T.C.

ALTINBAS UNIVERSITY

Information Technologies

USING ARTIFICIAL NEURAL NETWORK WITH COMPARATIVE ANALYSIS OF DIFFERENT TECHNIQUES FOR THE CLASSIFICATION OF SENTIMENT ANALYSIS

Omar Abdullah Saleh AL-BAYATI

Master Thesis

Supervisor

Prof. Dr. Oğuz BAYAT

İstanbul 2019

USING ARTIFICIAL NEURAL NETWORK WITH COMPARATIVE ANALYSIS OF DIFFERENT TECHNIQUES FOR THE CLASSIFICATION OF SENTIMENT ANALYSIS

by

Omar Abdullah Saleh AL-BAYATY

Information Technologies

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2019

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Oğuz BAYAT
Supervisor

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Prof. Dr. Oğuz BAYAT	School of Engineering and Natural Sciences, Altinbas University	____________
Prof. Dr. Osman Nuri UÇAN	School of Engineering and Natural Sciences, Altinbas University	____________
Prof. Dr. Hasan Hüseyin BALIK	Air Force Academy, National Defense University	____________

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

Asst. Prof. Dr. Oguz ATA

Head of Department

Approval Date of Graduate School of

Science and Engineering: ____/____/____

Prof. Dr. Oğuz BAYAT

Director

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

Omar Abdullah Saleh AL-BAYATI

DEDICATION

I give you this thesis in a form of appreciation and I may ask Allah to accept this work. Thank you to my academic adviser who guided me in this process and the committee who kept me on track. Thanks for my father who helped me in all things great and small. This dissertation is dedicated to my father who encouraged me to pursue my dreams and finish my dissertation. I dedicate this to my mother and father ABDULLH SALEH AL-BAYATI AND INTISAR HADI. This book is dedicated to, for his kindness and devotion you left fingerprints of grace on our lives. You shan't be forgotten.

ACKNOWLEDGEMENTS

First of all, thank to Almighty ALLAH who given us courage, wisdom to complete this work and patience to face all difficulties.

I wish to express my acknowledgements to my supervisor, Prof. Dr. Oğuz BAYAT who was abundantly helpful and offered invaluable support with his sincerity and belief in me. I must express my very profound gratitude to father Abdullah Saleh and mother INTISAR HADI for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

ABSTRACT

USING ARTIFICIAL NEURAL NETWORK WITH COMPARATIVE ANALYSIS OF DIFFERENT TECHNIQUES FOR THE CLASSIFICATION OF SENTIMENT ANALYSIS

AL-BAYATI, Omar Abdullah Saleh

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Prof. Dr. Oğuz BAYAT

Date: October 2019

Pages: 55

Sentiment Analysis means identifying the favorable, negative or neutral opinion or reviewer opinions expressed in a piece of job. In social media surveillance sentiment assessment is helpful to automatically characterize the general impression or mood of the customers as replicated on their social media for a particular brand or business and determine if they are regarded favorably or negatively on the Internet. This article examines the machine-based learning approaches to feeling assessment and highlights the key characteristics of methods. Prominently used techniques and methods are Naïve Bayes, Maximum Entropy and Support Vector Machine, the most near-neighbor classification of Machine Learning-based feeling assessment. Naïve Bayes ' depiction is quite easy but does not give rise to wealthy assumptions. The hypothesis that characteristics are independent is too restrictive. Maximum Entropy estimates the distribution of probability by information, but it does well only with dependent characteristics. For SVM the kernel is correct, but the way to deal with multi-class issues is not standardized. A method which combines neural networks and fuzzy logic often is used to improve the efficiency of correlations and dependencies between variables.

Keywords: Naïve Bayes, Neural Network, Sentiment analysis, Support Vector Machine, Machine Learning.

ÖZET

YAPAY SINIR AĞININ BENZER ANALIZLERIN SINIFLANDIRILMASINDA FARKLI TEKNIKLERIN KARŞILAŞTIRMALI ANALIZIYLE KULLANILMASI

AL-BAYATI, Omar Abdullah Saleh

Yuksek Lisans, Bilisim Teknolojileri, Altınbaş Universitesi,

Danışman: Prof. Dr. Oğuz BAYAT

Tarih: Ekim 2019

Sayfalar: 55

Duygu Analizi, bir işte ifade edilen olumlu, olumsuz ya da tarafsız görüşü ya da hakem görüşlerini tanımlamak anlamına gelir. Sosyal medyada gözetim hissi değerlendirmesi, müşterilerin sosyal medyalarında belirli bir marka veya işletme için kopyalanan genel izlenimini veya ruh halini otomatik olarak karakterize etmek ve İnternet'te olumlu mu yoksa olumsuz mu olarak değerlendirilip değerlendirilmeyeceğini belirlemek için faydalıdır. Bu makale, makine temelli öğrenme hissi değerlendirmesine yaklaşımları incelemekte ve yöntemlerin temel özelliklerini vurgulamaktadır. Belirgin olarak kullanılan teknik ve yöntemler, Makine Öğrenmeye dayalı duygu değerlendirmesinin en yakın komşu sınıflandırması olan Naif Bayes, Maksimum Entropi ve Destek Vektör Makinesi'dir. Naif Bayes'in tasviri oldukça kolaydır ancak zengin varsayımlara yol açmaz. Özelliklerin bağımsız olduğu hipotezi çok kısıtlayıcıdır. Maksimum Entropi, olasılık bilgisine göre dağılımını tahmin eder, ancak yalnızca bağımlı özelliklere iyi gelir. SVM için çekirdek doğrudur, ancak çok sınıflı meselelerle başa çıkma yöntemi standart değildir. Yapay sinir ağlarını ve bulanık mantığı birleştiren bir yöntem, değişkenler arasındaki korelasyonların ve bağımlılıkların etkinliğini geliştirmek için sıklıkla kullanılır.

Anahtar kelimeleri: Naif Bayes, Yapay Sinir Ağı, Duyarlılık analizi, Destek Vektör Makinası, Makine Öğrenmesi.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

1. INTRODUCTION

The neural system is arranged into hidden layers, input and output of the Artificial Neural Networks (ANNs). The neurons are joined together by a series of synaptic weights. An ANN is a powerful tool for identifying patterns, predictions and regressions in a variety of problems. The ANN continually changes its synaptic values during the learning process until sufficient acquired knowledge (until a certain number of iterations have been achieved or the error value of the target has been achieved). After completion of the learning or training stage, the ability of the ANN to generalize the problem with samples other than those employed during the training stage must be assessed. Finally, during training and testing, it is expected that the ANN will be able to accurately classify the patterns of a particular problem. In recent years several classic ANN algorithms have been suggested and developed. Many of them can however be trapped in unwanted solutions; in other words, they will be far from the best or the best solution. Moreover, the majority of the algorithms are multimodal or non-continuous.

Therefore, other methods are essential for training an ANN, such as bio-inspired algorithms (BIAs). As BIAs are powerful optimization instruments, they are well recognized by the artificial intelligence community and can fix very complex optimization issues. BIAs can navigate big multimodal and ongoing search regions and discover the optimal value for the best solution for a specified issue. BIAs are based on the conduct of nature defined as swarm intelligence. This idea is defined by [1] as owned by unintelligent agents with restricted individual capabilities, but smart collective conduct.

Many studies have been conducted using developmental and organic motivated algorithms to form ANN [2]. Metaheuristic approaches are based on local scanning, population and other techniques such as cooperative coevolutionary models [3] for training neural networks. The author presents an outstanding piece of literature on developing ANN algorithms [2]. However, most of the study studies concentrate on the growth of synaptic weight, parameters [4], or the growth of the amount of the neurons in the hidden layer. In addition, investigators do not include the development of transfer features as a significant component of an ANN to determine each neuron's output. For example, in [5] the authors proposed a combination of the ANN and PSO methodology for weight adjustment for ant colony optimization (ACO). Further studies like [6] alter the PSO for the purchase of the synaptic al weight and limit ANNs for Simulated Annealing. In [7], writers use the architecture and weight of Evolutionary Programming as a means to resolve classification and prediction issues. Another instance is [8] in which genetic programming is used to produce graphs of distinct topologies. In [9] a weather prediction issue was solved by applying the differential evolution (DE) algorithm to design an ANN. In [10] the writers use a PSO algorithm for altering their synaptic body weight to shape the relation between everyday rains in Malaya. In [11] writers only change the synaptic weights of ANN in order to solve classifications of issues by comparing the background technique to fundamental PSO [12]. In the case of differential evolution and fundamental PSO the series of weights is developed. In other works like [13], the three main components of the ANN were simultaneously created: architecture, transmissions and synaptic weights. In [14] the author solved the same problem with the Evolution (DE) differential algorithm, and proposed for the authors a new model with the PSO (NMPSO) algorithm. In addition, the writer [15] used an Artificial Bee Colony (ABC) algorithm to create the concept of an ANN with two distinct fitness functions.

1.1 PURPOSE AND THESIS IMPORTANCE

In this thesis, we're using artificial neural network for the classification of sentiment dataset that is taken for UCI Machine learning for the positivity prediction. The ability that artificial neural network has in the training according to the data and understanding its information might make this algorithm better than a lot of classification algorithms around us. In this study, we calculated the classification accuracy, Mean Square Error (MSE), Specificity, and sensitivity.

2. LITERETURE REVIEW

Sentiment relates primarily to feelings, emotions, thoughts or attitudes. With World Wide Web rapidly increasing, individuals often convey their feelings through social networks, blogs, ratings and reviews over the Internet. As textual information increases, the notion of expressing feelings needs to be analyzed and company insights calculated. Corporate owners and advertising businesses often use sentiment analysis to launch a fresh corporate and publicity strategy [89].

Sentiment analysis may be used for multiple reasons in separate areas. In online trade, for instance, sentimental analysis is widely integrated into e-commerce. Websites enable consumers to record their shopping and product quality experience. They provide a summary by assigning ratings or scores for the item and its distinct characteristics. Customers can read views and recommendations on the entire product and on particular product characteristics readily. Voice of the Market (VOM) is about how clients feel about their competitors ' goods or services. Voice of the customers (VOC) is concerned about what individual customers say about products or services. It involves an analysis of customers ' opinions and feedback. The management of Brand Reputation (BRM) is concerned with the management of market reputation. Customers' opinions or any other party may harm or reinforce business reputation [79].

Machine learning algorithms are very useful for classifying and for predicting the favorable or negative feeling of a specific document. Machine education is divided into two kinds of machine learning algorithms, known as monitored algorithms. Supervised learning algorithm is based on a labeled dataset in which each training document is marked with a suitable feeling. Unchecked learning includes unlabeled information if the text is not marked by suitable feelings [80].

The main focus of this article is the application of monitored training on a marked dataset. Analysis of sentiment is generally carried out at three stages, namely the level of the phrase, document level and aspect. Document The classification of the feeling at document level seeks to classify the whole document or subject as positive or negative. The sentiment classification of sentences takes into account the polarity of individual sentences, whilst the sentiment classification of the aspect level first identifies the various elements of a corporation and the polarity in relation to the elements acquired for company exploration [18] is calculated for each document.

The Artificial Neural Networks (ANNs) are the neural system that is structured into concealed input, input and output layers. An ANN is a powerful tool used to identify models, predictions and regressions in a variety of problems. The ANN continually changes its synaptic values during the learning process until sufficient acquired knowledge (until a certain number of iterations have been achieved or the error value of the target has been achieved). After the training stage, the capacity of ANN to generalize the problem must be evaluated with samples different from those used during the training stage. The problem should be generalized.

As BIAs are powerful optimizing instruments and can solve very complex optimizing issues, they have a positive support by artificial intelligence. BIAs can explore large multimodal and non - constant search areas for the best solution near the optimal value for a specific problem. BIAs are based on the behavior of nature described as the swarm. In [16] this concept is defined as owned by smart and limited personal agents with smart collective behavior.

Many works employ evolutionary and bio-inspired algorithms to train ANN as another basic type of education [17]. Metaheuristic training techniques for neural networks are based on local search, demographic techniques and others such as coevolutionary collaborative models [18].

The writers demonstrate a thorough overview of evolutionary algorithms used to develop the ANN [17]. But most study papers focus solely on synaptic weight growth, parameters [19] or the evolution of neuron number in hidden layers. The developer determines the amount of layers that are concealed. The research also does not include the development of transfer features, which are a significant component of an ANN that determines each neuron's output.

In [20], for instance, the writers proposed a technique for adjusting the weight of the syntactical to combine ANN and Particle Swarm Optimization (PSO) for a particular design. Additional research, including [20] conducted a modified Simulated Annealing PSO combination to achieve a number of ANN and ANN weights [21]. Author use evolutionary programming to fix issues of classification and forecasting in order to find out the architecture and weights. Another instance is Genetic Programming [22] where graphs representing various topologies are obtained. In [23], an ANN was designed to fix an weather prediction issue with the Differential Evolution (DE) algorithm. In [24] the writers modified the PSO algorithm to model the daily relations of precipitation and rain in Malaysia. In [25], writers compare the background propagation technique against fundamental PSO in order to adjust the weight of a synaptic ANN only to solve issues of classification. In [26] the weighing system is developed by means of the differential development and fundamental PSO.

