Sentiment Analysis using Convolutional Neural Network

VINIT KRISHNANKUTTY
Student Id: 1096016
Email ID: krishnankuttyv@lakeheadu.ca
Lakehead University

I. ABSTRACT

The paper demonstrates a summary on implementation of Convolutional Neural Network which is strongly built and upgradable. This implementation is used to perform a multiclass sentiment analysis for movie reviews, which is text based. The dataset used for the sentiment analysis is from Rotten Tomatoes movie reviews considering the train dataset. The training is done based on different larger number of epochs. This work clearly illustrates the working of Convolution and dense layers, with the collaboration of pooling and ReLu layer. The concepts of TF – IDF and countVectorizer are used for the implementation. The entire dataset is divided into training and test sets based on a definite proportion of 70:30. A graph for the entire dataset is plotted to give a better idea of the various components of the dataset. The accuracy rate is improved by making using more than one convolution layer. Where the output of one convolution layer is fed as the input to the second convolution layer used. Recall and precision is calculated to identify how effectively the prediction is performed. Figure of merit is calculated for understanding test's accuracy rate. In statistical model, figure of merit make use of both precision and recall values. F1 score is calculated based on the mathematical equation using both precision and recall. The python programming using Colab is implemented in a modular fashion. This helped in understanding the sub programs in an efficient way. The debugging and modification of each piece of code can be performed effectively. A githhub account is created and uploaded the contents such as python code, summary report, dataset and the trained dataset.

Keywords: Convolutional Neural Network (CNN), Stochastic Gradient Descent (SGD), term frequency–inverse document frequency (TF-IDF), Long Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM)

II. INTRODUCTION

A. Brief introduction about the dataset

A movie review dataset is analysed to perform classification of sentiments. This is done by creating a multi-layer, one dimensional CNN. Google Colab is used for the implementation. The movie reviews are classified into five labels of sentiments: 0 for negative, 1 for somewhat negative, 2 for neutral, 3 for somewhat positive and 4 for positive. The challenging factors are negation of the sentence, sarcasm, terseness, ambiguity of the language etc. Stanford parser is used for parsing each sentence. Every sentence is attached with a sentence Id. The repetition of phrases are avoided. The training set contains phrases and the labels for the sentiment associated. In order to identify the phrase belonging to a particular sentence, sentence id is also included. The testing set contains the phrases and each phrase must be assigned with sentiment label.

B. Google Colab

Google Colab standards for the abbreviation of Google Colaboratory. This helps programmers to perform the python coding based on no environment for configuration. As the size of the dataset increases and crosses the limit to be handled by the normal computer system CPU, Google Colab provides with GPU which is free of cost. GPU is the abbreviation for graphics processing unit. The situations where the multitasking is difficult to be handled by CPU, GPU performs easy and quick rendering for images and videos which are of high resolution. Google Colab platform is created in such a way that it is worth for secure network sharing.

C. Convolutional Neural Network

Artificial Neural Network helps to perform the processing of information in large scale. It was developed based on the idea of the human brain. The processing pattern in the neural network is exactly same as the working of human brain. There are multiple layers in the neural networks, including the input layer, hidden layers and the output layer. The data will be fed into the neural network through the input layer, the processing mechanism is done through the hidden layers and the output will be displayed through the output layer. The main drawback with the CNN were that every neuron of one layer should be connected to every neuron of the layer which proceeds. Hence CNN was introduced where only limited number of neurons of one layer is connected to the neurons of the next layer. Figure number 1 illustrates different layers of CNN, which are convolution layer, ReLu layer, pooling layer, fully connected layer.

Convolution layer make use of filters, pass the filter through the given input, find the dot product of the filter values and

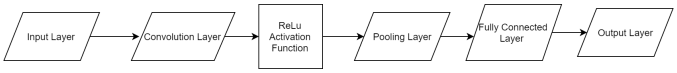


Fig. 1. CNN Layers

the input images part. Hence an activation map is generated. ReLu performs the activation function, the positive values will be maintained as it is and the negative values will be converted to zero. The dimensionality reduction is performed by pooling layer based on a window size selected randomly. The commonly selected window size is 2*2. Before the data is passed to fully connected layer, a flatten layer is introduced to convert the data into a single stack which is one dimensional. The output of flatten layer is fed into the fully connected layer, which exactly performs the classification of data.

