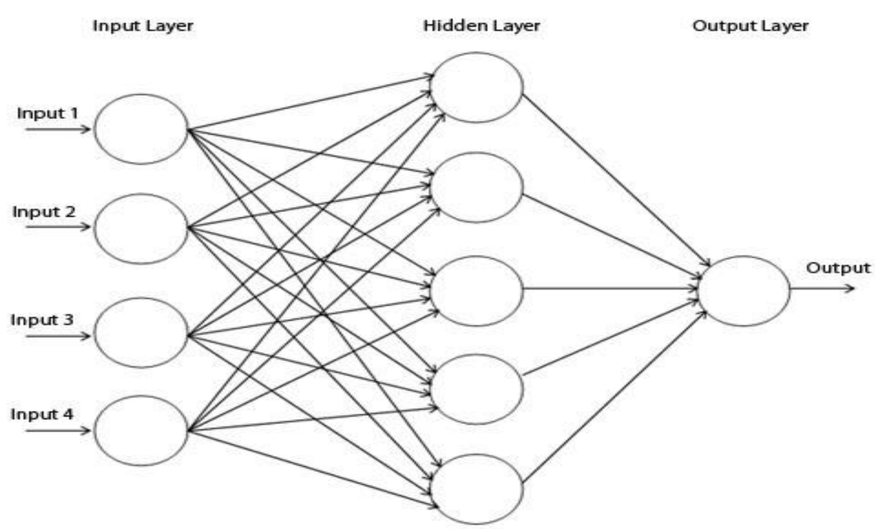

Fig. 23: Structure of a single- hidden layer

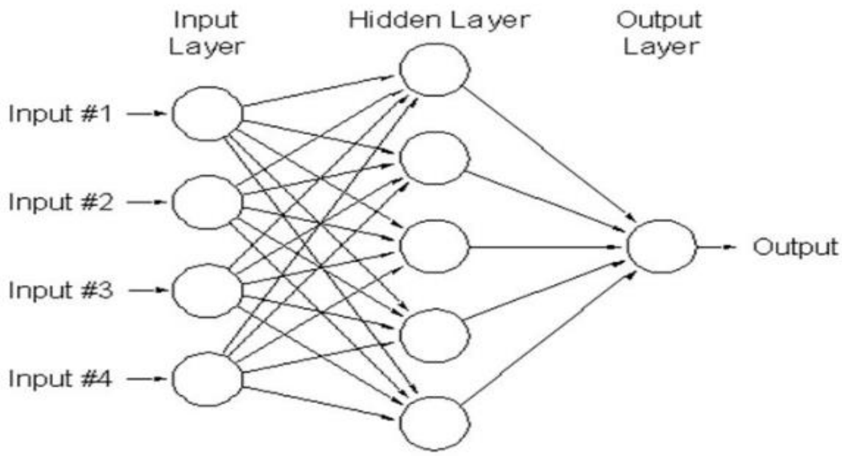

Fig. 24: Structure of a multilayer feed- forward neural network

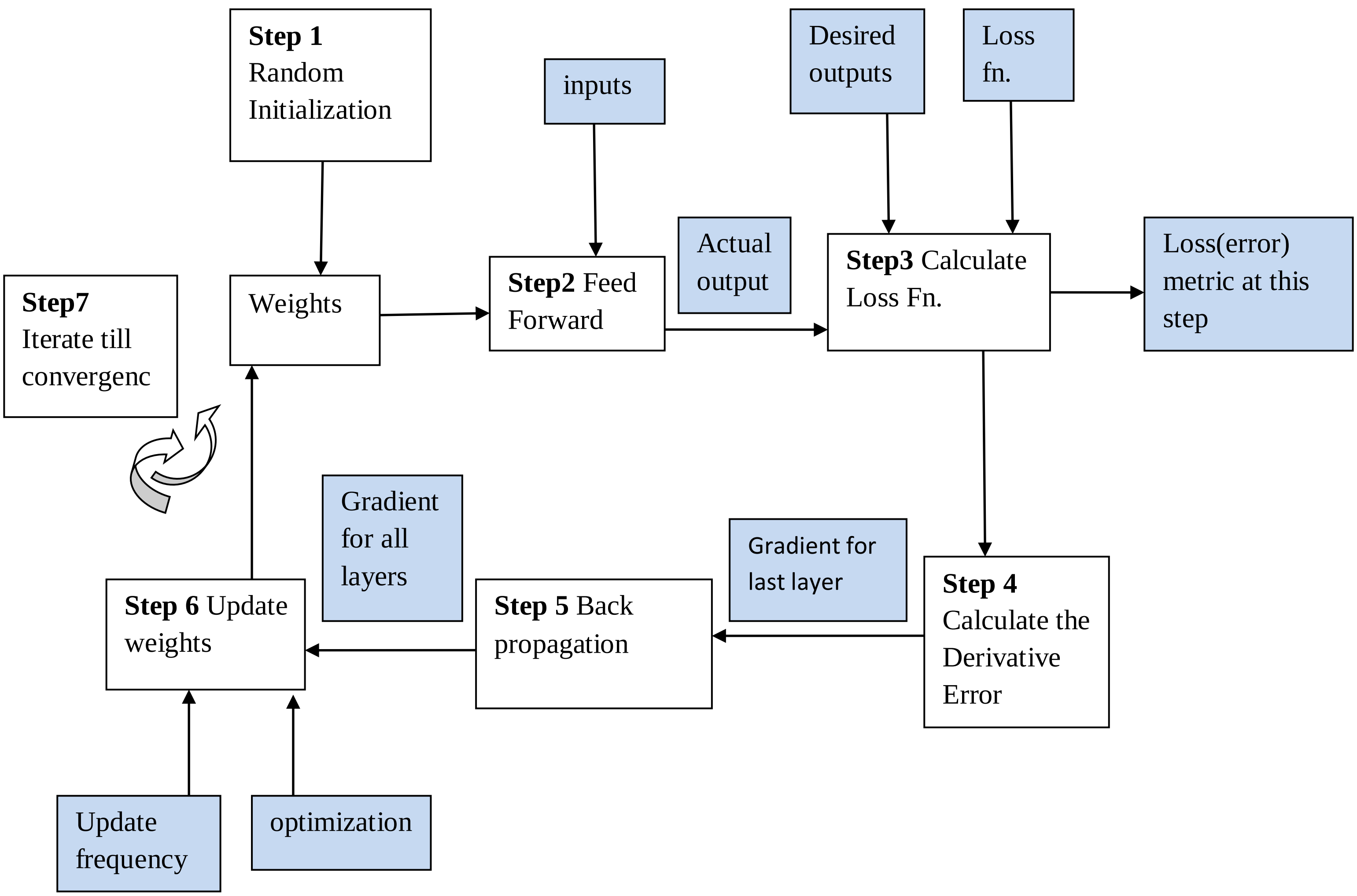

Fig. 25: Architecture of the proposed Model

This figure depicts the overall designing of the model step by step.

The test plan consisted of collecting the data set. Our data set has 178 different variables. The total number of entries in it was 11501. Among these 11501 entries, 11301 entries were chosen for training data and the rest 200 were testing data entries. Whenever we split the data into training and testing data sets, we mainly take a large portion of data set for training data and the rest of the small portion for testing data set.

To reduce the effects of data discrepancies, the data that is used for training and testing is similar. This is why we have divided or spitted the data from same data set only.

As the names themselves suggest, training data is the subset of data set which is used to train the model and testing data is that subset of our data set which is used to test our model.