The three main elements of the ANN have evolved simultaneously in other works such as [27]: architecture, transfer functions and synaptic weights. In [28], the authors solved the same problem using the Differential Evolution (DE) algorithm. They proposed a new PSO model. Another example is [29] in which authors develop an ANN design with two different fitness functions using an Artificial Bee Colony (ABC).

This research has significant contributions in comparison with these last three works. First and foremost, there are eight fitness functions to address three common problems arising from the ANN design: accuracy, overriding and reducing ANN [30]. In this respect, fitness functions take classification error into account, mean square error, validation error, architecture reduction and a combination of problems that arise while designing the ANN. The research in [31] further examines the behaviour, using different parameter values, of the three bioinspired algorithms. The values of the best parameter for these algorithms will be determined to achieve the best results during the experimentation phase. Furthermore, the best configuration for each selected classification problem is used to create a set of statistically valid tests. In the context of connection number, number of a neuron and the selected transfer functions, the results are also presented and discussed [32]. The results are presented by a proposed methodology. Another contribution to this study is a fresh measurement that enables the findings of an ANN produced with the suggested methodology to be effectively comparable [33]. This measures the detection rate during workouts and test phases, where the precision of testing is more weighted compared to the precision of practice. Finally, the findings of the three bio-inspired algorithms are likened to those of two classical learning algorithms [34]. This is based on the metaphor for the fundamental PSO method, because NMPSO is an algorithm that is relativement new (in 2009). It is essential to compare its efficiency with other similar algorithms.

In general, the problem to be resolved can be defined as a set of input patterns, and a set of the desired patterns, and an ANN is found represented by the function to be determined by so that the maximum number of neurons is minimized and defined [52]. It is important to note that three fields (architecture, synaptic weight and transfer functions) form part of the search space [35].

Automatic handwriting recognition was one of the most active fields of research in recent decades. Due to the cursive nature and the vast array of styles, vocabulary … etc., the task of recognition is a challenge [43, 44, 45, 46]. Although character recognition rates are high, it is not easy to recognize offline text. The words are separated in their character parts by the analysis of character shapes in a large number of text-recognition pieces. For the purpose of performing text recognition, stochastic approaches like Hidden Markov Models [54, 55, 56, 57, 58, 59] were used. HMM's avoid cursive word segmentation by joint segmentation and recognition into character/subwords [59]. Recent sequence classifications of Recurrent Neural Network (RNN) have been shown to perform better [47]. The BLSTM RNN architecture allows access both in the input direction to the long-range context.

A classifier combination approach can further improve recognition performance. Classification combination hinges on assuming that different classifiers are capable of compensating for each other by combining their own strengths and weaknesses. In order to record word assumptions [42], In a word recognition module, a character recognition system has been integrated. In order to classify characters as being part of isolated or continuous, man-written word recognizers, multi-layered perceptron (MLP) has been widely applied [40, 41, 48, 49, 53]. Recent studies in the field of the Deep Belief Network [51] education strategy have resulted in an improvement in the performance of machine learning and in the identification of patterns. In the deep networks, non-linear detector hierarchies are developed which better capture complicated data patterns. The DBNs offer superior performance than MLPs because of its deep learning mechanism. DBNs are used to upgrade existing handwriting systems in this work as a verification module [36].

In recent years, several classic ANN algorithms have been suggested and created. Many can, however, remain trapped in unwanted alternatives; that is, far away from the best or best alternative [50]. In addition, most of these algorithms cannot investigate multimodal and non-permanent surfaces. Consequently, other methods are needed to train the ANN, such as biologically inspired algorithms (BIAs) [37].

In the research, the implementation of bio-inspired algorithms can be automated through an ANN, notably by using Basic Particle Swarm Optimization (PSO), Second Generation PSO (SGPSO) as well as the New NMPSO model. This research provides a extensive analysis [39]. The proposed methodology simultaneously develops the architecture, the synaptic weights and the type of transfer function to create the ANNs that are most exact in a specific problem. In addition, a comparison with classic teaching techniques (back-propagation and Levenverg-Marquardt) of the Particle Swarm algorithm is provided. Moreover, a fresh way of selecting the highest amount of neurons (MNN) is provided in the study. The precision of the technique proposed is tested to solve certain actual and synthetic issues in pattern recognition.

The database or KDD often includes data mining in a wider context of knowledge research. The KDD process consists of several phases: the choice of the target, data processing, conversion where appropriate, data mining, pattern and interaction extraction and understanding and assessment of structures detected. [75, 76, 77, 78].

The state detection of Drowsy students helps understand the learning status of students, which is the necessary basic aspect of evaluation and evaluation of teaching activities [60, 61]. Due to external environmental factors, the quality of conventional processes can deteriorate considerably [62, 63, 64, 65].

Web-based learning environments offer new opportunities for students to learn [73]. These are rich in educational content and resources, which allow students to browse easily, based on their interests and needs. Therefore, students are responsible in these settings for preparing, carrying out and assessing their own training [71, 72, 73]. To increase the success of students and enhance the development of curriculum, education staff need to learn more. Yet conventional analysis methodologies cannot simply do that [74].

Pang, Lee and Vaithyanathan [83] classed sentiment by using three different machines: Naïve Bayes classification, Support Vector machine and Maximum Entropy classification. The categorized feelings are positive and negative. The use of n-grams increases these methods [81, 82]. Their research shows that SVMs work better than the technique used by Naïve Bayes.

For testing, instruction and identification of the properties, the structured reviews are used. Score techniques are used to determine if the assessments are positive or negative. The NB and SVM classifiers are utilized to classify the phrases acquired from web search via search queries using the product name. While working on individual phrases from internet searches, noise and ambiguity limit the output. However, the findings are qualitatively quite helpful in the context of a full web-based instrument and supported by a straightforward way of grouping phrases into attributes [84]. Naïve Bayes was discovered to accomplish better efficiency over SVM among SVM, NB and ME sentiment classification techniques [87].

C-classification of the nearest neighbor (kNN) is based on the assumption that an instance's classification is most like that of other instances in the vicinity of the vector space. KNN does not rely on the preliminary probabilities and is computationally effective in contrast to other text classification techniques such as Naive Bayes [89].

Jian, Chhen and Han-shi have suggested a strategy based on artificial neural networks in which the paper is divided into positive, negative and fuzzy tones. Speed and precision were evaluated in the assessment and feeling assessment on a wide range of tweets with Hadoop. This strategy utilizes the most recurrent back propagation algorithm. The findings indicate that the method is very effective when managing large-scale sentiment information sets compared to tiny data sets [88].

Chen, Liu and Chiu have suggested a Neural Network strategy that combines the benefits of the BPN and SO indexes to classify feeling in blogospheres. The suggested strategy produces more precise outcomes compared to traditional methods such as BPN and SO indexes. The precision of classification has been improved and training time has also been reduced [90].

Analysis of sentiments plays a significant role in opinion mining. It is usually used when customers must decide or choose a product with its reputation that derives from the view of others. Analysis of feelings can show what others believe of a product. In accordance with the intelligence of the crowd sentiment analysis, the product selection is indicated and recommended [66, 67, 68, 69, 70]. A single worldwide rating might alter this product's view. Another way for businesses who want to learn about customers' evaluation of their products is to use sentiment analysis. Analyzes of sentiment may also determine which characteristics clients consider more important. The human / machine interface field offers many opportunities to know what individuals believe. Sentiment assessment is a non-trivial stage in the assessment of the company operations, such as brand leadership, product scheduling etc. The overall process flow is shown in Figure 2.1.

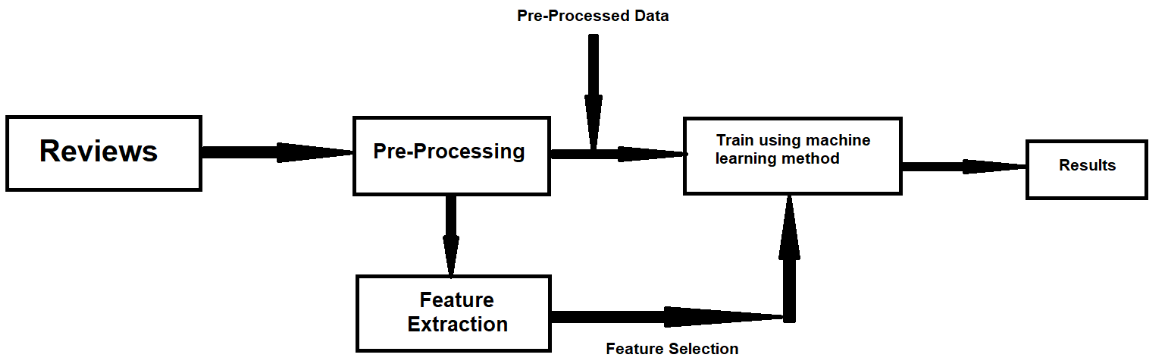

Figure 2.1: A general process flow using machine learning techniques [91].

3. METHODOLOGY

3.1 DATA RESOURCE

This dataset was created for [19], it contains sentences labelled with positive or negative sentiment. Score is either 1 (for positive) or 0 (for negative), the sentences come from three different websites/fields, IMDB, amazon, and yelp. For each website, there exist 500 positive and 500 negative sentences. Those were selected randomly for larger datasets of reviews. We attempted to select sentences that have a clearly positive or negative connotation, the goal was for no neutral sentences to be selected.

Following are steps in preprocessing.

- Stop word removal
- Symbol removal
- POS tagging (Part Of Speech).
    - Stanford POS tagging is used for our study.
    - This method finds actual parts of speech using the English parser mode.
    - The POS Tagging on the input sentence and uses Verb, Adverb and Adjectives only.
    - It uses the standard Penn Treebank POS tag sets.
- For example: The movie was not quite good. After the Removal of stop word Output is [Movie, not, quite, good] after POS Tagging result is [Movie/NN, not/RB, quite/JJ, good/JJ].

3.2 DATA PREPROCESSING

When the data input is too large to process, then it is called feature extraction to transform the data input into the feature set. If the functions are correctly extracted from the information, the job is anticipated. The function extraction technique extracts the function from the dataset (adjective). Then this adjective is used to display positive and negative polarity in a phrase that is helpful for determining the opinion / sense of people using the unigram model. It rejects the previous term in the phrases with the adjective.

Data preprocessing plays a vital role in the general process of knowledge discovery before information collection itself. One of the first measures is information standardization. In coping with parameters of distinct units and scales this step is very crucial. Some methods, for instance, use the Euclidean distance. For a reasonable comparison between them, therefore, all parameters should have the same scale.

In the following formula, all numeric variables are scaled in the range [0, 1]. One possible formula is given below (eq. (3.1)):

$$X_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

3.3 SUPPORT VECTOR MACHINES

A description of the unfamiliar dependence between measurements of objects and certain characteristics of these subjects is the primary purpose of statistic teaching. Measurements are considered to be observable on all objects of concern, also known as "input factors." On the other hand, the properties, or "output variables," of the objects are usually only available for a small subset of objects. It is intended to estimate the dependency between input and output variables so that the results for any item of concern can be determined. In pattern recognition, this relates to trying to estimate a function f: R N 7→ {±1} that can correctly classify new examples based on past observations. Vector machines are one of the common classification algorithms for these assignments and are renowned for their powerful theoretical background, widening efficiency and handling of high dimensional information. They also support vector machines [6] This chapter offers an overview of vector learning assistance, beginning with linear SVMs followed by the non-linear case expansion.

3.3.1 Linear SVMs

Let ((x1, y1)·· (xn, yn)) be the exercise data set in binary classification setting, where xi are the instance vectors (that means, the observations) and yi{ −1, + 1} are instance labeling.· (xn, yn) Support for vector training is the issue of discovering a hyperplane separating the favorable instances (labeled+ 1) from those with the larger margin (labeled-1). The hyperplane margin is described as the shortest distance from the hyperplane between the negative and the favorable cases. The idea that a hyperplane with the biggest range should be more noise resistant than a higher-level with a lower margin is behind the search for the hyperplane. Formally, assume the restrictions are met by all information (eq. (3.2) and eq. (3.3)):

$$w.x + b \geq +1 \quad y_i = +1 \tag{3.2}$$

$$w.x + b \leq -1 \quad y_i = -1 \tag{3.3}$$

Where w is normal to hyperplane, the distance perpendicular to the origin of is the Euclidean standard of w. Both of these limitations can easily be mixed into (eq. (3.4))

$$y_i(w.x_i + b) \geq 1 \tag{3.4}$$

The training examples for which eq. (3.4) holds lie on the canonical hyperplanes (H1 and H2 in Figure 3.2). The margin ρ in eq (3.5) can then be easily computed as the distance between H1 and H2.

$$p = \frac{|1-b|}{||w||} - \frac{|-1-b|}{||w||} = \frac{2}{||w||} \tag{3.5}$$

The following primary optimizing issue can therefore be solved to build the maximum hyperplane margin separating (eq. (3.6))

$$\min \tau(w) = \frac{1}{2}\|w\|^2 \ subject\ to\ y_i(w.x + b) \geq 1 \tag{3.6}$$

For two main reasons, we switch to Varangian wording on this issue: I constraints are easier to deal with, and ii) training data is only used between the vectors as a point’s product.

