Multi-class Sentiment Analysis using Deep Learning

Setul Patel
Department of Engineering
Lakehead University
Thunder Bay, Canada

***Abstract*—The analysis of opinions and emotions for the given form of text is defined as sentimental analysis. In other ways, it is also called ”opinion mining”. A huge amount of data is generated everyday and everyone shares their thought via social media websites like facebook, twitter, youtube and instagram. This paper includes a scalable and roburst CNN(Convolutional neural network) based solution for the problem of text-based movie review multi-class sentiment analysis. For that the raw trained data of Rotten Tomatoes movie reviews into a colab notebook is used. The model will be trained and evaluated on the basis of accuracy, recall, precision and F1 (Figure-of-merit) score.**

***Index Terms*—Sentimental analysis, opinion mining, CNN, deep learning, natural language processing, machine learning**

I. Introduction

Nowadays, people rely on each others views for the reference for their own needs. For example, people depend upon this user data generated data for analysis of the products they are buying or the movie they are going to watch. To analyse or conclude, overall data is too huge and wide ranged for normal human. Sentiment analysis is concerned with the identification and classification of opinions. Which is broadly concerned with the knowledge based approach and other machine learning techniques. As a result of this project, one will be able to find out whether the product or movie review is positive or negative. So the major steps to do the sentiment analysis are pre-processing of datasets, feature extraction , preparing and classifying the model. Here, pandas is used to manipulate with data and dataset which contains the data of 5 types of sentiment values according to the comments given. The dataset used here is divided in 4 columns such as phraseid, sentenceid, phrase and sentiment. First step in building the model is data cleaning in which main concerns are tokenization, punctuations, stopwords, stemming and lemmatization. For the cleaning part nltk(natural language is used which basically is a platform to build python program to work with human language data. Nltk is also called a specified tool for working and teaching in computational linguistic using python. Generally, data cleaning is defined as processing the data and provide the information which is going to be used to the machine. Then the next step is to split the database into training and testing part. The dataset is converted into arrays using numpy in order to work with keras. A numpy array basically is a grid of values, all of the same type and index by tuple of non-negative integers. The rank of the array is defined by the number of dimensions. Hence, the numpy array data is more precise to work with in order to get good results. Now, Sklearn is used for feature extraction from the data and then keras for training the model and testing it. Working with different amount of batch size, kernel size, number of layers and combination of activation function and optimizers to get optimum solution was last and challenging step of all.

II. Literature review

As stated by Sagar Chavan et al.[1] sentiment analysis alludes to the utilization of natural language processing, content analysis and computational semantics to recognize and remove abstract data in source materials. As of late, Opinion mining is at a hotspot in the field of natural language processing. Sentiment analysis is broadly applied to surveys and web based life for an assortment of uses, running from promoting to client care. Film surveys are a significant method to measure the presentation of a movie. The target is to separate highlights from the item audits and group surveys into positive, negative. To utilize Sentiment Analysis on a lot of film audits which is given by commentators and afterward attempt to comprehend what the general response to the film was by them, for example on the off chance that they preferred the film or they detested it. They intend to utilize the connections of the words in the survey to anticipate the general extremity of the audit. Since most opinions are available to us in the text format and its processing is easier than other formats, sentiment analysis has emerged as a sub-field of text mining.Sentiment analysis is used in different domains, ranging from analyzing tweets to comments on a particular website. Here they used naive-bayes multinomial method which works on the basis of bayes theorm with the assumption that features are mutually independent. In short here they are mining the text data given and then training the model. The accuracy on an average was around 73 percent.

III. Dataset

As the sentiment analysis is the process of identifying and classifying the expression of the data which may be shown in either text form or any other forms while the problem may be classification or regression. In binary classification problem, mostly the output may be either positive or negative. But if we think about the other outputs, here the problem is of multi class classification in which the dataset has 5 different types of solutions. The dataset used here is the Rotten tomatoes movie review dataset which is a corpus of movie reviews originally created by Pang and Lee[2] which was edited by Socher et al.[3] to get fine-grained labels. In the dataset is basically a table with four columns which is described in table 1. In which phrase id is number of rows in the dataset, Sentence Id is given by different types of sentences and different sentences gathered by the sentences, Phrase is the sentence phrase in which it may be a sentence or a part of sentence to get the original sentiment and sentiment is the category of the phrase given above is divided in 5 types of solutions (0 to 4) where 0 is for most negative and 4 is for most positive review.

