Multi-class Sentiment Analysis using Deep Learning

Karan Patel
Department of Computer Science
Lakehead University
Student ID: 1111107
kbhaskar@lakeheadu.ca
Instructor: Dr. T.Akilan

***Abstract*—The given literature describes about the implementation of scalable and robust Convolutional Neural network(CNN) in order to solve the problem of text-based movie review sentiment analysis. I have used the Rotten Tomatoes movie review dataset to estimate the sentiment value. This dataset is available on GitHub. The sentiment value of the given movie review is the y attribute. Thus, I have created a non-regression model to predict the y attribute. The given model is in the scope of Convolutional and dense layers, associated operations like average and max pooling and non-linear activation functions like Rectified Linear Unit(ReLU), Sigmoid, tanh, Softmax, Leaky ReLU. The defined CNN model consists of 2 convolution layers. To obtain maximum accuracy and minimum L1Loss, I retrieved the value of parameters by performing hyper-parameter tuning. When I received the best values of hyper-parameter, my model got the accuracy of 61.8%.**

***Index Terms*—CNN, Sentiment Analysis, Text Vectorization, Keras, Natural Language Processing(NLP)**

I. Introduction

Text Classification which is also known as text categorization is the process of categorizing the texts into their respective organized groups. Identifying the type of email whether it is spam or not is the example of text classification. To analyze the text the text classifiers are used which assigns the set of categories on based on its content. This task can be achieved by using NLP. The process of understanding if a given text is saying something positive or negative about a given subject is known as sentiment analysis[1]. In this process the user interprets an opinion about a given input or subject from given text. This also applies to NLP, text analysis to thoroughly identify and extract affective states and the subjected information. The implemented model is CNN as it is used to solve the problems related to NLP. It consists of various layers which uses activation functions like ReLU. CNN is a non-linear model because of ReLU as it is itself a non-linear activation function. Sentiment analysis is part of the interpretation of natural languages and text. Such views are used for suitable steps, such as marketing decision taking, growth of business and much more. Based on the meaning there various sentiment values assigned accordingly. The sentiment values are shown in table 1.

TABLE I
SENTIMENT MEANING AND ITS VALUES

Sentiment Meaning	Sentiment Values
Very Negative	0
Negative	1
Neutral	2
Positve	3
Very Positive	4

II. Literature Review

To extract the features from various forms of data Deep learning network is used which learns its own features. Deep Learning is the subsection of machine learning and the Deep Neural Networks has multiple neural networks where the output of one network is feed as input of the other network. This type of structure solves the disadvantage of having number of hidden layers in the neural network. Operating with data becomes more feasible and effective with the use of deep neural networks.

CNNs are like Neural Networks(NN) which consists of neurons with learnable weights and biases and also CNN has a wide range of application in the domains like Machine Learning, image classification and natural language. The neurons receives various inputs and further it takes weighted sum over the inputs and passes through an activation function and gives the desired output[2].The input from the previous layer is being forwarded to each layer and maintains the relationship between the attributes. When the number of convolutions are performed it is then passed through ReLU which performs the non-linear operation and it is also crucial because it is mainly responsible to bring non-linearity in our network. Amongst, the Sum Pooling, Average Pooling and Max Pooling the most used one is the max pooling. When pooling is performed, the matrix vectors are flattened which are feed into the fully connected neural network. To bifurcate the outputs, softmax function is used.

III. Proposed Model

The proposed model is the CNN model in which Rotten Tomatoes moview review dataset is used to estimate the sentiment values. There are various libraries used in the proposed model to load the data. Fig 1 displays the implementation workflow for the proposed model.

A. Dataset and Libraries

a) Dataset: I have used the Rotten Tomatoes movie review dataset which available on GitHub. In the given dataset reviews are labeled with tags which shows whether the review

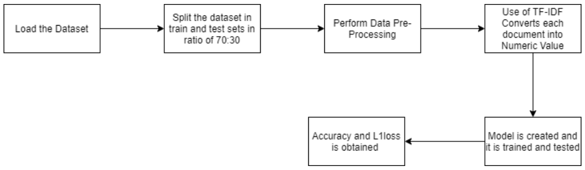
Load the Dataset
Split the dataset in train and test sets in ratio of 70:30
Perform Data Pre-Processing
Use of TF-IDF Converts each document into Numeric Value
Model is created and it is trained and tested
Accuracy and L1loss is obtained

Fig. 1. WorkFlow

is positive or negative. When the sentiment changes from negative to positive the labels have the given range from 0 to 4. The Rotten Tomatoes movie review dataset was originally collected by Pang and Lee[3]. Also, this dataset is a corpus of movie reviews which are basically used for the sentiment analysis. The random state value was assigned to 2003 while doing the division and the model is trained and tested in the ratio of 70:30. As the dataset was to be trained and tested according to the ratio of 70:30, the training instances were 109242 and testing instances were 46818 out of 109242 which are the total number of instances in the entire dataset.