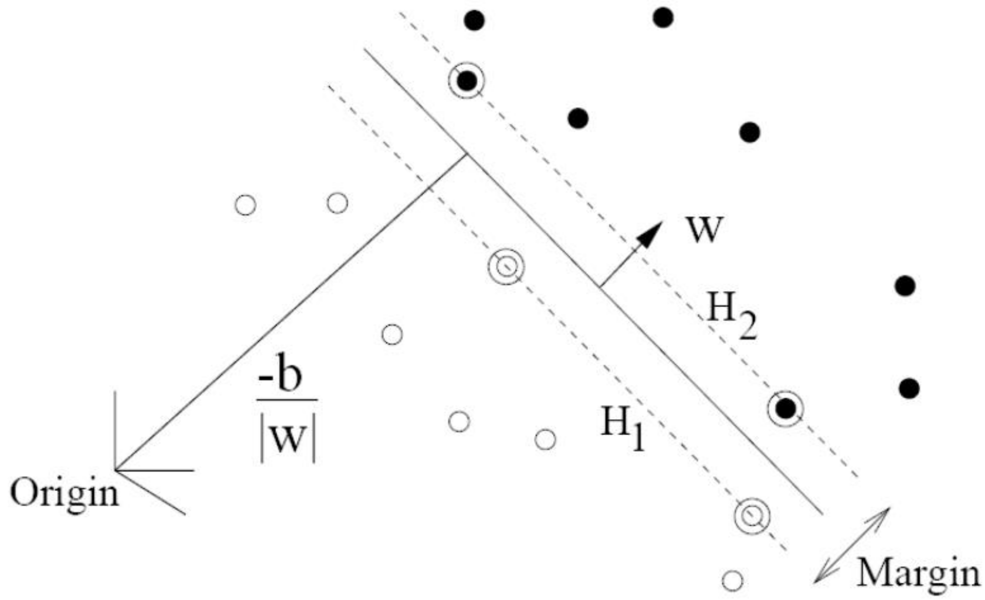

Figure 3.2: A hyperplane separating two classes with the maximum margin [88]

3.3.2 Nonlinear Support Vector Architecture

Suppose the training data are mapped with a mapping¯to another Euclidean room H: Rd7 to H. From the point of view of the training algorithm, this transformation is only affected in that the algorithm would depend on the data via the H points, namely on functions in form xi· xj at d, instead of computerizing dot products xi· xj at R d, < xj*· total (xj). Note that the computationally costly conversion of every instance is not necessarily necessary to us. In some expanded space feature (i.e. K(xi, xj)= äl (xi)·alf)(xj)), if the kernel K function were to be equivalent to the dot product in a certain space, we only need to calculate K value to find the dot products in os without even mapping xi and xj from Rd d to H. In the dual issue of (3.7), we substitute a kernel K for the conversion − kernel K, which corresponds to a dot product.

$$\max L_D = \sum_I \alpha_I - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right) \quad (3.7)$$

3.4 NAIVE BAYES CLASSIFIER

Classifier Naive Bayes is probabilistic, based on the theorem of Bayes. The theorem of Bayes defines the relationship between the hypotheses' conditional probabilities and the Eq (3.8) observations. The first time Suppose that h is the hypothesis and O is the observation.

$$P(h|o) = \frac{P(o|h).P(h)}{P(o)} \tag{3.8}$$

Naive Bayes Classifier is a monitored learning algorithm that requires training before it can be classified. It needs a training set therefore. The training package includes several observations and the courses they are categorized in. The goal of a Naive Bayes classification is to classify an invisible series of parameter values into one of the training classes. Assume that the classificatory values are B, A, A, C. This sequence of values must be classified in one of the groups of fire: YES or NO. In accordance with Eq. 3.5, a larger hypothesis must be chosen. 3.5. In order to compute the probabilities of each class, the classifier must refer to the instruction set according to the probability distribution in the training set. The classifier must count the data line number, in which T equates to B when the data row is classified as NO, for the determination of the probability of the Class NO. There is 1 row with this criterion and 3 rows with T being B and YES for the data row. There is 1 row with that criteria. The conditional likelihood of T is therefore equal to B provided that NO is equivalent to 1/4. All probabilities are calculated by the classifier.

3.5 ARTIFICIAL NEURAL NETWORKS

The neuron structure in the human brain inspires the Artificial Neural Network. The brain learns from experiences that go beyond the range of computers and thus adapts. This modeling also allows solutions to be less technologically developed in order to reduce human involvement.

The implementation of neural networks in computing gives us a key computational advance. Computers, like complex math and face recognition, do well. However, simple patterns are hard for computers. Computers cannot analyze, generalize and transform past patterns into future actions. The advanced neural network study allows people to understand the mechanism of thoughts, for example.

The study focusses on the' brain storage data as patterns method [17]. For individual faces to analyze and recognize, some of these patterns are very difficult. This process saves information as patterns, analyzes patterns and fixes a new computer field for problems. Training and creation of these networks is involved in the neural network to solve specific problems. We can perform different techniques such as learning, behaving, forgetting and reacting, etc. through our neural network.

"All aspects of this processor are known: The human brain is still mysterious to operate accurately. In particular, the most important element in the human brain is a certain type of cell which does not appear to regenerate as opposed to the whole body. Since the only portion of the body not slowly replaced by this type of cell, we are supposed to be able to remember, think and put past experiences into practice in all our actions. These cells are all known as neurons". The neural network as the human brain's neural network offers power through its complex components, its control mechanisms and its subsystems. Genetics programming and learning are also involved. In electro-chemical ways, neurons transmit the information between them. Depending on the classification process used, these neurons are classified into different categories. Current systems remain incompatible with the human brain. "The most basic elements of this complex and powerful organization can only replace these artificial neural networks. Neural computing has never succeeded the developer who is trying to solve problems with the human brain. Neural computing was never substituted to human brain by the development company which attempts to solve problems. It's a new way of solving problems and machinery.

Let's see the overview of human neurons. The basic element of the neural network is neuron. The biological neurons are input and subsequently subject to various non - light operations and then produce final output. Four nerve cells are typical: dendrites, somas, axons and synapses. It accepts inputs as the task of dendrites. The input is processed by Soma. Axon-transforms the processed inputs and synapses (contacts neurons). The biological neurons in structure are not too straightforward but complex.

Biology improves the understanding of neurons. By understanding the biological brain, network designers can enhance their systems further. Neural artificial networks (ANN) are computer tools which have been widely accepted in many different disciplines to model complicated problems in the real world. ANNs could be defined as structures consisting of strongly linked adaptive elements called neurons that provide massive computations for the representation of knowledge.

"The principal features of the ANN are the ability to handle inaccurate and inadequate information, the ability to manage inaccurate information, robustness, failure and failure tolerances, fault and failure tolerance". Artificial models possess following characteristics:

1. Nonlinearity makes it more suitable for data
2. Precise prediction for uncertain data and measurement errors due to noise insensitivity.
3. High parallelism means tolerance of hardware failure and quick processing.
4. Learn and adapt to the changing environment, allowing the system to update its internal architecture.
5. The model can be applied to unlearned data by generalization.

The main goal of ANN - based computing is to develop mathematical algorithms that enable artificial neural networks to study the processing of information and to gather human brain knowledge. Patterns based on ANN may provide virtually accurate solutions to specifically formulated problems and processes that only experimental and field observations understand. “In a number of applications from modeling, classification, design recognition and multi - variable data analysis, microbiological ANNs have been applied. The digital image processing interests are derived from two main applications: improvement for human - the reading of photo information; and processing of image information for storing, transferring and representing autonomy sensor machine; and defining a two - dynamic function as the image, f(x, y).

“Digital image processing is the processing of digital images via digital computers. A numerous elements with a specific position and value comprises a digital picture. They are called picture elements, elements, pixels and skins. An extensive and diversified application for digital image processing”.

3.5.1 General Properties of Artificial Neural Networks

3.5.1.1 Nonlinear I/O mapping

The analysis of high-dimensional data has an increasing importance due to the growth of sensor resolution and computer memory capacity. Typical examples are images, speech signals and multi-sensor data. In the analysis of these signals they are represented in their entirety or part by part in a sample space. As an example, a single data point may represent a 16x16 (sub) image. A collection of these images constitutes a cloud of points in a 256-dimensional space. In most multi-sensor data sets there is a large dependency between the sensors or between the sensor elements. This is certainly true for nearby image pixels. But also more fundamentally, it is not to be expected that any physical experiment contains hundreds of degrees of freedom that are of significant importance. Consequently, multi-dimensional data sets may be represented by lower dimensional descriptions. There are various reasons why such representations may be of interest. They may reveal the structure of the data or the problem, they may be used for relating individual data points to each other (e.g. finding the most similar one in a database) or they may be used for retrieving missing data values.

3.5.1.2 Generalization ability

Their capacity to generalize is one of the major advantages of neural networks. This means that a trained network can classify data from the same class that it never saw. Developers in real-life applications usually only have a small portion of all possible neural net generation patterns. The data set should be divided into three parts to achieve the best generalization:

The training is for the training of neural networks. During training, this data set error is minimized.

The validation set is used to determine the performance of neural networks in untrained patterns. A test set to finally monitor a neural net's overall performance. In the minimum validation error, learning should be stopped. The net generalizes best at this stage. If learning is not stopped, overtraining occur, and net performance decreases over all data, even if the error is still smaller in the training data. The network must finally be checked for the third dataset, the test set, after the completion of the learning phase. Each n training cycle, ANNS performs one validation cycle.

3.5.1.3 Fault-tolerance (graceful degradation)

Fault tolerance is the feature that permits the system, when certain components of it fail (or one or more inside failures), to continue to operate properly. The decline in operating quality, as opposed to an unnatural system, is proportional to the gravity to which the failure is causing a total failure if it does not. Fault tolerance is especially sought in high availability or life-critical systems. When parts of a system break up, features are known as graceful degradation.

A defect-tolerant design allows a system to operate at a lower level than to fail completely if some of the system fails. When a system fails. The term is used most often to describe computer systems designed to continue to operate more or less fully in the event of a partial failure, with reduced output or with an increased response time. In other words, the entire system is not stopped due to hardware or software problems. A motor car is designed to continue to drive in the presence of damage caused by fatigue, corrosion, manufacturing defects or impacts, when a tire has been punctured or when structural integrity is maintained. Another example is an automotive vehicle.

The tolerance of failures can be achieved by anticipating special conditions and building a system to deal with them within the scope of an individual system, and, in general, by trying to stabilize itself so as to converge the system in a faultless state. Nevertheless some kind of reproduction is better used when the consequences of an insufficient system are catastrophic or the costs of making it reliable enough are very high. In each case, the system must be able to reverse it in safe mode, if a system failure is so catastrophic. The recovery is like a reversal, but if people are present, it can be a human action.

3.5.1.4 Biological analogy

Analogy is a cognitive process where information or significance is translated to a different objective or linguistic expression from one particular topic–analogy or source. A less narrowly defined inference or argument from one specific to another is in comparison to the analogy of deduction, induction and abduction where at least one premise or conclusion is general. The word analogy could also refer to the relation of the source and the goal itself, often but not necessarily similar, as in the concept of biology. The atom model of Rutherford analogized the atom with the solar system. Analogy plays a major role in problem solving, including decision-making, discussion, perception, generality, remembrance, creativeness, invention, prevision, emotions, explanation, conceptualization and communication. The analogy was argued as "the cognitive nucleus." In particular, the analogical language includes, although not metonymies, exemplification, comparatives, metaphors, like, allegories and parable. It is not metonymies, but it does not contain the identification of places, objects or persons. As if it were, phrases like, etc.

and so on, and the same term also depends on the analogous understanding of the message by the receiver. Analogy is important not only in ordinary language and common sense in the fields of science, philosophy, law and humanity. Analogy is closely linked to concepts of association, comparative approach, correspondence, mathematical and morphological equality, homomorphism and iconicity. The concept of conceptual metaphor in the cognitive linguistics can be the same as the concept of analogy. Analogy also provides a basis for all comparative arguments and experiments whose findings are conveyed to objects not examined.

3.5.2 Structure of ANN

The neural network consists of three groups, layers and phases. The phases input, covered and output.

- The activity or input unit is the raw data that the network receives.
- The cached phase is based on the data entered and the weights of the connection between the input and the cached units.
- The phase of the output depends on the activity of the cloaks and weights between the cloaked and output units.

On the basis of the layer activity, different network types exist. A simple network type is called where the hidden units can build their own input representation. Whenever each hidden unit is active, the masses of the hidden and input units can be adjusted to allow for a hide unit to choose what it is. Other architectures such as single and multilayer are also available. In a single layer each layer is connected to the other. Overall, the network of the single layer includes only entries and outputs. The inputs are supplied by a number of weights to outputs. In several layers all units have inputs, hidden and outputs in different layers.

