Classification of Traffic Signs using Neural Network Algorithms and Comparison of Algorithms

Shahtaj Bano
Regn.# 330871

A thesis submitted in partial fulfillment of the requirements for the degree of **Master of Science**
in
Statistics

Supervised by: Dr. Tahir Mehmood

Department of Statistics

School of Natural Sciences
National University Of Sciences And Technology
H-12, Islamabad, Pakistan
Year 2022

FORM TH-4

National University of Sciences & Technology

MS THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: **Shahtaj Bano**, Regn No. **00000330871** Titled: **"Classification of Traffic Signs Using Neural Network Algorithms and Comparison of Algorithms"** accepted in partial fulfillment of the requirements for the award of **MS** degree.

Examination Committee Members

1. Name: DR. TARIQ SAEED Signature:

2. Name: DR. SHAKEEL AHMED Signature:

Supervisor's Name: DR. TAHIR MEHMOOD Signature:

Head of Department

19/08/2022
Date

COUNTERSINGED

Date: 22·08·2022

Dean/Principal

I Dedicate This Thesis To My Beloved Mother
And Late Grandfather.

Acknowledgment

Allah Almighty, the most beneficent and gracious, who created the whole universe, deserves all honor and glory. I am deeply grateful and indebted to Him for bestowing countless blessings upon me, including the courage and strength to complete my thesis effectively. Without a doubt, my sincerest appreciation goes to my supervisor, **Dr. Tahir Mehmood**, He is an amazing person indeed. May Allah bless him with an abundance of blessings.The research would not have been accomplished without his knowledge, experience , and support. My mother is the main pillar of my story. Whenever I failed in any task of my life she was there for me. Special thanks to my uncle Mumtaz Hussain for his support. I want to thank my teacher **Dr.Firdos Khan** who always gives me sincere and useful piece of advice. Furthermore, I want to thank my brother M.Shahzaib Ali, it is mainly because of his efforts and proper guidance and senior Shahid Khan, his appreciative responses to my questions that I have gained a complete grasp and respect of this field. They helped me throughout my research journey. I would like to pay my gratitude to my GEC members **Dr. Tariq Saeed** and **Dr. Shakeel** for their support and guidance in completing this thesis. Lastly, I want to thank the support of my sisters Shahbano and Shahrose and friend Hira Mehmood throughout my studies.

Abstract

The neural network algorithms are used to train different models for forecasting, classification, interpret, and analyze the results. The motive of the study was to train the data set with different neural network algorithms and check the classification accuracy and make comparison, which algorithm is best among all of them. The "German traffic sign" dataset is used to classify the different traffic signs. Each algorithm run for thirty times with different number of layers and hidden layers. A neural network depends on the learning rate, hidden layers, layers and activation function. The network gave calibration values, validation values, different results for different number of layers, and hidden layers. Smallest absolute gradient gave the best results for calibration (87.73) and validation (84.05) and at 2, 3,and 4 number of layers it gave the best classification accuracy. Smallest learning rate gave the least results for calibration (83.34) and validation (78.23). The outcome suggested that although the differences among the algorithms are not big, the SAG gave the highest classification accuracy.

Contents

List of Figures

Thesis

Shahtaj Bano

August 28, 2022

Chapter 1

Introduction To Machine Learning And Neural Networks

This chapter describes the importance of machine learning and Neural Networking in classification of traffic signs and strength of these models over classical statistical models. This chapter also describes the history of traffic sign classification in different countries.

1.1 Introduction

The power to recognize the traffic signs in an operational setting has been spotted as a critical need for intelligent transportation systems (ITS)[1]. Particularly , traffic sign recognition will support proactive, dynamic traffic control. Traffic sign classification first showed, in the design of speed limit sign recognition. First, it was in 2008 for the 2009 Vauxhall insignia then for the BMW 7 series in 2009 and the same year on the Mercedes-Benz s-class. At this stage these vehicles only detected the round speed limit board found all across Europe. And further, they worked on these systems because this was the main requirement for autonomous vehicles and self-driving cars. Blind people usually use self-driving cars and in these scenes, the detection system needs to recognize and classify the different traffic signs and not just the speed limit boards. The system indicates the driver when a speed limit sign or any other traffic sign is detected, keeping the driver informed of a turn ahead, speed limit changes, and other important road information. In this system a camera is attached that is front facing with a wild field of view that covers the entire road for any signs commissioned by traffic regularity bodies. When the camera captures road sign information, the road sign message is manifest on the multi-information display to keep the driver aware of all the situation and important information about the road signs and where to go. What is the limit? Etc. A traffic sign recognition system could be developed as a part of an intelligent transport system (ITS) that continuously checks the activity of the driver in time about upcoming decision points regarding risky traffic situations and navigation. The intelligent transport system centers on integrating information technology into transport infrastructure and also in the system of vehicles. This kind of system includes the in-vehicle navigation system, traffic management, road sensors, monitoring, and electronic message sign. This system increases transportation efficiency, to reduce the environmental impact with the use of modern communication technology and road safety issues[2]. The intelligent transport system has a big role in all this scene. This research effort focused on a traffic sign classification for the "German traffic sign recognition benchmark" in this, we aim to classify the traffic signs. It is our mirth to understand the characteristics of road and traffic signs and their implications for image processing for the traffic signs classification task. Different models and algorithms were developed and tested for the classification of traffic sign problems, which is defined as a technology by which a vehicle can recognize the traffic signs put on the road like turn ahead, speed limit, etc. This was the part of features collectively called ADAS which was being developed by a variety of automotive suppliers, that used image processing techniques to detect the traffic signs. The whole detection method can be parted into different criteria like shape-based, color-based, and learning-based methods and common are based on the shape of the sign board. Signboard shapes like circles, hexagons, and triangles describe different types of signs, which can be used for classification problems. In this research, the main strategy that has been adopted is to find the right combination of colors in the scene so that one color is located inside the convex frame of another color and make the combination with the right shape. It is very important to understand the color, color spaces, and also color space conversion. In this research many algorithms have been used that is based on neural network, these are deep learning algorithms that tune the model. They take an input image, assign importance to different aspects in the image and give the network indicator as an output. Machine learning is a sub-field of Artificial intelligence, in which divide the data set into two parts train data and test data and train our machines to work like humans. Nowadays Robots are probably the best example of machine learning which works like humans because of their artificial intelligence system and perform their duties without any tiredness. Automatic cars made by Tesla are also one of the best examples of machine learning in which car does not need any driver and the accident ratio of the automatic cars is also negligible. Machine learning has been used in this technology which helps the car in image recognition. There are different applications of neural networks, backpropagation neural networks which are used for classification and image recognition to detect objects, tune the parameters, recognize features, recognize faces, etc. They are made up of neurons with adjustable and learnable weights and biases. In these algorithms where input an image, each of its layers generates many activation maps. Activation maps show up the relevant features of the image. Each neuron takes a group of pixels as input, multiplies their color values by the assigned weights, sums them all, and run them through the activation function. These neural networks are commonly used to classify images, make clusters of them on the bases of their similarities, and then perform object recognition. It can be trained to take in these predefined traffic signs and learn how to use different deep learning techniques.

1.2 Machine Learning And Classification Models

This area starts with the concept of machine learning, simple and multi-layer neural networks, different types of classification models, and the selection of models based on their classification power.

Data Cleaning In Machine Learning

Nowadays data cleaning is very important as it is necessary for every model. Development in technology has been raised in a few days which has its pros and cons. Pros always come at a cost. Now industry has an immense amount of data, billions of bytes of data are available but it has been a real challenge to deal with this massive amount of data and use it for some statistical or analytical purposes in real-time. In the past few times, the demand for data scientists and data analysts has been raised. Large number of big technology companies are using data for their promotions, advertisements, and revenue purposes and for that, they hire data scientists and data analysts to handle the data for their purposes. Facebook advertisements are the best example of this, where let's say you search mobile phones on google for buying purposes and after some time you start watching ads of relevant websites on Facebook which are selling mobile phones. So, it is a very important task to use our data in the right way. The data is called primary sourced data because first of all, a person collects it by himself and then use it. Firstly, it is in raw form and because of that sometimes person faces the problem of some values. Some of them are missing, and some of them are outliers, so this kind of data also appear, and thus when multivariate analysis is working, for that system needs structured data with no missing observations. So in these kinds of situations, the data should be clean and for cleaning the data, there are different ways. At this place, human can use a very common method that is, use machines that take garbage as input and throw out the garbage, but try best to clean these unnecessary values as much as possible[3]. There are two main techniques for data cleaning i.e., quantitative cleaning technique and qualitative cleaning technique. In qualitative technique, it contains constraints, rules, and patterns to spot errors. And in the quantitative cleaning technique, there are some statistical methods for the identification of errors. When the errors are detected, then the system could use different explained ways to get the better of these errors.

- The main task in data cleaning is to first find the missing values in data and fill these out. There are many different ways to complete these missing values. If data is facing one or two observations that are missing then it is recommended to use earlier values if the data has a pattern of increasing or decreasing over time. If the data is homogeneous then system can use either mid-value means median or mean of data for those missing values. But for two or three missing values it cannot use median or mean values because they would be considered as an outlier in that situation.

- The other way which can be use is to remove all those rows which have missing values. This is not a good approach although but for the sake of cleaning system can use it as a last option. It is not recommend but could be used also study has been advised to eliminate those rows which do not have a connection with other data or system can erase those which seem to have a seasonality component. e.g., let's say there are historical data of some variables for 15 years. For some time, like data is available for Eid festival or any day like independence day in this case for specific years that way is not applicable so that system could remove missing rows because Eid effect or national days are already present in data.

- One more way could be an estimation of data using regression analysis or statistical techniques and making a prediction for the missing data using their classical models.

1.2.1 Machine Learning

Charles Babbage is a well-known name in history because he was the one who introduced the mechanism of the machine when no one knew that there will be a time when people can do their difficult tasks in one click. He invented the computer of huge sizes like a size of a room in the middle of 1830 and this faced huge criticism from the orthodox[4]. At that time all people were unaware of the power of this innovation and never knew it will become a massive part of human's daily life activities. Nowadays it would not be a surprising statement if we say, humans are machine driven and day after day humans are becoming more and more dependent on machines. We are all crowded by machines like a person can take a breath through a machine, from birth to death we are on machines. But there was a drawback in the invention of Charles Babbage that a computer can do only that task which human asked for. These program-based machines are only capable of doing those tasks for which human put information as input and get outputs but if human change anything or needs to know a little bit change of it they can't do that. They cannot exceed their defined limits. So study has machine learning, which is the scientific technology of making machines that can do tasks with more efficiency and in less time. This technology wanted to give the idea that humans and machines can do the same job but machines can do these tasks with high precision as defined these machines (computers) cannot do the task for which they do not have operating algorithms[5]. So, people work on artificial intelligence (AI) in machines, there has been so much hard work, so much struggle and it is expected that these machines would be capable of operating and doing and improving themselves in the coming future and they would not be dependent on humans dictation.

Machine Learning VS Deep Learning VS Neural Network

Artificial intelligence is the idea of making the smart intelligent powerful machines. Machine learning is a subspace of artificial intelligence which helps human to create AI-driven applications. Machine learning is a basic type of science but science is not originated from it. Most people used to think these three concepts machine learning, deep learning, and neural network are related to each other but in reality, these are sub-sections of artificial intelligence[6]. Machine learning comes first in this lane, deep learning is a sub-section of machine learning while neural network is a sub-section of deep learning. The algorithm learning techniques in both cases are the main difference between deep learning and machine learning. In machine learning human interference is more as compared to deep learning because it has a manual task system. So in this human make sure that from the data algorithm is learning, built-in nature is seeing when network is doing deep learning process, this process is more automated. In machine learning the different mechanisms to put the values into the system can be seen and explain the system about the algorithm.So in the deep learning process, it automatically learns algorithms from the data, adjusts the weights automatically in every iteration, gives a suitable learning rate, etc[7]. The importance of machine learning is that system can train any network without knowing the deep learning and same for the deep learning but when the working is with deep learning then to know about the machine learning makes the network easier and understandable.

To understand the relationship of artificial intelligence, machine learning, and deep learning we illustrated the figure 1.1. The artificial intelligence is the main branch in which the system mimics the brain system. Machine learning is a sub-branch of artificial intelligence which use the statistical ways to enable the system to improve and upgrade with experience and deep learning is a sub-section of machine learning.

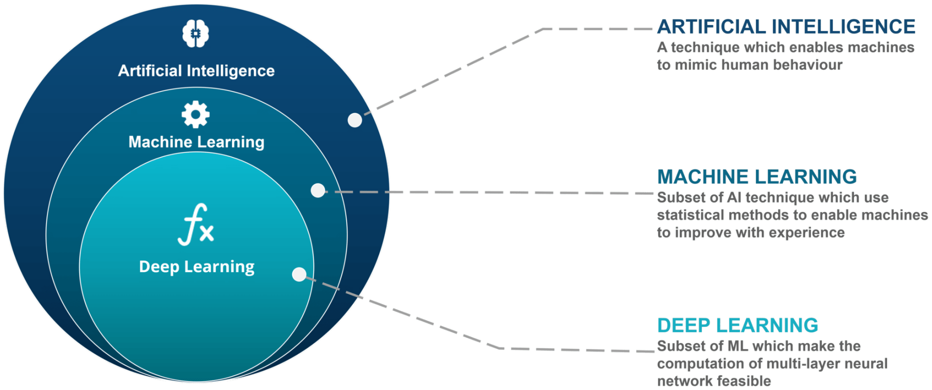
Artificial Intelligence
Machine Learning
f_x
Deep Learning
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour
MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Figure 1.1: This figure illustrates the relationship between AI, ML, DL

Deep learning has two types of models which are supervised learning and unsupervised learning. In supervised learning, system needs to have labeled data as input and output and the algorithms of supervised learning “learn” from the training data set and do iterations and adjust weights to minimize the error rate, but also keep in mind that labeled data is not always required for supervised learning. At the same time in unsupervised learning system do not need to have labeled input or output, it uses unlabeled or raw data sets. It does not make its algorithms. it depends on machine learning algorithms and makes groups of unlabeled data set. Deep learning is more powerful to handle unlabeled data sets (i.e., texts, images) and after that classify these data sets into groups or clusters according to their similar properties[8]. But in machine learning the situation is different if the system puts unlabeled or raw data e.g., of either texts or images, firstly, it needs to put information manually into an algorithm about classification, and then it becomes a banal and boring task to even control data in machine learning. Due to this problem or flaw of machine learning, deep learning and neural networks were introduced which have a massive number of uses in every area like speech recognition and language processing[9].