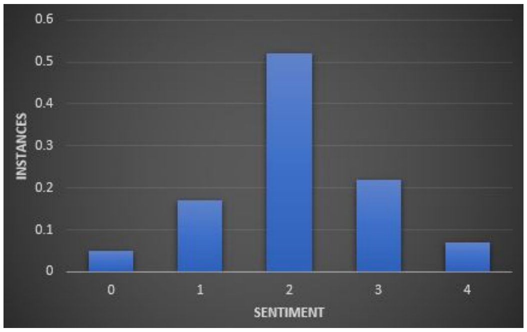

Fig. 2. Sentiment Distribution

b) Libraries: Keras, pandas, numpy, random, matplotlib, sklearn are the important libraries used in the implementation of this model. To perform different tasks I have used different libraries accordingly. Initially, to load the dataset pandas is used. Pandas is also used for performing classification using 1D-CNN. Mathematical calculations can be done using numpy library and it also converts pandas data-frame to array. Keras framework is used to build the CNN model. The random library generates a list like a range and then randomly returns one item from that given list.

B. Data Pre-Processing

The process of converting data to something a computer can understand is referred to as pre-processing. One of the major forms of pre-processing is to filter out useless data[4]. Removing Stop-Words, Removing Punctuations, Lemmatization are the major Pre-Processing steps which are included in the sentiment analysis. To perform the vectorization efficiently these steps are mandatory. The given dataset contains string so the time taken to analyze those strings increases as the dataset includes various punctuation and stop words. The Pre-Processing steps are described further.

a) Removing Stop Words: In natural language processing, useless words (data), are referred to as stop words[4]. I have removed stop-words(which include “the”, “a”, “an”, “in”) because these words don’t waste space in my database, and don’t take up valuable processing time. I can remove them easily, by storing a list of words that you consider to be stop words. I have used Natural Language Toolkit(NLTK) in python because it has a list of stop-words stored in 16 different languages.

b) Removing Punctuation: Punctuation won’t get removed while removing the stop words. Other than stop words, there various characters which should be removed. Punctuation removal is important as it may affect to the output also because the meaning of the sentence differs by using some punctuation. They can be removed manually. In my model I have removed these punctuation- ? : ! ; ( ) “ - . ,

c) Stemming: Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers[8]. To perform stemming NLTK library is used so that derivatives of the words are removed and inflexion is also eliminated.

d) Tokenization: Tokenization is the process of breaking down a sequence of strings into words, keywords, phrases and symbols called tokens. In this process the punctuation are discarded. Thereafter, these tokens become input for another process like parsing and text mining.

e) Lemmatization: Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item[5]. The purpose of stemming and lemmatization is to reduce inflexion forms and often derivatives of a word to a common basis. I have used the NLTK library to perform that.
So, these were the pre-processing steps used to eliminate irrelevant punctuation and words. Also, pre-processing reduces the number size of array as those words are removed from the dataset.

C. Text Vectorization

Machine learning algorithms take the numeric feature vectors as input. Thus, when working with the text documents, we need a way to convert each document into a numeric vector. This process is known as text vectorization[6]. I have used the Term Frequency-Inverse Document Frequency(TF-IDF) technique to perform the Word Vectorization. The reason behind using TF-IDF is that it functions in such a way that it reduces the values of common word which are used in different documents. Word Vectorization is used to map the words from vocabulary to corresponding vector of real numbers which is used to identify the word predictions or word semantics. TF-IDF mainly contains two concepts.

- Term Frequency(TF)
- Inverse Document Frequency(IDF)

TF means how frequently the word appears in the document. The mathematical expression for TF is as follows:

$$TF = \frac{Number\_of\_times\_word\_appear\_in\_the\_docs}{Total\_number\_of\_word\_in\_the\_docs}$$

IDF states the information about how to find the importance of the word. Here, the words which are less frequent are more informative and are very crucial in the document. The mathematical expression for IDF is as follows:

$$IDF = \log_{10} \frac{Number\_of\_docs}{Total\_No\_of\_docs\_in\_which\_word\_appears}$$

D. Inference time and Hyper Parameters

a) Inference time: Time taken to train the model is the Inference time. To calculate the inference time 'timeit' library is being imported.

b) Hyper Parameters: Hyper Parameters are those variables which are responsible for the model's accuracy because they determine that how a particular network is to be trained. Batch Size, Epochs, Drop-out rate, are the various parameters which I have used in my model.

E. Storing the model

At the end, I obtained my final results by performing various experimental analysis and doing many comparisons. As my keras CNN model was implemented, I stored the CNN model into drive by connecting my google drive with colab using mount function. Thereafter, I saved my model in h5 extension by using the keras model save function.