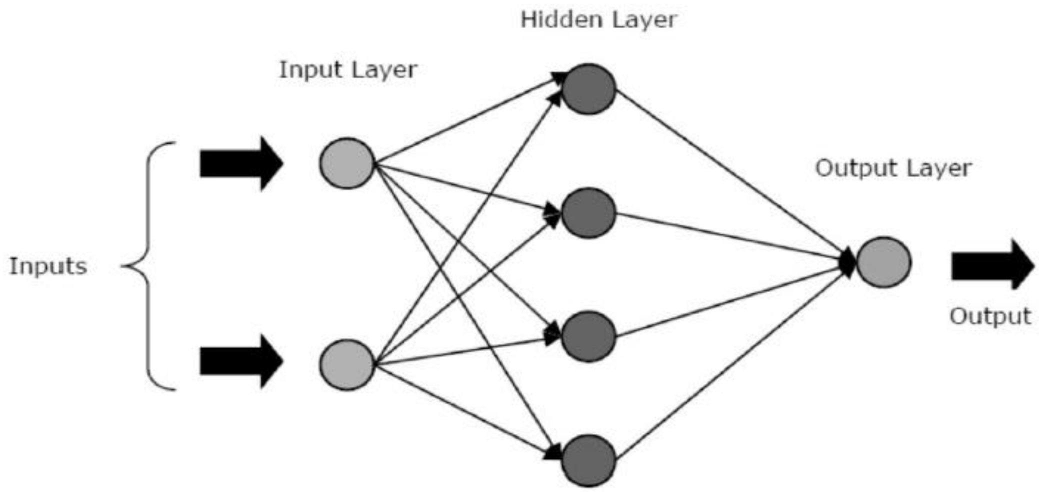

Figure 3.3: General neural network architecture [2]

3.5.3 Elements of Artificial Neural Networks

3.5.3.1 Inputs

The data, signal, feature, or outside environment measurements are received from this layer. These inputs are generally normalized within the limits of the activation functions. The values of samples or patterns. This standardization gives greater numerical accuracy for the network's mathematical operations.

3.5.3.2 Weights

The links between neurons are assigned to the number "weights" or "parameters" in artificial neural networks. These weights change as new data are fed into the neural net. So "learns the neural net.

The weights in an artificial neural network are an estimation of the combined multiple processes in biological neurons. Myelination is important, but not important. Myelination. Weights can be positive or negative in artificial neural networks. Weight: weight is comparable to an increased combination of dendrites between neurons, numbers of synapses between dendrites, density of postsynaptic terminal neurotransmitter receptors, as well as increased formation and fusion of neurotransmitters of the vesicles and of the pre-synaptic terminals.

Positive weights: The positive weights of the synapses releasing excitation neurotransmitters (i.e. glutamate) are analogous to the pre - synaptic terminals. The receiving cell will increase its likelihood of activation.

Negative weights: Negative weights are similar to those of the neurotransmitter synapse (i.e. GABA). The receiving cell is less likely to fire a potential action.

Myelination: Myelination increases the distance between the action potential and the axon. If the axon is not myelinated, the membrane voltage potential decreases far closer to the cell body. A garden hose is analogous. If the axon does not myelinize, the garden pants are leaking, and a lower water pressure (which cares for waves of water pressure, the potential action) leads to the end of the pants.

3.5.3.3 Additive function

An additive function is an arithmetic f(n) function in a positive integer that the product function is the sum of its functions when a and b are compressed. F(a) + f(b) holds a completely additive function even when they are not co-prime to all positive integral parts a or b if f(b) holds f(a)+ f(b). In this sense also, analogy with fully multiplicative functions is used with complete additive. F(1)=0 if f is a full additive feature. Each fully additive function is an additive, but not the other way around.

3.5.3.4 Activation function

The threshold or transfer feature is also known as activation functions. The activating functions have been used to transform neuron activation levels into output signals. Numerous activation functions are available in the neural network. Identity function, step function, part linear function and sigmoid function are various function types.

a. Identity activation function:

The activation function of identity is also referred to as "liner activation." The Network Activation function can be shown easily to fit a line regression model of the form if the ID is used in the network $Y_i = B_0 + B_1 + \cdots + B_k B_k$ where $x_1, x_2, \dots \dots \dots, x_k$ are the k network inputs, Y_i is the i-th network output $B_1, B_2, \dots \dots .., B_k$ are the coefficients in the regression equation. Consequently, a neural network with identity activation used in all its sensors is uncommon to be found.

b. Sigmoid activation function:

Nonlinearity in the model is used in the artificial neural network sigmoid functions. The result of a linear combination of its input signals is calculated by a network neuro element using a sigmoid function. The sigmoid function makes an interface between the product and itself easier and more popular in the Neural Network.

$$\varphi(v) = \frac{1}{1+\exp(-av)} \tag{3.9}$$

Sigmoid function results are generally used in learning algorithms. The Sigmoid graph is shaped as' S'. This function is defined as an expanding function which is commonly used for neural artificial network development. Sigmoid is a function that strictly increases, and shows a balance between linear and nonlinear functions.

One - polar – is the sigmoid function.

c. Step function:

This is a unipolar threshold, known as eq (3.10).

$$\varphi(v) = \begin{Bmatrix} 1 \; if \; v \; \geq 0 \\ 0 \; if \; v > 0 \end{Bmatrix} \tag{3.10}$$

The neuron K output with a threshold is eq (3.11)

$$y(k) = \begin{Bmatrix} 1 \; if \; v_k \; \geq 0 \\ 0 \; if \; v_k \; > 0 \end{Bmatrix} \tag{3.11}$$

$v_k \; is \; the \; induced \; local \; field \; of \; the \; neuron$ (eq. (3.12))

$$v_k = \sum_{j=1}^{m} W_{kj} X_j + b_k \tag{3.12}$$

When the neuron output is 1 when the local neuron field induced is not - negative, the neuron output is 0.

d. Piece wise linear function

It can be defined as a unipolar function (eq. (3.13))

$$\varphi(v) = \begin{Bmatrix} 1, & v \geq +1/2 \\ v, +\frac{1}{2} > & v > -1/2 \\ 0, & v \leq -1/2 \end{Bmatrix} \tag{3.13}$$

When it is expected that the amplification factor is within the linear area

1. The specific circumstances of linear functions are
    - If the linear operating area is maintained without saturation, a linear combiner is produced.
    - If the linear region's amplification factor is infinitely large, it reduces to a threshold feature.

e. Learning rules in neural network

There are many different types of study rules in the neural network, usually divided into 3 categories.

- Supervised Learning
- Unsupervised Learning

a. Supervised Learning

Training sets are available for supervised learning. This type of rule includes a set of examples with proper network behavior. The inputs are given as a training in controlled learning and the expected results are achieved. Parameters in this type of study are adjusted step by step by error signal; parameters are adjusted step by step by step by error signal.

A number of examples (trainings set) together with correct network conduct are provided for the learning rule.

$$\{x1, d1\}, \{x2, d2\}, \dots \dots \dots, \{xn, dn\} \tag{3.14}$$

The network input is xn in this case and dn is the required destination input. The output is generated by input. In order to make network outputs more exact, the Study rule is employed to change network biases and weights.

We commit ourselves with supervised learning to give the system the desired answer (d) when the entry is implemented. The distance between the real response and the desired response is used to correct the network parameter externally. For example, the error can be used to change weighing in the study of input patterns or circumstances where the answer to the error is recognized. For the learning mode, the training set, several input and output patterns are needed.

b. Unsupervised learning

In unexpected learning, self-organized learning is also known. Objective output is not available in uncontrolled learning. In that case, only network input changes weights and biases. For pattern reorganization, unattended study grouping is used. The answer required is not known in unattended learning, therefore explicit error information cannot be utilized for improving network behavior. Information of this type is not available to correct the wrong answers so that learning has to be done based on observations of marginalized or unknown responses to the data.

The algorithms in unchecked learning use redundant row data, which have no etiquette for class membership or associations. In order to identify its parameters in this way, the network needs to detect any existing patterns, properties, regulations, etc. Unattended study means learning without the teacher because it is not necessary for the teacher to participate, but the teacher must set objectives. Feedback on neural networks is important as well. Feedback is called progressive learning, which for uncontrolled learning is very important.

3.5.3.5 Outputs

This layer contains also neurons that generate and display the final neural network outputs at the previous layers. Thus the artificial neural networks ' principal architectures can be divided, taking into consideration the connection and the structure of the neuronal disposition. The following are possible:

(i) Single-layer feedforward network,

(ii) Multilayer feedforward networks,

(iii) Recurrent networks and

(iv) Mesh networks.

3.5.4 Classification of Artificial Neural Networks

3.5.4.1 Artificial neural networks according to constructions

a. Single Layer Feed Forward Network

A neural network with a source node projecting the neural output level, but not one single feed or an acyclic network. The single layer refers to the calculation node output layer in one network layer Figure 3.4.

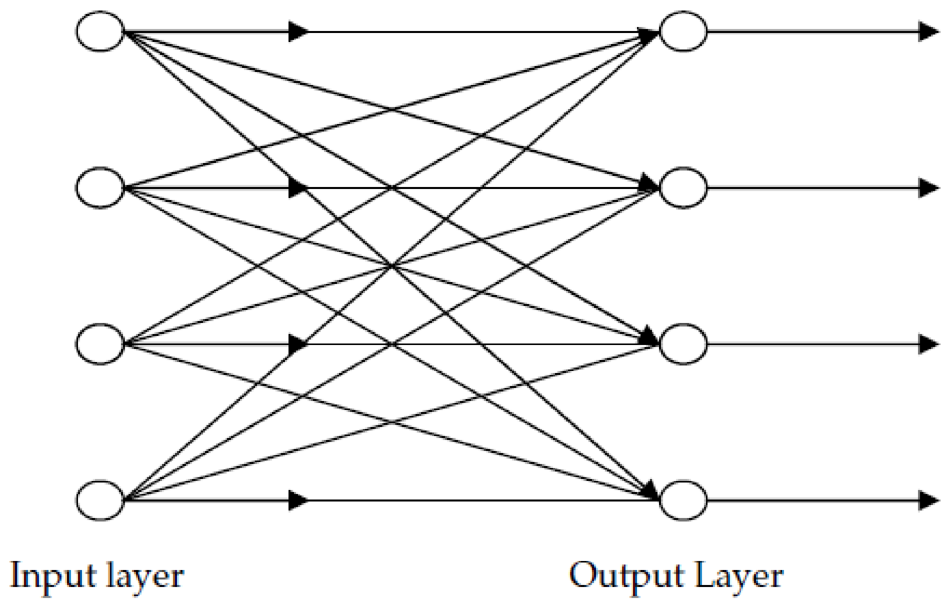

Figure 3.4: Single Layer Feed Forward Network [2]

b. Multilayer Feed Forward Network

The system consists of at least one cache layer known as clad neurons or clad unit hubs. The ability of clad neurons is to work between external information and system output and to get separate insights on a higher demand. In the system input layer, the source hubs provide the information flag for the neurons in the second layer. The third layer of inputs are the second layer of output signals, and so on. The general reaction of the system to the actuation design provided by sources in the main layer of the info is neuron output motions in the system yield level.

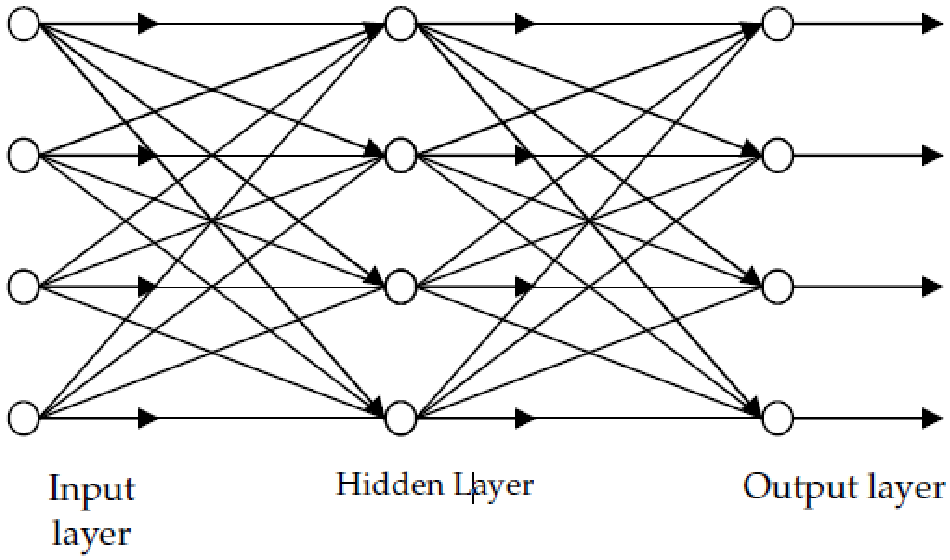

Figure 3.5: Multilayer Feed Forward Network [2]

Short feeding network characterization:

1. In general, activation is provided via ' hidden layers ' from input to output but many other architectures exist.
2. Static input - output mappings are implemented by mathematics.
3. Most popular algorithm for backpropagation supervised training:
4. Has proved useful in many practical applications as approximations of nonlinear functions and as a classification model.

c. Recurrent Network

The repetitive system in the figure is known as a forward neural system of something like the input circle, which includes at least one shrouded layer. 2.3.2. The first time. Critics could be your own critique, i.e. if your own information returns to neuronal yield. The use of unit deferring components, which leads to a unique conduct, has sometimes been criticized, since the neural system has non-direct units.