1.2.2 Machine Learning Working

Machine learning algorithms have three steps to work i.e., a decision process, an error function, and last model optimization process.

- Most of the time machine learning is used either in the classification of data or prediction of data. Machine learning starts the process by inputting training data into the already chosen algorithm and it trains the system to make difference between two objects suppose a lion and a cat. There are billions of characteristics that made a lion different from a cat. From this, it concludes, that it requires learning every characteristic of an object. Every characteristic is important to machine so that it can make a difference between two objects and recognize them successfully. So the sum up is machine learning will take the data as input, then train this data manually, after that algorithm will learn from it and then do classification [10].

- Later an error function is defined which finds the error between actual and predicted outcomes.

- In the last estimate the model by weights adjustment and try to reduce the error. And this whole process will happen n number of times until the required accuracy is achieved[11].

1.2.3 Machine Learning Methods

The classification of machine learning is in three subcategories i.e., supervised machine learning, unsupervised machine learning, and semi-supervised machine learning. Supervised machine learning defines the labeled data only. It does not work with raw data and unsupervised learning defines the raw or unlabeled data but semi-supervised learning has the qualities of both learning, supervised learning and unsupervised learning.The detailed work on these learning techniques are given below.

1.2.4 Supervised Machine Learning

There are many different types of data sets and so study cannot define a general mechanism for everyone. Each data set has different properties so due to different types of data sets and their properties machine learning has been divided into different methods for various types of data.

Supervised machine learning gets information through specified or labeled data to observe the native nature of data and make future predictions easier. The condition which is mandatory for supervised machine learning is only to mention the type of data, whether the data is from labeled data or unlabeled data where a label can define as a data set that already has been attached. If the data is in raw form or not labeled then supervised machine learning is not a good approach to apply to data to make predictions. The predictions which network wants to make will not give accurate results with raw data in the supervised technique. Supervised learning data sets are created to train algorithm into classifying data and they use training data set which contain labeled input and give a correct output which helps the model to learn fast[12]. The example which is best to explain supervised learning is the "text classification problem", in this the aim is to classify and predict the class label of a given text. Another example related to text classification is to classify or predict the sentiments of text pieces same as tweets and product reviews.

In given below figure 1.2 the whole mechanism of supervised machine learning can be seen . There is a labeled data, in the next step it will go for the model training to ready the system, and go for testing and prediction of the data to check whether it predicts the right or wrong data.

Different shapes can be seen in labeled data. Due to different geometrical shapes the data has been labeled according to that and after that algorithm learns from it and in the last the categories prediction of geometrical shapes correctly.

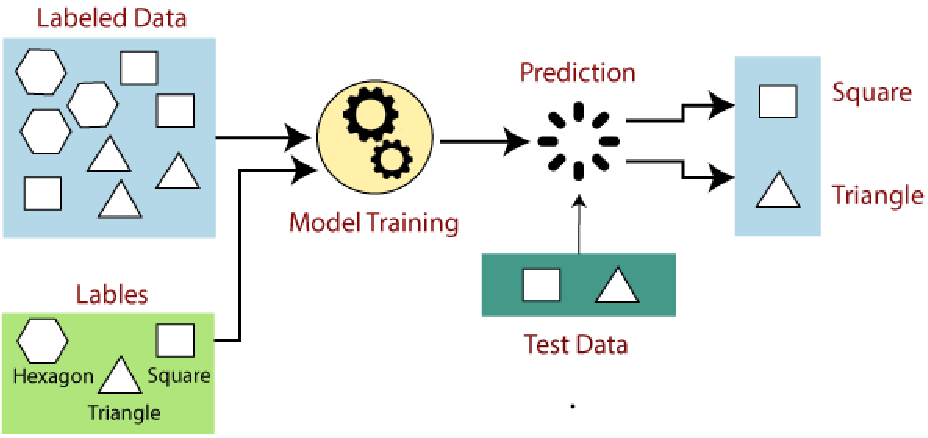

Figure 1.2: This figure illustrates the working of supervised machine learning

Supervised learning has different types of uses in the context of statistical modeling.

- In regression analysis, there is immense use of supervised learning ways in which it observe and find the effect of unlike independent variables on a dependent variable and in the analysis make predictions that depend on resultant coefficients.
- When the system faces binary output data then it uses a logistic function to fit the data and this is familiar as logistic regression. The logistic regression function and simple regression are somehow similar to each other. Any technique and method has its pros and cons same as supervised machine learning way also has some positives and some negatives and the following are those.
- Historical data has a high weight in supervised learning so it makes it easy to do predictions because statistics and statistical analysis are almost dependent on past manners.
- The optimization power of supervised learning techniques is quite high

because it depends on the memory of the data or also known as the memory-based models so it optimizes the data very well.

- The real world have a massive number of problems in which human can use supervised machine learning techniques to solve these problems.
- The main disadvantage of supervised machine learning is it dependent on labeled data if they do not have labeled data so it could not be in the right situation to use its algorithms and by this, the decision boundaries could be affected.
- Supervised machine learning algorithms are best than unsupervised but training these algorithms is not an easy task because the system has to select many samples from every class while training therefore making time is a big arrangement [13].

1.2.5 Unsupervised Machine Learning

The main point of supervised learning is network can only deal when the data set is labeled but when the unlabeled data set present then study goes for unsupervised machine learning. These are the extensions of machine learning and they are good for searching hidden patterns, making groups or clusters inside the data, customer segmentation, cross-selling strategies, and image recognition. Human interaction is not involved but has massive use in data analysis[14]. It is best in finding patterns but does not know exactly what is the objective. What is the main purpose behind all the processes? What does system need to do? What other is thinking? So it is very helpful in cyber-security where in the system attacker always go for the different methods. So the summary points are: It learns the patterns, designs, and structure from the unlabeled datasets, make predictions, and do analysis. In real life it is a more suitable technique because the system usually has raw data.

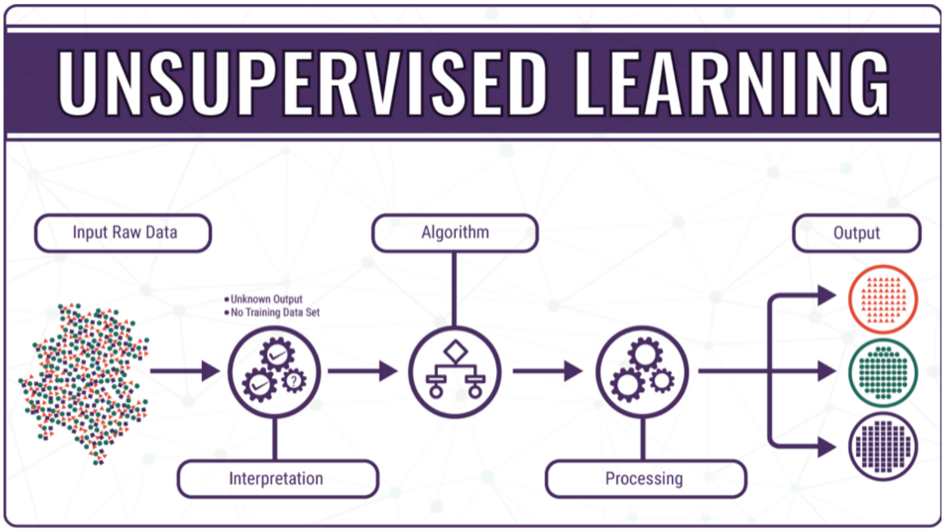

Figure 1.3: This figure illustrates the working of unsupervised machine learning

In figure 1.3 the design of unsupervised machine learning can be seen where the data, which is in a raw form go for interpretation so in this there is no training of data is happening then application of algorithm happens so the process of the method starts and in last system gets output.

Unsupervised machine learning has a massive amount of uses in many areas but in clustering, association, and dimension reduction it is best. Here are some of them:

- clustering is a purely statistical concept but can be used in data science. In clustering algorithm, use unlabeled data, process it, and transform it into groups with the help of their patterns. Cluster algorithms could further be classified into overlapping, hierarchical, and probabilistic types.
- Initially the data values are separate and then made into clusters based on their similarities and this until one cluster unit is achieved is hierarchical clustering. In probabilistic clustering, solve the complications of soft clustering or estimation, and the method of making and select-

ing groups is based on likelihood e.g., through the Gaussian mixture model. Association is used to check the relationship between variables. This method is valuable in industries like when companies need to check the relationship between their products and on this basis, they increase or decrease the sale of product and also works in recommendations of products on an online store where if someone buys a table then program recommend them a chair, lamp, diary, etc as relevant products[15].

- It is a common perception about data if the system has more data means more information but it can be a problem at some points where the problem of over-fitting and extra values, which are not related can be seen. So it makes it complicated for an algorithm to learn. For that situation, dimension reduction techniques are best. They eliminate some features of data to reduce the dimensions of data. This only reduces the unnecessary information and keeps the integrity of the data[16].

1.2.6 Semi-Supervised Machine Learning

Semi-supervised machine learning is the mixture of both supervised and unsupervised machine learning. If the data set is the combination of labeled and a large portion of it is unlabeled then go for semi-supervised machine learning in which it gets the benefits of both supervised and unsupervised learning so it gives the bridge between two approaches[17]. Semi-supervised Machine Learning has various uses in real time some of them are given here: It is used for clustering the objects which helps to classify the class of that object, an outcome variable, cluster labels, or information about the relationship when it is known. The main advantage because of the combination it is easier to understand and makes applications more simple. If the system works with semi-supervised learning the application of that algorithm becomes very easy. It is very helpful in the reduction of the annotated data.

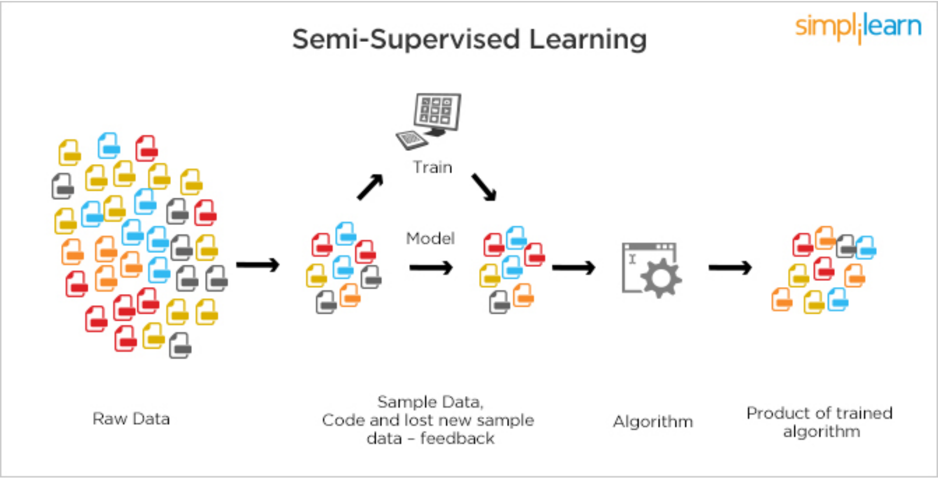

Figure 1.4: This figure illustrates the working of semi-supervised machine learning

In figure1.4 the algorithm of semi-supervised machine learning can be seen. It has data in a raw form then the system goes for the training of model so that it can work accurately for the network then its algorithm works and system gets the desired output or predictions from the network. An example of semi-supervised learning is a text documents classifier[18]. This helps to remove the noisy items from the data which makes the computations more efficient and also best for the model. It is a more suitable algorithm and the computation is very simple and the system gives the result with very high accuracy. There are different algorithms which are based on semi-supervised technique some of them are listed below:

- Self Training

  It is the re-sampling method which again and again labels the unlabeled training samples.

- Graph Based semi-supervised machine learning

  The more significant sub-section of semi-supervised learning and nowadays one of the most famous algorithms. The steps to make it are as follows:

- Graph development
- One the subset of nodes, system has to infuse the seed marks
- The unlabeled nodes in the graph have to be mentioned with labels.
- Low-Density Separation

  In this system has decision boundaries and these decision boundaries should be in between the low-density region.

1.3 Objective Of The Research Study

Following are the objectives of the research:

- Classification of traffic signs by using algorithms of neural networks.
- Understand the characteristics of road and traffic signs
- Comparison of the algorithms.

1.4 Organization Of The Study

The research gets going as follows, chapter 2 will give a detailed literature review about neural networks and algorithms of neural networks, and chapter 3 will tell the different algorithms of neural networks after that the description of data will be writing, applications of algorithms of ANN, for the final chapter conclusion of the study and future direction of research.

Chapter 2

Literature Review

2.1 Introduction

In this chapter we commence our study with the concept of the neural network and then further jump into the pond of algorithms of neural networks. The algorithms of neural network are great in the tuning of the networks like a back-prop, resilient-prop, etc.