D. Sentiment Analysis

It is the technique for identifying the emotion behind a collection of words based on attitudes, emotions and opinions which are registered online. The sentiment analysis mainly focus on business based on emotions of various people. An example can be demonstrated based on the various comments posted for a particular product. The posting can vary based on the usage of the product by different people. Based on the understanding the genuine comments, the products can be improved. To make a genuine analysis of the emotions, natural language processing can be used. The precision polarity can be enhanced to many categories if more efficient minute classification of emotions are to be made. The emotion detection which is a part of sentiment analysis, detects various moods such as happy, frustrated, sad, angry etc. Most of the systems used for emotion detection make use of lexicons, which explains the relation between words in the form of list and the emotions that correspond to each word. They also make use of deep learning or machine learning techniques. The different types of sentiment analysis are fine grained, aspect based, multilingual sentiment analysis.

1. Fine grained sentiment analysis considers wider classes of sentiments. It instead of just considering normal positive, negative and neutral sentiments classes, considers more detailed granules of sentiment such as very positive, positive, negative, very negative and neutral. The star ratings usually assigned to fine grained sentiment analysis are five star rating for very positive and 1 star rating for very negative.

2. Aspect based sentiment analysis is based on understanding certain aspects of the comments, like defining certain product as with short life indicates that the product is not good enough and has minimal life span.

3. Multilingual sentiment analysis, requires more resources and has to undergo more preprocessing steps, since it considers more than one language as the medium of exhibiting the emotions.

The main benefits with sentiment analysis is related to the organising and analysing the unstructured raw data. It helps the business in processing large volume of data in an effective and profitable manner. The sentiment analysis when applied to real time can help solve critical issues, such as handling an angry customer. Hence the remedies can be taken on spot. The centralised system for sentiment analysis can be incorporated in the companies to make sure that the strategy is applied to all the data while performing the process of sentiment analysis. This helps to improve the prediction accuracy rate.

E. The mechanism of sentiment analysis

The process of sentiment analysis make use of natural language processing and certain well defined algorithms. The various algorithms used by the analysis strategy are rule based, automatic and hybrid.

1. Rule based approaches make use a fixed set of definite rules to classify the emotions. These rules are defined by the programmers based of certain proven ideas. This intensively helps to identify polarity and the subject of the human opinion. The rules are framed based on various techniques. These ideas are obtained from linguistics study. The various techniques can be summarised as stemming, part of speech tagging, tokenisation and parsing. Lexicons are widely used too. The working methodology of rule based approach is initiated by defining two broad words with extreme polarity which are positive and negative. Proceeded by counting number of words which are positive and negative. If the count of words which are positive is more than that of negative, it returns a positive sentiment. A neutral sentiment will be generated when the number of positive and negative words are in balance. The rule based system is identified as unsophisticated as they do not care about how the words are framed into a sequence. Since the rule based approaches are defined by the human, it needs all time updating and fine tuning.

2. Automatic approaches do not rely on human developed rules instead make use of machine learning algorithms to define the rules and classify the emotions for sentiment analysis. Figure No 1 illustrates, the machine learning techniques which make use of a manual feature extraction unit handled by the experts in the domain.

