“Sentiment Analysis using Deep Neural Networks”

A PROJECT REPORT

SUBMITED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
SOFTWARE ENGINEERING

Submitted by

Anupam Khera

(2K19/SWE/02)

Under the supervision of

Dr. Shailender Kumar
Associate Professor
Department of Computer Science & Engineering
Delhi Technological University, Delhi

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2021

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Anupam Khera (2K19/SWE/02), student of M. Tech (SWE) hereby declare that the project report entitled **"Sentiment Analysis using Deep Neural Networks"** which is submitted by me to Department of Computer Science, Delhi Technological University, Delhi in partial fulfilment of requirement for the award of the degree of Master of Technology in Software Engineering, has not been previously formed the basis for any fulfilment of requirement in any degree or other similar title or recognition. This report is an authentic record of my work carried out during my degree under the guidance of Dr. Shailender Kumar.

Anupam Khera

Place: Delhi **Anupam Khera**

Date: 16th June, 2021 **(2K19/SWE/02)**

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Sentiment analysis using deep neural network" which is submitted by Anupam Khera, 2K19/SWE/02, Department of Computer Science, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

26/06/2021

Place : Delhi

Date : 16th June, 2021

Dr. Shailender Kumar

SUPERVISOR

Associate Professor

Dept. of Computer Science & Engineering

ACKNOWLWDGEMENT

I am very thankful to **Dr. Shailender Kumar** (Associate Professor, Department of Computer Science) and all the faculty members of the Department of Computer Science and Department of Software Engineering at DTU. They provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing me with the laboratories, infrastructure, testing facilities and environment which allowed me to work without any obstructions.

I would also like to appreciate the support provided to me by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Anupam Khera

(2K19/SWE/02)

ABSTRACT

Sentiment analysis is a context based mining of text which extract and identifies subjective information from text or sentence provided. It is the most common text classification technique that analyse incoming text and interprets if the given data produces positive, negative or neutral sentiment. Sentiment analysis can help in understanding the data by using several learning algorithms which were earlier only understood by humans. Different organisation can grow their businesses by understanding behind their brands, products or services.

Identifying sentiments of customer about products and services is a game changing strategy if company wants to take an edge over its competitors.

CONTENTS

List of Figures

List of Tables

List of symbols and abbreviations

Abbreviations	**Full Form**
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
BPTT	Back Propagation Through Time
NLP	Natural Language Processing
CNN	Convolutional Neural Networks

CHAPTER 1

INTRODUCTION

1.1 General

The 21st century is an information era. I have lot of unstructured and unorganised data which travel from one place to another. There is billions of conversation happening right now between among people, lots of tweets are currently taking place on Twitter, millions of emails are sent every day lots of post are there on Facebook and millions of reviews are present on different websites such as Amazon, Flipkart. Now this data is waiting to get processed. There is lots of information present in this data but as the data is unstructured and unorganised different techniques and mediums are required to get information from this data.

Every text statement made by humans have sentiment value to its which could be understood and then can harvest great meaning in terms of business opportunity it can help business grow so as I can understand the customers' demands. Lot of people on different websites reviews products of different brands, those reviews could be analysed and taken into consideration while planning the future growth of a company this can give an edge to a company over its competitors is they can easily understand what a consumer really demands. Sentiment values are very useful to understand and carry the meaning.

1.2 Problem Formulation

Sentiment analysis is the prediction of emotions behind sentences, text, words which can generate a lot of information about opinions different decisions that particular sentence wants to say this information can have great value if used properly for businesses or study of human behaviour. Today lots of analytical companies use data processing to gain information about people and their opinion about different products, take an example of Twitter there are millions of tweets which travel from one place to other carrying different meanings this could be understand, computer does that job in fraction of time. This information can be useful to know about the sentiments of people about their opinion about different products they purchase about the different destinations they travel around the world about different government policies what they think about as people express lot of things on social media today not everything is hidden so those statements or text could be analysed and used as a tool to understand sentiments values of the people carrying those statements.

In this project, I take the idea to analyse those text or statements made by any human and try to get valuable information out of those text or statements information which could be useful to understand the sentiment behind that statement in this project on the basis of their value in three categories that are positive neutral and negative.

There are lot of techniques available in the market right now through which a classifier could build is Deep neural network. Deep neural networks have been present for over two decades now and already very advance but still there is a lot of scope for more improvement one such deep neural network that is good with sequential data has already producing excellent results which is long short term memory neural network that comes under the recurrent neural network.

Now these mathematical models do not understand text or any other sequential data only take numerical values as input so natural language Processing concepts will help in retrieving information from text data so that that information could be fed as an input to our model which will after all classify particular statement into a sentiment that are positive neutral negative.

1.3 Objective of the Project

In this project, the following objectives need to be achieved:

1. Importing popular dataset for the purpose of classification of data in positive, negative, neutral on the basis of its sentiment value.
2. Training and testing a deep neural network on a given dataset.
3. Finding the performance of deduced model on different aspects.
4. Comparing the performance of different deep neural networks available.

Understanding the data has always been very complex but with the help of right techniques, I have prepared the classifier which will produce desired sentiment value associated with the textual data. This project takes IMDB dataset which contain thousands of movie reviews published by lots of people train and test the efficiency of classifier which is based on Deep neural networks.

This project takes use of keras library which is built over tensor flow from which I will import different classes to pre-process and model our data. Google colaboratory has been used for its processing power to train deep neural network.

CHAPTER 2

LITERATURE REVIEW

Sentiment analysis is not a new task it has been done over a long period of time with Different techniques analysis of textual data is not an easy task it requires a lot of pre-processing and based on that pre-processing appropriate results can be produced

Lot of studies present on analysis of movie reviews on IMDB which have produced good results over the period of time with the advancement of Technology and increasing knowledge of datasets and their analysis has lead us to greater understanding in the field of sentiment analysis.

Research papers out there which used classical or traditional machine learning techniques such as support vector machine, K nearest neighbour, naive Bayes algorithm and sometime unsupervised learning to detect patterns and clusters among data but with the advancement of Technology and humans understanding of deep neural networks I have come to great results compared to the previous time.

Researchers have come all the way from feed forward artificial neural networks to Specialist neural network for classified images searches convolutional neural networks and for classifying sequential data such as recurrent neural networks and long short term memory networks. IMDB data in it contain millions of movie reviews and therefore millions of sentiments. Now these sentiments can be analysed and again used for businesses to predict opinion and also for human behaviour.

This section, detailed information related to the variety of research papers concentrating on sentiment analysis using different methodologies, challenges present in sentiment analysis, and which techniques can perform better, etc. has been listed.

- *Hameed, Zabit et al (2020)* [1]: This paper presents the idea of using single layered BiLSTM and generated an accuracy of 90.585% on IMDB dataset.
- *Fu, Xianghua et al (2018)* [2]: This paper presents the idea of lexicon-enhanced LSTM model for the purpose of sentiment analysis where the model is proposed using LSTM to combine word embedding for text representation. Model presented in this paper gave 89.50% accuracy on IMDB movie review dataset.
- *Long, Yunfei et al (2017)* [3]: This paper presents the idea of using two different attention mechanisms with LSTM. Model presented in this paper gave 90.10% accuracy on IMDB movie review dataset.

- *Collados, Jose Camacho et al (2018)* [4]: This paper presents the idea of using fusion of CNN and LSTM with different cleaning process. Model presented in this paper gave 88.90% accuracy on IMDB movie review dataset.
- *Ma, Yaohong et al (2019)* [5]: This paper presents the idea of using attention mechanism and adversarial training with BiLSTM. Model presented in this paper gave 89.22% accuracy on IMDB movie review dataset.
- *Goldberg, Yoav (2016)* [6]: This study presents neural network models from the angle of NLP.
- *Huang, Po-Yao et al (2016)* [7]: This paper presents the idea of a neural machine translation (NMT) architecture, different features are fuse with text to create sequences which are implemented over parallel LSTM.
- *Zhang, Lei et al (2018)* [8]: This paper presents a study on deep learning & gives uses in the field of sentiment analysis.
- *Zhang, Honglun et al (2017)* [9]: This paper presents a multi-task learning architecture with 4 types of RNN layers to concatenate information across multiple related tasks.
- *Hochreiter, Sepp et al (1997)* [10]: This paper introduces an efficient and gradient based technique called LSTM short for Long short term memory.