2.1.1 Review Of Studies Related To Recognition Of Traffic Signs Using Artificial Neural Network And Deep Learning

Traffic signs play an important role in road safety and prevention from accidents. Traffic signs are designed to give drivers and pedestrians necessary information and also warn about dangers. These are placed along the roadside and construction areas to guide and warn the drivers, as well as to regulate the flow of traffic for all people. All signs like traffic signs, parking lot signs, and including construction signs are designed to provide a simple and clear message so that anyone can understand. Nowadays autonomous vehicles are in trend and somehow it has become a necessity for many people. Most vehicles in modern times have traffic sign recognition censors but those do not work well because for some reason sometimes they don't recognize traffic signs due to light, weather conditions, maybe any object is hiding it or color has faded away, and so many reasons[19]. In research, researchers made many algorithms, and models that used different neural networks to work like a human brain. It gets considerable interest lately, this interest has become famous for intelligence applications, like advanced driver assistance systems (ADAS), autonomous driving, mobile or advanced mapping, and the releases of large traffic signs datasets such as German and Belgian. Traffic sign recognition can also cover two problems: traffic sign classification and traffic sign detection. Traffic signs detection tells where to localize the traffic sign in the image space accurately. It allows the driver to be a little more at ease while going on unknown new roads or tricky cuts. Traffic sign classification controls the labeling of traffic signs, recognizing the traffic signs, and classifying what is the sign and with which class it belongs[20]. In 1968 the Vienna Convention on signals and road signs firstly showed which was able to standardize traffic signs across different countries. In the starting 52 countries signed this treaty, including which 31 European countries included. The convention has mainly classified the road signs into seven categories labeled with letters A to H. This way of standardization has been the main rule for helping the development of traffic-sign recognition systems which can be used worldwide. In the past traffic sign recognition was only based on speed limit signs in 2008 for 2009 VAUXHALL INSIGNIA after that in 2009 the new BMW 7series showed a speed limit, and the same year on the Mercedes-Benz S-Class. That was the time when their systems only recognize the round speed limit signs placed all across Europe. With time Second-generation systems came up with overtaking restrictions. The first time appeared in 2008 in OPEL INSIGNIA after this many companies followed OPEL INSIGNIA and had this in their systems. By the time 2011 VOLKSWAGEN also had this technology since 2012, A technology came that is called Road Sign Information that was present in V70, VOLVOS80, XC70, S60, V40, and V60. They were only able to recognize direction signs. They were not good at the recognition of city limit signs, while most European countries are associated with speed limit signs. Nowadays technology turned into modernization and they converted their systems into modern systems. In Europe and other countries also in Pakistan most of the vehicles are autonomous and self-driving in which traffic sign recognition technology is important and also have different cab systems in which the drivers use GPS. They are not aware of many places so the system recognizes the traffic signs and shows indicators on board to help them to reach their desired place safely. Traffic sign recognition is very important for the safety of pedestrians, drivers, roads, and nature. Traffic signs can be classified or recognized by using forward-facing cameras in modern vehicles, autonomous or self-driving cars, and trucks. One of the most basic and important samples of a traffic-sign recognition system is the speed limit. People face many accidents because they don't follow the speed limit signs or their vehicles don't recognize the sign and they drive over the speed limit sometimes. Most of the GPS data would pick up speed information, but further speed limit traffic signs can also be used to gather the information and display it on the dashboard of the vehicle to alert the driver about the road sign. This available feature is an advanced driver-assistance feature present in most high-end vehicles, mainly in European cars[21]. Deep neural networks are somehow the best in all fields but in classification, they do amazing work. These networks have different algorithms after dividing the data set into two parts the training and testing part, and the application of the algorithm set according to their work requirement. There are multiple algorithms for traffic-sign classification. Few of them are those which are based on the shape of the sign board. Typical sign board shapes like circles, hexagons, and rectangles describe the different types of signs, which can be used for classification. Some main algorithms for object recognition include AdaBoost detection, Haar-like features, Freeman Chain code, and deep learning neural network methods. Haar-like features are used to create cascaded classifiers which are used to help to detect the sign board objects or characters. The shape representation system has the Freeman chain code method which is, in an image the system describes the shape of the boundary of an object. In this, the main idea is to pass over the boundary of the object, and for every new pixel of an image, record the direction by which the system reaches the object. It has two kinds of chains, one is absolute and the other is s relative chain code. The neural networks use computer vision and image processing to train the network and give its potential outcomes. After that these trained neural networks can be used in real-time to detect or recognize new traffic signs in real-time. Self-driving vehicle companies including uber are making and outsourcing traffic-sign datasets along with navigation and map companies like TOM TOM. These modern computer vision and neural network techniques make this super and global efficient and easy to get a task in real-time. Convolutional neural networks are working best in modern traffic-sign recognition systems, mainly for the requirements of autonomous vehicles and self-driving cars. In these situations, the recognition system needs to identify most traffic signs and not just speed limits. This is where the Vienna convention on road signals and signs comes to help. A convolutional neural network can be trained using deep learning techniques to learn and understand the predefined traffic signs[22]. Deep learning methods can be comprised of traffic-sign recognition or detection. Digital curves Polygonal approximation using the Ramer-Douglas-Peuker algorithm can be used to detect the shape of the signboard and use methods like support vector machine (SVM) and Byte-MCT with the classifier of an AdaBoost have been used in one of the ways to detect traffic signs. Neural Networks have a lot of algorithms but we choose Backpropagation, Feed-forward propagation, and some more to classify the traffic signs and recognize them correctly[23]. In this model, weights were involved, adjust the weights to minimize the statistical error and give a high accuracy. The purpose of the study was to show that we can get the best accuracy by using the different algorithms of ANN (artificial neural networks).

2.2 Neural Networks And ANN Algorithms

2.3 Neural Networking

Neural networking is called the mimicry of human brain system and it is a unit in neuroscience. Back in history, the people used computers in which machines do only those tasks that was mentioned to do but due to neural networking now machines are able to work like the human brain works, how the human brain commands, operates different tasks, and much more. Neural networking due to its capability of doing tasks like the brain has many benefits, people are using it in many fields like in medical the professionals are using it in neuroscience and solving many neurological complications and it helps to do processes smoothly as study knows neurons are complexly attached with each other[24]. A neural network is a compound set of input layers, an output layer, hidden layers, weights, neurons, etc, and these units or layers are combined in an exact way that copies the human brain[25]. In neural networking, the study has various algorithms to find the underlying relationships in a data set and have a set of neurons that could be either natural or artificial[26]. Classification of neural network model depends on the nature of the data set. Based on the nature of the data set it has Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and more[27].

2.3.1 Understanding Of Neural Networks

The basic structure and concept of neural networks will be discussed below. As we mentioned earlier in neural networking system works like a human brain working.

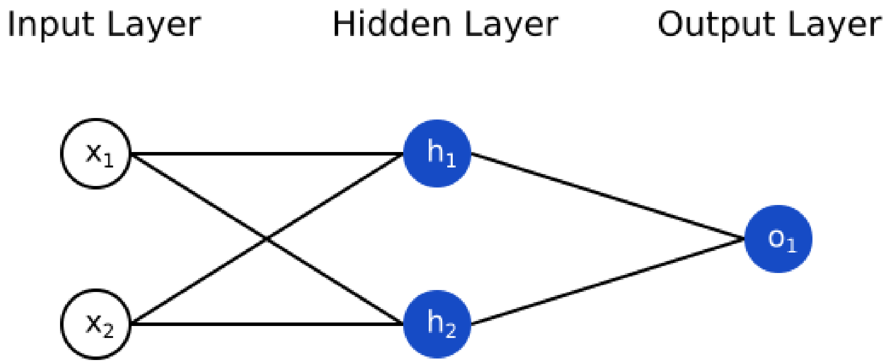

Figure 2.1: This figure illustrates the structure of a simple neural network

In the given figure2.1 the structure of a simple neural network can be seen. This network depends on two inputs which are x1 and x2 then these inputs go to the hidden layers but in between they had some weights and some activation functions which activated the function to further procedure. This model is a single perceptron model because it has a single hidden layer with two nodes which are h1 and h2. And in last we have our output value which is represented by o1. It is very different from the computers like computers can not mimic the working of the brain but neural networking is mimicry of brain work. Neural networks are made up of basically three components which are input which system gets from its features and also called them independent variables because inputs are always based on system's desire there are hidden layers which are the main part of the network and output which are the result or conclusion it gets through the process and also outputs are dependent. The whole process seems simple but this is not as simple as it is looking there are a lot of components between them like there are some weights attached to them, and some activation functions work here to further proceed the process, then there is a bias which is usually 0.5 but it also differs according to the wish and system, some transfer functions which transfer the information through nodes, etc. In a neural network, there is a combination of different neurons which are attached through different interconnected nodes and make a layer of those. So by attaching the nodes the system makes a network that works properly. In networks Inputs layer has one or more neurons depending on desire like the weight of a person is a single neuron input but the height of all the students is not a single neuron input same as for output neurons they are mostly one but could be more than one based on specific needs, it is known as a logistic outcome if two outputs are present for example with height input the system wants to predict the IQ level or age of the students[28].

The main process part is where the network has hidden layers. Hidden layers could be one, two, three, or more than these but there is one concept for that when there is only one hidden layer in the system then that is called a single perceptron But if there will be two or more than two hidden layers then call it multi-perceptron and they used to apply a non-linear transformation with the help of activation function and weights which adjust on the independent variables. Further, will see the concept and implementation of weights mean how to implement the weights on the network and also how to adjust those weights to minimize the error rate, activation function, etc[29].

2.3.2 Activation Function

In a network there are many neurons and each neuron has an activation function and a threshold value[30]. Different types of activation functions work and it depends on the nature of the data which function will work. The limit of the activation function is sometimes between 0 to 1 and sometimes between -1 to +1. Selection of activation function is important and the fixed one use in all layers of the network[31]. Some of the activation functions are explained here,

- first one is a sigmoid function which is the most popular and frequently used activation function, and it gives the reflection of the S alphabet, once the system concluded through the whole process of input, adjustment of weights, this sigmoid function gives the final value between 0 and 1 [32]. The given function is sigmoid function:

$$f(u) = \frac{1}{1 + \exp^{-cu}} \tag{2.1}$$

The benefit of the sigmoid function is, It's computation is very fast and have less economical worth for this it is the most preferable one. Once the system takes the derivative of the sigmoid activation function .

$$\frac{\partial(f(u))}{\partial u} = f(u)(1 - f(u)) \tag{2.2}$$

As it can be seen that derivative is simply the derivative function multiplied by 1 minus f(u) so saw that no major and complicated computations required and when the network is using gradient descent to find out the maxima, this derivative is used to learn the weight vector. It works well in the case of classifiers.

- The next activation function is an alternative to the sigmoid function which is called the Tanh activation function. It can be displayed as

$$f(u) = \tanh(cu) \tag{2.3}$$

And in shape, this function is also similar to the sigmoid S-shaped function but the difference is it ranges between -1 to +1[33]. These two have similar properties but these are more excellent for non-linear functions due to their wide range sometimes system avoids both of the functions because of the vanishing gradient problem and to avoid this vanishing gradient problem, go for the Relu activation function which is better in computation and performance[34].There are a lot more activation functions that usually don't use because of less competency.

2.3.3 Working Of Neurons

Neural networks are made up of the combination of neurons set in layers[35]. Every neuron is a mathematical operation in which takes values as inputs, multiplies them with their weights then sends the sum through an activation function to another neuron. Here is the sample of mathematical computation of a neuron by which it gets the output.

$$\sum_{j=1}^{n} w_{ij}x_j + b_j \tag{2.4}$$

Here in equation bj is biases of the function which are very important because in modeling of fitting the data weight goes up and down and if there is bias then there is no need for a line to pass through the origin and it is same like the intercept in a regression model. It gives space for the activation function to go either in an upward direction or in downwards and also it provides flexibility which is good in machine learning. In the neural network, there is an assigned magnitude of error with the calculation of weights and inputs which is called the cross entropy loss function. The objective is to have the least error so that the accuracy for the model gets high. The model's accuracy just depends on its error rate so try to minimize the error rate as much as system can.

In given figure2.2 the working of neurons has been shown. There is a single input layer and in it, there are three input neurons which are x1, x2, and x3 which take values then the arrow between them shows the activation functions which are working for the process there are three weights w1, w2, and w3. firstly, the first neuron which is x1 goes with the weight and bias to the hidden layer then moves towards the output then again the first neuron is now attached to the second weight and again goes to the end. And this process goes for the second input neuron and then goes for every neuron in the network. And in last system gets the error rate for the networks.

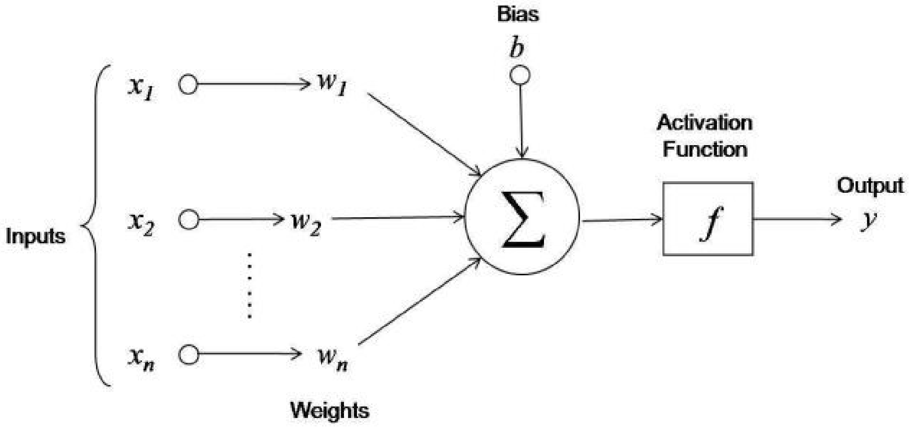

Figure 2.2: This figure illustrates the working of neurons in neural network

The main objective is to construct a neural network that minimizes the cross entropy loss function with the help of adjustment of weights and learning rate. Learning rate is a tuning parameter that ranges from 0 to 1 and tells the step size at each iteration or epoch while the network is getting to minimize the cross entropy loss function[36]. There are different algorithms for different tasks. Discussing below:

2.3.4 Feed-Forward Neural Network

The feed-forward was the simplest network that works in one direction, it is also called a uni-direction network[37]. Feed-forward is based on a simple way just take inputs to go through the hidden layer nodes and go to output nodes no need for any kind of loops or cycles for this network. This network mechanism is very popular because the computation power is good as it calculates relatively fast and almost could solve any accountable calculations. Feed-forwards are more accurate in the prediction of things like they work amazing with historical data[38].