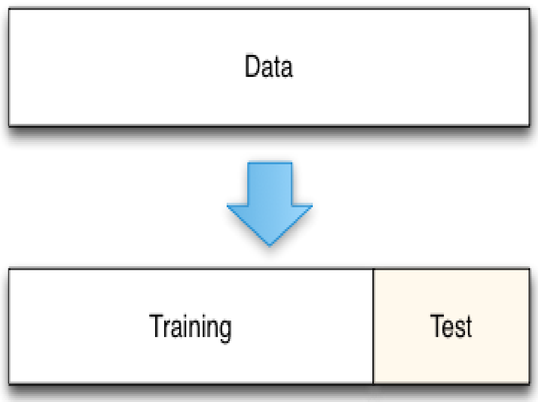

Fig. 26: Test Plan

The matrices used are categorized as:

Accurate

Reliable

Useful

Basically, metrics used are the sets of measurements which are used to determine the efficiency of our model.

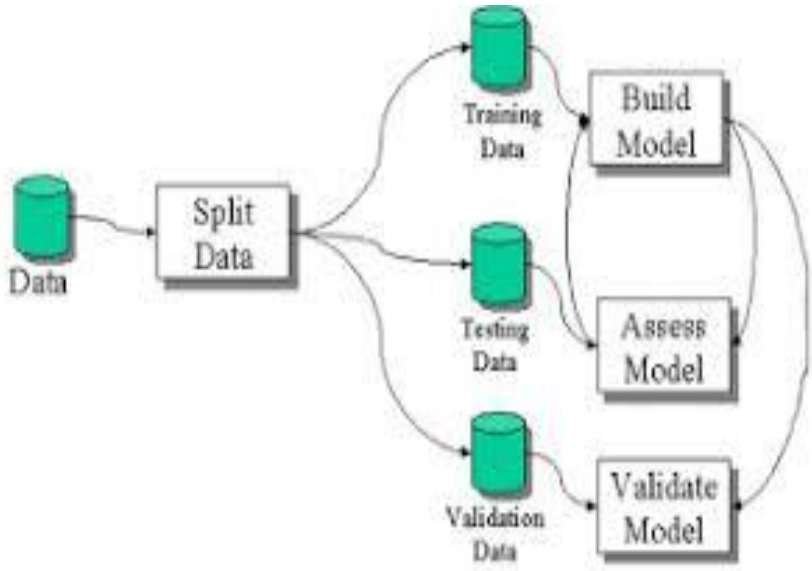

Fig. 27: Test Plan for model development

CHAPTER- 4

PERFORMANCE ANALYSIS

The data has been loaded and trained:

```
Total no. of rows: 11501

First 5 rows are:

['X21.V1.791', '135', '190', '229', '223', '192', '125', '55', '-9', '-33', '-38', '-10', '35', '64', '113', '152', '
```

Fig. 28: Loading of Data Set

```
[ 135.  190.  229.  223.  192.  125.   55.   -9.  -33.  -38.  -10.   35.
   64.  113.  152.  164.  127.   50.  -47. -121. -138. -125. -101.  -50.
   11.   39.   24.   48.   64.   46.   13.  -19.  -61.  -96. -130. -132.
 -116. -115.  -71.  -14.   25.   19.    6.    9.   21.   13.  -37.  -58.
  -33.    5.   47.   80.  101.   88.   73.   69.   41.  -13.  -31.  -61.
  -80.  -77.  -66.  -43.    5.   87.  129.  121.   88.   12.  -76. -150.
 -207. -186. -165. -148. -103.  -33.   40.   94.   75.    8.  -81. -155.
 -227. -262. -233. -218. -187. -126.  -65.  -12.   27.   61.   49.    9.
  -46. -124. -210. -281. -265. -181.  -89.   -4.   53.   53.   38.   43.
   31.   34.    9.   -7.  -34.  -70.  -84. -101.  -70.  -11.   42.   62.
   66.   74.   64.   59.   56.   36.  -11.  -30.  -43.  -23.    8.   42.
   77.  103.  135.  121.   79.   59.   43.   54.   90.  111.  107.   64.
   32.   18.  -25.  -69.  -65.  -44.  -33.  -57.  -88. -114. -130. -114.
  -83.  -53.  -79.  -72.  -85. -109.  -98.  -72.  -65.  -63.  -11.   10.
    8.  -17.  -15.  -31.  -77. -103. -127. -116.  -83.  -51.]
4.0
```

Fig. 29: Trained Data Set

After loading the data, the data is normalized and divided into components such as training data and testing data.