Out of all the paper reviewed in this section following concluding points are noted:

1. Most of the papers introduces some variant of LSTM to be used for sentiment analysis and gives the highest accuracy.
2. Initial papers results will be used to compare the performance of proposed model in later stages.
3. Simple variants of LSTM provide higher accuracy in case of highly polarised reviews.
4. Focus on word embedding is also provided as it acts as the initial step of understanding natural data before applying deep learning.
5. Deep learning methods are much more effective and provide an edge over traditional approaches in understanding sequential data.

CHAPTER 3

THEORETICAL CONCEPTS

This section presents the basic theoretical concepts required to understand the key processes and working of the experiment studied in this project. This section familiarizes the concept deep neural networks (ANN, RNN, LSTM). The concepts introduced in this section help to understand the proposed architecture for sentiment analysis.

3.1 Sentiment Analysis

Sentiment analysis is the prediction of emotions behind sentences, text, words which can generate a lot of information about opinions different decisions that particular sentence wants to say this information can have great value if used properly for businesses or study of human behaviour. Today lots of analytical companies use data processing to gain information about people and their opinion about different products, take an example of Twitter there are millions of tweets which travel from one place to other carrying different meanings this could be understand, computer does that job in fraction of time. This information can be useful to know about the sentiments of people about their opinion about different products they purchase about the different destinations they travel around the world about different government policies what they think about as people express lot of things on social media today not everything is hidden so those statements or text could be analysed and used as a tool to understand sentiments values of the people carrying those statements. Fig. 1 shows multiclass classification into negative, neutral and positive sentiment value.

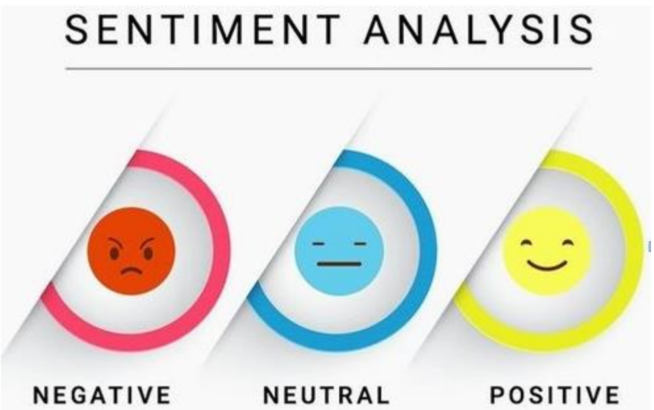

Fig. 1: Negative and Positive Sentiments

3.2 Natural Language Processing

Natural language processing and processing of human languages in such a way that important information could be gathered which will act as an input to many computer models available out there are lot of computer models which only deals with mathematical values so to perform sentiment analysis on IMDB data first reviews must be understood by the model therefore the concept of NLP comes in. NLP is an interaction between human language and other computing devices and falls under the realm of artificial intelligence as well as computer science.

Tokenization : It is the first step When we start a data processing so in our data set we have given reviews which are in the form of statements which combine of many words not to understand these statements efficiently and effectively they have to be broken down into tokens in our case tokens are words so this whole process of breaking the complete statement into single entity is known as tokenization.

Normalisation : Is the next step after performing tokenization, the name suggest in normalisation we normalise our text by converting that text into same case and removing punctuations and generating uniformity

Stemming : English vocab provides lot of words which may have or may not have the same Root word, so to decrease the complexity we only care about the root word for example root word of calling and called is call, so in this process we try to remove suffixes prefixes infixes in order to obtain a word stem.

Lemmatization : It is somewhat related to stemming but is more complex and time-consuming. In many cases is stemming fails to produce the desired output for example if we take a word ‘was’ the stem generated by it will be ‘wa’ but it is actually not correct Lemmatization is more intelligent technique. Lemmatization of word better will be good

Stop words : There are several words in a statement that does not produce significant result for our very really help in generating output such words should be removed so as to decrease the amount of data to be processed and make it less complex words such a ,an, the falls in the category of stop words.

Word embedding : Word embedding is a task in which a string is converted into a vector because computer models to not take string as input only take numerical data such as vectors as input and also produce vectors as output so to simplify the process of understanding of movie reviews and generating positive results every word is associated with the number that will define its significance. The words which are related to each other in any form such as synonyms antonyms are given close numbers so as to define their relationship and differentiate with other and unrelated words.

Bag of words : This is a representational model which simplifies selection of words based on content in the statement words count the frequency of every word irrespective of their order and grammar, for example

"Hello hello," said Suresh.

"There, there," said Radha.

The resulting bag of words representation as a dictionary:

```
{
    'hello': 2,
    'said': 2,
    'suresh': 1,
    'there': 2,
    'radha': 1
}
```

3.3 Feed forward Artificial Neural Networks

Deep neural networks are a kind of classifier which mimic the structure of the human brain, they consist of neurons which act as nodes in this complicated structure deep neural networks consists of a number of layers which are divided in three categories which are input layer one or more hidden layers and output layers.

Unlike the simple classifier such as support vector machine, linear regression, Logistic regression, naive Bayes algorithm, K nearest neighbour are very simple to understand but neural networks are more advanced and more complex than any other machine learning classifier. Fig. 2 illustrates a feed forward artificial neural network with one hidden layer.

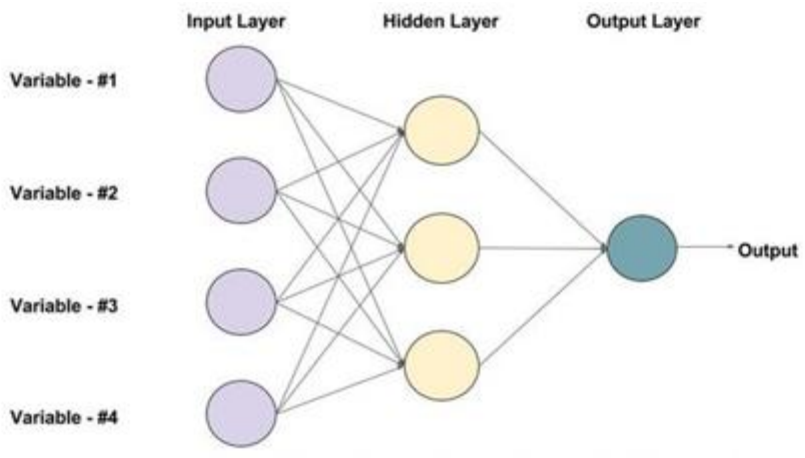

Fig. 2: Feedforward Neural Network with single hidden Layer

For sentiment analysis neural networks will serve as a classifier for strings to generate binary reviews that are positive and negative. First that data will be trained and based on that training the weights and biases associated with neurons learn using an algorithm called backtracking. Cost functions are used so as to check the difference between expected output and desired output based on these cost functions hyper parameters are tuned accordingly using various techniques present.

Feed forward artificial neural network is the simplest among all neural networks, they do not consist of a large number of hidden layers and easier to understand. The input flows from input layer to output layer and weights are generated accordingly.

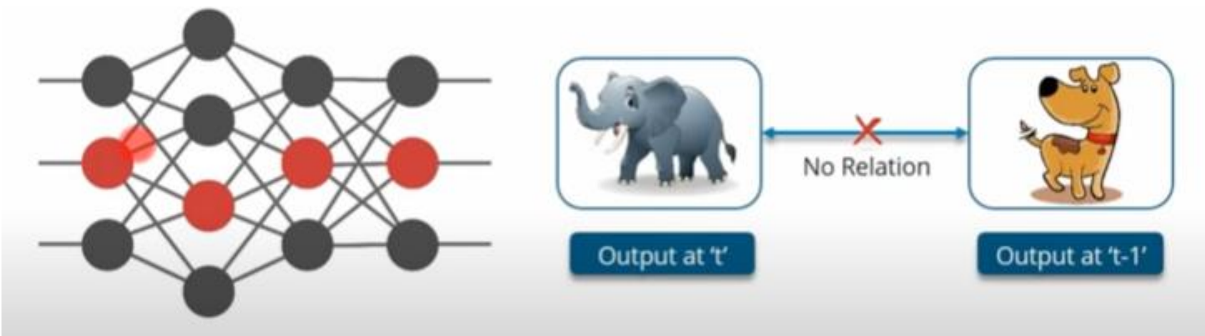

Fig.3: Feedforward network showing no relation between output

For the purpose of sentiment analysis where I deal with dependencies in which you can not understand the next word without reading all the words coming before it. As in feed forward network the output generated at time T is not related with output generated at T-1 or T+1, they all are independent therefore for this project that is sentiment analysis, feed forward network will not produce efficient and effective results. Fig.2 and Fig. 3 illustrate ANN with no relation between output at different time stamps. Therefore I will move to a better neural network that is recurrent neural network which also takes feedback from the output generated previously and fed in with the input and to generate the next output.

