Multi-class Sentimate Analysis

Jay Sukhadiya (1110211)
Department of Computer Science
Lakehead University,
Thunder Bay, Ontario
Email: jsukhadi@lakeheadu.ca

***Abstract*—In the past several years, Sentiment analysis has become a trending topic of market research and of scientific in the field of Natural Language Processing (NLP) and Machine Learning (ML). It is used in many applications related to Social Media, data sets, ML, Visualizations, and Evaluation methods used in market studies. There are many algorithms and classification techniques that are destined to do a binary classification of sentiment of a text or a paragraph weather it is a positive or a negative sentence. Here we are using labels for phrases on a scale of five values: negative, somewhat negative, neutral, somewhat positive, positive. Obstacles like sentence negation, sarcasm, terseness, language ambiguity, and many others make this task very challenging. In this paper, We have used Pytorch library for model building and NLTK for text pre-processing. In this study, Sentiment is estimated using 1D Convolutional Neural Network (CNN). The database we used is Rotten Tomato Movie Reviews which contains PhraseId, SentenceId, Phrase, and Sentiment as columns.**

I. Introduction

Movie reviews are an important way to evaluate the performance of a movie according to audience. While providing a numerical or star rating to a movie tells us about the success or failure of a movie quantitatively, and a collection of these movie reviews is what gives us deeper qualitative insight of different aspects of movie. A textual information or a review gives us strong and week points of the movie and deep learning and analysis of these textual information of the movie tells us if the movie in general meets the expectation of the reviewer. The main task in sentiment analysis is to classify the polarity of a given text at the document, feature or sentence level-Weather the expressed opinion in the format of the text is positive negative or neutral. It is relatively easy to understand the success or failure of the movie using numerical or star system where a user gives his own opinion on movie as a score, but, when it comes to having a textual information as a movie review we have to take into consideration of features, sarcasm, actual meaning of words, and the sentiment behind it. Sentiment Analysis is a major subject in machine learning which aims to extract the subjective information and features from any textual information. The field of sentiment analysis relates to natural language processing and text mining. Using the sentiment analysis we can predict the likes and dislikes of user and weather the person was ”happy”, ”sad”, ”angry” or so on. There are many previous studies that uses only bi-classification problem where only polarized examples are considered.however, sentiment is often not unequivocal, and a bipolar view of sentiment has only limited application. In our study, we consider the problem of multi-class classification sentiment analysis. In total five labels are considered: negative, somewhat negative, neutral, somewhat positive, and positive.

In this paper:

- We have used Rotten Tomatoes movie reviews data set to train our model and get the accuracy score of movie reviews..
- Then we have used Google colab to connect to a GPU run-time and process out movie review data.
- Then we split the data into 70:30 ratio and used NLTK to pre-process the training data set.
- Then we used Pytorch library to create a cnn model to train our data set in batches and let it run for a set of epochs.
- Finally, we evaluated our model on the testing data set to see the performance of the model.

The remaining of this paper is organized as follows: we start by reviewing related work, followed by a description of our newly created data set. We then present our proposed baseline NN model. The experimental results section demonstrates the accuracy score, precision score, recall score and F1 score of our proposed model. Finally, we close with some concluding remarks.

II. Literature Review

The Convolutional CNN were used mainly on the 2D data like images and videos which lead to refereed them as “2D CNNs”. As an alternative, the modifies version of 2D CNN called 1D CNN has recently been developed. There are many application in which 1D Convolutional neural network are advantageous than its counterparts 2D CNNs in dealing with 1D signals. The reasons behind this are:

- For FP (Forward propagate from the input layer to the output layer to find outputs of each neuron at each layer) and BP (Compute delta error at the output layer and back-propagate it to first hidden layer to compute the delta errors) in 1D CNN requires simple array operations rather than matrix operations.
- 1D CNN with less number of hidden layers and neurons has ability to learn many challenging tasks compared to 2D CNNs with many number of hidden layers to handle same tasks.

- Training 1D CNNs takes very less memory due to its less number of hidden layers and few neurons, whereas 2D CNNs will require some special kind of hardware and software setup.
- Due to their less computational requirements , they are well suited for real-time and low-cost applications.

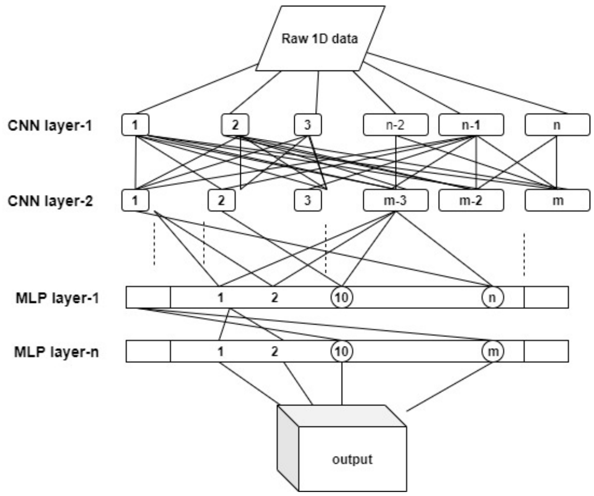

Fig. 1: 1D Convolutional neural network steps

As shown in fig.1 1D Convolutional Neural network consist of following hyper-parameters.