Once this is done, one hot encoding is done and then the architecture of the model is designed.

An important criterion is important to be predicted and designed for validating the performance and working of the proposed model.

In this part, to achieve the desired outputs from the model designed, the proposed methodology and techniques are applied on the database collected.

Out of the 11501 data set values, 11300 are used for the training purpose and rest are assigned as the testing data. Then once the architecture of the network is successfully developed and the predicted output is attained, the model is analyzed for its performance and accuracy.

The output data is divided into 5 different classes using one hot encoding method and the results obtained are:

One hot Encoding the output data into 5 classes

```
[[0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 ...
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]]
[0. 0. 0. 0. 1.]
```

Fig. 30: One hot Encoding

The 0s and 1s in the above output depict the presence and absence of the respective order of the epilepsy in the patient.
'1' means that the patient is having epilepsy of the respective order of the class and '0' refers to absence of epilepsy of that particular order.

6. Evaluating the model

Getting almost 90% accuracy on testing data

```
[0.2340556496950849, 0.8954773715992069]
Accuracy:  89%
```

Fig. 31: Accuracy of the proposed model

On evaluation, the accuracy of the ,model comes out to be approximately 90.

Predicting values for given input

```
56
[[0.35036668 0.35036668 0.35036668 0.35036668 0.64963335]]
5
[0. 0. 0. 0. 1.]
5
```

Fig. 32: Prediction of input values

Here, this outcome depicts that after training the neural model completely, for a given input the various probabilities of the seizure falling into various classes is depicted. The highest otcome probability is the final output for the given input depicting into which class the epileptic seizure of the particular patient has been diagnosed.