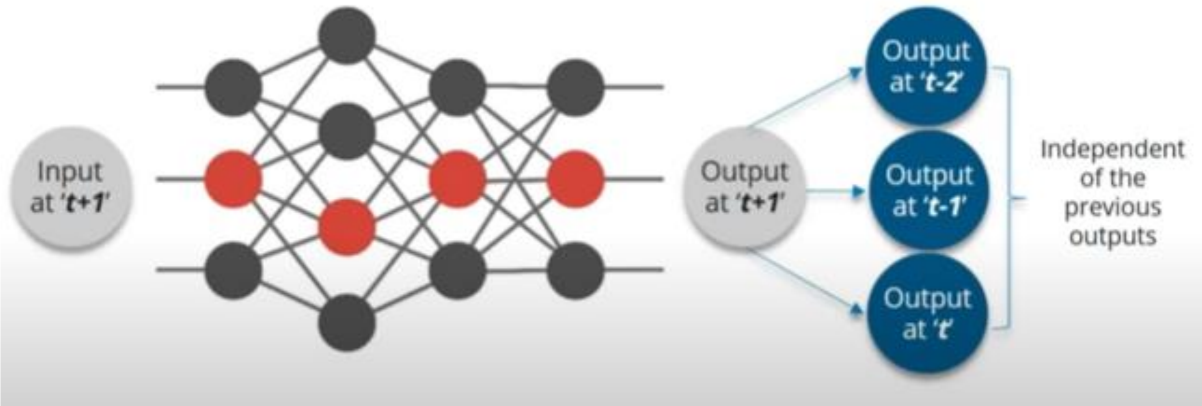

Fig. 4: Feedforward Network showing independent outputs

3.4 Recurrent Neural Network

RNN are best when it comes to sequential data analysis, they have been used to analyse sequential data for years now and are doing the job perfectly. Sentiment analysis includes dependencies which require neurons to learn as well as forget such results could only be produced by using RNN. Recurrent neural networks have two issues when dealing with long-term dependencies that are exploding gradient descent and vanishing gradient descent. As a result, I switched to a recurrent neural network, which is a type of neural network.

This issue is addressed by recurrent neural networks. These kind of networks contains loops in them, so that persistence of information can take place. Fig. 5 shows feedback from previous output to current input where input vector at time t is shown by x_t and output vector at time t is shown by h_t .

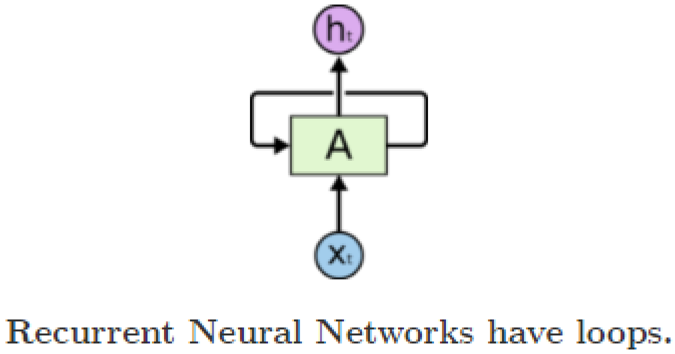

Fig. 5: A basic RNN

If RNN shown in Fig. 6 is unfolded it will look like Fig. 5 in which x and h shows input and output respectively at continuous time stamps where output at t=1 is calculated using input at t=1 and output at t-1.

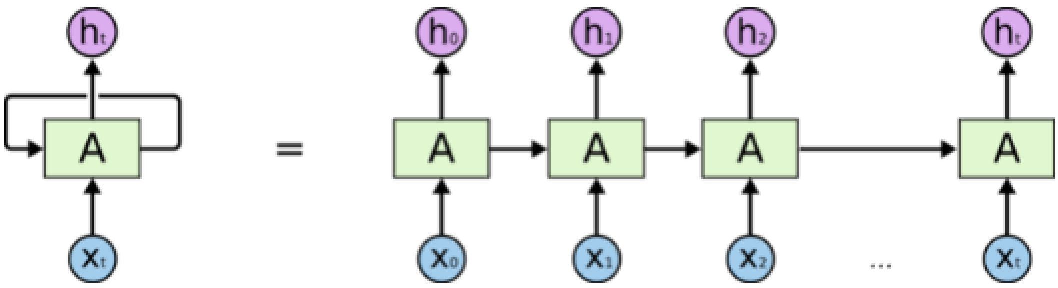

Fig. 6: An unfolded RNN

Let us take an example of next word predictor based at previous words, the statement is "cloud are in the", now it is obvious that the next word will be sky. Any further context, in not needed in such cases, RNN can learn to use past information when the distance between relevant information and the location where it's needed is short.. Fig. 5 shows a RNN learning through past information.

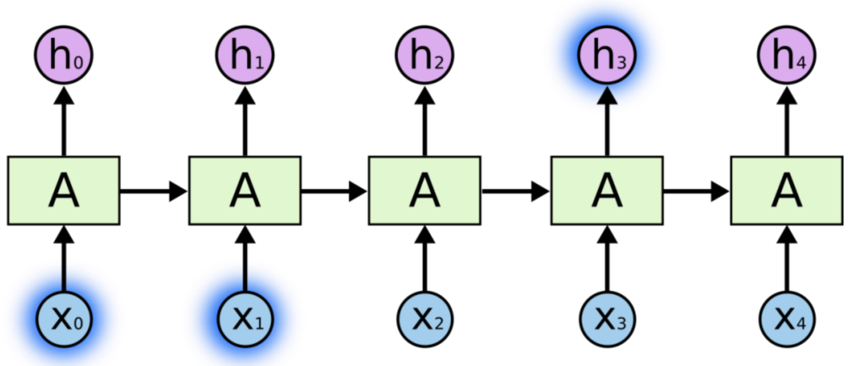

Fig. 7: RNN learning through past information

Let us take another example “Kunal and his dog Oreo are best friends, they live in a village near delhi”, In this case previous words suggest that the word ”they” is referring to Kunal and Oreo both. It's entirely feasible for the distance between relevant data and the point at which it's required to grow significantly.

Unfortunately, as the distance widens, RNNs lose their ability to learn to connect the dots. Fig. 8 illustrates a RNN with lot of previous information in which information at time t=0 and t=1 is relevant in generating output at t+1 but there is a possible gap between it.

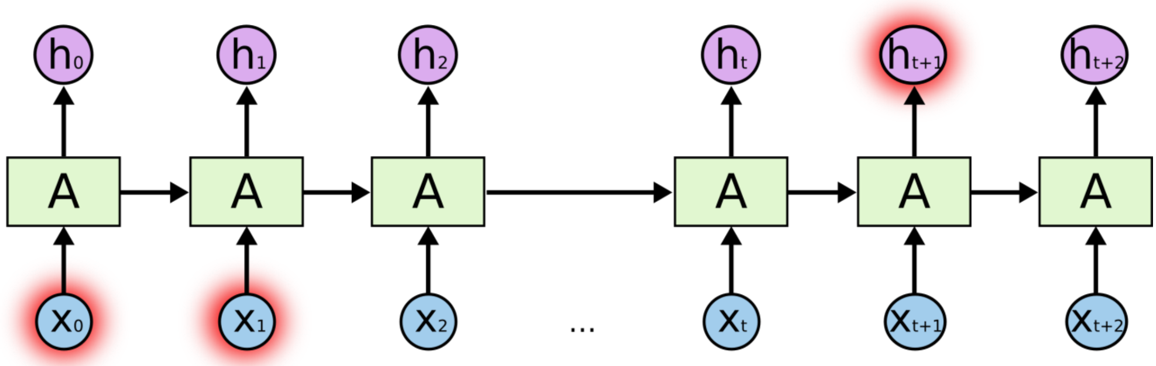

Fig. 8: RNN with lot of previous information

Fig. 9 illustrates how when a RNN move forward on learning through training data, it tends to forget information from early in the sequence.