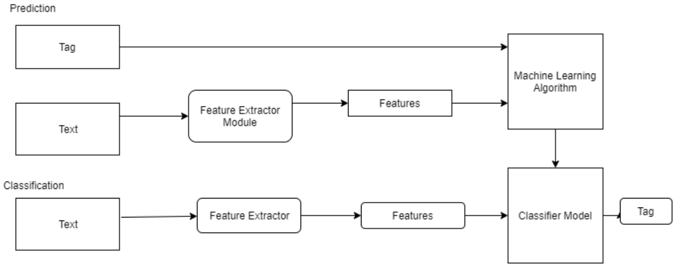


Fig. 2. Feature Extraction

The model so developed learns to associate the input sentence to a tag. The tag can positive, negative and neutral. The input is converted to corresponding vector by the feature extractor module. The tags along with the feature vector are fed into machine learning algorithm. The initial stage in the machine learning approach is text transformation. The ancient technique to implement this is using bag of words. The bag of words is also referred to as bag of ngrams. The various classification algorithms used are Naïve Bayes, Linear Regression, Support Vector Machine, Deep Learning techniques. The Naïve Bayes approach make use of the Naïve Bayes Algorithm to classify the dataset. Linear Regression is same as supervised learning, but predicts some values instead of labels. Support Vector Machine is a non-probabilistic model. It defines the text in a multidimensional space. A hyperplane draws the border between the two classes of data. The main advantage of using a CNN Deep Learning technique is that it does not require the feature extraction module as in machine learning. The features will be extracted by the layers in CNN, based on pre training. One a training is performed with a specific set of data, a new classification does not require another separate feature extraction. The main drawback with Deep Neural Network is that, it needs a huge dataset. The more the training, the more accurate the prediction will be. It is time consuming and cannot be run on normal platform, as it needs GPU for training and prediction of data.

3. Hybrid approaches make use of a combined effort of both rule based and automatic approaches for sentiment analysis. Hence it produces more accuracy than using rule based and automatic approaches independently.

III. Literature Survey

The first research work considers sentiment analysis improvisation based on classification of sentence. This is done based on BiLSTM-CRF and CNN [1]. The paper mainly focus on sentence classification based on divide and conquer approach, based on count of opinions, an automatic data driven mechanism that helps in extracting the features from the sentence taken as the input. It consider the traditional sentiment classification based on sentence level. This mainly consider a single technique for all types of data. The technique proposed in the research work is a divide and conquer methodology, which separates sentences based on various criterion and apply sentiment analysis technique individually on each of the sentences. A sentence is defined to be more complex, if it has more targets or outputs of sentiments. The work proposed make use of a model which sequence based implemented using neural network. Every opinionated sentence is classified into three categories. A one dimensional CNN is used and every sentence is given as the input to CNN independently.

The second research work focussed to get basic idea about implementation of sentiment analysis is a survey carried out on sentiment analysis based on various deep learning techniques [2]. It explains various social media, sites that demonstrate reviews, blogs creating data in the form of heaps. Sentiment analysis helps to process the unstructured raw data and identifies some meaning to help judge the context of comments posted. The paper defines sentiment analysis as classification of feelings or mind set in the form of mannerisms such as positive, negative, thumbs up etc. The work clearly explains various drawbacks with sentiment analysis which include the non-availability of labelled data. Considering the self-learning capacity of Deep Neural Networks, sentiment analysis is combined with them to overcome the shortcoming of labelled data.

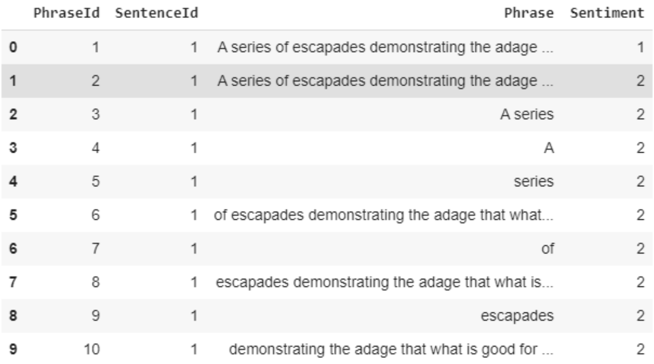
The research paper on sentiment analysis based on regional CNN-LSTM model [3], illustrates dimensional analysis of sentiments. This consider a multidimensional space for example Valence arousal space to identify numerical values which are non-discrete in nature. The dimensional approach of sentiment analysis considers the data for sentiment analysis on a wider dimension, and helps to perform classification based on fine grained approach. The research work depicts a clear comparison with the categorical approach which considers a binary classification of dataset. The proposed system illustrates a two way implementation using both CNN and LSTM. There is a comparison with the normal CNN, which considers the entire input text as a single unit, the CNN based on LSTM make use of division mechanism of input text into individual collection of words. Since there is a merging of CNN and LSTM, the prediction helps to consider the factors such as regional data and the sentence dependencies which are far in distance.