IV. EXPERIMENTAL ANALYSIS AND COMPARISONS

Hyper parameter tuning is very important as it could give the best accuracy to the model. The accuracy of my model varied as per the changes done in the parameters. At a certain point I got my optimum parameters which resulted into a good accuracy. Table 2 shows the optimum hyper parameters responsible for obtaining good accuracy.

Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses[7]. There are various optimizers like Adadelta, RMSprop, adam. Use of appropriate optimizer with proper batch size results into highest r2 score and lowest L1loss. I have compared the batch size with the optimizers and therafter observed the accuracy. When I allocated the batch size 64, epoch to 25 and the optimizer was Adadelta I got the highest accuracy of 61.8%. The comparison of optimizers with respect to batch size is being displayed in Table 3.

TABLE II
OPTIMUM HYPER PARAMETERS

Hyper-Parameter	Value
Optimizer	Adadelta
Epoch	15
Batch Size	128
Number of Layer	2
Loss Function	categorical_crossentropy

TABLE III
COMPARISON OF OPTIMIZER AND BATCH SIZE AND ACCURACY

Batch Size	Optimizer	Inference Time	Accuracy
264	Adadelta	494.72	0.592
128	Adadelta	503.39	0.601
64	Adadelta	483.17	0.618
264	RMSprop	471.89	0.561
128	RMSprop	494.52	0.572
64	RMSprop	483.03	0.57

The accuracy and L1loss also depends on the number of epochs assigned. When I carried out experimental analysis I came to a conclusion that accuracy increases as the epochs value increases which is shown in figure 3. On the other hand, the L1loss value will decrease when the values of epochs are increased.

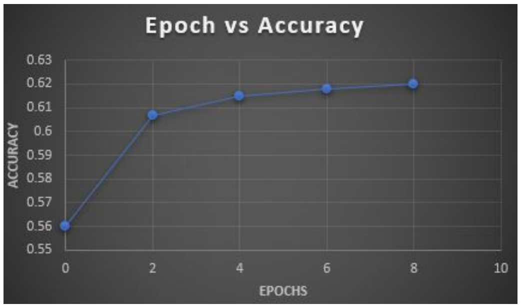

Fig. 3. Increase in Accuracy

V. APPLICATIONS OF SENTIMENT ANALYSIS

Sentiment analysis is widely applied to reviews and social media for a variety of applications ranging from marketing to customer service. This is also used to measure public opinion about an event or product. Also, used to track customer reviews, survey responses and competitor analysis. Customer analysis is also a major application of Sentiment analysis[9].

VI. FINAL RESULTS

I got my model's highest accuracy by performing appropriate parameter tuning and allocating batch size, epoch and optimizer accordingly. I carried out various trials by changing the parameters and obtained accuracy respective to the hyper parameters. Finally I got the optimum parameters which resulted into a good accuracy. Initially I loaded the dataset and performed training and testing accordingly on it. Therafter, I converted the movie review dataset into numerical form using TF-IDF. Table 4 displays the results of my trained model.

TABLE IV
RESULTS

Metrics	Values in %
Accuracy	61.58
Recall	53.84
F1 score	59.16
Loss	1.002
Precision	65.9

ACKNOWLEDGEMENT

I am very much thankful to Professor Dr. Thangarajah Akilan who assisted me in understanding the concept of CNN and othe NLP concepts and also helping me in the class labs by solving my doubts regarding some concepts. Also, I would like to thank the TA's Punardeep and Andrew who taught the sentiment analysis and other deep learning practical concepts in the lab and also solved my problems regarding the implementation of CNN.

VII. CONCLUSION

This assignment is based on predicting sentiment values based on movie review dataset. To perform the predictions, I have used the Rotten Tomatoes movie review dataset which consists of 5 sentiment values based on the meaning and also has more than One Lakh instances. Further, to make the predictions of the sentiment value based on the movie review I have used the CNN model. TF-IDF method is used to convert the sentences into the matrix format. Using the Keras framework, I used the matrix to construct the CNN model and then calculated it after accurate hyperparameter tuning. Eventually, I got the highest accuracy value of 61.85% after the hyper parameters are altered.

REFERENCES

[1] https://monkeylearn.com/what-is-text-classification/

[2] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[3] https://github.com/xingziye/movie-reviews-sentiment/blob/master/sentiment-analysis-movie.pdf

[4] https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

[5] https://www.geeksforgeeks.org/python-lemmatization-with-nltk/

[6] https://www.kaggle.com/edchen/text-vectorization

[7] https://medium.com/@sdoshi579/optimizers-for-training-neural-network-59450d71caf6

[8] https://www.geeksforgeeks.org/python-stemming-words-with-nltk/?ref=lbp

[9] https://theappsolutions.com/blog/development/sentiment-analysis-for-business/