2.3.5 Step By Step Process Of Feed-Forward

- Let's suppose there are two input values that are associated with ten weights and two biases functions and the system wants binary output. So initial biases will be

  B1=B2= 1 or 0

  Here the system can fix the biases according to the nature of the data. A bias permits the network to either shift the activation function to the right or left. The first step is to find the product of weights and inputs in hidden nodes.

$$H1 = I1W1 + I2W3 + B1W5 \tag{2.5}$$

$$H2 = I1W2 + I2W4 + B1W6 \tag{2.6}$$

  Next is to select the activation function which suits to the data best. The network selected the sigmoid activation function here.

$$S = \frac{1}{1 + \exp(-x)} \tag{2.7}$$

Now go for hidden nodes activation function.

$$HA1 = \frac{1}{1 + exp(-H1)} \tag{2.8}$$

$$HA2 = \frac{1}{1 + exp(-H2)} \tag{2.9}$$

$$O1 = HA1W7 + HA2W9 + B2W11 \tag{2.10}$$

$$O2 = HA1W8 + HA2W10 + B2W12 \tag{2.11}$$

Now firstly choose activation function for output layer; $y = f(x) = n$, then calculate output node activation function

$$OA1 = O1 \tag{2.12}$$

$$OA2 = O2 \tag{2.13}$$

After the first stage, the error will be valuable but we'll use an algorithm called[39]. Backpropagation for adjustment of weights and minimize the error.

2.4 Conclusion

The study has gone through an extensive literature review and came to know the efficiency and benefits of using artificial, simple neural networks and then deep learning with multiple layers. It has been observed that although deep learning is recommended to use in many cases due to its flexibility to change weights but in some cases, ANN also gives good classification. We also observed that these models give very high predictions when it comes to short data set classification but are not that accurate in long run because of the hierarchical structure of the series.

Chapter 3

Methodology

3.1 Back-Propagation

Backpropagation is the most frequently used and easiest network of neural networking and is also called the essence of neural network training and so this is the standard method for training artificial neural networks[40]. When the system faces a loss function in networking, then it does weights adjustment and by backpropagation, tune the weights of the network based on the error rate which the network obtained from the previous epoch or iteration. When the network fine-tune the weights that allow the network to reduce error rates and make the model more reliable and efficient by increasing its generalization. Basically, backpropagation starts from the very last node and goes to the very first node weights adjustment so that is why it is also called “Backward propagation of error”[41].

3.1.1 BackPropagation Algorithm Working

For backpropagation firstly there is a need to feed input values forward through the network, calculate the total net input to every hidden neuron, use the activation function and repeat the algorithm with output neurons and find out the total cross loss entropy function. It computes the gradient of the loss or error function for every single weight with the help of chain rule and has the computation of gradient but that does not define how the system used gradient[42]. Once the network find out the loss now use the backpropagation method to go back to every node's weight and adjust every weight in the network so that the actual output value would be closer to the target output means minimizing the error rate. But in backpropagation, one parameter which is known as the learning rate has a very important place because it is a tuning parameter that defines the step size at each iteration when the system wants to find the point of local minima. The network works like computation of one layer at a time efficiently, unlike a native one-way computation. It gives the computation generally in the delta rule[43].

3.1.2 Algorithm Steps

1. Input comes through the already made path.
2. The input computed using real weights W. Firstly weights are randomly selected.
3. Done calculation of output for every node from the input layer nodes, to the hidden layers nodes, to the output layer.
4. Calculate the total error function.

   Error function= Actual Output – calculated Output

5. Move back from the very first output layer to the hidden layer to adjust the weights in such a way that the error will decrease.

Keep this process in a loop until the wanted output is achieved. As the mathematical computations of feed-forward is explained now will move towards the computation of the backpropagation. method[44].

3.1.3 Process Of Back-Propagation

To do backpropagation, always need to do feed-forwardpropagation first then the study can move towards backpropagation. The main thing in backpropagation is the adjustment of weights. The initial weights of the network were guessed weights, (one theory related to weights adjustment is, there is an ocean of different weights the system drops the network into that ocean and which weight is suitable to the network it comes with that this is called weights harmony). There are many different techniques for updating the weights, using the weights adjustment method by which weights would be adjusted. The adjustments of network weights using the stochastic gradient decent optimization.

$$\frac{d}{dx}[f(g)(x)] = f'g(x)g'(x)\frac{\delta e}{w1} = \left(\frac{\delta e}{\delta OA1}\frac{\delta OA1}{\delta O1}\frac{\delta O1}{\delta HA1}\frac{\delta HA1}{\delta H1}\frac{\delta H1}{\delta W1}\right) + \left(\frac{\delta e}{OA2}\frac{\delta OA2}{O2}\frac{\delta O2}{\delta HA1}\frac{\delta HA1}{\delta H1} \tag{3.1}$$

$$= \frac{\delta e}{\delta OA1} = \delta\left(\frac{1}{n}\frac{\sum_{i=1}^{2}(y-OA)^2}{\delta OA}\right) \tag{3.2}$$

in the above equation, y is derived output, whereas OA is the obtain output and total expression is known as total error.

$$= \delta\frac{1}{n}\frac{[(yi-OA)^2(y2-OA)^2]}{\delta OA1} \tag{3.3}$$

$$= \frac{1}{n}\frac{\delta(yi-OA1)^2}{\delta OA1} \tag{3.4}$$

$$= \frac{1}{n}[yi-OA1](-2) \tag{3.5}$$

$$= \frac{-2}{n}(yi-OA1) \tag{3.6}$$

from the output nodes bounces to the first activation node of the last hidden layer.

$$= \frac{\delta OA1}{\delta HA1} = \delta\frac{HA1W7+HA2W9+B2W11}{\delta w7} \tag{3.7}$$

lastly take the sum of the product of the individual derivatives to calculate the formula for the specific weights.

$$\frac{\delta e}{\delta e7} = \frac{\delta e}{\delta OA1}\frac{\delta OA1}{\delta oA}\frac{\delta O1}{\delta W7} \tag{3.8}$$

Using weight update equation.

$$w1^{k+1} = w1^{k} - \eta\frac{\delta e}{\delta w1^{k}} \tag{3.9}$$

As we know that

$$\mathrm{E} = (\mathrm{T} - \mathrm{y})$$

Keep working on this error until the error becomes 0 or near to 0, and use this for classification[45].

$$\theta = \sum yiwi \tag{3.10}$$

$$\frac{\delta\theta}{\delta y} = wi \tag{3.11}$$

$$\frac{\delta\theta}{\delta wi} = yji \tag{3.12}$$

$$\frac{\delta E}{\delta yj} = \frac{1}{2}[2(t-y)(-1)] \tag{3.13}$$

$$\frac{\delta E}{\delta y} = -(tj - yj) \tag{3.14}$$

$$y = \frac{1}{1 + e(-\theta)} \tag{3.15}$$

$$y = \left(1 + e(-\theta)^{-1}\right) \tag{3.16}$$

$$\frac{\delta y}{\delta \theta} = (-1)\left(-e^{-\theta}\right)\left(1 + e(-\theta)^{-2}\right) \tag{3.17}$$

$$\frac{e^{-\theta}}{(1+e^{-\theta})^2} \tag{3.18}$$

$$\frac{1}{1+e^{-\theta}}\frac{e^{-\theta}}{1+e^{-\theta}} \tag{3.19}$$

$$= y\frac{e^{-\theta}}{1+e^{-\theta}} \tag{3.20}$$

$$\frac{e^{-\theta}}{1+e^{-\theta}} = \frac{1+e^{-\theta}-1}{1+e^{-\theta}} \tag{3.21}$$

$$= \frac{1+e^{-\theta}}{1+e^{-\theta}} - \frac{1}{1+e^{\theta}} \tag{3.22}$$

$$= (1 - \text{y})$$

$$\frac{\delta y}{\delta \theta} = y(1-y) \tag{3.23}$$

Find the change in error w.r.t change in transfer potential.

$$\frac{\delta E}{\delta \theta j} = \frac{1}{2}(tj - yj)^2 \tag{3.24}$$

$$\frac{\delta E}{\delta \theta j} = \frac{\delta E}{\delta y}\frac{\delta y}{\partial \theta_j} \tag{3.25}$$

$$= -(tj - yj)\frac{\delta Yi}{\delta \theta j} \tag{3.26}$$

$$\frac{\delta E}{\delta \theta} = -(tj - yj)yi\left(1 - yj^{i}\right) \tag{3.27}$$

$$\frac{\delta E}{\delta WHi} = \frac{\delta E}{\delta \theta_j} \frac{\delta \theta_j}{\delta u} \tag{3.28}$$

$$= \frac{\delta E}{\delta \theta} \eta \eta^{\circ} \tag{3.29}$$

$$\frac{\delta E}{\delta w} = -y\dot{y}yj(1 - yi)(tj - yj) = \delta w \tag{3.30}$$

3.1.4 Need Of BackPropagation

Backpropagation method is popular in many ways some of them are given below:

- Backpropagation technique is comparatively fast, simple, easy to handle, and process technique.
- It does not need to tune every parameter of the network just needs to tune the numbers of inputs.
- When the system does not need to look back and have prior information about anything then the study come up with a very flexible modeling same case with backpropagation which is a flexible network so it does not require prior knowledge about anything in the network.
- Backpropagation is a standard method to minimize the error function that generally works excellent.
- No special selection of the features which the system needs to learn[46].

3.2 Globally Convergent Algorithm

The main algorithm is a globally convergent algorithm that is basically based on the resilient-backpropagation but without weight backtracking and further modifies the learning rate, there are only two options for learning rate either the learning rate attached with the smallest absolute gradient (sag) or sometimes the smallest learning rate (SLR) itself[47]. The learning rates depend on their predefined boundaries in the learning rate limit. The globally convergent algorithms are used to maximize the likelihood of the models or functions and help to minimize the error rate[48]. Firstly, the study will discuss Resilientpropagation and then further methods.

3.3 Resilient-Propagation

Resilient-prop was made in 1992 by Martin and Heinrich. It was a famous gradient descent algorithm in which the network needs only the signs of gradients to do computations and updates. It has the ability to adapting the step size dynamically and works on every weight independently[49].

Resilient-propagation working can be done through weights tracking and also it can be done without weights tracking. There is no huge difference between them but sometimes system face difficulty in minimizing the error rate so one of the best techniques to minimize the error is to adjust the weights for which there are also some techniques but if the system is working resilient without weight tracking so it will be difficult to find the local minima. The resilientpropagation which are having weight tracking is also called the resilientpropagation positive, it only depends on tuning the weights not other parameters and the same when there is no weight tracking is known as the resilientpropagation negative, in this only other parameters like gradient and rate will be update, both have equal importance but study prefer to work with resilient-propagation with weight tracking because it has various techniques for updating weights.

In the given figure3.1this is the same functioning as it is discussed in the backpropagation method, same it has input neurons, activation function then hidden layers and in last output but here is the difference is R-prop. with or without weight tracking.

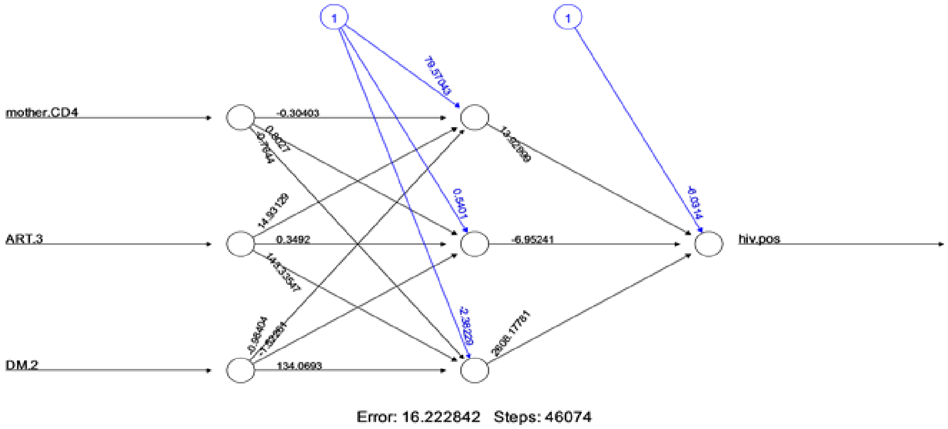

Figure 3.1: This figure illustrates the structure of resilient propagation

Somehow the backpropagation and the resilient-propagation are similar to each other but the backpropagation method has a fixed learning rate so to minimize the error rate the network only can do the adjustment of weights. Because of the parameter and when in the process of the backpropagation the weights did not adjust in a good manner and showing in steepest descent direction means opposite or negative to the gradient, so it effects from the flaw of local minima and the convergence rate slows down[50]. So then move to the resilientpropagation method to cover up the error rate of backpropagation. sometimes it updates the weights to find the local minima for the error function.

In R-prop the main ingredient is gradient descent which is used to train the neural networks and different machine learning models. Mostly when the system uses the gradient descent variants, they work with the sign and the magnitude of that gradient[51]. The gradient points mostly work in the direction of the steepest trek. Because the network mostly wants to find a minimum, when moving the gradient in the opposite direction we get a minimum. But the thing is the direction of the gradient points depends on the sign of the gradient. The resilientpropagation is well known as Rprop which has various variants like RProp-positive, RProp-negative, IRProp-, and IRProp-positive[52]. There are some good chunks of resilientprop. which are mentioned below:

- Training of the network is easier.
- No need to specify any parameter.
- More accuracy.