TABLE I
ROTTEN TOMATOES MOVIE REVIEWS DATABASE

PhraseId	SentenceId	Phrase	Sentiment
1	1	A series of escapades demonstrating...	1
2	1	A series of escapades...	2
3	1	A series	2
4	1	A	2
5	1	series	2

Google colab is the environment used here to implement the model. To read the dataset and manipulate with it, pandas library is being used. The table 2 shows the number of entries in the dataset along with category of particular entry in which number 0 got least entries and number 2 got highest number of entries that means that there are less most negative reviews and most neutral reviews.

TABLE II
SENTIMENT VALUE COUNTS

Number of phrases	Sentiment
79582	2
32927	3
27273	1
9206	4
7072	0

IV. DATA PRE-PROCESSING

Now starts data preprocessing in which the data will be cleaned in order to detect and correct corrupt or inaccurate records from dataset and refer to identify incomplete, incorrect or irrelevent part of data and after all modify the data accordingly. The cleaning process includes tokenization, stemming and lemmatization. Tokenizing is the task of chopping the document unit into pieces called tokens which eventually means to throw away the useless characters.There are several types of tokenization as well including unigrams, bigrams and trigrams. The main purpose of it is to calculate the probability of sequence of tokens. Stemming is used to remove the suffix and prefix of a word to get the core word which is called stem. Lemmatization is used to remove inflectional endings and to remove base or dictionary form of word which is called as lemma. Data cleaning also includes removing punctuation as well.

Defining the data into training data and testing data is also an important step. So, now the data is split into 70-30 for training and testing portions respectively. Splitting the data is followed by vectorization. Regularly, the algorithm is used to operate on a single value at a time but to optimize the algorithm should work on a set of values which is called vectors. The CPUs nowadays provide support for vector operations where a single operation is applied to multiple data. This process is called vectorization and it has different types such as BoW (bag of words), TF-IDF (Term frequency-inverse document frequency) and Word2vec. Bag of words is representation of occurrence of word and has vocabulary of known words and measure of presence of known words. TF-IDF just calculates a term frequency that means the frequency of a word occurrence. It is usually used in information retrieval and text mining. Processing the word by vectorizing words and creating a group of related models that are used to produce word embeddings is called word2vec. Here I tried using all of them one by one and then also tried the combinations but at last, the code contains TF-IDF as vectorization module. The code for the vectorization module is shown in listing 1.

```
from sklearn.feature_extraction.text
import CountVectorizer , TfidfVectorizer
from keras.utils import to_categorical

 vectorizer = TfidfVectorizer(max_features = 2500)
X = vectorizer.fit_transform(dataset[”join”])
Y = dataset['sentiment']

 X_train = vectorizer.transform(X_train).toarray()

 X_test = vectorizer.transform(X_test).toarray()
```

Listing 1. Vectorization using sklearn and keras

V. PROPOSED MODEL

The most important step in any NLP or deep learning project is to train a model in a way that the model is optimal for getting best results with the given database. A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation. Figure 1 shows an example of a CNN in which there are certain inout levels and hidden layers between it and output layers. Usually in CNN, the connections between the units do not make cycles and neural networks comprise a hierarchy of processing levels. Each level is called a network layer and consists of a number of processing nodes. The CNN learns to map a given image to its corresponding category by detecting a number of abstract feature representations, ranging from simple to more complex ones. These discriminating features are then used within the network to predict the correct category of an input image. The function of a CNN is to automatic learning of a hierarchy of useful feature representations and its integration of the classification and feature extraction stages in a single pipeline which is trainable in an end-to-end manner. This reduces the need for manual design and expert human intervention.