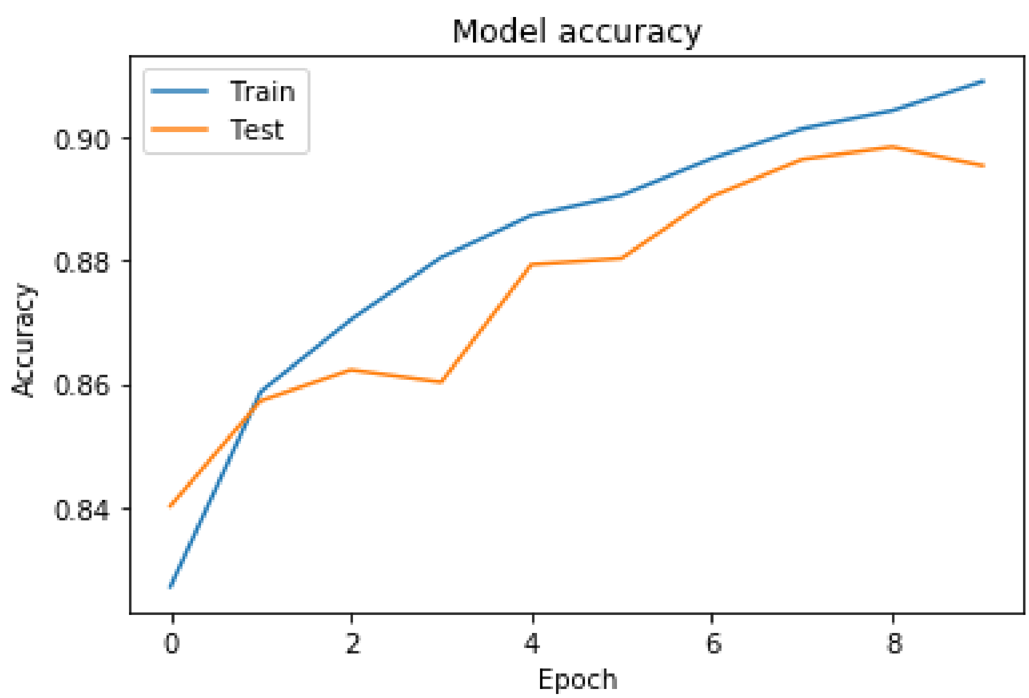

Fig. 33: Plot of Model Accuracy

This graph depicts that when the epoch values are increased , the accuracy of the training data set increasing continuously but the value of accuracy of the testing data fluctuates. It first increases, then decreases and at last this can be incurred that due to increase in the values of epochs, the accuracy of the testing data decreases at the last stage.

Epoch means to train a network on every item of the dataset once. It basically means the repetitions that are done forwards and backwards to train a network for a particular dataset.

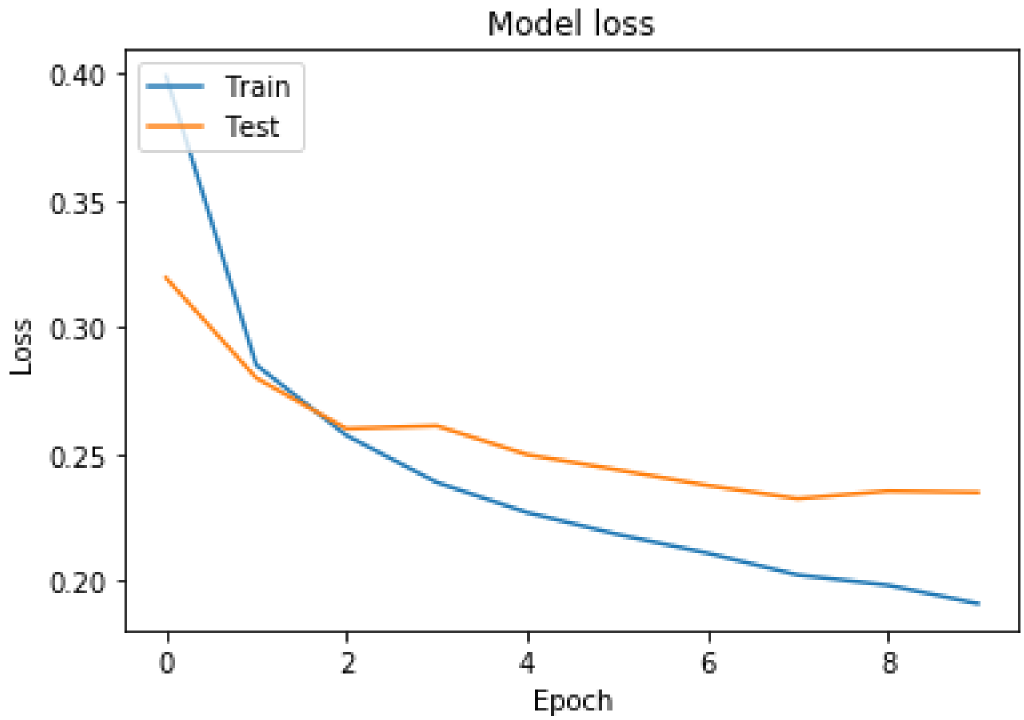

Fig. 34: Plot of Model Loss

This graph depicts that with increase in the number of epochs, the model loss for both training and testing data decreases which is a good remark for better efficiency and accuracy of the neural network model thus designed.

CHAPTER-5

CONCLUSIONS

It can be concluded from the respective project work that epilepsy represents certain neurological disorders and are quite common and can affect people of any ages. Epilepsy is as same as unpredictable seizure disorder. It may be related to a brain injury but most often, the cause behind the epileptic seizures s unknown. There may be more than one types of such seizures in case of a single patient and the symptoms of epileptic seizures can affect other parts of the body as well.