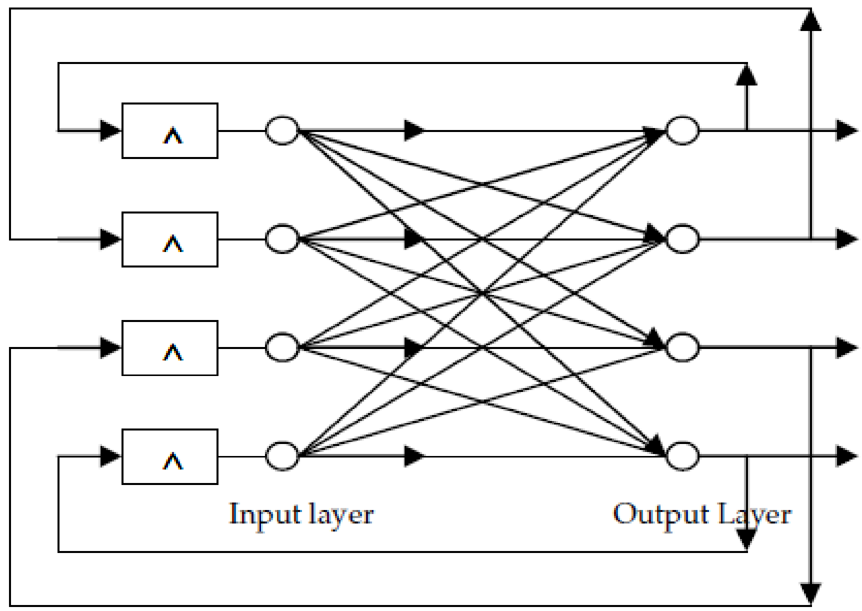

Figure 3.6: Recurrent Network [2]

Other network types are available ; Delta - bar - delta, hop field, quantifying vectors, counter propagation, probabilistic , hamming, memory boltzman , spatium - temporal patent, adaptive resonance, auto - order, recirculation etc.

The cyclic path of synaptic connections is a recurring neural network.. Basic characteristics:

1. Recurring are all biological neural networks

2. They implement dynamic systems mathematically

3. Various types of algorithms, no clear winner, are known

4. In general, theoretical and practical problems have prevented practical applications up to now.

3.5.4.2 Artificial neural networks according to learning algorithms

If a network is structured for a particular application, it will be ready for formation. At start and beginning of training, initial weights are selected randomly. Two approaches are available; controlled and unattended.

a. Supervised Training

The inputs and outputs are provided for supervised training. The data is processed by the network and the results are compared with the results. The system spreads errors to adjust network weights. This process takes place time and again when weight is constantly changed. The same data set is processed many times during networking training, as the weight of connections is always improved. The data set for the training is referred to as the "training set."

Sometimes a network can never learn. This can take place as there are no specific information in the input data to produce the required output. Networks also do not converge when there are not enough data to enable complete learning. In order to retain part of the data as a test, there should ideally be sufficient data. Multiple nodes in many layered networks can store data. To monitor the network, the supervised training must retain data for the system to test after training of a system, to determine whether the system simply saves its data.

Then, if a network cannot simply solve the problem, the designer must examine inputs and outputs, number of layers, numeric elements per layer, layer connections, transfer and training capabilities and also initial weights. The training rules govern another part of the creativity of the designer. Many laws (algorithms) are used in the training session to make the required weight adjustment feedback. The most common method is back-propagation. It is a conscious analysis and not only a technique to prevent overtraining of the network. General statistical trends in the data are the initial setup of the artificial neural network. It then' learns' about other aspects of the data which are generally falsified.

The weights can be frozen upon request, if finally the system is properly trained and no further training is necessary. This network can then be converted on certain systems into hardware. During production, other systems do not lock in, but continue learning.

b. Unsupervised or Adaptive Training

Uncontrolled (learning) is the other kind of training. The network has inputs but not desired outputs in this kind of way. The system must then determine its own functions in order to group the input data. Often this is termed autonomy or adaptation. These networks do not adjust their weight using external influences. Their performance is instead monitored internally. The networks seek the regularity or trends of the input signal and adjust it to the network functions. Although the network still needs to be able to provide information on how to organize itself without knowing whether or not this is right. These data are incorporated into the network's topology and study rules. The cooperation between the processing element clusters may be emphasized by an unchecked learning algorithm. The clusters would work together under such a scheme. If any external input activated a node within the cluster, it could increase the whole activity of the cluster. Furthermore, it can inhibit the whole cluster if external input to the cluster nodes has been decreased. Competition between processing elements could also provide a learning basis. Competitive cluster training could enhance the response of particular groups to special incentives. In that sense, these groups would be linked and a special appropriate response would be provided. The weight of the winning processing element is normally only updated when the learning contest is effective. Unattended learning is not well understood at the present time and a lot of research is in progress.

c. L**earning Laws (Algorithms)**

There are many common uses for learning laws. The majority are a variation of ' Hebb's rule ' which is the best - known and oldest.

Hebb's Rule: In the Compatibility Organization, Donald Hebb introduced this. The fundamental rule of thumb is that the weight should be strengthened if the neuron comes from another nerve and both neurons are very active (the same sign).

Hopfield Law: Increase connectivity weight by the study rate and when both active and inactive the desired outcomes and inputs.

The Delta Rule: The idea is that the strengths of the input connections can constantly be modified so that the difference of the desired output to the actual output of the element is reduced (delta).

The Gradient Descent Rule: This is similar to the Delta rule because the transfer function derivative is still used to change the delta bug prior to the application of connection weights. However, a proportional additional constant associated with the learning rate accompanies the final weight - based change factor.

Kohonen's Law: This allows processing parts to acquire or update their weight. The strongest item is declared the winner and its competitors can be inhibited and its neighbors excited. Only the winner can adjust his weight and only the winner plus neighbors can adjust it.

3.5.5 Artificial Neural Network's Output Calculations

Sensitivity, septicity and accuracy are preferred statistics for determining the performance of a classifier. Susceptibility is the estimation rate for patients with epileptic diseases, specialty is the estimation speed for healthy people and accuracy is true. Equality. These statistical numbers are calculated using (3.15), (3.16) and (3.17).

$$Sensitivity = \frac{TP}{TP+FN} \tag{3.15}$$

$$Specificity = \frac{TN}{TN+FP} \tag{3.16}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.17}$$

In the above equations, the number of TP - diagnosed epileptic patients, the total number of normal epileptic patients, for whose epileptic disease was diagnosed, and the total number of normal epileptic patients, for which FN was diagnosed.

3.6 BACKPROPAGATION

Backpropagation is a method used to compute the gradient required in artificial neural networks for the calculation of network weights. Backpropagation means "retroactive error propagation" as an error is calculated at the output and reversed across all layers of the network. It is frequently used to train deep neural networks.

Discharge is a common application of the Delta rule in multi - layer feed systems that permits iterating gradients to be measured using the chain rule for each layer. It is closely linked to Newton's Gauss algorithm and forms part of the ongoing research on the neural background.

The reverse propagation is a special case for an automatic differentiation technique. In learning, backpropagation is commonly used to adjust the weight of neurons through the gradient downward optimization algorithm to calculate the degree of loss function.

Consider, for example, the network for one training case: (1), 1,0 and therefore x1 and x2 inputs are 1 and 1. The result is parabolic, when y output is tracked against the error E on the vertical axis of the horizontal axis. The y output which reduces the E-error is the lowest parable. In a single training case, the minimum affects the horizontal axis, which means the error is zero and the network can produce an output y exactly matching the desired output t. In order to optimize the search for least errors, this reduces the problem of mapping inputs to outputs as can be seen in eq (3.18).

$$y = x_1 w_1 + x_2 w_2 \tag{3.18}$$

where x1 and x2 weights are linked to the input unit connection with the output unit. The error therefore depends on the input weights of the neuron that must ultimately be changed over the network to enable learning. The result is a parabolic bowl with a separate horizontal axis, with each weight having a vertical axis fault. The same tract would require an elliptical paraboloid k+1 for a neuron with k weights.

Any complex system can at least be abstracted in its basic abstract parts or at least separated. The build-up of several easy layers generates complexity. This article is meant to clarify how the simplest abstraction is done by neural networks. We attempt to decrease the machine learning system in NN to its basic abstract parts. We attempt to use as few mathematical equations and codes as possible and concentrate only on abstract ideas in the contrary to other messages that explain neural networks. In the largest and easiest abstract representation a monitored neural network can be described as a black box, using 2 techniques which is shown in figure 3.7.

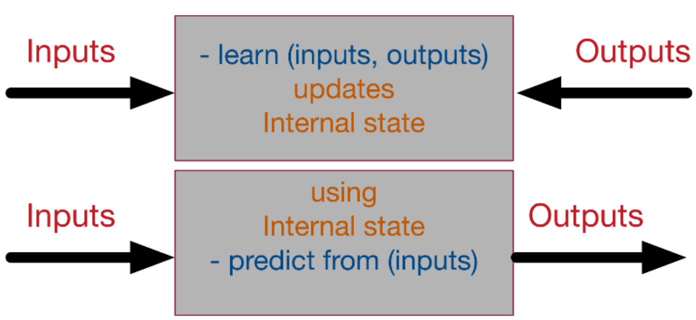

Figure 3.7: Backpropagation Training Method [24]

3.6.1 Learning as a Problem of Optimization

It helps us first of all to understand the link between the actual neuron output and the exact output of a specific training example in order to obtain a mathematical derivation of the back propagation algorithm. The neural network has 2 inputs, a output unit and no hidden unit. Each neuron uses a linear output [note 1] as opposed to most neural networks, which is the weighted amount. At first, before training, weights are altered. In this instance, the neuron will learn from examples consisting of the tuples x1, x2, t where x1 and x2 are the network inputs and t is the correct output (network output should be produced when these inputs are trained). The first network calculates a y output which can differ from t (random weights) when x1 and x2 is specified. Squared error measurement is a common method for measuring differences between the expected t and the current y output (eq (3.19)):

$$E = (t - y)^2 \tag{3.19}$$

where E is an error or discrepancy.

Consider the network for one training case, for instance: (1), 1,0 and therefore x1 and x2 inputs are 1 and 1. The result is a parabolic when the y output on the vertical axis of the horizontal axis is tracked against the E error. The y output that minimizes the E error is the minimal parabola. The minimum also affects the horizontal axis in a single training case, which means the error is zero and that the network can produce an output y that exactly matches the desired output t. This reduces the problem of mapping inputs to outputs in order to optimize the search for the least error.

The neuron output depends, however, on the sum of all the inputs weighted (eq (3.20)):

$$y = x_1 w_1 + x_2 w_2 \tag{3.20}$$

where the weights of x1 and x2 are connected to the output unit by the input unit connection. The error depends therefore on the incoming weights of the neuron, which must be changed over the network ultimately in order to allow learning. The result is a parable bowl with a separate horizontal axis and a vertical axis fault with each weight. For the same tract, an elliptical paraboloid k+1 dimensions would be required for a neuron with k weights.

3.7 TRAINING OF ARTIFICIAL NEURAL NETWORKS

The neural biological computing systems that make up animal brains are artificial neural networks or connecting systems. Such systems learn tasks by following examples, usually without task-specific programming (a step towards improvements in performance). You can learn how to identify images that contain the Cats and use the results to identify the Cats in the other images through the analysis of flames that manually mark "cat" or "no-cat." They don't know cats, for example, fur, tails, whiskers and cat - like faces. Rather, from the learning material they are processing they develop their own set of relevant features. An ANN is based on an artificial neuronal collection which is linked with the unit or node (analogous to biological neurons in the animal brain). Any link between artificial neurons (similar to a synapse) can signal them. The artificial neuron receiving the signal can proceed and signal the connected artificial neurons.

In the common ANN implementation, the signal is an actual value in the relation between artificial neurons, and the output is calculated by a nonlinear input sum function from each artificial neuron. Artificial neurons and connections usually weigh in accordance with learning. The weight increases or lowers the signal strength of the connection. An artificial neuron threshold can only be present if the aggregate signal crosses this threshold. Artificial neurons are usually arranged in layers. Inputs can be converted into different types by different layers. Signals travel from the first layer to the last layer (output), maybe several times after the layers have been passed through.

The original objective of the ANN approach was to solve problems like the human brain. The focus was over time on matching specific mental skills that lead to biodiversity. Different tasks such as computer vision, speech recognition, machine translation, social network filtering, video games and playboards and medical diagnosis are performed using ANNs. This section shows a training process that optimizes ANN performance through BP and GA algorithms.

3.7.1 Training Artificial Neural Networks with Backpropagation.

The method of gradient descent includes the derivatives of the squared error function with respect to the weight of the network. This is usually done in return. The squared error function takes the following:

$$E = \frac{1}{2}(t - y)^2 \tag{3.21}$$

where

E is the error squared,

t is the target output for a sample training, and

y is the current output of neuron output.

The 1/2 factor in the exponent is distinguished. Then the term is increased by an arbitrary learning rate. Whether a continuous coefficient is implemented now is not important.

For each neuron j, its output o_j is defined as eq (3.22)

$$o_j = \varphi(net_j) = \varphi\left(\sum_{k-1}^{n} w_{kj} o_k\right) \tag{3.22}$$

The netj input for the neuron is the weighted sum of previous neuron outputs. The ok of the input layer is simply a xk input in the network. If the neuron is in the first layer after the input layer. The input units in the neuron are numbered by n. The wkj variable shows the weight from the neuronal to the j.