3.3.1 Process

Resilientpropagation is basically the extension of the backpropagation so the whole process is the same for r-prop but in the last step when the system has to update the weights then:

$$w(t) = w^{(t-1)} - \eta^{(t-1)} \cdot (sign)\frac{\partial E^{(t-1)}}{\partial w(t-1)} \tag{3.31}$$

In this step the partial derivative sign of the error w.r.t the respective weight is evaluated. Backpropagation simply keeps the gradient descents and measures them by learning rate. Then the measured gradient descents are directly applied to those weights. In R-PROP for every weight it takes an individual value but only uses the sign of that gradient descent to rise or drop the individual value. Then that value is applied directly to the weights. There are some points on how to increase or decrease that value:

- When for the two successive iterations the derivative w.r.t weights of the model have the same sign, then the updated value will be increased.
- From the last iteration if the derivative with respect to the weight changes sign, then the updated value will be decreased.
- If the derivative value comes to zero, then the updated value does not change it remains the same.
- When the weights keep changing, then the change in weights will be reduced.

- For many iterations if the weights continue to change in the same way, then the magnitude of the weight change will be increased.

3.3.2 Types

There are different scenarios for resilient propagation sometimes it works with weights and sometimes it does not. In resilient the network considers the signs which depend on weights, gradient, and parameter, etc.

- **Resilient Propagation With Weight Tracking**

  As the study mentioned earlier sometimes it works with weights tracking, in the network, they check the error rate and go back to adjust the weights to minimize the error and find the local minima point.

- **Resilient Propagation Without Weight Tracking**

  Sometimes to find the local minima the network only works with gradient and other parameters to reduce the error rate. It does not track the weights to check the local minima point[53].

3.3.3 Resilient Propagation Working

First thing about rpropagation is, that it has more complexity as compared to the backpropagation but the best part is it gives efficient results and speedy training work. Resilient propagation (Rprop) at some platforms is called resilient backpropagation because of its similarities to the most common algorithm for training a neural network which is the backpropagation method. The only difference between resilient and backpropagation is the training of the network with Rprop is way faster than the backpropagation training and resilientprop doesn't need to define or explain any free parameter value as compared to backpropagation in which the network needs to specify some values like the learning rate and sometimes an optional momentum term[54].

3.3.4 Gradient Magnitude

For resilientpropagation there is a need to decide what will be the step size means which will be the scaled version of gradient magnitude and which version will be used by most the gradient descent algorithms. There are various versions of gradient's magnitude one of them is heuristic which works well but there is not a full surety of work sometimes it is not a good choice[55]. When the study has the same optima function then these functions should be similar in step updates by using gradient descent but in the case of gradient's magnitude, the network works differently, the step sizes of function depending on the order of magnitude. So basically gradient's magnitude is not a good approach for determining the step size because it does not hold the important and full information about the step size. Learning rate is supposed to be a fixed in backprop. but in rprop. sometimes it is not a good way to handle the network. If the network measured some patches of the function with a fixed learning rate then it could also fail[56].

3.3.5 Weight Adjustment

The problem of selecting the step size was a major one in many cases so modern variants of gradient descent tried to get around this issue by dynamically selecting or adapting the step size. The modernization is depending on the sign of the gradient, the sign will decide what will be the best choice for the network and just need to ignore the magnitude of the gradient. There will be no issue in the R-Prop network if the function has steep places. R-Propagation has different step sizes for every dimension. For example, we suppose $\eta(t)$ is the step size of some random ith-weight in the ith-iteration of gradient descent. $\eta(0)$ is the value of the first iteration and same as $\eta(1)$ is the second iteration value. These values of iterations are hyper-parameter that needs to be selected in advance. Depending on the gradient the step size is then dynamically adjusted for every weight[57]. The weights are updated by themselves using the following equation

$$w(t) = w^{(t-1)} - \eta^{(t-1)} \cdot (sign) \frac{\partial E^{(t-1)}}{\partial w(t-1)} \tag{3.32}$$

Here in the last step the partial derivative sign of the error w.r.t the respective weight is evaluated. By using the help of a defined step size the network moves in direction of descent.

3.3.6 Adapting The Step-size

The algorithm is to compute the gradients in every iteration or cycle of resilient-propagation and for every dimension, update the step size individually. In this, compare the signs of the gradient of current and a gradient of the previous iteration[58]. The main criteria of the network for adapting the step size are the following:

- When the signs of gradients are comparing, the sign of current and the previous gradients are the same then the network moves in the direction of the previous gradient's direction. Hence the network is following the same direction as the previous one so this seems a good approach in which the step size is increasing and achieve the local optimum point more quickly.
- When the network faces the opposite signs like if the network has a different previous gradient's sign and the current gradient's sign is different then it will move in an opposite direction which means that it immediately passed the local optimum point. Therefore, step size would be decreased to stay away from passing over the optimum point again.
- And if the gradient approaches zero then for this weight the local optimum could be found and step size could not be changed.

3.3.7 Hyper-Parameter

The hyper-parameter that the network needs to choose is a question but in reality, there are already well-known values present for them that work excellent. In resilientprop there is a concept of clipping step sizes which are η mini and η maxi, these are used to avoid the situation of gaining too large a unit or way too small[59]. If there are clipping values, that are smaller and bigger than the wanted or somehow necessary values it won't be problematic for the network because the temporary step size generated immediately.
Where α scales the step size which will be greater than 1 and β scale step size should be less than 1, based on the speed situation means will increase or not increase. The most frequently written value for α is 1.2 and for β is 0.50. Basically, this helps well to increase step size gradually, and same if there is a possibility of immediately decreasing trend is when passing around the local optima point.
If there are various step sizes then it will not be a problematic situation for the network and if the network wants fine-tuning of the weights then it is the main point that β values are not the reciprocal of the value of α[60].

3.3.8 Explanation

When the topic is about convergence speed, validity w.r.t the parameters training and also the accuracy point then resilientpropagation is a great algorithm choice. Basically, resilientprop is an algorithm of local adaptive learning, the main idea of rprop. is to remove the bad and harmful influence which happens just because of the partial derivatives of weight step sizes. As mentioned before, about the backpropagation so, compared to the backpropagation the resilient-prop convergence is faster and does not need more or high training. Because of the advantages of resilientprop, nowadays many companies are taking benefits from this algorithm for their purposes. One of the main objectives to use is the study has various step sizes for each weight.

If it runs the network and get to know about one of the weights which are already closer to its optimal value while for the other weight there is still a need for many changes, then it will not be an issue for ResilientProp. But if the system deals with different gradient descent variants then it could be more complex to handle the network in this situation, especially when the dealing is with gradient magnitudes which can be a misleading situation here. Resilientprop has many advantages but with advantages, have to face some disadvantages as well one of them is network needs large batch updates which means rprop does not excel in all situations. Another disadvantage of this algorithm is if stochastic gradient descent (SGD) has more randomness than the issue of step size arrives, it jumps around way too much and badly influences the updates.

3.3.9 Advantages Of The Resilient-propagation

It is most frequently used method because of its speedy computations. It has many advantages over other methods but it feels good to use this method over the back-propagation method because of these advantages:

- In the resilientpropagation method, the training of the network is faster than the training of backpropagation.
- Resilient-prop doesn't need any specification for any free parameter value but in comparison to back propagation, the back-prop method requires values for the parameter which is the learning rate.

3.3.10 Disadvantages

There are always some pros and some cons of anything same this resilient method is good but also has the following disadvantages:

- The implementation of the resilient algorithm is difficult as compared to backprop. implementation.

- The computation of the algorithm is a more complex process than backpropagation.

3.4 Smallest Learning Rate (SLR)

Neural networks always need to train by using different training algorithms like backpropagation, feed-forwardpropagation, resilientpropagation, and so on. The neural network is all about backpropagation because every other algorithm starts with the backpropagation. The backpropagation generally needs more time when data is on large scale and also for complex problems[61]. These kinds of algorithms usually have a tuning parameter, for backpropagation, it is the learning rate parameter that decides how much change can be made in weights in response to an observed loss on the training set. Many hyper-parameters are used to tune or build the training of different neural networks. One of them is "learning rate" which is a very important parameter so the selection of the learning rate can have a huge effect on the speed of training and also generalization accuracy. There are different techniques or systems which tells different learning rates like large learning rates, small learning rate, and cyclical learning rates. With a large learning rate, the model learns the system faster but when the learning rate is smaller it takes more time to converge but gives more optimal solutions than a larger one. And in the cyclical learning rate, the learning rate moves from one value to another value to find the best place where they get the most optimal solution in less time and with adjusted weights, within the band of values but this is not a good technique because there is a confusion, may be the system will get the right one in starting or in last[62]. Most of the networks who have worked with these kinds of training algorithms has been gone through the issue of selecting the learning rate, but the study has rarely much guidance on which value is suitable, and what value is good because the study also showed that the best value for learning parameter depends on the work.

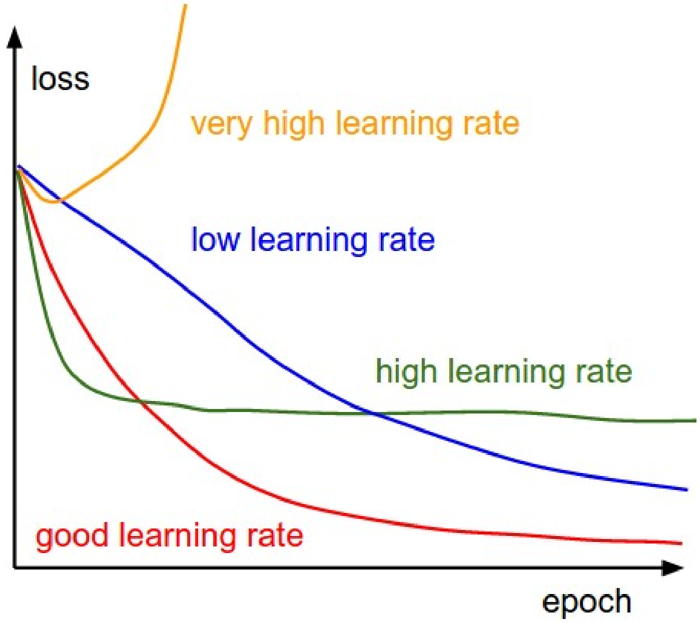

Figure 3.2: This figure illustrates different learning rate behaviour

In figure3.2 different points define different coloring lines are having different learning rates. The orange line shows the very high learning rate but the green line shows the less high learning rate as compared to the orange line the blue line shows the low learning rate and in last red line is best on learning rate it gradually decreases when then epoch size increases and loss becomes less. It can be seen as loss decreases with the increasing epoch size and the good learning rate system gets. Mostly, most neural networks like to work with the largest learning rate which allows convergence and also speed up the training part[63]. But since there is no big evidence or scientific research about this so human can use different learning rates and at which network gets more accuracy that should be good but just for that algorithm.

Many algorithms are present for tuning the learning rate parameter automatically but most of them algorithms only concentrate on upgrading or enhancing the quality of convergence means to improve the speed and for that, they forget to concentrate on generalization accuracy. It controls the weights of network w.r.t loss gradient so the selection of learning rate has huge effect. Whenever the data is on large scale and have complex problems, the high learning rate affects the generalization accuracy but speed up the training. if the learning rate is not high means small enough, without any additional improvement and upgrading in generalization accuracy the additional cutting in size throws away computational resources. It slows down the speed of training then the computation takes more time and also money. There are different training patterns like online training in which after the modeling or presentation of every training instances the weights are updated. Another is batch training which is opposite to previous training like in this weight changes but these changes accumulated and the application worked at the end of the process. But the drawback of batch training is it needs more time for training than on-line training time[64].

3.4.1 Process

To find the best learning rate these are following given steps:

- Find the learning rate with the smallest total sum of squares.
- Continuously train the network by using that learning rate until the selected accuracy starts to decline.
- In the next step, the network has to observe the maximum generalization accuracy for that learning rate and also save the accuracy.
- To get the smallest learning rate where the accuracy is near perfect, network should gradually decrease the learning rate by a constant factor and then, again and again, train the network.
- Try the last two steps until obtain the maximum accuracy which stops to upgrade and improve.

$$w(t) = w^{(t-1)} - \eta^{(t-1)} \cdot (sign)\frac{\partial E^{(t-1)}}{\partial w(t-1)} \tag{3.33}$$

Then use that learning rate in given equation to update the weights.

3.4.2 The Learning Rate Effects On Training Speed And Generalization Accuracy

When the network is working and to find the parameter which minimizes the loss function, the system go for the algorithm gradient descent learning, then at the present place in weight space it calculates the error gradient, then the weights will change in the opposite way of a gradient in an aim to minimize the error. Sometimes the gradient may tell the network or indicate in which direction the weights should have to go. But it does not conform and is a safe thing to say, before the error changes its pattern like quits the pattern of decreasing trend and converts into the increasing pattern, specifically how long the weights can move safely in that direction. So it could be like if a learning rate that is too high often goes too far in the positive direction means the correct direction, thus affecting the accuracy. Because of all the effects, it takes more time to train a learning rate which is too large, because of its continually overshooting problem of objective and the reverse process means unlearn the pattern which it already has been learned, so, the need for expensive and rich backtracking[65]. This instability situation causes bad generalization accuracy since before jumping back again the weights of the network can never be organized or set enough to go to the minimum. If the learning rate is not high and small enough to keep away such over-corrections, it leads the process in a comparatively smooth path by the error loss landscape, after all, settling in the minimum. If the Reduction of the learning rate goes more then it can make the path more and more smooth, and this can lead to significantly improving the generalization accuracy. So after reducing the learning rate so many times there comes a time where the reduction of the learning rate any more is simply a waste of time, comes in taking several more steps to get the same path with the same minimum than necessary steps[66].

3.4.3 Selection Of A Learning Rate

There are different types of learning rates and it is very important to choose the right learning rate.

- The learning rate should not be too large.

  Learning rate value can not be too large because it is not a good approach for a good method.

- The learning rate should not be too small.

  The learning rate value can not be too small for the method because when there is too small learning rate then there is a need for many updates for the rate which is not good. After all, it takes more time.