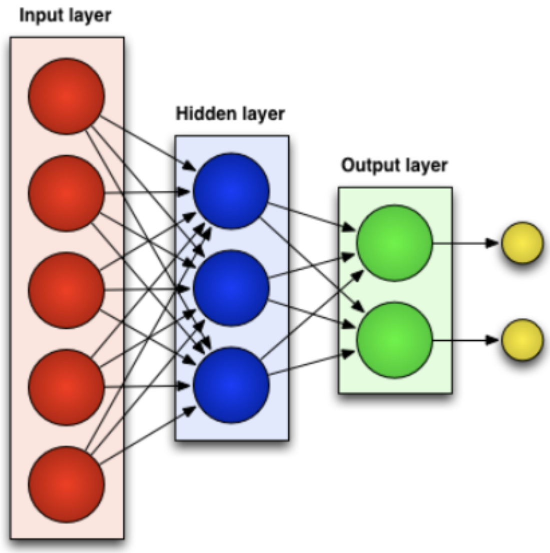

Fig. 1. Convolutional neural network

Source: Lecture notes, Dr.Emad Mohammad[6]

Keras is defined as a high level neural network API which was written in Python. It works as a wrapper for low level libraries like TenserFlow and others. The focus of making keras to enable fast experimentation for this kind of programs.In keras there are two kinds of APIs to work with. The sequential API permits you to make models layer-by-layer for most issues. It is restricted in that it doesn't permit you to make models that offer layers or have various sources of info or yields. Alternatively, the functional API permits you to make models that have significantly greater adaptability as you can without much of a stretch characterize models where layers interface with something beyond the past and next layers. Truth be told, you can associate layers to (truly) some other layer. Thus, making complex systems, for example, siamese networks and residual networks become conceivable. Here we are using sequential API so that we can go one by one in most of the problems of our dataset. I am importing dense, dropout, flatten, Convolutional 1Dimentional and max pooling layers to complete the model for training and testing. Dense layer in a neural network is like a regular neurons in which each neurons receives an input from all the neurons in the previous layer. Having 64 units and relu and sigmoid as an activation function, dense layer comes at last in the model as shown in listing 2. The main reason for dropout is to drop units during training randomly in which each unit retained with fixed probability and is independent of other units. The pooling layer is used at the end of convolutional layer. A pooling layer operates on blocks of the input feature map and combines the feature activations. This combination operation is defined by a pooling function such as the average or the max function. Similar to the convolution layer, we need to specify the size of the pooled region and the stride as defined here as 1. The pooling operation effectively down-samples the input feature map. Such a downsampling process is useful for obtaining a compact feature representation which is invariant to moderate changes in object scale and pose. In the model, pooling layer is applied on the data here which is text and making the data smaller but the meaning and any other attributes of the data does not change. Table 3 given above shows the summary of the model which was provided by the code given in listing 2.

TABLE III
MODEL SUMMARY

Layer(type)	Output Shape	No. of parameters
Conv1D$_1$9($Conv1D$)	(None, 2498, 64)	256
conv1d$_2$0($Conv1D$)	(None, 2496, 64)	12352
max$_{pooling1d_1}$0	(None, 2496, 64)	0
dropout$_1$0($Dropout$)	(None, 2496, 64)	0
flatten$_1$0($Flatten$)	(None, 159744)	0
dense$_1$9($Dense$)	(None, 64)	10223680
dense$_2$0($Dense$)	(None, 5)	325

Total Params : 10,236,613

Trainable params: 10,236,613

Non-trainable params: 0

```
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=3,
                 activation=’relu’,
                 input_shape=(2500,1)))

model.add(Conv1D(filters=64, kernel_size=3,
                 activation=’relu’))

model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(rate = 0.25))
model.add(Flatten())
model.add(Dense(64, activation=’relu’))
model.add(Dense(num_classes, activation=’sigmoid’))
model.summary()
```

Listing 2. Convolutional model formation

One of the most popular approaches for neural network regularization is the dropout technique (Srivastava et al.[5]). During network training, each neuron is activated with a fixed probability. This random sampling of a sub-network within the full-scale network introduces an ensemble effect during the testing phase, where the full network is used to perform prediction. Activation dropout works really well for regularization purposes and gives a significant boost in performance on unseen data in the test phase. Hence, the model and layers comes to an end with all the parameters provided and layers initiated.