β is non - linear and distinctive with the activation function. The logistic function is a commonly used activation feature as defined in eq (3.23).

$$\varphi(z) = \frac{1}{1+e^{-z}} \tag{3.23}$$

which has a convenient derivative of eq (3.24)

$$\frac{d\varphi}{da} = \varphi(a)(1 - \varphi(a)) \tag{3.24}$$

Identifying the error derivative

Twice by the chain rule is calculated the partial derivation of an error in relation to a wij eq (3.25).

$$\frac{\vartheta E}{\vartheta w_{ij}} = \frac{\vartheta E}{\vartheta o_j} \frac{\vartheta o_j}{\vartheta net_j} \frac{\vartheta net_j}{\vartheta w_{ij}} \tag{3.25}$$

In the last right side factor of the foregoing w_{ix} depends only one term in the sum net_h which is defined eq (3.26)

$$\frac{\vartheta E}{\vartheta net_j} = \frac{\vartheta}{\vartheta w_{ij}} \varphi\left(\sum_{k-1}^{n} w_{kj} o_k\right) = \frac{\vartheta}{\vartheta w_{ij}} w_{ij} o_i = o_i \tag{3.26}$$

When the first layer of the neuron after the input layer is, o_i is only X_I.

With respect to the input, only the active function parameters are a derivative of neuron j output (if logistics function is used) which is defined as eq (3.27)

$$\frac{\vartheta o_j}{\vartheta net_j} = \frac{\vartheta}{\vartheta net_j} \varphi\left(\vartheta net_j\right) = \varphi\left(\vartheta net_j\right)\left(1 - \varphi\left(\vartheta net_j\right)\right) \tag{3.27}$$

That is why a differentiated activation function is required for back propagation. (The activation function of ReLU, which cannot be differentiated by 0, is very popular in the recent past, for instance in AlexNet)

The first factor to evaluate is simple, because then oj = y and if the neuron is in the output layer as eq (3.28).

$$\frac{\vartheta E}{\vartheta o_j} = \frac{\vartheta E}{\vartheta y} = \frac{\vartheta}{\vartheta y} \frac{1}{2} (t - y)^2 = y - t \tag{3.28}$$

But if j is within an arbitrary inner network layer, it is less obvious to determine derivative E with respect to the oj which is defined in eq (3.29).

$$\frac{\vartheta E(o_j)}{\vartheta o_j} = \frac{\vartheta E(net_1, net_2 \ldots net_u)}{\vartheta o_j} \tag{3.29}$$

Taking into account E as a function where all neurons are inputs (L=1, 2 ... u) as can be seen in eq (3.30)

$$\frac{\vartheta E}{\vartheta o_j} = \sum_{\partial \in L} \frac{\vartheta E}{\vartheta net_j} \frac{\vartheta net_j}{\vartheta o_j} = \sum_{\partial \in L} \frac{\vartheta E}{\vartheta o_j} \frac{\vartheta o_j}{\vartheta net_j} w_{j\partial} \tag{3.30}$$

Thus, if all derivatives are known with respect to oj that which is closer to the neuron output, the derivative with regards to oj can be calculated. Combining everything (eq(3.31)):

$$\frac{\vartheta E}{\vartheta w_{ij}} = \partial_j o_i \tag{3.31}$$

In 1960 Henry J. Kelley and 1961 Arthur E. Bryson derived the basics of continuous back-propagation from control theory. They have used dynamic programming principles. A simple derivation based only on the chain rule was published in 1962 by Stuart Dreyfus. This is a multi - phase method used to optimize dynamic systems by Bryson and Ho in 1969.

In the beginning of the 1960's, background propagation was derived by a number of researchers and launched by Seppo Linnainmaa as early as 1970, for instance Arthur E. Bryson and Yu-Chi Ho, among the researchers of the 1960s. After a careful analysis in his PhD dissertation in 1974, the first person to propose use for neural networks in the United States was Paul Werbos. The prize for work of David E. Rumelhart was awarded in 1986 to Geoffen E. Hinton, Ronald J. Williams & James McClelland.

The general automated differentiation (AD) method of discernible connected networks for nestled differentiating functions was issued by Linnainmaa in 1970. This is the backbone that even with sparsely networked networks is efficient. Backpropagation was used in 1973 by Dreyfus to adapt controller parameters in proportion with error gradients. Werbos noted the potential to apply this principle to neural artificial networks in 1974, and in its current application on neural networks employed the Linnainmaa AD Method in 1982.

In 1986, Rumelhart, Hinton and Williams proved that the process could produce useful internal images of inbound data in hidden layers in neural networks. Wan was the first winner in 1993 of an international competition for pattern recognition. In the 2000s it was disadvantageous, but it came back in 2010 with cheap, high-performance computer systems based on GPUs. In particular, research was conducted into linguistic structures, where connectivity models could explain a number of first language and second language learning phenomes using this algorithm.

4. IMPLEMENTATION AND RESULTS

4.1 THE METHODS USED FOR THE CLASSIFICATION

It includes machine learning algorithms to deduce the feeling from a knowledgeable data set. This approach to the classification of feelings is monitored and enables for efficient text classification. The classification of machine learning requires two distinct records, specifically for practice and testing. An automatic classifier uses a training set to know and distinguish document characteristics and a test set for verification of the automatic classifier's results. There are several methods used to classify reviews by the machine. Technical machine learning like NB, ME and SVM has improved text categorization performance.

4.1.1 Naïve Bayes

The strategy for classification of text is one of the most efficient, commonly used, and easy. The first thing to do with this approach is to calculate the preceding probability for an entity as a class, and the final probability is to multiply the previous probability by the probability. The method is Naïve in that all the words in the text are supposed to be separate. This hypothesis makes implementing it easier, but less precise.

$$P\left(\frac{C}{D}\right) = \frac{P(\frac{D}{C})P(C)}{P(D)} \qquad (4.1)$$

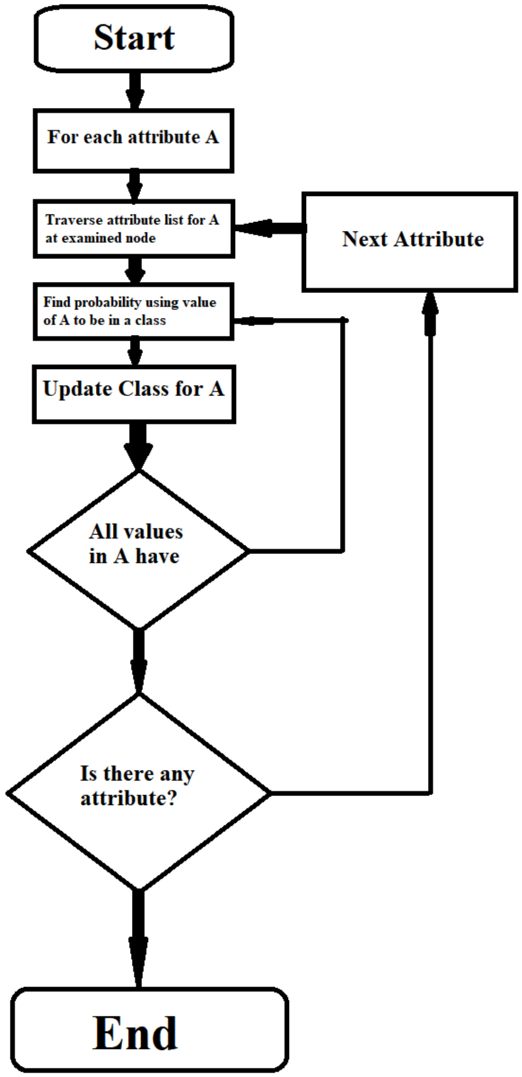

Figure 4.1: Naïve Bayes Classification process

4.1.2 Support Vector Machines (SVM)

It is also used for classifying text according to a discriminatory classifier. The strategy is based on the structural risk reduction principle. First of all, the training data points are divided into two classes on the basis of a defined choice or surface. The choice builds on the vectors of assistance chosen in the training set. The multiclass SVM is used for sentiment analysis among the various versions of SVM. The centroid classification algorithm calculates for each training class the centroid vector. Similarities are then calculated between a document and all central points and a class based on these values is allocated to the document.

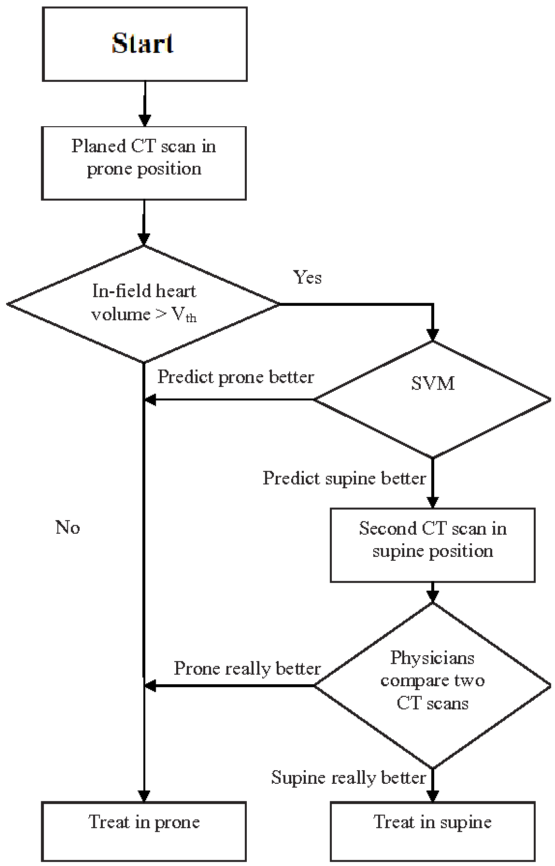

Figure 4.2: Support Vector Machine Classification process

4.1.3 Artificial Neural Network

A significant instrument for classification has appeared as a neural network. In the last ten years the classification of neural networks has become a promising solution to different standard classification techniques. The neural network can manage the correlation / dependency between input factors with the suitable network structure. Neural networks have the benefit of the following theory. First, neural networks are auto adaptive techniques driven by information, because their adaptation to the information can be done without an explicit functional or distributional form specification for the underlying model. Secondly, they are universally functional approximates because neural networks can arbitrarily approximate any feature. Since each classification process aims a functional link between the member group and the object's characteristics, it is undoubtedly essential to accurately identify this fundamental function.

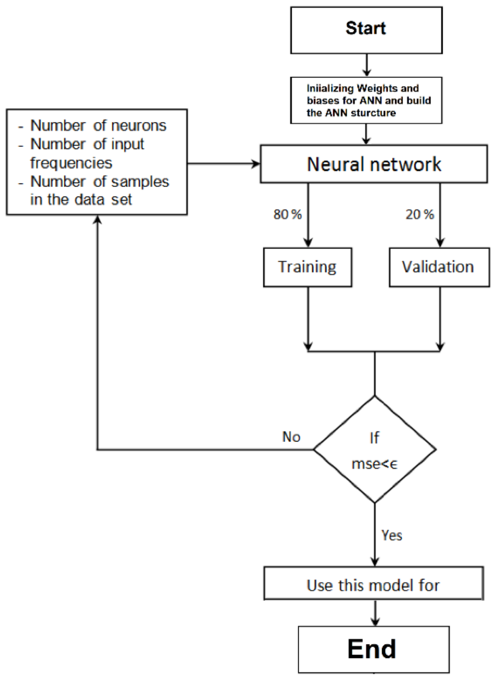

Figure 4.3: Artificial Neural Network Process

Backpropagation is a way to calculate the gradient needed for network weight calculation in artificial neural networks. Backpropagation implies "retroactive mistake spread" as the result is calculated and reversed over all network layers. It is often used for the formation of profound neural networks.

Discharge is a common application of the Delta rule in multi-layer feeding schemes which allows the measurement of the iterating gradients using each layer's chain rule. It is strongly related to the Gauss algorithm of Newton and is an integral component of continuing neural studies.

Reverse propagation is a unique situation for the method of automatic differentiation. Back propagation is frequently used when learning, in order to calculate the loss level by adjusting weight of the neurons using the gradient downward optimization algorithm.

For example, for one case of training, consider the network: (1), 1.0 and hence inputs from x1 and x2 are 1 and 1. The outcome is parabolic if y output is monitored for E mistake on the horizontal axis vertical. The smallest parable is the y-output that reduce the e-error. The minimum impacts the horizontal axis in a single training situation, which results in a null error and network y output that matches the required t output may be generated. This decreases the issue of mapping inputs to outputs in order to optimize the search for fewer mistakes (eq (4.2)).

$$y = x_1 w_1 + x_2 w_2 \tag{4.2}$$

where the weights x1 and x2 have to be connected to the output unit input unit connection. This mistake thus relies on the weight of the input of the neuron to be altered over the network in order to allow learning. The outcome is a parabola with a distinct horizontal axis and a vertical axis defect for each weight. For a neuron with k weights, the same tract would involve an elliptical paraboloid k+1.