Neural system is the most broadly utilized strategy and demonstrates great in epileptic seizure identification. The exactness rates are high yet the main downside in utilizing neural organize progressively application is that the preparation time is longer.

Thus, we have developed a model to detect and analyze the same. The proposed neuro fuzzy model makes the use of ANNs for the classification of various types of epileptic seizures.

Future scope:

The proposed methodology is an initiative for modeling the prediction of epilepsy by classifying the order of epilepsy the patient is suffering from.

To bring up a better prediction model depicting the best possible accuracy for classification of the same.

Applications:

ANN Based Faulty Line Detector

Traditionally, the approaches that were used to determine the faulty line by tracking the change in three phases of the signals that were injected. By using the NN based fault line identification the occurrence of fault is directly determined by using current and voltage trails as the features.

Load Forecasting using an ANN:

Electrical load forecasting is a cornerstone of planning and operation of electrical power. The routine maintenance and scheduling of electricity generation on a daily basis, the energy estimation country wise and the planning of new plants all depends upon precise load forecasting in the future.

Load Forecasting is of the following two types:

Spatial Forecasting:

It mainly constitutes future load distribution forecasting in a region which can be a country, a state or a part of the state etc.

Temporal Forecasting: In this the load is forecasted for a specific supplier in future days, months, hours or years. Depending upon the forecasting length, temporal forecasting is of three types:

Medium-Term Load Forecasting (MTLF)

Short-Term Load Forecasting (STLF)

Long-Term Load Forecasting (LTLF)

REFERENCES

- Fullér, R. (1995). Neural fuzzy systems.
- Sood, M. (2017). Performance analysis of classifiers for seizure diagnosis for single channel EEG data. *Biomedical and Pharmacology Journal*, *10*(2), 795-803.
- Sood, M., Kumar, V., & Bhooshan, S. V. (2014). Comparison of Machine Learning Methods for prediction of epilepsy by Neurophysiological EEG signals. *Int J Pharm Bio Sci*, *5*(2), 6-15.
- Valluru, S. K., & Rao, T. N. (2011). Introduction to Neural Networks, Fuzzy Logic and Genetic Algorithms.
- Lamba, V. K. (2008). *Neuro fuzzy systems*. University Science Press.
- Sivanandam, S. N., & Deepa, S. N. (2006). *Introduction to neural networks using Matlab 6.0*. Tata McGraw-Hill Education.
- Rajasekaran, S., & Pai, G. V. (2003). *Neural networks, fuzzy logic and genetic algorithm: synthesis and applications (with cd)*. PHI Learning Pvt. Ltd.
- Srinivasan, V., Eswaran, C., & Sriraam, N. (2007). Approximate entropy-based epileptic EEG detection using artificial neural networks. *IEEE Transactions on information Technology in Biomedicine, 11*(3), 288-295.
- Prince, P. G. K., & Hemamalini, R. (2010, December). A survey on soft computing techniques in epileptic seizure detection. In *INTERACT-2010* (pp. 377-380). IEEE.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 8.5.2019

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: Ishu Priya Department: CSE Enrolment No 151424

Contact No. 9805460431 E-mail. ishurangra16@gmail.com

Name of the Supervisor: Dr. Hemraj Saini

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): EPILEPSY DETECTION USING NEURO-FUZZY SYSTEMS

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 57
- Total No. of Preliminary pages = 47
- Total No. of pages accommodate bibliography/references = 1

Ishupriya

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....10....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

8/5/19.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
08.05-2019	• All Preliminary Pages • Bibliography/Images/Quotes • ~~14~~ 20 Words String	17%	Word Counts	11,380
Report Generated on			Character Counts	59,487
09.05.2019		**Submission ID**	Total Pages Scanned	57
		1127106703	File Size	1.1M

SBhalla

Checked by
Name & Signature

Ashok

Librarian 09.05.2019

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com