A REPORT ON SENTIMENT ANALYSIS

Shiv Hansoti (ID:- 1105615)

***Abstract*— Sentiment analysis is the evaluation of people's reviews, attitudes, opinions, sentiments, and emotions expressed in textual form. It is trending topic in the field of natural language processing in present days. Its is has gained attention due to it's wide range of applications in living world. For instance, many time third party opinion or suggestion of other's are required to take decision in buying product or selecting most value by money gadgets. It is also a good dimension to explore as there are my topics which could be evolved if rigorous and through research is carried out. In this experiment Multi-class sentiment analysis is done on data of Rotten Tomatoes movie reviews.**

I. INTRODUCTION

Sentiment analysis is the method of natural language processing which uses normalization, text mining and analysis, statistics and probability to analyze human sentiment. The best businesses understand the sentiment of their customers—what people are saying, how they're saying it, and what they mean. Customer sentiment can be found in tweets, comments, reviews, or other places where people mention your brand. Sentiment Analysis is the domain of understanding these emotions with software, and it's a must-understand for developers and business leaders in a modern workplace. As with many other fields, advances in deep learning have brought sentiment analysis into the foreground of cutting-edge algorithms. Today we use natural language processing, statistics, and text analysis to extract, and identify the sentiment of words into positive, negative, or neutral categories.The main purpose is to understand the sentiment of their customers. It means that they should understand what the people are saying, how they are saying it and what it really means. These types of customer sentiment can be found in their tweets, comments, reviews or any other places where they mention the brand name. Now we are doing sentiment analysis and it is mainly done for understanding these emotions of the customers but with software. Sentiment analysis should be well understood by the software developers and the business leaders in the world. It has been brought to the foreground with the advancement in the deep learning algorithms. In this world, we are using NLP's, statistical modeling and text analysis to identify sentiments into different categories i.e. positive, negative, or even neutral categories.

II. LITERATURE REVIEW

A typical CNN architecture generally comprises of alternate layers of convolution and pooling followed by one or more fully connected layers at the end. In some cases, fully connected layer is replaced with global average pooling layer. In addition to different mapping functions, different regulatory units such as batch normalization and dropout are also incorporated to optimize CNN performance [45]. The arrangement of CNN components play a fundamental role in designing new architectures and thus achieving enhanced performance. This section briefly discusses the role of these components in a CNN architecture.

A. Convolution Layer

Convolutional layer is composed of a set of convolutional kernels (each neuron acts as a kernel).However, if the kernel is symmetric, the convolution operation becomes a correlation operation. Convolutional kernel works by dividing the image into small slices commonly known as receptive fields. Division of an image into small blocks helps in extracting feature motifs. Kernel convolves with the images using a specific set of weights, by multiplying its elements with the corresponding elements of the image receptive field.

B. Pooling Layer

Feature motifs, which result as an output of convolution operation can occur at different locations in the image. Once features are extracted, its exact location becomes less important as long as its approximate position relative to others is preserved. Pooling or downsampling is an interesting local operation. It sums up similar information in the neighborhood of the receptive field and outputs the dominant response within this local region.

C. Activation Function

Activation function serves as a decision function and helps in learning of complex patterns.Selection of an appropriate activation function can accelerate the learning process.

D. Batch Normalization

Batch normalization is used to address the issues related to internal covariance shift within feature-maps. The internal covariance shift is a change in the distribution of hidden units‘ values, which slows down the convergence (by forcing learning rate to small value) and requires careful initialization of parameters.

E. Dropout

Dropout introduces regularization within the network, which ultimately improves generalization by randomly skipping some units or connections with a certain probability. In NNs, multiple connections that learn a non-linear relation are sometimes co-adapted, which causes overfitting. This random dropping of some connections or units produces several thinned network architectures, and finally one representative

network is selected with small weights. This selected architecture is then considered as an approximation of all of the proposed networks.

F. Fully Connected Layer

Fully connected layer is mostly used at the end of the network for classification purpose. Unlike pooling and convolution, it is a global operation. It takes input from feature extraction stages and globally analyses output of all the preceding layers [61]. This makes a non-linear combination of selected features, which are used for the classification of data.

III. Data-set Explanation

There are many online movie reviewing platforms in which the assigned or enrolled user express this emotion or sentiment about a specific movie. And these platforms are open source so one can scrap the data or request sample data for performing experiments like sentiment analysis or opinion analysis on the comments or text that are posted by user for movies or TV-series which they have watched recently. Rotten Tomatoes is one of such platform. The selected Rotten Tomatoes data-set contains 3 columns

- Phrase ID
- Sentence ID
- Phrase sentiment

Phrase ID is the unique number or index number arrange in incremental order. Sentence ID is the number of different class of sentence. Phrase sentiment is the sequence of words which can be used to train model. The below shown sample table is a small part of data-set.

TABLE I
An Example of a Data-set

Phrase ID	Sentence ID	Phrase sentiment
1	1	A series of escapades demonstrating
67	2	quiet , introspective

IV. Code and Method Explanation

A. Loading Dataset

First, lets get view of data-set in google-colab. As the data file is a tab-separated file(tsv), we can use pandas to read data-set by passing arguments to tell the function that the delimiter is tab and there is no header in our data file. Then we set the header of our data frame.