IV. Proposed Model

The proposed model illustrates a Convolutional Neural Network which is strongly built and upgradable. This implementation is used to perform a multiclass sentiment analysis for movie reviews, which is text based. The dataset used for the sentiment analysis is from Rotten Tomatoes movie reviews considering the train dataset. The model make an effective combination of different layer of CNN such as Convolution layer, dense layer, pooling which helps to reduce the dimension, and the ReLu layer which standardizes the data. The model is trained with 50 number of epochs. This make use of Bag of Words, TF – IDF, and Word2Vec. The entire dataset is split into training and test sets. The dividing is done based on the proportion 70:30, where 70 percent is used for training and 30 percent is used for testing. Various performance metrics used are based accuracy, recall, precision and figure of merit. Accuracy for a CNN is defined as the fraction of total number of accurate prediction to the total number of overall predictions. Recall indicates the ratio between total number of true positive to the sum of total number of true positive and total number of false negatives. Precision is the ratio between true positives to the sum of true positive and false positive. Figure of merit helps to categorise the performance of a device or a model. F1 score is calculated based on the mathematical using precision and recall values. It helps to validate the analytical model.

V. Experimental Analysis

For implementing CNN for sentiment analysis, python programming language is used with Colab platform. The various libraries used are pandas, sklearn.model-selection, numpy, random, nltk, nltk.tokenize, nltk.corpus, nltk.stem, sklearn.feature-extraction.text, torch, torch.optim, torch.nn, sklearn.preprocessing, spacy. Import csv, urllib.request, matplotlib. Read the dataset in the tsv format by providing the link and using tab as the separator. Then 10 data are printed using the data.head(10) command. The figure below illustrates the 10 data displayed. The display of data is done after clearing all the empty values in the dataset using the term drop=TRUE.



	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2
5	6	1	of escapades demonstrating the adage that what...	2
6	7	1	of	2
7	8	1	escapades demonstrating the adage that what is...	2
8	9	1	escapades	2
9	10	1	demonstrating the adage that what is good for ...	2

Fig. 3. Sample Data from Dataset

Training of the dataset is done after the process of splitting the dataset into training and testing data. Here in the proposed model, the proportion of division of dataset is 70:30, where 70 percent of the data is used for training and remaining 30 percent is used for testing. The train values are added to a list along both the axes. After performing the word tokenisation on the training data from the dataset, which helps to divide the sentences into simple tokens, the tokenised words from the training data is stored into a document using document.append function. Similarly the remaining 30 percentage for testing is added to the list. The sentences from the dataset for testing is tokenised to form individual smallest units of the sentence and appended to the document. These documents are there after printed. TfidfVectorizer is used to convert the text into word frequencies. The text is transformed into word count vectors using CountVectorizer. The TfidfVectorizer perform the tokenization of the documents, analyse and find out vocabulary and frequency weighting of the inverse document. Hence they perform the encoding of the new documents. CountVectorizer helps to tokenize the huge set of documents in the text format. This is proceeded by building vocabulary of words which are known. vectorizer.fit-transform identifies itself as feature extractor where it identifies the features based on which transformations in the future will be performed. vectorizer.transform helps in the conversion of feature dictionary data into a 2D matrix of features. In order to work with Keras model, the dataset is converted to numpy array, this includes both the training and the testing data. Figure number 4 explains the text classification mechanism.