Any complex system can at least be abstracted into its basic abstract parts. Multiple layers accumulate, creating complexity. This article explains how neural networks function with the simplest abstraction. We attempt to decrease the machine learning system in NN to its basic abstract elements. We attempt to use as few mathematical equations and coding as possible as opposed to other articles explaining neural networks and concentrate solely on abstract ideas. A monitored neural network can be described as a blackbox using 2 learning and prediction techniques with the largest and easiest abstract representation:

It helps us first of all to understand the link between the actual neuron output and the exact output of a specific training example in order to obtain a mathematical derivation of the back propagation algorithm. There are 2 inputs, an output unit and no concealed unit in the neural network. In contrast to many neural networks, each neuron utilizes a linear output [note 1], which is the weighted quantity. At first, before training, weights are altered. In this instance, the neuron will draw attention from examples that are made up of x1, x2, t where the x1 and x2 are the network inputs and t the correct output. The first network calculates a y output which can differ from t (random weights) when x1 and x2 is specified. Squared error measurement is a common method for measuring differences between the expected t and the current y output (eq (4.3)):

$$E = (t - y)^2 \tag{4.3}$$

where E is an error or discrepancy.

Consider the network for one training case, for instance: (1), 1,0 and therefore x1 and x2 inputs are 1 and 1. The result is a parabolic when the y output on the vertical axis of the horizontal axis is tracked against the E error. The y output that minimizes the E error is the minimal parabola. In a single training case, the minimum affects the horizontal axis, which means that the error is zero and the network can produce a y output that exactly matches the desired t output. This decreases the issue of mapping inputs to outputs, so that searches for the least mistake are optimized.

The neuron output depends, however, on the sum of all the inputs weighted (eq (4.4)):

$$y = x_1 w_1 + x_2 w_2 \tag{4.4}$$

where the weights of x1 and x2 are connected to the output unit by the input unit connection. The error depends therefore on the incoming weights of the neuron, which must be changed over the network ultimately in order to allow learning. The result is a parable bowl with a separate horizontal axis and a vertical axis fault with each weight. For the same tract, an elliptical paraboloid k+1 dimensions would be required for a neuron with k weights.

4.2 RESULTS

This experiment proves the performance of the ANN-BP against Search Vector Machine (SVM) and Naïve bias algorithms which was developed using MATLAB 2015, the construction of ANN-BP method consists mainly of 817 inputs, two hidden layers of 30 and 10 neurons respectively, and output layer is consisted of one output where in this study sigmoid function has been used.

This dataset is originally taken from UCI machine learning repository which contains 817 features and 1 result attribute which in this proposed study was used as a target output where the instance's target is negative the output is 0 otherwise the output is 1.

The features of the dataset were taken according to the frequency of each word from the whole dataset which is showed in figure 4.4. Each sentence contains several (one, two, three … etc) amount of words to the total words from the whole dataset.

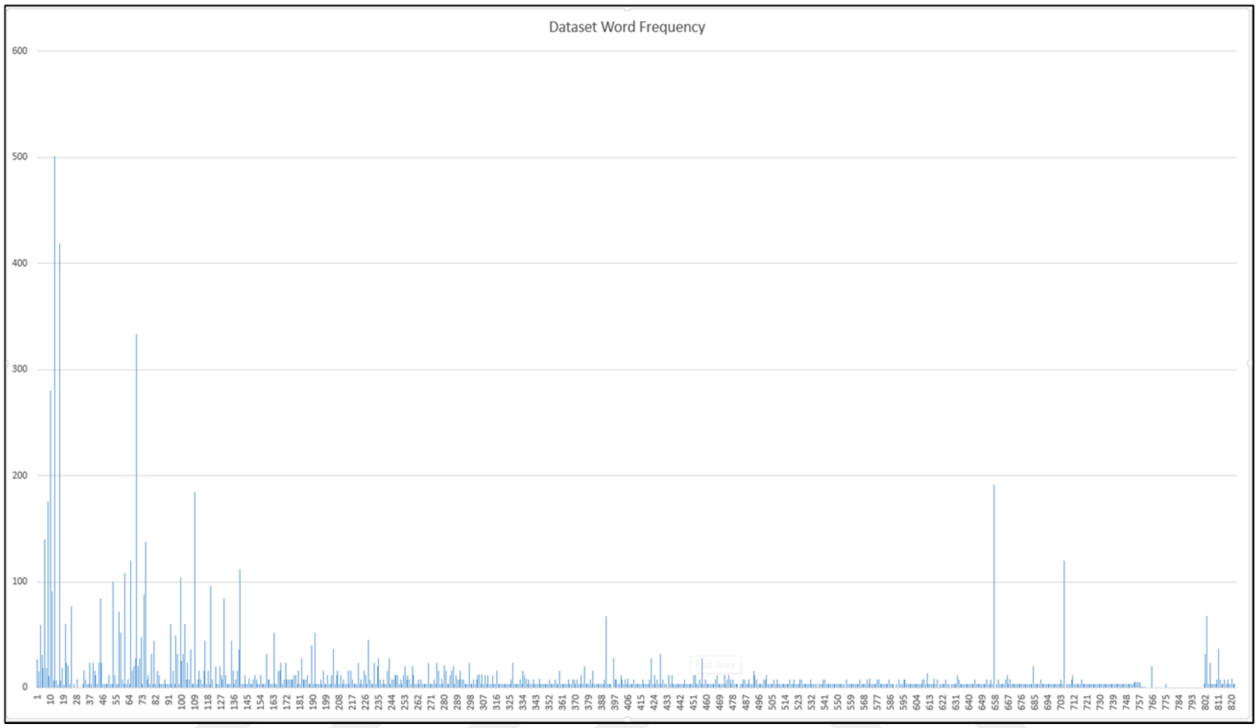

Figure 4.4: The dataset after converting it to a word frequency form

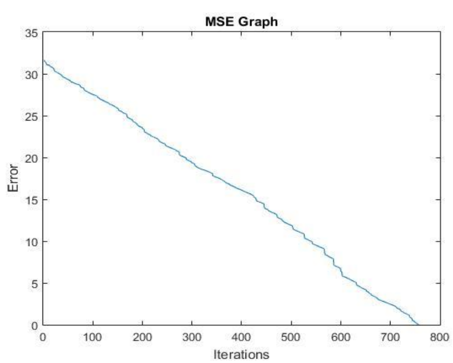

Figure 4.5: The results of the implementation using Naïve Bias

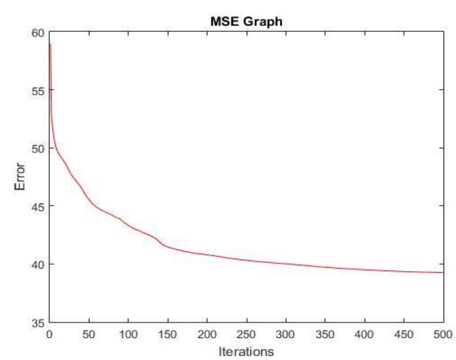

Figure 4.6: The results of the implementation using SVM

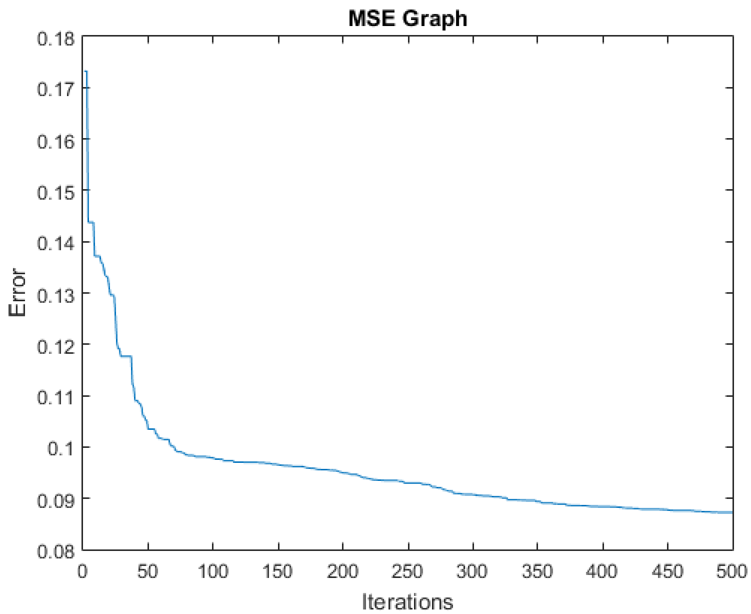

Figure 4.7: The results of the implementation using ANN

Table 4.1: Performance Comparison between the Three Methods

Method	Search Vector Machine	Naïve Bias	Artificial Neural Network
MSE	0.124799844236611	0.153262518968134	**0.095935**
TRAINING ACCURACY	94.216468187084922	84.673748103186639	**95.406539**
TESTING ACCURACY	93.816468187084922	79.724770642201833	**94.981685**

Table 4.1 shows the capabilities of the neural networks (compared to Naïve Bias and Search Vector Machine) on how to analyze sentiments according to specific word within the sentence. Our Neural network approach showed that the standard results for zero sentence is positive which make sense when we talk about the satisfaction of the costumer who never commented on the product. This makes ANNs in general better than a lot of classification methods (including Naïve bias and Support Vector machine), due to their focus on the relationships between attribute to another.

5. CONCLUSION

Sentiment analysis has become a significant research problem for the measurement of the big number of unstructured data. Companies and people are now making attempts to find the best feel analysis scheme. In sentiment analysis, some of the algorithms are used to produce excellent results, but no method can solve all the issues. Most of the scientists have noted that SVMs have elevated precision than other algorithms, but have restrictions as well. Our research focuses on master teaching methods and the use of neural artificial networks (ANN) for sentiment classification and analysis in order to overcome the limitations of certain methods. In our study, implementations of ANN suggest that the classification of the best artificial neural network would be improved and the logic fused.

5.1 SUGGESTIONS

In this research work, we perform the Naïve bias, SVM, and ANN for sentiment analysis using the research datasets. For future suggestions we want to perform below points:

- The current evaluation is based on single class problems for classification, however under the real time scenario this will not be the case always. So we suggest evaluating the performance of proposed model using multi-class datasets.

Second point is, the consideration of more real time will be the interesting future direction for this research work.

REFERENCES

[1] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in Robots and Biological Systems: Towards a New Bionics? vol. 102 of NATO ASI Series, pp. 703–712, Springer, Berlin, Germany, 1993.

[2] X. Yao, "Evolving artificial neural networks," Proceedings of the IEEE, vol. 87, no. 9, pp. 1423–1447, 1999.

[3] E. Alba and R. Martí, Metaheuristic Procedures for Training Neural Networks, Operations Research/Computer Science Interfaces Series, Springer, New York, NY, USA, 2006.

[4] J. Yu, L. Xi, and S. Wang, "An improved particle swarm optimization for evolving feedforward artificial neural networks," Neural Processing Letters, vol. 26, no. 3, pp. 217–231, 2007.

[5] M. Conforth and Y. Meng, "Toward evolving neural networks using bio-inspired algorithms," in IC-AI, H. R. Arabnia and Y. Mun, Eds., pp. 413–419, CSREA Press, 2008.

[6] Y. Da and G. Xiurun, "An improved PSO-based ANN with simulated annealing technique," Neurocomputing, vol. 63, pp. 527–533, 2005.

[7] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 694–713, 1997.

[8] D. Rivero and D. Periscal, "Evolving graphs for ann development and simplification," in Encyclopedia of Artificial Intelligence, J. R. Rabuñal, J. Dorado, and A. Pazos, Eds., pp. 618–624, IGI Global, 2009.

[9] H. M. Abdul-Kader, "Neural networks training based on differential evolution algorithm compared with other architectures for weather forecasting34," International Journal of Computer Science and Network Security, vol. 9, no. 3, pp. 92–99, 2009.

[10] K. K. Kuok, S. Harun, and S. M. Shamsuddin, "Particle swarm optimization feedforward neural network for modeling runoff," International Journal of Environmental Science and Technology, vol. 7, no. 1, pp. 67–78, 2010.

[11] B. A. Garro, H. Sossa, and R. A. Vázquez, “Back-propagation vs particle swarm optimization algorithm: which algorithm is better to adjust the synaptic weights of a feed-forward ANN?” International Journal of Artificial Intelligence, vol. 7, no. 11, pp. 208–218, 2011.

[12] B. Garro, H. Sossa, and R. Vazquez, “Evolving neural networks: a comparison between differential evolution and particle swarm optimization,” in Advances in Swarm Intelligence, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds., vol. 6728 of Lecture Notes in Computer Science, pp. 447–454, Springer, Berlin, Germany, 2011.

[13] B. A. Garro, H. Sossa, and R. A. Vazquez, “Design of artificial neural networks using a modified particle swarm optimization algorithm,” in Proceedings of the International Joint Conference on Neural Networks (IJCNN '09), pp. 938–945, IEEE, Atlanta, Ga, USA, June 2009.

[14] B. Garro, H. Sossa, and R. Vazquez, “Design of artificial neural networks using differential evolution algorithm,” in Neural Information Processing. Models and Applications, K. Wong, B. Mendis, and A. Bouzerdoum, Eds., vol. 6444 of Lecture Notes in Computer Science, pp. 201–208, Springer, Berlin, Germany, 2010.