- CNN and MLP layers and neurons
- Filters in each layers
- Sub-sampling factors in each layers
- Pooling and activation functions

Sentiment analysis and opinion mining in social networks present nowadays a hot topic of research. However, most of the state of the art works and researches on the automatic sentiment analysis and opinion mining of texts collected from social networks and micro-blogging websites are oriented toward the binary classification (i.e., classification into “positive” and “negative”) or the ternary classification (i.e., classification into “positive,” “negative,” and “neutral”) of texts. [1] proposed a novel approach that, in addition to the aforementioned tasks of binary and ternary classifications, goes deeper in the classification of texts collected from Twitter and classifies these texts into multiple sentiment classes. With sentiment analysis techniques, we can automatically analyze a large amount of available data, and extract opinions that may help both customers and organization to achieve their goals. This is one of the reasons why sentiment analysis has been spread in popularity from computer science to management and social sciences. Sentiment analysis has also applications in review-oriented search engines, review summarization, and for fixing the errors in users ratings (such as for cases where users have clearly accidentally selected an incorrect rating when their review otherwise indicates a different evaluation) [2].

PhraseId	SentenceId	Phrase	Sentiment
73596	3763	the comic touch	3
50099	2461	A very slow , uneventful	1
73395	3751	look down on their working...	2
3773	143	completely	2
78691	4049	make it as much fun as re...	3
37283	1770	is either a more rigid ,...	3
114996	6124	romantic triangle	2
114899	6119	The story that emerges has...	3
21271	951	The Player , this latest...	2
114947	6120	bath house	2

TABLE I: Example of our database

III. Proposed Methodology

Data we used in this paper is publicly available to Kaggle users under the competition titled “Sentiment Analysis on Movie Reviews”. In the training file, there are 156,060 rows and 4 columns: Phrase Id, Sentence Id, Phrase, and Score (class). A phrase is a concatenation of words separated by a space, and is assigned a score of 0 through 4 which represents: negative, somewhat negative, neutral, somewhat positive, and positive according to increasing numbers. Here I shows several random results of our database after shuffling the rows. Here we can see the Phrase column as a text field containing a sentence, which we use as our input to get the column named as sentiment as our output. In fig. 2, the basic structure of our methodology is given.

Here we have used pandas library to read the Sentiment Analysis on Movie Reviews data set that we are going to use. The data set contains in total 156,060 data entries containing the review with it’s Id and Sentiment presentation as a number. Here each entry has different attributes like PhraseId, SentenceId. Phrase, Sentiment.

- We have imported numpy library to work with and manipulate the data. To split the data we have imported sklearn library. And by using numpy library we have read csv file using its internal function and we have also discarded any incomplete entries in the data set.
- We have taken sentiment as our output variable so that we can get the sentiment as result. We have taken Phrase columns from the data set as our input to use it to get the results which contains a sentence or a review that describes the movie according to a user.
- The next step is to check for the class imbalance in our training data tokenized sentences to get the idea of our dataset.
- We then split our data into training and testing sets using sklearn library.
- The next step is to get only the needed data to do the classification, so we remove some columns like PhraseId and SentenceId from our dataset. We used some preprocessing steps on this data like: stemming, lemmatization, removing stop words and punctuations, Tokenize the words.
- The main part of our class must be a subclass of the module of torch in which we defined the input layers,