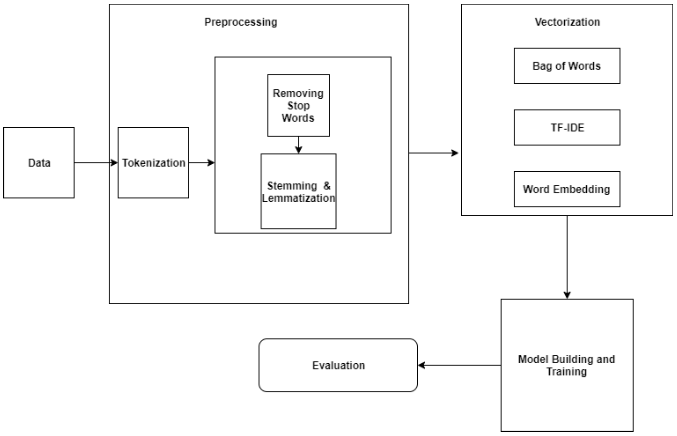


Fig. 4. Text Classification

The data is pre-processed using the concepts of stemming and lemmatization. For making use of the Keras, import to-categorical from keras.utlis. The method to-categorical, is used to convert vector class into a matrix of binary class values. Both the methods help to minimize the variation of data in terms of morphology. Stemming converts the words into stems, which does not care about the semantic of the data instead considers only syntax. Lemmatization is the process of converting the words into the root words, by taking into account the conceptual meaning of the root word in terms of semantics. The smallest indivisible word that is framed as a result of applying lemmatization is called as lemma, the lemma is defined to be indivisible further more. WordNetLemmatizer, PorterStemmer, LancasterStemmer are imported from nltk.stem library. Porter stemmer make use of porter algorithm which is defined as the algorithm which is minimum aggressive. Lancaster algorithm is defined as the most aggressive and the fastest algorithm. In order to adjust the outcome in the most efficient way, stopwords and punctuations are removed from the full sentences.

Import keras, from keras.datasets import mnist. Import sequential from keras.model. Minst dataset is the abbreviation for modified national institute of standards and technology database. It is often referred to as a smaller set of a huge NIST dataset. People who work in the field of research related to data science, make use of this dataset for the comparison with the dataset they use in their research work. Sequential model is a linear collection of layers in the pattern of stack. Backend as ‘k’ is imported from keras, in order to determine the current backend. This supports keras which is multi backend. In order to make the model understand about the shape of the dataset, it is going to handle with, sequential layers in the CNN should mention the shape. From keras.layers, the layers of CNN are imported, which are Conv1d, Maxpool1d, flatten, dense and dropout. Conv1d is the simple convolution layer that generates the filter and this filter is passed on to different parts of the input data. A dot product is preformed between the input data and the filter. MaxPool1d performs dimensionality reduction by passing a two by two matrix of stride. The output of the MaxPool1d is passed through the flatten layer, which converts the data into single list which is to be fed into the fully connected layer. The exact classification of data is performed in the fully connected layer. torch.nn.functional is used to import relu, softmax, and sigmoid activation functions.

Relu is an activation function also referred to as the step function, which helps to standardise the data. Such that negative values will be minimised to zero and positive values will be retained as it is. Softmax function makes the final output in the form of probability distribution, such that the total output values will add up to 1. Sigmoid function is also referred to as squashing function. This limits the values in the range between 0 and 1. It generates an S shaped curve known as the sigmoidal function. DataLoader and TensorDataset are imported from torch.utilis.data. DataLoader is responsible to read the data and store them into the memory. It can add, update, and remove data into the csv files. TensorDataset is based on tensor. Tensor is a mathematical unit, which indicates the correlation between the objects which are algebraic in nature with respect to a vector space. The number of convolution layer is increased to two so that the training can be performed in a better way. Hence the output of first convolution layer will be fed as the input to the second convolution layer. The convolution layer is defined based on three parameters which are number of input channels, number of output channels and the batch size. Linear layer takes number of inputs and the outputs. The output of the linear layer is fed as the input to the output layer. input.reshape((self.batch-size, self.inputs, 1)) is performed while the implementation, this helps to reshape the data which is fed into the input layer. The data is processed in form of one dimension, but the layer can accept the data in the form of a three dimensional array. The final output is fed into the softmax layer. This layer helps to convert the data into a probability distribution, so that the sum of the output values sum up to one.