VI. Analysis and results

After initiating model, the next step would be to train the model and analyse results based on different criteria such as accuracy, F1 score, precision measure, recall measure and loss. In the model, I tried to manipulate with the activation functions, number of layers for all the layers and then also with different number of epochs, batch size, filters, dropout rate and optimizers. The different type of activation functions here used to get better results are sigmoid, ReLU, softmax, tanh and leaky ReLU. Sigmoid takes a real-valued number and squashes it into range between 0 and 1. Sigmoid neurons saturate and kill gradients, thus neural network will barely learn when the neuron's activation are 0 or 1. Where as ReLU Takes a real-valued number and thresholds it at zero.

$$Relu(x) = max(0, x) \tag{1}$$

$$h_\theta(x) = \frac{1}{1 + e^{-}\theta^{T}x} \tag{2}$$

Equation 1 and equation 2 above shows the mathematical notation for ReLU and sigmoid activation functions which is used here in the model to get better results. At the first point, tanh, softmax were the activation functions I was using but the accuracy of the model was lower down to around 15 to 30 percent only. Not only the accuracy but other factors were also low. The importance of optimizer is also as high as defining a model and the optimizer is algorithm used to change the attributes of the neural network as learning rate and weights in order to reduce losses and get optimal results. The optimizers used in for evaluation and get different results to find the optimal solution were SGD, Adam, Adamax, RMSProp, Nadam and Adadelta. The results got by using these optimizers are provided in table 3 given below.

TABLE IV
Results based on different optimizers

Optimizer	loss	Accuracy	F1 score	Precision	Recall
SGD	1.25	0.51	0.51	0.51	0.51
Adamax	1.035	0.63	0.58	0.60	0.56
RMSProp	1.197	0.42	0.37	0.48	0.55
Adagrad	5.038	0.51	0.48	0.36	0.72
Adam	1.093	0.63	0.59	0.59	0.60
Adadelta	1.025	0.63	0.40	0.61	0.30
Nadam	1.14	0.63	0.62	0.58	0.66

VII. Challenges

The main challenge behind getting the best result was to choose right optimizer and build a proper model with the number of epochs, batch size, layers and other parameters to be arranged properly. The time when working with the model first to be perfect, I was working with one convolutional layer and one max pooling layer only. It did not give me the result I wanted to be precise. After that I added some layers and tried different activation function for all the layers and after all, the model came up with convolutional layer with ReLU as activation function and dense layer with sigmoid as activation function. Alternatively, setting up the optimizer was also a big issue such as while using RMSProp as optimizer, at the stage after 4th epoch, the accuracy of model decreased to 4 percent after being consistent to be at 50 percent till 3rd epoch. When working with Adadelta, the accuracy, loss and precision measure were not any issues but the F1 score and recall found to be 0.4 and 0.3 respectively. Nevertheless, for any deep learning models or any problem in which the model has to be trained, most probable issue is with overfitting and underfitting. Underfitting happens when machine learning model can not adequately get the underlying structure of the data. It means the data does not fit properly according to the model.

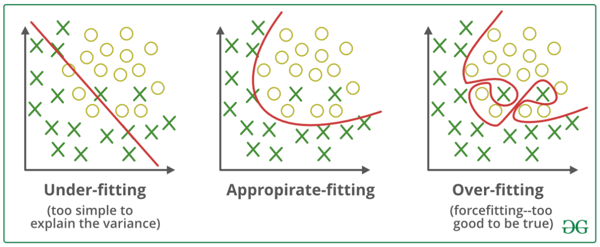
Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Fig. 2. Overfitting

Source: https://towardsdatascience.com

Overfitting happens when the noise of the data from dataset is captured by a statistical model or machine learning model. As shown in figure 2, underfitting happens when the number of parameters to train the particular model is less and overfitting occurs when it is high and model is trained for more number of epochs. To conclude, the model here contains the proposed model and Adamax as an optimizer with the results shown in table 4.

References

[1] Sentiment Analysis of Movie Ratings System. Sagar Chavan, Akash Morwal, Shivam Patanwala, Prachi Janrao. Department of Computer Engineering, Thakur College of Engineering and Technology, India.

[2] Pang and L. Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In ACL, pages 115–124.

[3] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Chris Manning, Andrew Ng and Chris Potts. Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).

[4] Rotten tomatoes movie review database, https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/overview.

[5] Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsk, Ilya Sutskever, Ruslan Salakhutdinov. University of toronto, 2014.

[6] Lecture notes, Dr.Emad Mohammad, Lakehead University, 2020.

[7] Lecture notes, Dr. Thangarajan Akilan, Lakehead University, 2020.