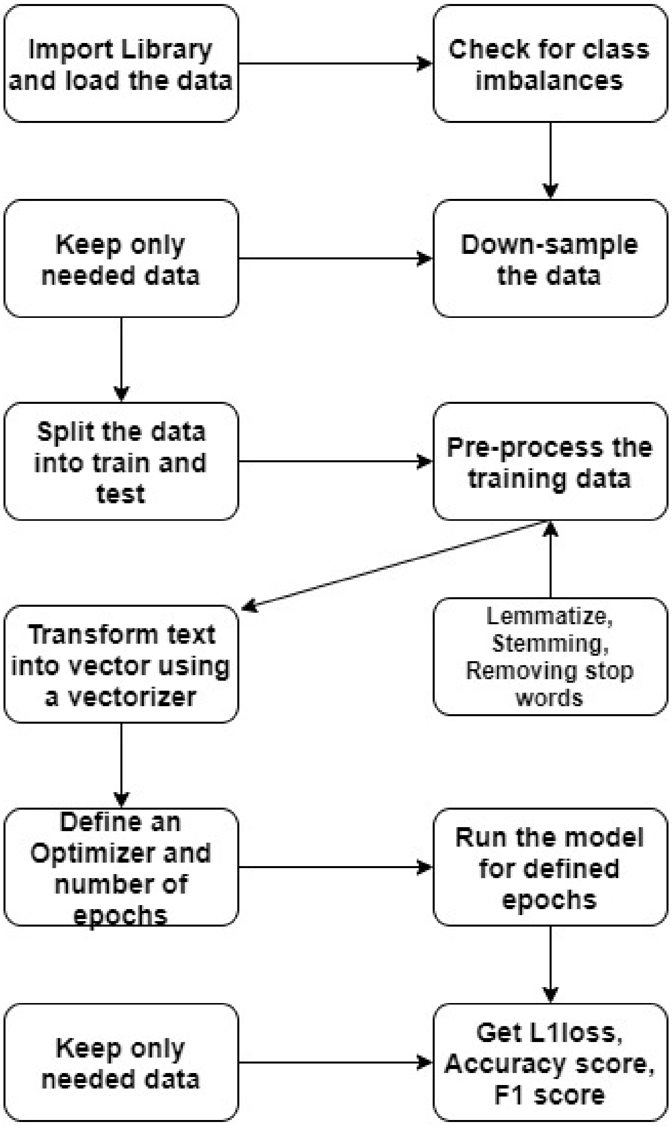

Fig. 2: General flow of our method

max pooling layer, convolutional layer, flatten layer, and linear layer.

- The next step is to define a method to feed inputs through input layers.
- Here we have tried with Adam, Admax, SGD etc. optimizer provided by a package imported from Pytorch library. Followed by importing L1Loss package and CrossEntropyLoss package from pytorch for our score measurement.
- Then we define a ModelLoss method which will return the average L1 loss and F1 score, accuracy of the passed model on the passed DataLoader. Here we define the number of batches we want to use to pass through every time to get the average values og L1Loss and F1 score.
- We have used different batch size to test our data to get the best result and finally the best batch size of 64.
- Then Finally we trained model for 100 epochs and got the trained model to use it in testing.
- Then finally we tested the model with our trained model and got the F1 score, Accuracy and L1Loss value.

Here we have shown a graph fig. 3, which shows different L1Loss, F1 score and Accuracy score obtained from each epochs from 1 to 100.

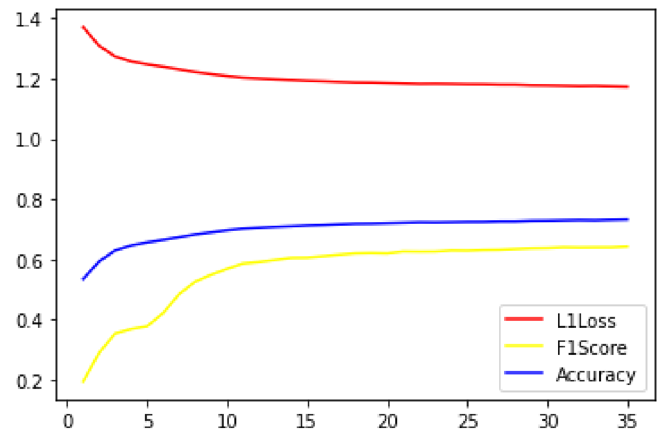

Fig. 3: General flow of our method

IV. Experimental Analysis

Here we have tried different outcomesto get the accuracy score and F1 score with respect to different Optimizers, Vectorizers, BatchSize, Learning rate, maxFeatures and many other parameters. Here Table.II shows different outcomes using different combinations of vectorizers and optimizers with training for different number of epochs.

Epochs	Vectorizer	Optimizer	Accuracy	F1 Score
100	TfidfVectorizer	Adam	0.39	0.21
75	CountVectorizer	Adamax	0.58	0.38
35	TfidfVectorizer	Adam	0.46	0.22

TABLE II: Experimental Table

V. Conclusion

In this paper, we have proposed an approach for sentiment analysis, where a set of reviews is to be classified into 5 different classes. The obtained results show some potential: the accuracy obtained for multi-class sentiment analysis in the data set used was 58. However, we believe that a more optimized training set would present better performances. Throughout this work, we demonstrated that multi-class sentiment analysis can achieve a high accuracy level, but it remains a challenging task. Our result shows that using right combination of Optimizers, Vectorizers, BatchSize, Learning rate, maxFeatures and many other parameters in the proposed model we can achive higher accuracy and required F1 score.

References

[1] MONDHER BOUAZIZI, TOMOAKI OHTSUKI, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter", Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, IEEE, 2014

[2] B. Pang and L. Lee, “Opinion mining and sentiment analysis,”Foundations and trends in information retrieval, vol. 2, no. 1-2, pp. 1–135, 2008.