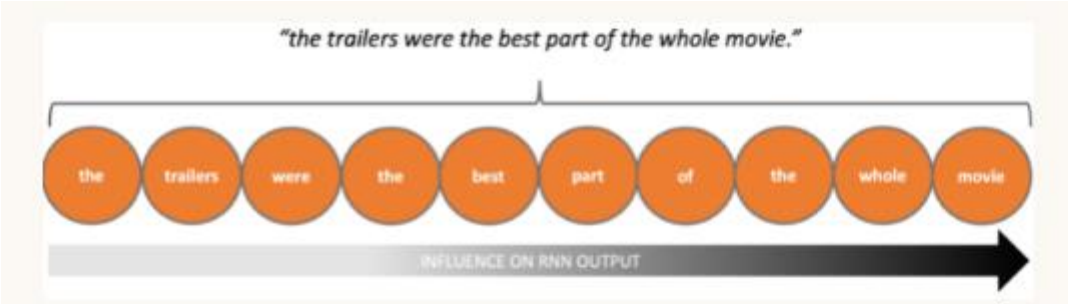

Fig. 9: RNN forgetting information from early in sequence

There are two major problems with RNN i.e. vanishing and exploding gradient decent which can be addressed by using LSTM, a special type of RNN.

3.5 Long Short Term Memory

There are many variants of RNN available which can solve the problems like exploding and vanishing gradient descent problems whereas LSTMs are most popular.

LSTM hidden layer contains multiple neural networks each is assigned an indivisual task. LSTMs are optimised with an algorithm called back propagation through time, BPTT. Fig. 10 shows components in LSTM

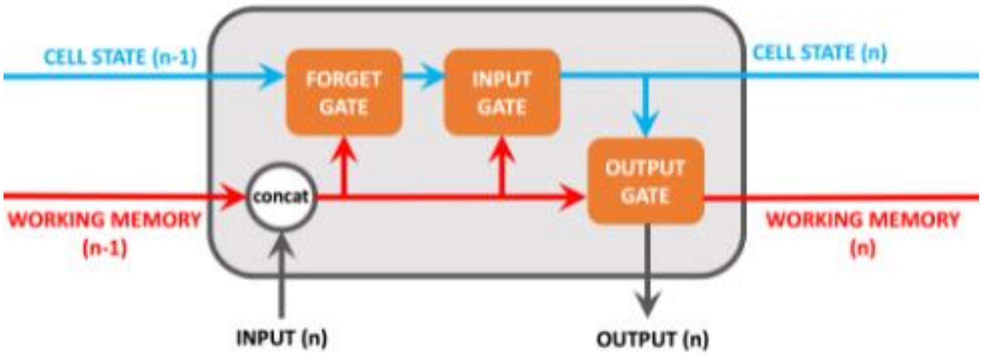

Fig. 10: Components in LSTM

Cell State: Its job is to serve as a long-term memory for the node, storing information (if needed) through all iterations.

Forget Gate: Its role is to Identify which data should be eliminated from the Cell State.

Input Gate: This is going to decide what information should be added to the Cell State.

Output Gate: This is going to decide on what working memory this node will output.

Fig. 11 illustrates block diagram of LSTM network.

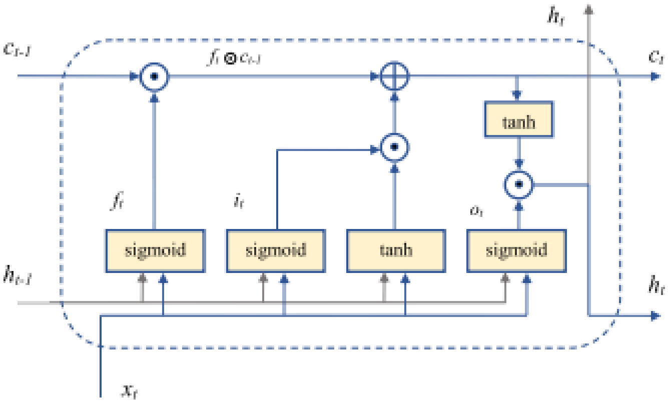

Fig. 11: An illustration of LSTM network

In Fig. 11, the forget gate (f_t) evaluates if the information from the previous state (c_{t-1}) should be forgotten or retained by seeing input (x_t) and also the h_{t-1} i.e. the hidden state and its output value can be 0 or 1. Exactly also the i_t i.e. input gate evaluates how many information from x_t and h_{t-1} must pass to change the state of the cell, and the output can be 0 or 1. The cell state formed by mathematical operations on ct - 1, and f_t, and it is represented by the c_t value. The output gate (o_t), whose value might be 0 or 1, monitors the stream of information from the present state of the cell to the hidden state. (3.1) to (3.5) shows mathematical details. Where $x_t \in R^n$, $W \in R^{v*n}$, b $\in R^v$, and the indices n & v respectively represent the size of the x_t and. At any moment t, the input vector x_t, h_{t-1} and c_{t-1}, while the the current hidden state h_t and the present state of the cell c_t are the outputs. $\odot$ describes multiplication.

$$f_t = sigmoid(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{3.1}$$

$$i_t = sigmoid(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{3.2}$$

$$c_t = c_{t-1} \odot f_t + i_t \odot \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{3.3}$$

$$o_t = sigmoid(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{3.4}$$

$$h_t = o_t \odot \tanh(c_t) \tag{3.5}$$

CHAPTER 4

PROPOSED MODEL FOR SENTIMENT ANALYSIS

A LSTM based artificial neural network is proposed so as to perform sentiment analysis on IMDB movie review dataset. Fig. 12 shows different layers of proposed model.

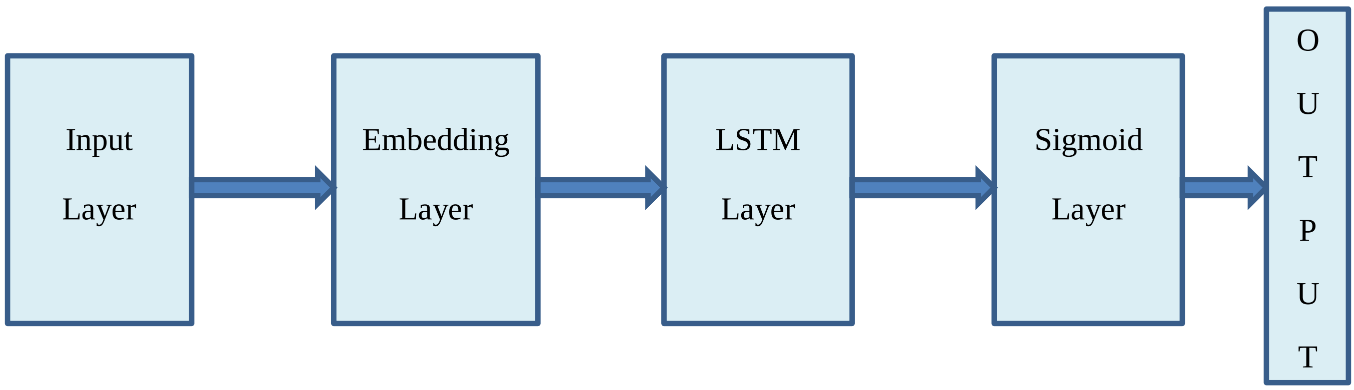

Fig. 12: Layers of proposed model from input to output

5.1 Input Layer

The input layer is the beginning of this network. Keras contains the IMDB dataset in form of a input layer such that data samples are represented as a series of unique indices of the same length i.e. 500 is length of every review such as long reviews are truncated and small reviews are padded.