SGD stands for Stochastic Gradient Descent. Stochastic Gradient Descent is an optimizer which helps to overcome the slow gradient and cost computation and provides an easy method to introduce a new data with respect to the settings which are online. SGD takes model parameters and the learning rate as the inputs. The learning rate is a parameter for tuning in machine learning and deep learning algorithm for optimization, determining the size of step in every iteration when the model proceeds to a minimum value of loss function. Adam is another optimizer that helps to overcome the drawbacks of SGD. It is defined as an optimization algorithm, which is adaptive learning. Since Adam is more efficient that SGD, Adam is used in the model proposed and implemented. In order to handle noisy issue in space gradient, Adam make use of two different algorithms such as RMSProp and AdaGrad.

Furthermore CrossEntropyLoss is calculated to measure the performance of the model. Cross - entropy loss helps to evaluate the classification model performance, which is expressed as a value between 0 and 1. Since Cross – entropy express the value of the output in terms of probability distribution, the implementation of softmax as the final layer of the CNN is not essential. It is calculated based on the entropy, and involves calculation of bit count needed for a data to get transmitted between two different distributions. The distributions can be any of two different mathematical distributions. keras.losses.categorical-crosssentrpy is used to calculate the cross entropy for the CNN proposed using the keras. The batch size is defined as 128. The model is created based on the CnnClassifier. The CnnClassifier includes the parameter such as batch size, number of X columns and number of Y columns. Categorical accuracy is calculated which is the measure of index of maximum actual value is balanced with the maximum predicted value. F1 score is the measure of accuracy associated with the test data. It is calculated based on the values of precision and recall. Recall indicates the ratio between total number of true positive to the sum of total number of true positive and total number of false negatives. Precision is the ratio between true positives to the sum of true positive and false positive. The model score is calculated and assigned to the variable score. It is calculated based on model.evaluate. The values of model performance determining factors are :

1. Cross entropy loss value: 1.18
2. Accuracy Value: 0.53
3. Recall Value: 0.37
4. Precision Value: 0.64
5. F1Score Value: 0.47

VI. Conclusion

This summary report provides the details on implementation of CNN to perform a multiclass sentiment analysis for text based movie reviews. The implementation was done in python language using Google Colab as the platform as the training needs GPU. The sentiment analysis was performed by CNN on Rotten Tomatoes movie reviews considering the train dataset. Based on the requirement of implementation, a model is created using CNN with strong bond between various layers such as convolution, ReLu, pooling, flatten and fully connected layer. The concepts of TF – IDF and countVectorizer are used for the implementation. The entire dataset is divided into training and testing data, with 70 percent for training and 30 percent for testing. SGD and Adam are the two optimizers used in the implementation of sentiment analysis. Once the model is implemented, accuracy, recall, precision, F1 score and Cross entropy values are calculated for analysing the loss values and to make the model more accurate. The python programming using Colab is implemented in a modular fashion. A githhub account is created and uploaded the contents such as python code, summary report, dataset and the trained dataset. My github link: https://github.com/vinitotp/Natural-Language-Processing-

References

[1] 1. Tao Chen, RuifengXu, Yulan He, Xuan Wang," Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN", Volume 72, 15 April 2017, Pages 221 – 230, Expert System with Applications 72 (2017) 221 – 230

[2] 2. Qurat Tul Ain, Mubashir Ali, Amna Riaz†, Amna Noureen‡, Muhammad Kamran‡, Babar Hayat and A. Rehman," Sentiment Analysis Using Deep Learning Techniques: A Review", Vol. 8, No. 6, 2017, (IJACSA) International Journal of Advanced Computer Science and Applications

[3] 3. Jin Wang1, 3, 4, Liang-Chih Yu2, 4, K. Robert Lai3, 4 and Xuejie Zhang1 Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model", August 7-12, 2016, Pages 225 – 230, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics

[4] 4. Natural Language Processing Lab 4, documents prepared by Punardeep Sikka and Andrew Fisher