- import pandas as pd (Importing data using url)
- url = 'https://raw.githubusercontent.com/cacoderquan/Sentiment-Analysis-on-the-Rotten-Tomatoes-movie-review-dataset/ master/train.tsv'
- response = urllib2.urlopen(url) (Function to open URL)

TABLE II
Output or view of dataset

Phrase ID	Sentence ID	Phrase sentiment
148904	8101	one Hollywood cliche
31670	1485	Old people will love this movie

B. Splitting data-set

Data-set was divided in 70:30 ratio. 70 percent data-set for training and 30 percent data-set for testing.

- Xtrain, Xtest, Ytrain, Ytest = traintestsplit(df ['Phrase'], df ['Sentiment'], testsize=0.3, randomstate=2560)

C. Getting data-set in array

- documents=[]
- Xtrain = np.array(Xtrain.values.tolist())
- Ytrain = np.array(Ytrain.values.tolist())

D. Pre-processing of data

Pre-processing and cleaning the data helps model to understand data more easily and it enhances the performance. For processing textual data there are various methods like Tokenization, Stemming, Lemmatization, Substitutions etc.

- porter = PorterStemmer()
- lancaster=LancasterStemmer()
- wordnetlemmatizer = WordNetLemmatizer()
- stopwordsen = stopwords.words("english")
- punctuations="?:!.,;'÷()"

E. Parameters defined in assignment

- removestopwords = True
- useStemming = False
- useLemma = False
- removePuncs = True

F. Vectorization transformation and TFIDF

In order to perform machine learning on text, we need to transform our documents into vector representations such that we can apply numeric machine learning. This process is called feature extraction or more simply, vectorization, and is an essential first step toward language-aware analysis.Representing documents numerically gives us the ability to perform meaningful analytics and also creates the instances on which machine learning algorithms operate. In text analysis, instances are entire documents or utterances, which can vary in length from quotes or tweets to entire books, but whose vectors are always of a uniform length. Each property of the vector representation is a feature. For text, features represent attributes and properties of documents—including its content as well as meta attributes,

such as document length, author, source, and publication date. When considered together, the features of a document describe a multidimensional feature space on which machine learning methods can be applied.The simplest encoding of semantic space is the bag-of-words model, whose primary insight is that meaning and similarity are encoded in vocabulary.

- vectorizer = TfidfVectorizer(maxfeatures = 2500), ngramrange=(1, 1))
- X = vectorizer.fittransform(df["text"])
- Y = df['sentiment']

G. Building a Convolution Neural Network for training

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme.

- making model sequential
- Adding 1D convolution layer with 64 filters and kernel size of 3 with and activation function of "RELU". Input shape= 2500,1
- Adding one more convolution layer and making it little complex with 128 filters, kernel size of 5 and giving activation function as "RELU"
- After Convolution layer adding a layer of MaxPool with pool size of 1.
- Adding a dropout layer with rate of 0.25.
- making model flat by flattening the layer
- At last making model dense by adding dense layer with classes and activation function of "SOFTMAX" as an input.

H. Implementing Loss Calculation

- Loss Function used:- Categorical CrossEntropy
- Optimizer:- Adam Optimizer
- Matrics:- Accuracy paramameter

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0.

Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks.The algorithms leverages the power of adaptive learning rates methods to find individual learning rates for each parameter. It also has advantages of Adagrad, which works really well in settings with sparse gradients, but struggles in non-convex optimization of neural networks, and RMSprop, which tackles to resolve some of the problems of Adagrad and works really well in on-line settings.

I. Result (Accuracy, F1 Score, Precision)

Accuracy	F1	Precision	recall
0.6291	0.6069	0.6852	0.5458
0.6189	0.5992	0.6744	0.5251
0.6423	0.6221	0.6983	0.5632

V. OTHER METHODS

There are various methods which could be used for more precise and accurate score like Recurrent Neural Network, Long Short term Memory but they were beyond the score of assignment.

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data

VI. CONCLUSIONS

The give document includes working of Sentiment Analysis using Convolution Neural Network. Statements or comments were taken from the Rotten Tomato Dataset which is online platform for posting movie reviews.It is open source and freely available to everyone for experimental purpose.

ACKNOWLEDGMENT

I would like to thank Dr. Thangarajah Akilan for providing qualitative assignments and for giving us a good knowledge for optimizing and making us understand practically trending topics Sentiment Analysis.

REFERENCES

[1] https://towardsdatascience.com/cnn-sentiment-analysis-1d16b7c5a0e7

[2] Jin Wang1,3,4, Liang-Chih Yu2,4, K. Robert Lai3,4 and Xuejie Zhang1 Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model

[3] Sentiment Analysis Using Convolutional Neural Network 4 Xi Ouyang ; Pan Zhou ; Cheng Hua Li ; Lijun Liu, 2015 IEEE International Conference

[4] https://en.wikipedia.org/wiki/Recurrentneuralnetwork.

[5] https://en.wikipedia.org/wiki/Longshort-termmemory.

[6] Lexicon Integrated CNN Models with Attention for Sentiment Analysis,Bonggun Shin, Timothy Lee, Jinho D. Choi,In Proceedings of the EMNLP Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, of WASSA'17, 2017.