5.2 Embedding Layer

The embedding layer is the second level of the proposed architecture, and it converts each index corresponding to a unique word in the dataset into a vector of real-valued characteristics. An embedding matrix is formed by combining these real-valued vectors. Every row in this matrix represents vocabulary, and the matrix has a dimension of x * y, where x is the size of the vocabulary collection., in this case it is 5000 and y is the dense vector dimension which is 32. In this project, keras embedding layer is used.

5.3 LSTM Layer

A LSTM network has been taken into account in which information moves forward hence output at value of time t is solely determined by the information delivered before it. to comprehend sentiment of an input review correctly the relevance of previous information is also considered.

5.4 Output Layer

Output layer gives a probability of a positive review. If probability comes near to 0 the IMDB movie review generate negative sentiment value otherwise positive sentiment value. A cost function called binary cross entropy is used to examine the cost of model.

$$\text{Binary Cross Entropy} = -\frac{1}{m}\sum_{i}^{m}\left(y_i * \log\left(p(y_i)\right) + (1 - y_i) * \log\left(1 - p(y_i)\right)\right) \quad (4.1)$$

CHAPTER 5

EXPERIMENTAL SETUP

5.1 Analysis of Dataset

This project uses IMDB Movie Reviews Dataset imported from Keras which contains 50000 highly polarised reviews equally classified as a "positive" or "negative". Fig. 13 shows the distribution of IMDB reviews dataset over positive and negative value.

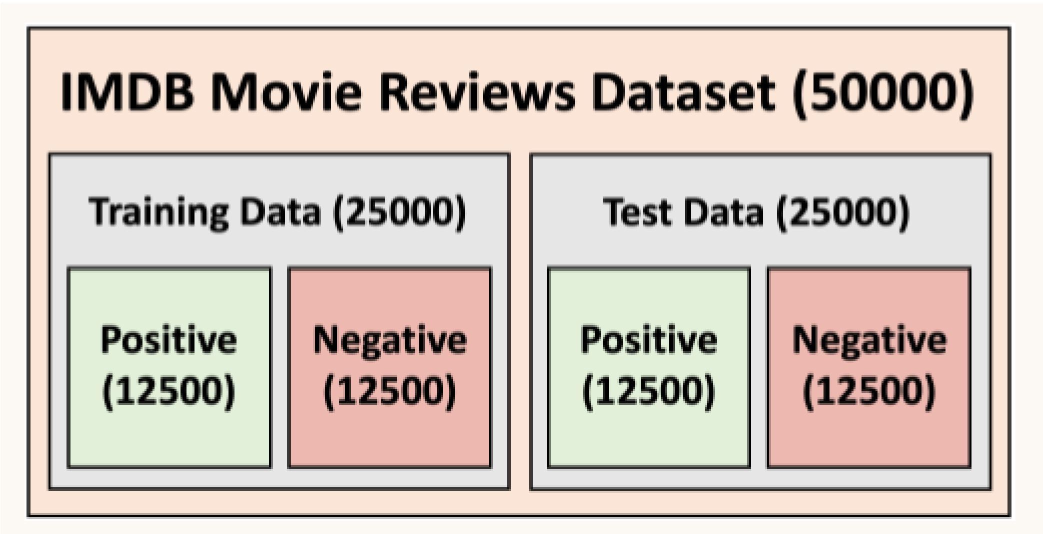

Fig. 13: Distribution of 50000 records in the IMDB dataset

Every word in the dataset is assigned a unique integer that represents its overall frequency in the dataset. Without punctuation marks, all reviews are represented as a series of integers.The most popular vocabulary of 5000 words is taken. This helps filter away potential noise and outliers from the dataset to reduce training time.

Ensure that each review is the same length of 500 words. This implies either a truncation or a reduction in length, filling in views based according to their original length. The objective is to keep all records in a standardized format.

Distinct ID's have been assigned to every unique word in our vocabulary by Keras itself. Python command such as print(word2id) and print(id2word) can be used to check id's assigned to words and vice versa.

```
print(word2id)
```

{'fawn': 34701, 'tsukino': 52006, 'nunnery': 52007, 'sonja': 16816, 'vani': 63951, 'woods': 1408, 'spiders': 16115, 'hanging': 2345, 'woody': 2289, 'trawling': 52008, "hold's": 52009, 'comically': 11307, 'localized': 40830, 'disobeying': 30568, "'royale": 52010, "harpo's": 40831, 'canet': 52011, 'aileen': 19313, 'acurately': 52012, "diplomat's": 52013, 'rickman': 25242, 'arranged': 6746, 'rumbustious': 52014, 'familiarness': 52015, "spider'": 52016, 'hahahah': 68804, "wood'": 52017, 'transvestism': 40833, "hangin'": 34702, 'bringing': 2338, 'seamier': 40834, 'wooded': 34703, 'bravora': 52018, 'grueling': 16817, 'wooden': 1636}

Fig.14: word to id

```
print(id2word)
```

{34701: 'fawn', 52006: 'tsukino', 52007: 'nunnery', 16816: 'sonja', 63951: 'vani', 1408: 'woods', 16115: 'spiders', 2345: 'hanging', 2289: 'woody', 52008: 'trawling', 52009: "hold's", 11307: 'comically', 40830: 'localized', 30568: 'disobeying', 52010: "'royale", 40831: "harpo's", 52011: 'canet', 19313: 'aileen', 52012: 'acurately', 52013: "diplomat's", 25242: 'rickman', 6746: 'arranged', 52014: 'rumbustious', 52015: 'familiarness', 52016: "spider'", 68804: 'hahahah', 52017: "wood'", 40833: 'transvestism', 34702: "hangin'", 2338: 'bringing', 40834: 'seamier', 34703: 'wooded', 52018: 'bravora', 16817: 'grueling', 1636: 'wooden'}

Fig 15:id to word

5.2 Hyper-parameter Setting

The optimal values of hyper parameters are mentioned in Table I

Table I: Hyper parameter setting for IMDB dataset

Hyper parameter	IMDB dataset
Train approach	Cross Validation
Optimizer	Adam
Loss function	Cross entropy
Batch size	64
Epochs	20

Dropout	0.2

A. Training Approach

Cross-validation is a technique for assessing the model's efficiency by training it on a subset of input data and testing it on a portion of input data that has never been seen before.

B. Optimizer

Using a gradient descent optimization process, neural networks discover the underlying and sophisticated patterns from data. I used a variation of Adam optimizer for this. Adaptive Moment Estimation is a methodology for optimising gradient descent algorithms. When working with huge problems with a lot of data or parameters, the method is quite efficient. It is efficient and takes minimal memory. It appears to be a hybrid of the 'gradient descent with momentum' and the 'RMSP' algorithms.

C. Loss Function / Cost Function

Optimizing the proposed model is an important task. Parameters such as weights corresponding to every neuron in the the neural network has to be optimized in order to minimize the loss functions value. In this project, As a loss function, I utilised binary cross-entropy, which is an excellent choice for binary classification.

D. Batch Size

Batch size is number of sample which is processed at a time by a neural network such that after every batch weights and biases change based on the value of cost function. I used the batch size of 64 in this project.

E. Epoch

Epochs represents the number of iterations a neural network is going to follow before stopping. In this project I have used 20 epochs. Table II shows accuracy and loss associated with every iteration. Fig. 12 shows model accuracy on training and testing data.