- The learning rate is cyclical In this the learning rate moves from one value to another value to find that place where they will get the most optimal solution and with well-adjusted weights but in less time and also within the band of values.

If the learning rate gradually decreases then the generalization accuracy upgrades and improves the quality and also upgrades the training speed. If continue to the decreasing pattern of the learning rate then the accuracy continues to upgrade slightly but the speed of the training network starts to get worse once again. Finally, there comes a point at which accuracy flattens out and does not improve with a smaller learning rate, and further reductions in the learning rate only waste computational resources. If the network has a large learning rate then the accuracy will be affected, which will give us poor accuracy and also affect the training speed. It concluded in different studies that the learning rate should be optimal, not too high or not too low. The optimal learning rate is so far the best one because it does not take more time for convergence and also gives the accurate solutions.

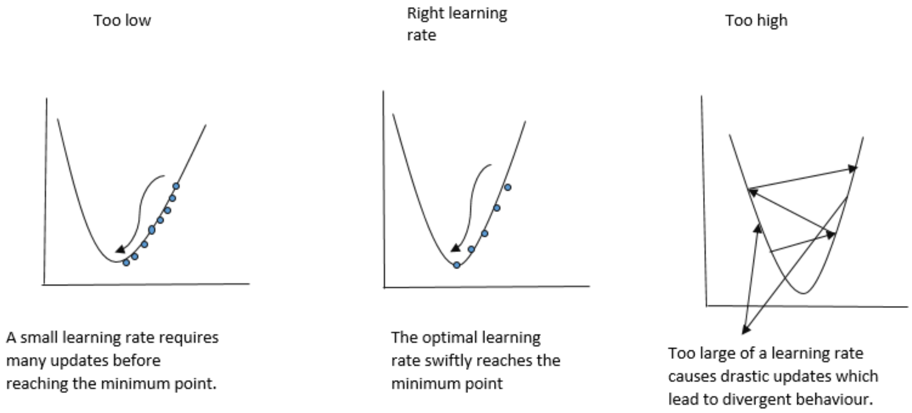
Too low
Right learning rate
Too high
A small learning rate requires many updates before reaching the minimum point.
The optimal learning rate swiftly reaches the minimum point
Too large of a learning rate causes drastic updates which lead to divergent behaviour.

Figure 3.3: This figure illustrates different learning rate convergence

The given figure3.3 shows the different learning rate graphs. In right graph where it has a large learning rate, it shows very drastic updates so it has weird convergence. The another graph in which the learning rate shows a very small value there is a need for many updates for the learning rate to get the optimal point but in the last situation where the learning rate is optimal means the almost right learning rate there we smoothly get the optima point. There is no need for many updates to get a feasible solution or obtain the local minima point.

After seen all the rates the obvious question arises, is how to choose a learning rate that is small enough to achieve good generalization accuracy without wasting computational resources. There are some useful techniques for the selection of the learning rate in a network:

- Grundwald's Safebayes Method:

  In 2017 Grunwald and van Ommen noticed that there is always a need for parameter and hyper-compression, for those models which are non-convex and misspecified.

- Lyddon Et Al. Method:

This method is obtained by the Newton and Raftery approach which was purposed back in 1994 and it was the weighted likelihood bootstrap approach(this is used in the Bayesian framework). It was generally for the making of bootstrap samples that have the same asymptotic distribution.

- Syring And Martin Method:

  In 2019 this method was shown for the tuning of the learning rate to obtain the nominal frequentest coverage probability[67].

3.4.4 Avoid Large Learning Rates

Firstly, need to know how to watch out for learning rates that are way too large. How the training of those learning rates is slower than a learning rate which is smaller and also in accuracy how they are less accurate than smaller one. This can be seen by observing either on a keep-off set the generalization accuracy, more suitably the system can also examine it by checking the total sum squared (TSS) error which was calculated during the process of training. Suppose the system sets the learning rate at 0.05 and if the learning rate comes larger than 0.05 then network has to discard that learning rate from further computation and process. Because this learning rate is slower and gives less accurate responses, and thus has no plus point over the "fastest" and more accurate learning rate. If system wants to avoid learning rates that are way too large, then algorithm has to train the set of neural networks just for one iteration by using the learning rates but with the same initial weights at different orders of magnitude and also for every network with the same order of presentation. If system got less TSS error of any learning rate then it will consider that learning rate fastest and also set that as a base and compare other learning rates with that learning rate. If any of them get larger then discard that learning rate.

3.4.5 Smaller Learning Rates Advantages

If any of the learning rates are the fastest then that is also the most accurate one but this approach is not true, the fastest learning rate does not always give most accurate responses. The advantage of a smaller learning rate is if the system is working with complex problems and with large-scale data, the learning rate that is smaller than the "fastest" one learning rate often concludes in the most amount of generalization accuracy, so it is usually worth and rich in addition to upgrade and improve the accuracy, spending further more training time using small learning rate.

3.5 Smallest Absolute Gradient (SAG)

A gradient is an algorithm that is used for the training of machine learning models and neural networks. These training methods help the models to learn the algorithm over time and within the gradient, the cost function acts as the barometer and computes the accuracy with every iteration and update parameter. In optimal solutions there is a need to smooth out the data functionality so for that system uses different gradient decent. system has to achieve an optimal point, and an absolute gradient which gives more accuracy, optimal point and least error rate. Gradient decent is the most frequently used algorithm for the training of machine learning models and neural networks. When the system is working with this optimization algorithm, the algorithm works until the model function is close to zero or equal to zero.It continuously updates the parameters within the model to get the smallest possible error.

The given figure3.4 shows that in graph loss is present horizontally and vertically it has a value of weight. In this, the point from where it started the network, and then gradually it comes closer to the minima point and shows the point of convergence where it gets optimally position means the error rate is near to zero, can be seen.

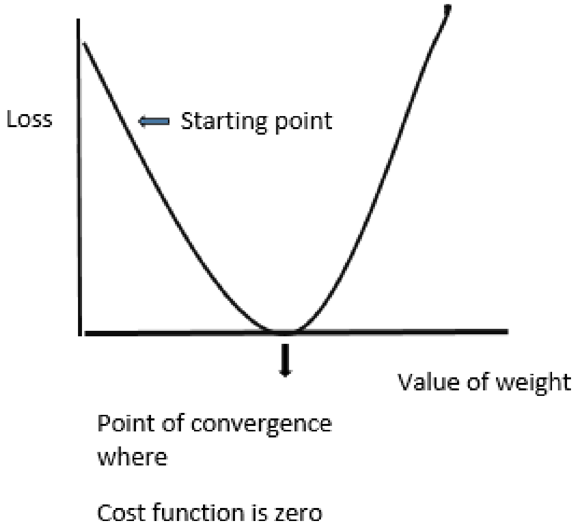

Figure 3.4: This figure illustrates the gradient behaviour

The gradient which gives the near optima or optimum point is so far the best. Once the model is fully optimized for accuracy, it can be the most powerful tool for computer science projects and artificial intelligence(AI) applications[68].

3.5.1 Gradient Types:

Gradient is very important for training. It has different types:

- Batch Gradient Descent:

  For every node, it collects the error in a training set but after the whole training it only updates the model once. This process started as a training epoch[69]. While this method provides efficient computations, at the same time for large training data sets it still has a long processing time as it still has to store all the data into the memory[70]. Because the system is doing all this just for a stable error rate means equivalent to zero or close to zero so batch gradient descent is also the source of a stable error gradient and convergence, but sometimes it gets the convergence point that is not the most ideal convergence point and searching the local minimum point versus the global point[71].

- Stochastic Gradient Descent (SGD):

  In the SGD method, within the data set it runs a single training epoch for each example and also it updates at each epoch each training parameter of every task one at a time[72]. Since the system has to hold only one training example because it is more convenient to store in memory. Comparative to the batch gradient decent these continuous updates give more detail and speed but in the output it losses the computational efficiency[73]. Gradient's frequent updates give the noisy gradients, but it is not that bad because it can help us in run away through the local minimum and find the global one point[74].

- Mini-Batch Gradient Descent:

  It has more properties because it is the combination of the concepts of both batch gradient descent and stochastic gradient descent[75]. It divides the whole training data set into small batch sizes like small groups and performs the function on each batch and updates each one. It is a faster process to find the optimal solution and convergence[76]. This approach gives a balancing behavior between the efficient computations of batch gradient descent and the stochastic gradient descent's speed[77].

if the network has non-smooth data where it has to optimize the model so there are some techniques that are use for smoothing the network[78].

3.5.2 Standard Methods For Non-Smooth Optimization

To see the quality of different techniques of machine learning some methods are mentioned here which are available to optimize the non-smooth dual functions[79]:

- One is a simple sub-gradient technique
- The other is a simple sub-gradient-level technique and the last is incremental sub-gradient technique.

3.5.3 The Simple Sub-gradient Method:

This method consists of a simple and easy algorithm to minimize the function of non-differentiable convex[80]. This method is somehow similar to the ordinary gradient's method which is for differentiable functions, but in this, there are some notable exceptions. Every relaxed problem can get an optimized solution through this[81].

3.5.4 The Simple Sub-gradient Level Method:

The motive of using the sub-gradient method is convex optimization which draws projections on the successive estimation of level sets which corresponds to the main idea of optimal value[82]. The system makes the whole sample of convergence and efficient estimation for simple level controls in which they do not need a feasible set to be condensed[83].

3.5.5 The Incremental Sub-Gradient Method:

This is one of the classes of sub-gradient methods which is used for minimization of a convex function that has the sum of big numbers of component functions[84].

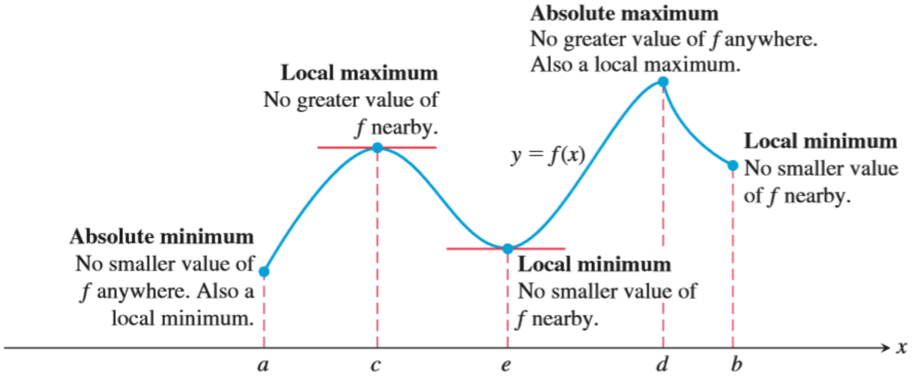

Figure 3.5: This figure illustrates the different positions for gradient

The given figure3.4 shows the positions of absolute minimum, local maximum, absolute maximum, and local minimum. we always desire to find the local minima point because that point gives the best accuracy and less error rate.

When the system is dealing with the lagrangian relaxation method for coupling constraints of big-scale separable problems then this kind of minimization comes in a dual context. In this making, each sub-problem can be solved to get optimality condition and when each problem gets optimization then update multipliers and also step sizes but the updating formulation of this method is similar to previous one mean similar to a simple sub-gradient level method[85].

Once the significant oscillations are indicated, condition full-filled then k is reset to back at 0. To see the fair and best comparison of methods, at each iteration only one sub-problem is being solved exactly for once.

Chapter 4

Data Explanation

4.1 Data Source

The data we used in this research is "German Traffic Sign recognition benchmark"(GTSRB) data we took from the Kaggle website. https://www.kaggle.com/data german-traffic-sign

Data consists of different images of traffic signs from different angles. The traffic signs were like the speed limit, parking lot, turn a head, and more.

4.1.1 Pre-Processing

The images which we used from the German traffic sign benchmark data set were not clear, the sizes were different for different signs, some of them were not visible, some of their colors faded away, and some were blurred so we decided to rescale the data images, and then all images were at the same size. The rescaling process was done by R studio.

The given figure4.1 shows the different traffic signs which we used to see daily but really do not know what is the meaning of that sign. There are different signs like speed limit signs, no turn, turn right or left, one-way traffic, and many more.

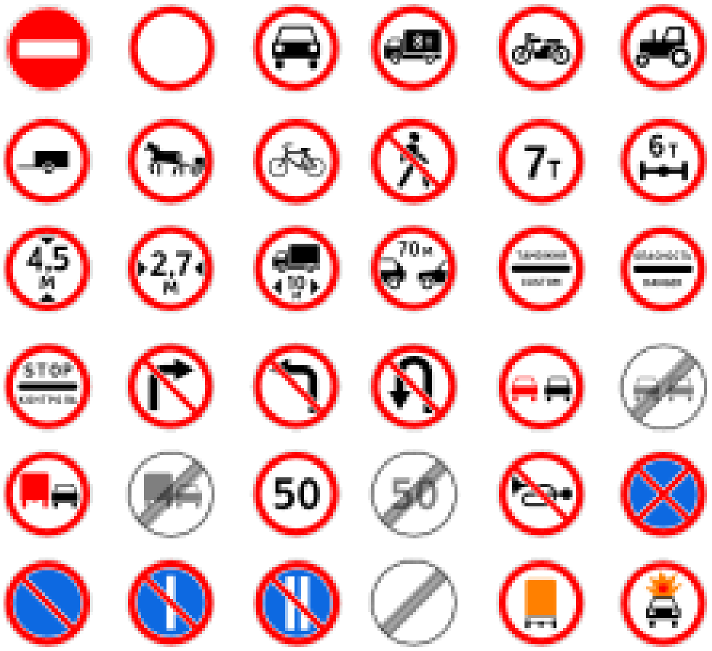

Figure 4.1: This figure illustrates different traffic signs

4.2 Computation

R studio is used for computations statistical analysis, modeling, and also for graphical representation of data. There are several packages for training the neural networks for different algorithms like nnet package, neuralnet, neuralnet tools, and neuralsens. We used a neuralnet package to train our network. We train our network with five different algorithms. We used backpropagation, resilient propagation with weight tracking and without weight tracking, smallest learning rate (SLR), and last smallest absolute gradient(SAG). We train each method thirty times and find the calibrated and validated values at a different number of layers and a different number of hidden layers. Each algorithm showed different calibrated and validated values with different layers and hidden layers.