[15] B. A. Garro, H. Sossa, and R. A. Vazquez, “Artificial neural network synthesis by means of artificial bee colony (abc) algorithm,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC '11), pp. 331–338, IEEE, New Orleans, La, USA, June 2011.

[16] R. C. Eberhart, Y. Shi, and J. Kennedy, Swarm Intelligence, The Morgan Kaufmann Series in Evolutionary Computation, Morgan Kaufmann, Boston, Mass, USA, 1st edition, 2001.

[17] M. Chen, “Second generation particle swarm optimization,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08), pp. 90–96, Hong Kong, June 2008.

[18] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in Proceedings of the Congress on Evolutionary Computation (CEC '99), vol. 3, IEEE, Washington, DC, USA, July 1999.

[19] D. Kotzias, M. Denil, N. D. Freitas, P. Smyth, From Group to Individual Labels using Deep Features, University of California, 2015, link: https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences [Access:21/10/2019]..

[20] N. Higashi and H. Iba, “Particle swarm optimization with Gaussian mutation,” in Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03), pp. 72–79, IEEE, April 2003.

[21] S. Mohais, R. Mohais, C. Ward, and C. Posthoff, “Earthquake classifying neural networks trained with random dynamic neighborhood PSOs,” in Proceedings of the 9th Annual Genetic and Evolutionary Computation Conference (GECCO '07), pp. 110–117, ACM, New York, NY, USA, July 2007

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, D. E. Rumelhart, J. L. McClelland, and PDP Research Group, Eds., pp. 318–362, MIT Press, Cambridge, Mass, USA, 1986.

[23] J. A. Anderson, An Introduction to Neural Networks, The MIT Press, 1995.

[24] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” Proceedings of the IEEE, vol. 78, no. 10, pp. 1550–1560, 1990.

[25] D. N. A. Asuncion, UCI machine learning repository, 2007.

[26] R. A. V. E. De Los Monteros and J. H. Sossa Azuela, “A new associative model with dynamical synapses,” Neural Processing Letters, vol. 28, no. 3, pp. 189–207, 2008.

[27] B. A. Garro and R. A. Vázquez, “Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms,” *Computational Intelligence and Neuroscience*, 2015.

[28] V. G. Gudise and G. K. Venayagamoorthy, “Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks,” in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS’03 (Cat. No.03EX706)*, 2003, pp. 110–117.

[29] “Pima Indians Diabetes Database.” [Online]. Available: https://www.kaggle.com/uciml/pima-indians-diabetes-database. [Accessed: 20-Dec-2018].

[30] BAGLEY, J. D. The Behavior of Adaptive Systems Which Employ Genetic and Correlative Algorithms. PhD thesis, University of Michigan, Ann Arbor, 1967.

[31] BAUER, P., BODENHOFER, U., AND KLEMENT, E. P. A fuzzy algorithm for pixel classification based on the discrepancy norm. In Proc. 5th IEEE Int. Conf. on Fuzzy Systems (New Orleans, September 1996), vol. III, pp. 2007–2012.

[32] BODENHOFER, U. Tuning of fuzzy systems using genetic algorithms. Master's thesis, Johannes Kepler Universit¨at Linz, March 1996.

[33] BODENHOFER, U., AND BAUER, P. A formal model of interpretability of linguistic variables. In Interpretability Issues in Fuzzy Modeling, J. Casillas, O. Cord´on, F. Herrera, and L. Magdalena, Eds., vol. 128 of Studies in Fuzziness and Soft Computing. Springer, Berlin, 2003, pp. 524–545.

[34] BODENHOFER, U., AND HERRERA, F. Ten lectures on genetic fuzzy systems. In Preprints of the International Summer School: Advanced Control—Fuzzy, Neural, Genetic, R. Mesiar, Ed. Slovak Technical University, Bratislava, 1997, pp. 1–69.

[35] BODENHOFER, U., AND KLEMENT, E. P. Genetic optimization of fuzzy classification systems — a case study. In Computational Intelligence in Theory and Practice, B. Reusch and K.-H. Temme, Eds., Advances in Soft Computing. Physica-Verlag, Heidelberg, 2001, pp. 183–200.

[36] BONARINI, A. ELF: Learning incomplete fuzzy rule sets for an autonomous robot. In Proc. EUFIT'93 (1993), vol. I, pp. 69–75. 121 122 BIBLIOGRAPHY

[37] BONARINI, A. Evolutionary learning of fuzzy rules: Competition and cooperation. In Fuzzy Modeling: Paradigms and Practice, W. Pedrycz, Ed. Kluwer Academic Publishers, Dordrecht, 1996, pp. 265–283.

[38] BONARINI, A. Anytime learning and adaptation of hierarchical fuzzy logic behaviors. Adaptive Behavior 5 (1997), 281–315.

[39] BULIRSCH, R., AND STOER, J. Introduction to Numerical Analysis. Springer, Berlin, 1980.

[40] E. Augustin, M. Carre, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux. Rimes evaluation campaign for handwritten mail processing. In IWFHR, pages 231–235, 2006.

[41] A. B. Bernard, F. Menasri, R. H. Mohamad, C. Mokbel, C. Kermorvant, and L. Sulem. Dynamic and contextual information in HMM modeling for handwritten word recognition. IEEE Trans. PAMI, 33:2066–2080, 2011.

[42] A. Britto, R. Sabourin, F. Bortolozzi, and C. Suen. A two-stage HMM-based system for recognizing handwritten numeral strings. In ICDAR, pages 396–400, 2001.

[43] Y. Chherawala, P. P. Roy, and M. Cheriet. Feature design for offline Arabic handwriting recognition: handcrafted vs automated?, 678-682. In ICDAR, 2013.

[44] P. Doetsch, M. Hamdani, H. Ney, A. Gimenez, J. Andres-Ferrer, and A. Juan. Comparison of Bernoulli and Gaussian HMMs using a vertical repositioning technique for off-line handwriting recognition. In ICFHR, page 37, 2012.

[45] A. El-Yacoubi, R. Sabourin, C. Y. Suen, and M. Gilloux. An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. IEEE T-PAMI, 21:752–760, 1999.

[46] H. Fujisawa. Forty years of research in character and document recognition an industrial perspective. Pattern Recognition, 41:2435–2446, 2008.

[47] A. Graves. RNNLIB: A recurrent neural net-work library for sequence learning problems. http://sourceforge.net/projects/rnnl/.

[48] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. IEEE T-PAMI, 31:855–868, 2009.

[49] E. Grosicki and H. E. Abed. ICDAR 2009 hand-writing recognition competition. In ICDAR, page 13981402, 2009.

[50] S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, 1998.

[51] G. E. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. Neural Comput., 18:1527–1554, 2006.

[52] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. IEEE Trans. PAMI, 20(3):226–239, 1998.

[53] A. L. Koerich, R. Sabourin, and C. Y. Suen. Recognition and verification of unconstrained handwritten words. IEEE Trans. PAMI, 27:1509– 1522, 2005.

[54] U. V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. IJPRAI, 15:65–90, 2001.

[55] F. Menasri, J. Louradour, A. L. Bianne-Bernard, and C. Kermorvant. The A2iA French handwriting recognition system at the RIMES-ICDAR2011 competition. In DRR XIX, 2012.

[56] R. Mohamad, L. Likforman-Sulem, and C. Mokbel. Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. IEEE Trans. PAMI, 31:1165–1177, 2009.

[57] M. P. Schambach and S. F. Rashid. Stabilize sequence learning with recurrent neural networks by forced alignment. In ICDAR, pages 1270–1274, 2013.

[58] A. Vinciarelli and S. Bengio. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Trans. PAMI, 26:709–720, 2004.

[59] S. Young. The HTK Book, Version 3.4. Cambridge Univ. Eng. Dept., 2006.

[60] N. R. Stepanek and B. Dorn, “Automated Analysis of Lecture Video Engagement Using Student Posts,” in *Artificial Intelligence in Education*, 2017, pp. 565–569.

[61] R. L. U. Cazarez and C. L. Martin, “Neural Networks for Predicting Student Performance in Online Education,” *IEEE Latin America Transactions*, vol. 16, no. 7, pp. 2053–2060, Jul. 2018.

[62] G. Zhao *et al.*, “Deep convolutional neural network for drowsy student state detection,” *Concurrency and Computation: Practice and Experience*, vol. 30, no. 23, p. e4457, 2018.

[63] C. Lee *et al.*, “PSO-Based Fuzzy Markup Language for Student Learning Performance Evaluation and Educational Application,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2618–2633, Oct. 2018.

[64] Y. Ma, C. Cui, X. Nie, G. Yang, K. Shaheed, and Y. Yin, “Pre-course student performance prediction with multi-instance multi-label learning,” *Sci. China Inf. Sci.*, vol. 62, no. 2, p. 29101, Feb. 2019.

[65] P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.

[66] Heikki, Mannila, Data mining: machine learning, statistics, and databases, IEEE, 1996.

[67] U. Fayadd, Piatesky, G. Shapiro, and P. Smyth, From data mining to knowledge discovery in databases, AAAI Press / The MIT Press, Massachusetts Institute Of Technology. ISBN 0–262 56097–6, 1996.

[68] J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” Morgan Kaufmann, 2000.

[69] Alaa el-Halees, “Mining students data to analyze e-Learning behavior: A Case Study”, 2009.

[70] U. K. Pandey, and S. Pal, “Data Mining: A prediction of performer or underperformer using classification”, (IJCSIT) International Journal of Computer Science and Information Technology, Vol. 2(2), pp.686-690, ISSN: 0975-9646, 2011.

[71] S. T. Hijazi, and R. S. M. M. Naqvi, “Factors affecting student’s performance: A Case of Private Colleges”, Bangladesh e-Journal of Sociology, Vol. 3, No. 1, 2006.

[72] Z. N. Khan, “Scholastic achievement of higher secondary students in science stream”, Journal of Social Sciences, Vol. 1, No. 2, pp. 84-87, 2005..

[73] Galit.et.al, “Examining online learning processes based on log files analysis: a case study”. Research, Reflection and Innovations in Integrating ICT in Education 2007.

[74] Q. A. AI-Radaideh, E. W. AI-Shawakfa, and M. I. AI-Najjar, “Mining student data using decision trees”, International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan, 2006.

[75] U. K. Pandey, and S. Pal, “A Data mining view on class room teaching language”, (IJCSI) International Journal of Computer Science Issue, Vol. 8, Issue 2, pp. 277-282, ISSN:1694-0814, 2011.

[76] Shaeela Ayesha, Tasleem Mustafa, Ahsan Raza Sattar, M. Inayat Khan, “Data mining model for higher education system”, Europen Journal of Scientific Research, Vol.43, No.1, pp.24-29, 2010.

[77] M. Bray, The shadow education system: private tutoring and its implications for planners, (2nd ed.), UNESCO, PARIS, France, 2007.

[78] B.K. Bharadwaj and S. Pal. “Data Mining: A prediction for performance improvement using classification”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 9, No. 4, pp. 136-140, 2011.

[79] J. R. Quinlan, “Introduction of decision tree: Machine learn”, 1: pp. 86-106, 1986.

[80] Vashishta, S. (2011). Efficient Retrieval of Text for Biomedical Domain using Data Mining Algorithm. IJACSA - International Journal of Advanced Computer Science and Applications, 2(4), 77-80.

[81] Kumar, V. (2011). An Empirical Study of the Applications of Data Mining Techniques in Higher Education. IJACSA - International Journal of Advanced Computer Science and Applications, 2(3), 80-84. Retrieved from http://ijacsa.thesai.org.

[82] B. Kumar and S. Pal, “Mining Educational Data to Analyze Students Performance,” International Journal of Advanced Computer Science and Applications, vol. 2, no. 6, 2011.

[83] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002, pp.79–86.

[84] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,”in Proceedings of the 12th international conference on World Wide Web. ACM, 2003, pp. 519–528.

[85] Neethu M S and Rajasree R,” Sentiment Analysis in Twitter using Machine Learning Techniques”, 4th ICCCNT 2013 July 4 - 6, 2013, Tiruchengode, India.

[86] S.B.Kotsianta, I.D. Zaharakis and P.E. Pintelas.” Machine Learning: a review of classification and combining technique”,Springer,(pp.159-190).-2006.

[87] G. Vinodhini and RM. Chandrasekaran,”Sentiment Analysis and Opinion Mining: A survey”, International Journal of Advanced.

[88] S. Ertekin,” Learning in extreme conditions: Online and active learning with massive, imbalanced and noisy data”, Electronic Theses and Dissertations for Graduate School.

[89] Eui-Hong (Sam) Han, George Karypis and VipinKumar, “Text Categorization Using Weight Adjusted k-Nearest Neighbour Classification”, 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), (pp. 53-65).-2001.

[90] ZHU Jian, XU Chen, and WANG Han-shi, “Sentiment classification using the theory of ANNs”, The Journal of China Universities of Posts and Telecommunications, July 2010, 17(Suppl.): 58–62.

[91] Pranali Borele, Dilipkumar A. Borikar, “An Approach to Sentiment Analysis using Artificial Neural Network with Comparative Analysis of Different Techniques”, IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. V (Mar-Apr. 2016), PP 64-69.