Table III: Accuracy and Loss for first 20 epoch

Epochs	Training_Loss	Training_Accuracy	Validation_Loss	Validation_Accuracy
1	0.3063	0.8807	0.3205	0.8658
2	0.2789	0.8889	0.3272	0.8609
3	0.2208	0.9163	0.3073	0.8702
4	0.1904	0.9260	0.3108	0.8787
5	0.1682	0.9398	0.3249	0.8723
6	0.1493	0.9469	0.3398	0.8600
7	0.1383	0.9513	0.3513	0.8734
8	0.1250	0.9547	0.3621	0.8631
9	0.1635	0.9382	0.4316	0.8520
10	0.1180	0.9584	0.3919	0.8637
11	0.0981	0.9651	0.4531	0.8673
12	0.2353	0.9076	0.4821	0.7600
13	0.3007	0.8715	0.3916	0.8701
14	0.1244	0.9548	0.3686	0.8691
15	0.1021	0.9635	0.4409	0.8690
16	0.0989	0.9648	0.4686	0.8136
17	0.2247	0.9131	0.4248	0.8623
18	0.1333	0.9497	0.4584	0.8560
19	0.1096	0.9612	0.5735	0.7309
20	0.2050	0.9172	0.4601	0.8704

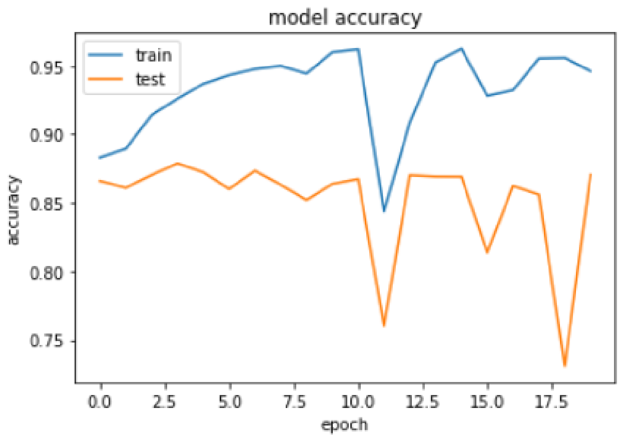

Fig. 16: Model accuracy on training and testing data

F. Dropout

As the name suggest dropout layer is used to reduce over fitting in artificial neural networks. This is the most straightforward method for reducing over fitting in artificial neural networks. It occurs at random and disables in the learning phase, given the number of nodes to make network slim. In this layer, I used a dropout of 0.2, and also in LSTM, I used a dropout value of 0.2

5.3 Model Architecture

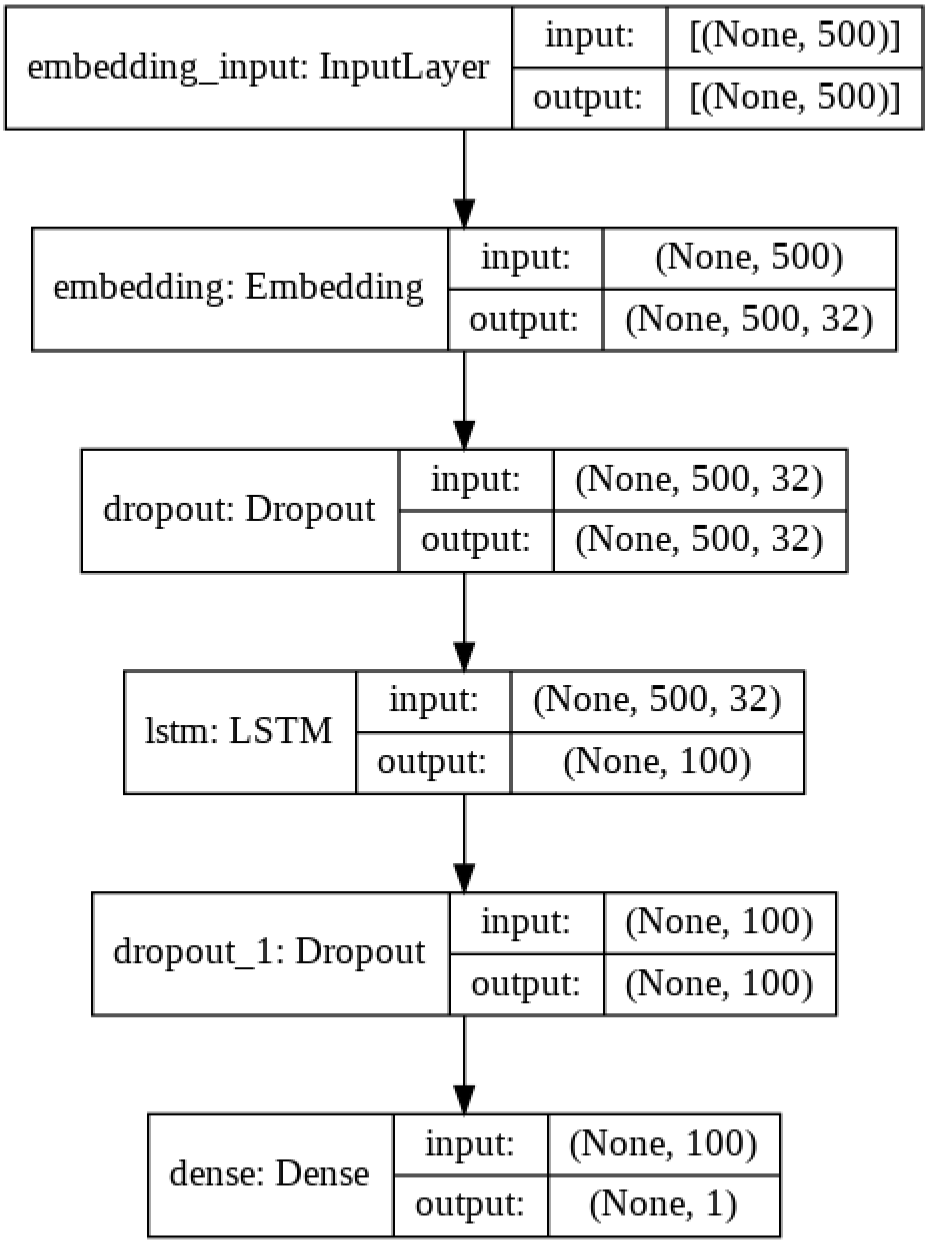

Fig. 17: LSTM model plot

A. Input Layer

The Input Layer takes every IMDB movie review as a one dimension vector. Every review is either truncated or padded to a length of 500. Fig. 18 show the input layer

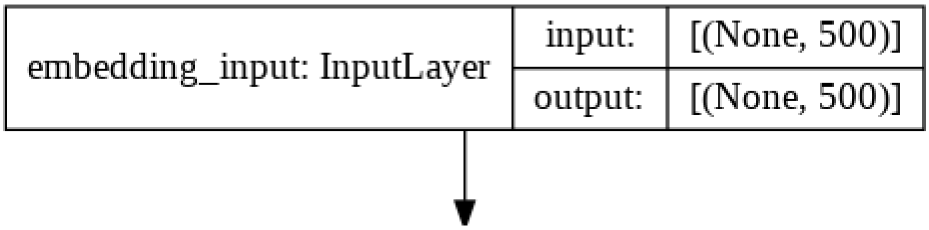

Fig. 18: Input Layer

B. Embedding Layer

The second layer that comes in the sequence of model is embedding layer which operate on input layer and generate a word embedding space of 500*32 , where 500 is the movie review length and 32 is the word embedding dimension. Fig. 19 shows the embedding layer

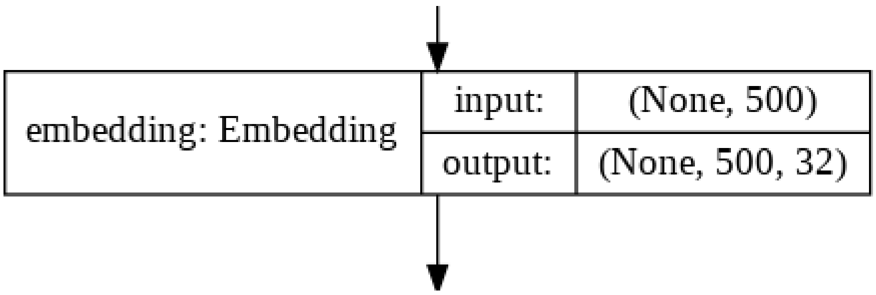

Fig. 19:Embedding Layer

C. Dropout Layer 1

Dropout of value 0.2 is used between embedding layer and LSTM layer to remove over fitting in model. Fig. 20 shows dropout layer.