This table shows the most calibrated, least calibrated, and same most validated and least validated values with a different number of layers and hidden layers in backpropagation. From this, we can see that the most val-

Table 4.1: Backpropagation calibrated and validated values

	Most calibrated	Least calibrated	Most validated	Least validated
Layers	1	2	3	3
Hidden layers	7	5	5	5

idated and least validated values came with the same number of layers but the most calibrated and least calibrated values came with a different number of layers and hidden layers. so we do not fix the layers for any algorithm. We can get accuracy for an algorithm at any layers and hidden layers so we should try our method with different numbers of layers and hidden layers and choose that which gives the best accuracy.

Table 4.2: Resilientpropagation +ve validated and calibrated values

	Most calibrated	Least calibrated	Most validated	Least validated
Layers	3	2	2	2
Hidden layers	5	4	6	8

In this table we see the resilient+ least calibrated, most validated, and least validated values shared the same number of layers but hidden layers are changed for every item. That means we can get the most and least values at the same number of layers.It is not a good approach to have exactly these number of layers and hidden layers for another algorithm and data so that you will get the again most calibrated values at these points. Everything is different in this case.

Table 4.3: Resilientpropagation -ve validated and calibrated values

	Most calibrated	Least calibrated	Most validated	Least validated
Layers	3	1	3	3
Hidden layers	2	9	2	5

The same thing is happening in this table when we used resilientpropagation-ve then the most calibrated, most validated, and least validated values are coming with a same number of layers. So if we want to get local minima and reduce the error rate the one option is to train your model with a different number of layers and hidden layers. Because at different layers you get different accuracy and error rate so choose the best out of them all.

Table 4.4: Smallest Learning Rate calibrated and validated values

	Most calibrated	Least calibrated	Most validated	Least validated
Layers	4	3	4	3
Hidden layers	6	7	6	7

In this table when we are working with the smallest learning rate algorithm then we got all the most values and least value means most calibrated, most validated, and least validated and least calibrated with a same number of both layers

Table 4.5: Smallest Absolute Gradient validated and calibrated values

	Most calibrated	Least calibrated	Most validated	Least validated
Layers	2	1	3	3
Hidden layers	8	2	8	5

Same happened here when we are working with smallest absolute gradient we got the most and least validated values at a same number of layers but different hidden layers and remaining points got at different numbers of layers so we can say that selection of a number of layers and hidden is also an important task while we are training any neural network.

Chapter 5

Analysis And Results

The neural network algorithms were applied to the German traffic sign benchmark data and the algorithms of the neural network showed us the average calibrated and validated value of the network with different numbers of layers and hidden layers and also conveyed us the idea of the best classification method for the traffic signs. if we have a look at the average values of calibration and validation of the algorithm's training of network for classification of traffic signs then we came to know the SAG is the algorithm that gave us the highest calibrated value, SAG gives us the high classification accuracy. The calibrated value was 87.73 by SAG, which is a good accuracy point. and when we talk about the least calibrated algorithm then SLR gave us the least calibrated value. The accuracy of SLR is not that good. The backpropagation algorithm gave us the good results like its results are close to the SAG results which means the backpropagation methodology and the SAG methodology both can be used for the classification of the German traffic signs and we will get above 80 percent accuracy.

The given graph5.1 shows the calibration value of all the algorithms which we used in our work. Calibration value tells about the measurement accuracy of the system or model when it meets a known parameter means it meets the known standard value of the model. The horizontal axis shows the algorithms and vertically we represent the calibration values.

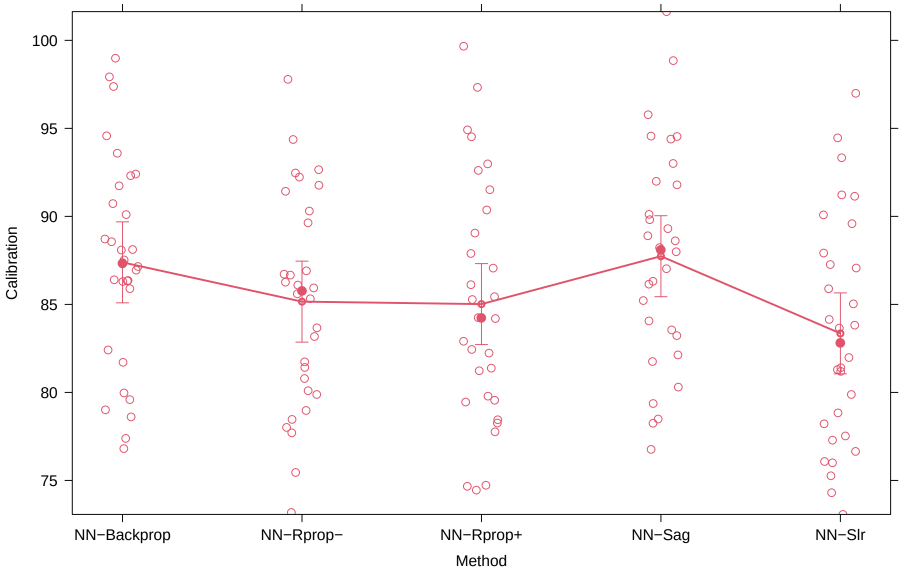

Figure 5.1: This graph shows the calibration of network

The first algorithm Backpropagation shows an 87.38 calibration value, Resilientpropagation –ve shows 85.15 value, R-prop. +ve shows 85.03, SAG represents 87.73 calibrated value, and the last SLR shows 83.34. Secondly, The Backpropagation gave us the best results which were 87.38 calibrated, the difference between SAG and backpropagation is a little bit so we can also consider the backpropagation as the best classifier for the traffic sign classification. Although we have good calibrated values for all the algorithms the graph shows the most calibrated points are in SAG which has the highest calibrated value which is 87.73 and also the other algorithms gave us the good calibrated values like 85.15 for resilientpropagaton -ve and for positive resilient it is 85.03, so both have over 80 percent values which is good. The lowest is 83.34 which is the SLR value. Individually this value is good but comparative to other values, it has least calibrated value.

The graph 5.2 shows the validation value of all the algorithms which we used in our work. Validation value tells about the validity of the function means when a system satisfies its defined functional intent.

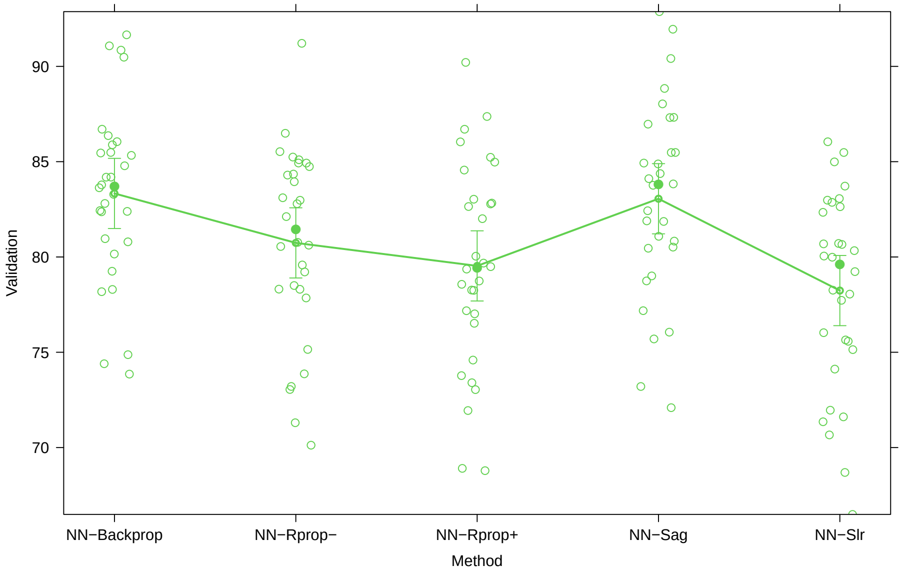

Figure 5.2: This graph shows the validation of network

It is the testing part of same data set from which we already obtained the training set. The horizontal axis shows the algorithms mean the methods and vertically we represent the validation values.

The first algorithm Backpropagation shows 83.32 validated values, R-propagation –ve shows 80.73 values, R-prop. +ve shows 79.52, SAG represents 84.05 validated value, and the last SLR shows 78.23. In this graph, the most validated point is also SAG which is 84.04 as we have seen most calibrated value is also from SAG, and the least validated method was SLR which has 78.23 value. In this we validated our calibrated results that we did right calibration for our dataset. As we are getting the same results for calibrated and validated values. It is not a problem that validated values are coming below than eighty percent but the only thing that matters, the most and least calibrated values and most validated and least validated values are coming from the same algorithm.

This graph 5.3 shows us that at a different number of layers they give us accuracy at different levels.

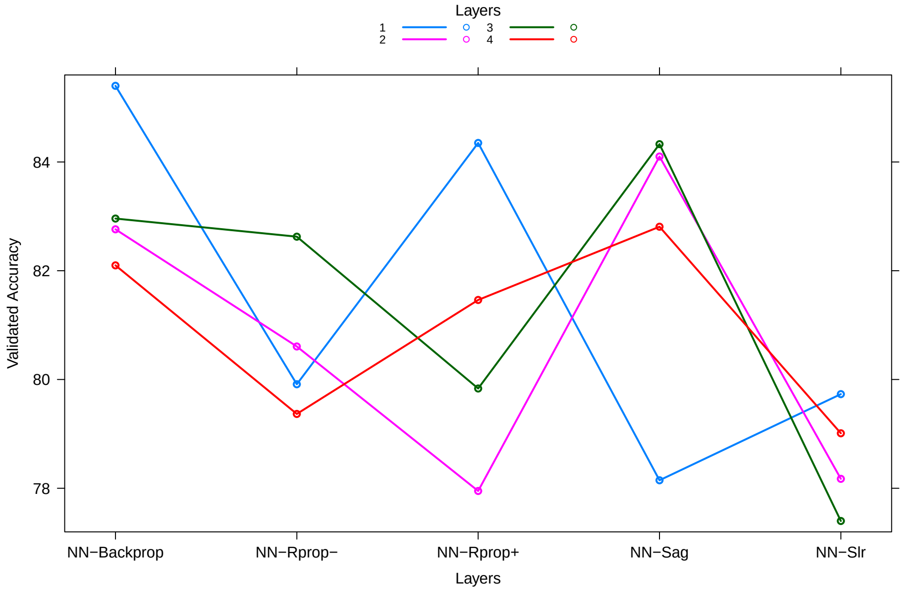

Figure 5.3: This graph shows the validation accuracy at different number of layers

In this, we represent the number of layers with different colors. The horizontal axis is again for methods and the vertical axis is for validation accuracy. The blue line shows that we have a single layer, the pink line shows the two layers model, for three layers we have a green color, and last for four layers we have an orange color line. As we can see from the graph when we have a single layer in our model then backpropagation gives us the best results and SAG gives us the lowest value at a single layer. At 2 layers SAG gives the best results and with a minimum margin with SLR, the resilient+ gives the worst values. At 3 layers again SAG was best and SLR was poor. AT 4 layers outcome was the same means the SAG was highest and the SLR was lowest.

The graph 5.4 shows the algorithm at different hidden layers how much they are classifying accurately. We are presenting at how many hidden layers an algorithm gives us validation accuracy. We do change the number of layers and hidden layers to find the best model where we get more accuracy.

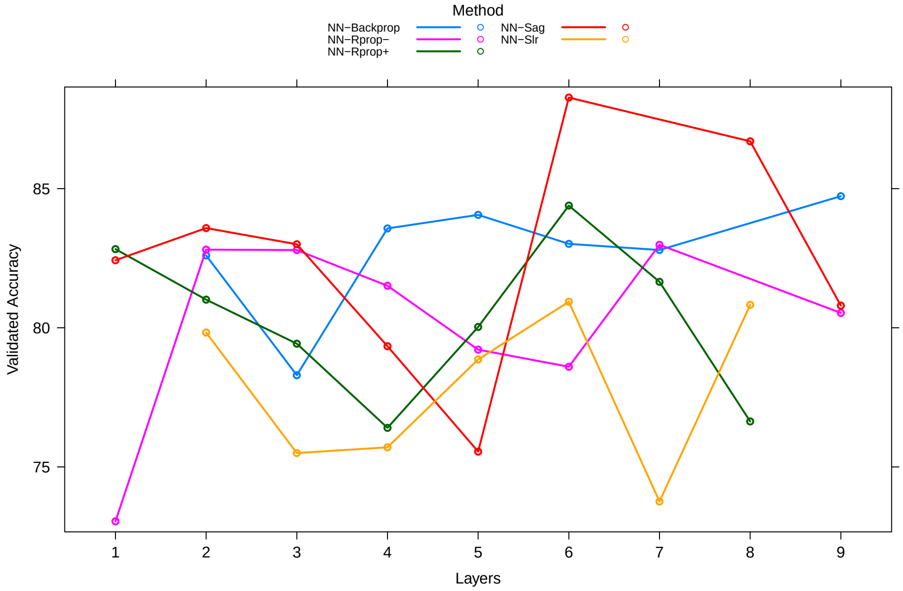

Figure 5.4: This graph shows the validation accuracy at different number of hidden layers

Here horizontal axis shows the number of hidden layers and different colors represents the different methods and vertically we have validation accuracy. In this back-prop. is represented by blue color, the pink for R-prop.-ve, the green for R-prop. +ve, for the SAG we had red and we choose the orange color for the SLR.

when the hidden layers are highest means 9 then back-propagation gave the best results. At 6 hidden layers, SAG was excellent. At 8 and 6 hidden layers the SLR gave the same results. At 7 hidden layers R-prop+ was best but at 6 R-prop- was best which means to say at different number of hidden layers the different algorithms shows the different results.