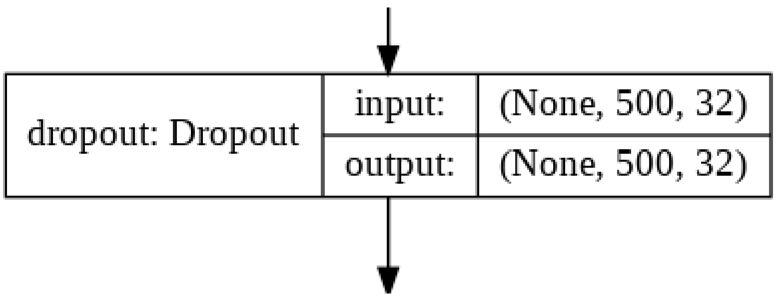

Fig. 20: Dropout Layer

D. LSTM Layer

LSTM layer is responsible for learning the sentiments of IMDB movie review dataset. It takes input as embedding vector and output a vector of 100 values which will be later mapped to single value by dense layer. Fig. 21 shows the LSTM layer

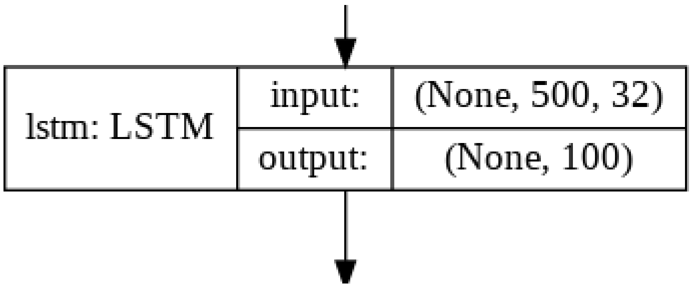

Fig.21: LSTM Layer

E. Dropout Layer 2

Dropout of value 0.2 is used between LSTM and dense layer

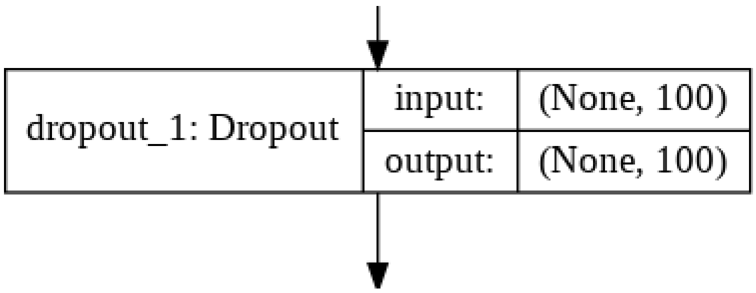

Fig. 22: Dropout Layer 2

F. Dense Layer

The final layer of the proposed model is Dense layer with task to map 100 nodes generated by the previous layer into single node and using a sigmoid function over it to map the output number between 0 and 1, where predictions close to zero will map to negative sentiment and prediction closer to one will map to positive sentiment. Fig. 23 shows the graph of sigmoid function and fig. shows dense layer. Sigmoid function is equal to

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{5.1}$$

where z is the input value

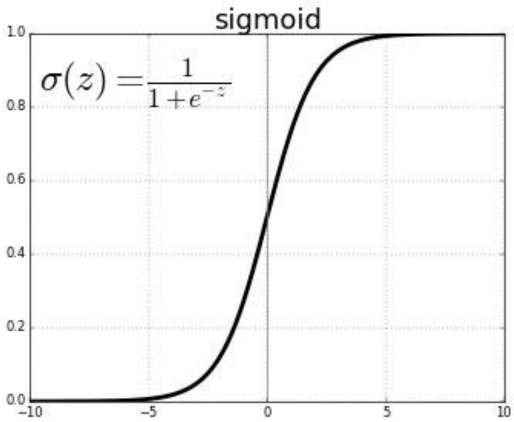

Fig. 23: Sigmoid Function

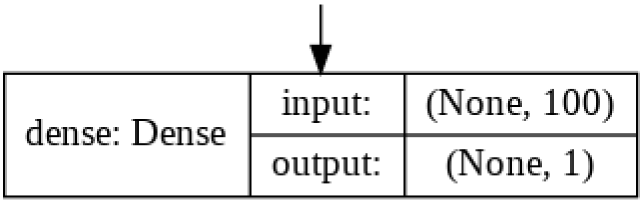

Fig. 24: Dense Layer

Table III: Model Summary

Type of Layer	**Output Dimension**	**Parameters**
embedding layer	(None, 500, 32)	1,60,000
dropout layer 1	(None, 500, 32)	0
lstm layer	(None, 100)	53,200
Dropout layer 2	(None, 100)	0
dense layer	(None, 1)	101
Trainable Parameters = 213301		
Non Trainable Parameters = 0		
Total Parameters = 213301		

5.4 Performance Evaluation

The following matrices have been used to calculate performance of the proposed model using confusion matrix. Fig. 25 shows confusion matrix that includes true positive (TP), false positive (FP), true negative (TN), false negative (FN).

- **True Positive**

  It tells the total number of reviews which are correctly classified as positive reviews among all reviews.

- **False Positive**

  It tells the total number of reviews which are wrongly classified as positive review among all reviews

- **True Negative**

  It tells the total number of reviews which are correctly classified as negative review among all reviews.

- **False Negative**

  It tells the total number of reviews which are wrongly classified as negative review among all reviews.

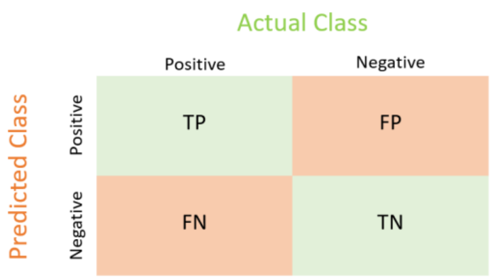

Fig. 25: Confusion Matrix

- **Precision**

  It is the ratio of correct positive outcomes to the total positive outcomes for a class. It is given by the following formula.

$$Precision = \frac{TP}{TP+FP} \tag{5.1}$$

- **Recall**

Recall or sensitivity is defined as the ratio of correctly classified positive outcomes to the total number of actual positive outcomes. It is given by the following formula.

$$Recall = \frac{TP}{TP+FN} \tag{5.2}$$

- **Accuracy**

It is the ratio of correct outcomes to the total outcomes of the experiment. It is given by the following formula.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{5.3}$$

- **F1 score**

It is the harmonic mean of precision and recall. It is given by the following formula:

$$F1\ score = 2 * (Precision * Recall)/(Precision + Recall) \tag{5.4}$$

Table IV contains the values of all the performance matrices mentioned above.

Table III: Performance matricesl for IMDB dataset

Dataset	Normalised Confusion Matrix			Parameters for evaluation of performance of model on IMDB dataset			
IMDB	Predict→	Positive	Negative	Precision (%)	Recall (%)	F1Score (%)	Accuracy(%)
	Positive	0.89	0.11	86.40	89	87.68	87.5
	Negative	0.14	0.86				

CHAPTER 6

RESULTS AND DISCUSSION

In this project I used deep neural network that is long short term memory for classification of movie reviews dataset into negative and positive based on their sentiment values. The model which is trained in this project gives an accuracy of 87.5 % over testing data.

Fig. 26 gives training data's and testing data's accuracy over number of epochs

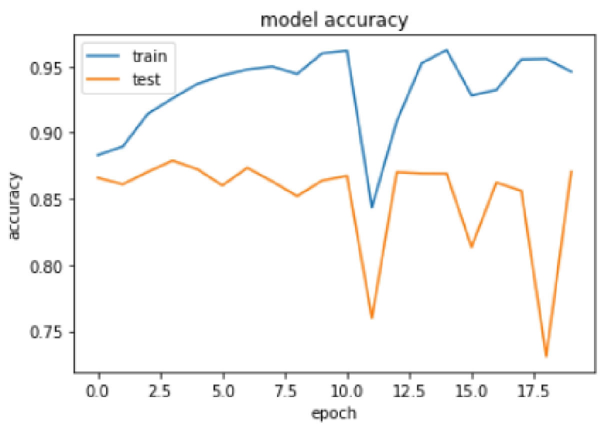

Fig. 26: Model accuracy on testing and training data

Fig 27 shows the loss generated by the cost function of testing and training data over number of epochs.