5.1 Abbreviation List

In this study there were different words. Sometimes abbreviation of words has been used. Here is the list for help to understand the words. ANN

Artificial neural network

ML

Machine learning

AI

Artificial intelligence

ADAS

Advanced driver assistance systems

CNN

convolutional neural network

RNN

Recurrent neural network

R-prop

Resilient propagation

SLR

Smallest learning rate

SAG

Smallest absolute gradient

SGD

Stochastic gradient decent

Bibliography

[1] De la Escalera, A., Armingol, J.M. and Mata, M., 2003. Traffic sign recognition and analysis for intelligent vehicles. Image and vision computing, 21(3), pp.247-258.

[2] Stallkamp, J., Schlipsing, M., Salmen, J. and Igel, C., 2011, July. The German traffic sign recognition benchmark: a multi-class classification competition. In The 2011 international joint conference on neural networks (pp. 1453-1460). IEEE.

[3] Lee, G.Y., Alzamil, L., Doskenov, B. and Termehchy, A., 2021. A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance. arXiv preprint arXiv:2109.07127.

[4] Mano, M.M., 1993. Computer system architecture. Prentice-Hall, Inc..

[5] Pannu, A., 2015. Artificial intelligence and its application in different areas. Artificial Intelligence, 4(10), pp.79-84.

[6] Choi, R.Y., Coyner, A.S., Kalpathy-Cramer, J., Chiang, M.F. and Campbell, J.P., 2020. Introduction to machine learning, neural networks, and deep learning. Translational Vision Science and Technology, 9(2), pp.14-14.

[7] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks, 61, pp.85-117.

[8] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.

[9] Shinde, P.P. and Shah, S., 2018, August. A review of machine learning and deep learning applications. In 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-6). IEEE.

[10] Mitchell, T.M. and Mitchell, T.M., 1997. Machine learning (Vol. 1, No. 9). New York: McGraw-hill.

[11] Bell, J., 2022. What Is Machine Learning?. Machine Learning and the City: Applications in Architecture and Urban Design, pp.207-216.

[12] Nasteski, V., 2017. An overview of the supervised machine learning methods. Horizons. b, 4, pp.51-62.

[13] Jiang, T., Gradus, J.L. and Rosellini, A.J., 2020. Supervised machine learning: a brief primer. Behavior Therapy, 51(5), pp.675-687.

[14] Gentleman, R. and Carey, V.J., 2008. Unsupervised machine learning. In Bioconductor case studies (pp. 137-157). Springer, New York, NY.

[15] Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.L.A., Elkhatib, Y., Hussain, A. and Al-Fuqaha, A., 2019. Unsupervised machine learning for networking: Techniques, applications and research challenges. IEEE access, 7, pp.65579-65615.

[16] Huang, X., Wu, L. and Ye, Y., 2019. A review on dimensionality reduction techniques. International Journal of Pattern Recognition and Artificial Intelligence, 33(10), p.1950017.

[17] Zhu, X. and Goldberg, A.B., 2009. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), pp.1-130.

[18] Zhou, X. and Belkin, M., 2014. Semi-supervised learning. In Academic Press Library in Signal Processing (Vol. 1, pp. 1239-1269). Elsevier.

[19] Qingsong, X., Juan, S. and Tiantian, L., 2010, April. A detection and recognition method for prohibition traffic signs. In 2010 international conference on image analysis and signal processing (pp. 583-586). IEEE.

[20] Paulo, C.F. and Correia, P.L., 2007, June. Automatic detection and classification of traffic signs. In Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'07) (pp. 11-11). IEEE.

[21] De La Escalera, A., Moreno, L.E., Salichs, M.A. and Armingol, J.M., 1997. Road traffic sign detection and classification. IEEE transactions on industrial electronics, 44(6), pp.848-859.

[22] Wu, Y., Liu, Y., Li, J., Liu, H. and Hu, X., 2013, August. Traffic sign detection based on convolutional neural networks. In The 2013 international joint conference on neural networks (IJCNN) (pp. 1-7). IEEE.

[23] Walczak, S., 2018. Artificial neural networks. In Encyclopedia of Information Science and Technology, Fourth Edition (pp. 120-131). IGI global.

[24] Krenker, A., Bešter, J. and Kos, A., 2011. Introduction to the artificial neural networks. Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech, pp.1-18.

[25] Gershenson, C., 2003. Artificial neural networks for beginners. arXiv preprint cs/0308031.

[26] Basheer, I.A. and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), pp.3-31.

[27] Zupan, J., 1994. Introduction to artificial neural network (ANN) methods: what they are and how to use them. Acta Chimica Slovenica, 41, pp.327-327.

[28] Dayhoff, J.E., 1990. Neural network architectures: an introduction. Van Nostrand Reinhold Co..

[29] Wang, S.C., 2003. Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.

[30] Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. towards data science, 6(12), pp.310-316.

[31] Elliott, D.L., 1993. A better activation function for artificial neural networks.

[32] Pratiwi, H., Windarto, A.P., Susliansyah, S., Aria, R.R., Susilowati, S., Rahayu, L.K., Fitriani, Y., Merdekawati, A. and Rahadjeng, I.R., 2020, February. Sigmoid activation function in selecting the best model of artificial neural networks. In Journal of Physics: Conference Series (Vol. 1471, No. 1, p. 012010). IOP Publishing.

[33] Karlik, B. and Olgac, A.V., 2011. Performance analysis of various activation functions in generalized MLP architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems, 1(4), pp.111-122.

[34] Agarap, A.F., 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[35] Hammerstrom, D., 1993. Working with neural networks. IEEE spectrum, 30(7), pp.46-53.

[36] Luo, H. and Hanagud, S., 1997. Dynamic learning rate neural network training and composite structural damage detection. AIAA journal, 35(9), pp.1522-1527.

[37] Bebis, G. and Georgiopoulos, M., 1994. Feed-forward neural networks. IEEE Potentials, 13(4), pp.27-31.

[38] Sazli, M.H., 2006. A brief review of feed-forward neural networks. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 50(01).

[39] Frean, M., 1990. The upstart algorithm: A method for constructing and training feedforward neural networks. Neural computation, 2(2), pp.198-209.

[40] Sathyanarayana, S., 2014. A gentle introduction to backpropagation. Numeric Insight, 7, pp.1-15.

[41] Rojas, R., 1996. The backpropagation algorithm. In Neural networks (pp. 149-182). Springer, Berlin, Heidelberg.

[42] Du, S., Lee, J., Li, H., Wang, L. and Zhai, X., 2019, May. Gradient descent finds global minima of deep neural networks. In International conference on machine learning (pp. 1675-1685). PMLR.

[43] Wythoff, B.J., 1993. Backpropagation neural networks: a tutorial. Chemometrics and Intelligent Laboratory Systems, 18(2), pp.115-155.

[44] Erb, R.J., 1993. Introduction to backpropagation neural network computation. Pharmaceutical research, 10(2), pp.165-170.

[45] Li, J., Cheng, J.H., Shi, J.Y. and Huang, F., 2012. Brief introduction of back propagation (BP) neural network algorithm and its improvement. In Advances in computer science and information engineering (pp. 553-558). Springer, Berlin, Heidelberg.

[46] Deng, W.J., Chen, W.C. and Pei, W., 2008. Back-propagation neural network based importance–performance analysis for determining critical service attributes. Expert Systems with Applications, 34(2), pp.1115-1125.

[47] Toivonen, H.T., 1985. A globally convergent algorithm for the optimal constant output feedback problem. International Journal of Control, 41(6), pp.1589-1599.

[48] Jensen, S.T., Johansen, S. and Lauritzen, S.L., 1991. Globally convergent algorithms for maximizing a likelihood function. Biometrika, 78(4), pp.867-877.

[49] Anastasiadis, A.D., Magoulas, G.D. and Vrahatis, M.N., 2005. New globally convergent training scheme based on the resilient propagation algorithm. Neurocomputing, 64, pp.253-270.

[50] Prasad, N., Singh, R. and Lal, S.P., 2013, September. Comparison of back propagation and resilient propagation algorithm for spam classification. In 2013 Fifth international conference on computational intelligence, modelling and simulation (pp. 29-34). IEEE.

[51] Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

[52] Mosca, A. and Magoulas, G.D., 2015. Adapting resilient propagation for deep learning. arXiv preprint arXiv:1509.04612.

[53] Igel, C. and Hüsken, M., 2000, May. Improving the Rprop learning algorithm. In Proceedings of the second international ICSC symposium on neural computation (NC 2000) (Vol. 2000, pp. 115-121).

[54] Riedmiller, M. and Rprop, I., 1994. Rprop-description and implementation details.

[55] Li, J. and Rhinehart, R.R., 1998. Heuristic random optimization. Computers and chemical engineering, 22(3), pp.427-444.

[56] Xue, W., Mou, X., Zhang, L., Bovik, A.C. and Feng, X., 2014. Blind image quality assessment using joint statistics of gradient magnitude and Laplacian features. IEEE Transactions on Image Processing, 23(11), pp.4850-4862.

[57] Xie, R., Wang, X., Li, Y. and Zhao, K., 2010, June. Research and application on improved BP neural network algorithm. In 2010 5th IEEE Conference on Industrial Electronics and Applications (pp. 1462-1466). IEEE.

[58] Zhang, Z., 2018, June. Improved adam optimizer for deep neural networks. In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS) (pp. 1-2). Ieee.

[59] Feurer, M. and Hutter, F., 2019. Hyperparameter optimization. In Automated machine learning (pp. 3-33). Springer, Cham.

[60] Claesen, M. and De Moor, B., 2015. Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127.

[61] Günther, F. and Fritsch, S., 2010. Neuralnet: training of neural networks. R J., 2(1), p.30.

[62] Smith, L.N., 2017, March. Cyclical learning rates for training neural networks. In 2017 IEEE winter conference on applications of computer vision (WACV) (pp. 464-472). IEEE.

[63] Jacobs, R.A., 1988. Increased rates of convergence through learning rate adaptation. Neural networks, 1(4), pp.295-307.

[64] Wilson, D.R. and Martinez, T.R., 2001, July. The need for small learning rates on large problems. In IJCNN'01. International Joint Conference on

Neural Networks. Proceedings (Cat. No. 01CH37222) (Vol. 1, pp. 115-119). IEEE.

[65] Li, Y., Wei, C. and Ma, T., 2019. Towards explaining the regularization effect of initial large learning rate in training neural networks. Advances in Neural Information Processing Systems, 32.

[66] Takase, T., Oyama, S. and Kurihara, M., 2018. Effective neural network training with adaptive learning rate based on training loss. Neural Networks, 101, pp.68-78.

[67] Wu, P.S. and Martin, R., 2020. A comparison of learning rate selection methods in generalized Bayesian inference. arXiv preprint arXiv:2012.11349, 6.

[68] Mustapha, A., Mohamed, L. and Ali, K., 2020, June. An overview of gradient descent algorithm optimization in machine learning: Application in the ophthalmology field. In International Conference on Smart Applications and Data Analysis (pp. 349-359). Springer, Cham.

[69] Chen, Y. and Shi, C., 2022. Network revenue management with online inverse batch gradient descent method. Available at SSRN 3331939.

[70] Haji, S.H. and Abdulazeez, A.M., 2021. Comparison of optimization techniques based on gradient descent algorithm: A review. PalArch's Journal of Archaeology of Egypt/Egyptology, 18(4), pp.2715-2743.

[71] Wilson, D.R. and Martinez, T.R., 2003. The general inefficiency of batch training for gradient descent learning. Neural networks, 16(10), pp.1429-1451.

[72] Amari, S.I., 1993. Backpropagation and stochastic gradient descent method. Neurocomputing, 5(4-5), pp.185-196.

[73] Ketkar, N., 2017. Stochastic gradient descent. In Deep learning with Python (pp. 113-132). Apress, Berkeley, CA.

[74] Hardt, M., Recht, B. and Singer, Y., 2016, June. Train faster, generalize better: Stability of stochastic gradient descent. In International conference on machine learning (pp. 1225-1234). PMLR.

[75] Hinton, G., Srivastava, N. and Swersky, K., 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, 14(8), p.2.

[76] Khirirat, S., Feyzmahdavian, H.R. and Johansson, M., 2017, December. Mini-batch gradient descent: Faster convergence under data sparsity. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 2880-2887). IEEE.

[77] Konečný, J., Liu, J., Richtárik, P. and Takáč, M., 2015. Mini-batch semi-stochastic gradient descent in the proximal setting. IEEE Journal of Selected Topics in Signal Processing, 10(2), pp.242-255.

[78] Bagirov, A., Karmitsa, N. and Mäkelä, M.M., 2014. Introduction to Nonsmooth Optimization: theory, practice and software (Vol. 12, p. 13). Cham, Heidelberg: Springer International Publishing.

[79] Mäkelä, M.M. and Neittaanmäki, P., 1992. Nonsmooth optimization: analysis and algorithms with applications to optimal control.

[80] Boyd, S., Xiao, L. and Mutapcic, A., 2003. Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004, pp.2004-2005.

[81] Bagirov, A.M., Jin, L., Karmitsa, N., Al Nuaimat, A. and Sultanova, N., 2013. Subgradient method for nonconvex nonsmooth optimization. Journal of Optimization Theory and Applications, 157(2), pp.416-435.

[82] Goffin, J.L. and Kiwiel, K.C., 1999. Convergence of a simple subgradient level method. Mathematical Programming, 85(1), pp.207-211.

[83] Nesterov, Y., 2014. Subgradient methods for huge-scale optimization problems. Mathematical Programming, 146(1), pp.275-297.

[84] Nedic, A. and Bertsekas, D.P., 2001. Incremental subgradient methods for nondifferentiable optimization. SIAM Journal on Optimization, 12(1), pp.109-138.

[85] Geary, A. and Bertsekas, D.P., 1999, December. Incremental subgradient methods for nondifferentiable optimization. In Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304) (Vol. 1, pp. 907-912). IEEE.