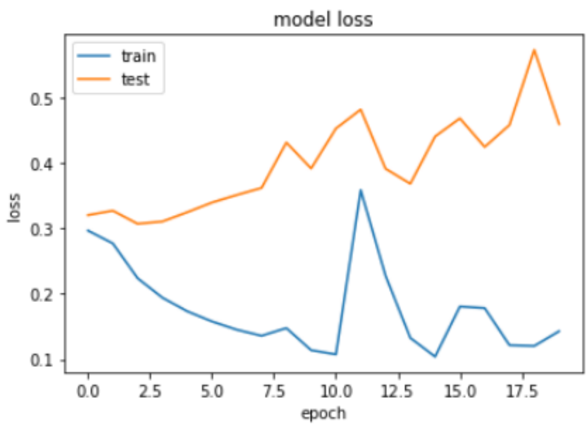

Fig. 27: Model loss on testing and training data

Fig 28 illustrates training data's and testing data's precision over number of epochs

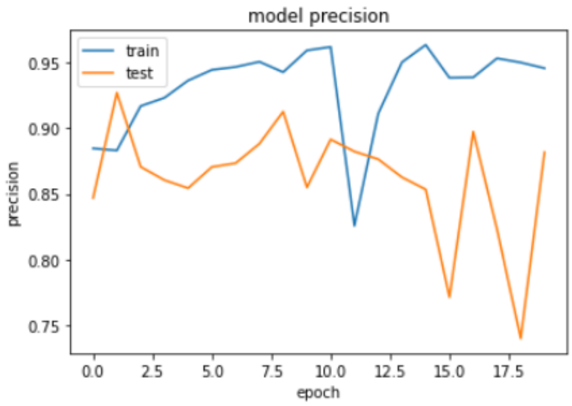

Fig. 28: Model precision on training and testing data

Fig 29 shows the recall of testing and training data over number of epochs

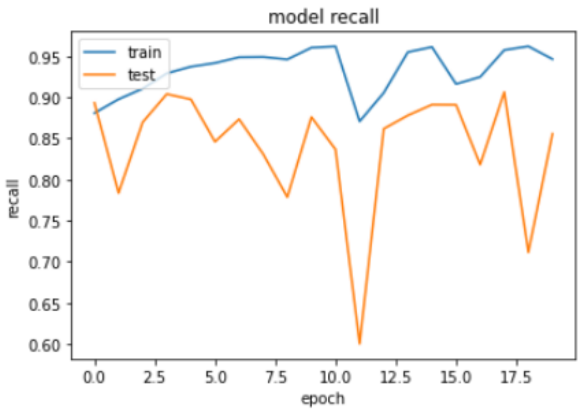

Fig. 29: Model recall on testing and training data

Fig 30 and Fig 31 illustrates training data's and testing data's confusion matrix.

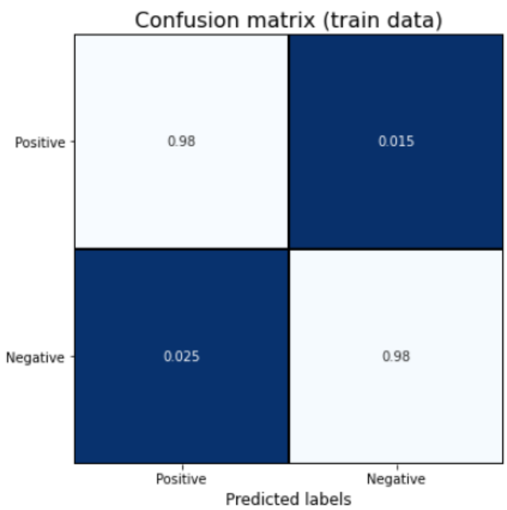

Fig. 30: Confusion matrix of training data

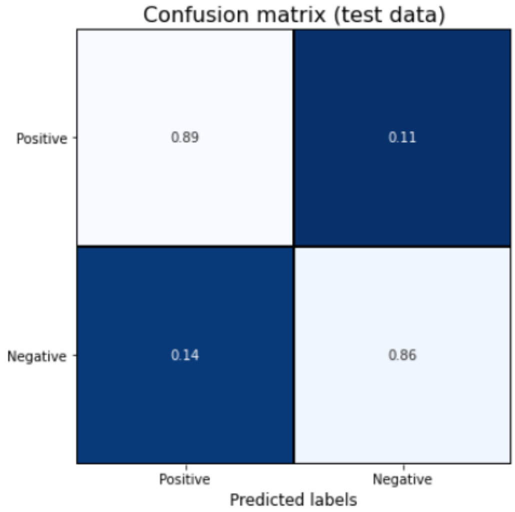

Fig. 31: Confusion matrix of testing data

Table V compares the accuracy of the proposed model with recently published models. Proposed model's accuracy is comparable to accuracy of other complex models. All these models are complex as they contain fusion of two or more learning mechanism. Proposed model only contains a single layer LSTM network and able to produce 87.5 accuracy which is only around 3% lesser than BiLSTM model which provides highest accuracy among all. LSTM being the simplest among all provided comparable results with IMDB dataset.

Table IV: Performance Comparison for different models

Methods	Mechanism Used	Accuracy(%)
BiLSTM [1]	Single Layered	90.58
LSTM+CBA+LA [3]	Combination of different attention mechanisms with LSTM	90.10
CNN+LSTM [4]	Fusion of Convolutional NN and LSTM with several cleaning processes	88.90
WALE-LSTM [2]	Combination of lexicon & attention mechanism	89.50

FARNN-Att [5]	Attention Mechanism & adversarial training with BiLSTM	89.22
Proposed	**Single Layered LSTM**	**87.5**

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Through this study I have concluded that LSTM deep neural networks can be used efficiently and effectively to evaluate sentiment values behind text. Long short term memory is able to do this job and produced a good accuracy on testing data. There are different modifications which can be made in this technique which provides us the room for improvement and the number of iteration which are used for back tracking the model through which it learns the weights and biases is 20 due to limited computation power, but in Real world applications more than thousand iterations are done in order to evaluate the best possible values for weights associated with every neuron the accuracy could be fairly increased.

Long short term memory help very well and able to resolve complex sequences and understanding sequential data. Long short term memory require a lot of computation and therefore lot of resources they have been used majorly in all kinds of machine translations and specifically for sequential data analysis. Google for spell check grammar check, synonym and antonym. In our daily life during typing a message in whatsapp from our keyboard in mobile phone the system already predicts the next word before I type it this is a very good example of LCM. Different companies have used it for their benefits and are able to produce valuable and effective output but due to their Complex nature they require more computational power compared to simpler neural networks.

The future scope of this project lies in developing this model to analyse the sentiment value behind statements of hindi language and using multiclass classification.

There is always room for improvement, deep neural networks keeps on becoming complex when the number of layers and also the number of neurones in that particular layer increase.

One can always work on models which are more complex by containing fusion of two or more learning mechanism.

REFERENCES

[1] Z. Hameed and B. Garcia-Zapirain, "Sentiment Classification Using a Single Layered BiLSTM Model," in *IEEE Access*, vol. 8, pp. 73992-74001, 2020, doi: 10.1109/ACCESS.2020.2988550.

[2] X. Fu, J. Yang, J. Li, M. Fang, and H. Wang, ''Lexicon-enhanced LSTM with attention for general sentiment analysis,'' IEEE Access, vol. 6, pp. 71884–71891, 2018.

[3] Y. Long, L. Qin, R. Xiang, M. Li, and C.-R. Huang, ''A cognition based attention model for sentiment analysis,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Copenhagen, Denmark, 2017, pp. 462–471.

[4] J. Camacho-Collados and M. T. Pilehvar, ''On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis,'' in Proc. Empirical Methods Natural Lan. Process. (EMNLP), Brussels, Belgium, 2018, pp. 1–7.

[5] Y. Ma, H. Fan, and C. Zhao, ''Feature-based fusion adversarial recurrent neural networks for text sentiment classification,'' IEEE Access, vol. 7, pp. 132542–132551, 2019.

[6] Y. Goldberg, ''A primer on neural network models for natural language processing,'' J. Artif. Intell. Res., vol. 57, pp. 345–420, Nov. 2016.

[7] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer, ''Attention-based multimodal neural machine translation,'' in Proc. 1st Conf. Mach. Transl., Shared Task Papers, Berlin, Germany, vol. 2, 2016, pp. 639–645

[8] L. Zhang, S. Wang, and B. Liu, ''Deep learning for sentiment analysis: A survey,'' Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery, vol. 8, no. 4, p. e1253, Jul. 2018.

[9] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, ''A generalized recurrent neural architecture for text classification with multi-task learning,'' in Proc. 26th Int. Joint Conf. Artif. Intell., New York, NY, USA, Aug. 2017, pp. 1–7.

[10] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.