Sentiment Analysis using Convolution Neural Network

Pathikkumar Patel(ID: 1117477)
Department of Computer Science
Lakehead University
Thunder Bay, Canada
ppatel73@lakeheadu.ca

***Abstract*— During the last few years, CNN models have been extensively used and they have proven to be better for image and video related problems. However, 2D CNNs are not designed to operate as efficiently and effectively on a time series or one dimensional data. 1D CNNs have achieved state-of-the-art performance levels in several applications such as personalized biomedical data classification and early diagnosis, structural health monitoring, anomaly detection and identification in power electronics and motor-fault detection. In this article we will review Convolution Neural Network and discuss its architecture along with its applications. A specific application of predicting sentiment using CNN will be discussed in detail along with the results obtained.**

***Keywords*—Convolution Neural Networks, Machine Learning, Deep Learning, Sentiment Analysis**

I. Introduction

Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. The dataset selected for this experiment is the Rotten tomatoes movie review dataset. It is an opensource dataset and is easily accessible on the internet. The dataset contains two files train.csv and test.csv which contains the train and test dataset respectively, However, for this experiment we will only use the train.csv file for training and testing purposes. The dataset has four columns in total namely: PhraseId, SentimentId, Phrase and Sentiment. The PhraseId column has Ids for different phrases for the same sentence and the SentenceId remains the same for all the phrases belonging to a same sentence i.e a movie review. There are no missing values in the dataset.

This article will use a Convolution Neural Network model to predict the *Sentiment* of these phrases. Pytorch framework has been used to develop and test the model. The pipeline of the experiment is stated as follows:

- Data Preprocessing
- Splitting the data
- Numerical Representation of data
- Model Creation
- Evaluating the model

Data pre-processing, in this case means tokenization, stop-word removal, punctuation removal and stemming. The data loading and pre-processing steps have been carried out using the torchtext library. The data has been randomly split into 70:30 ratio belonging to the training and testing set respectively with random seed of 2003. Torchtext uses two parts namely TEXT and LABEL to read the data. Here, TEXT contains all the phrases and the LABEL contains the Sentiment of these phrases. A preview of the dataset is as shown in figure-1.

	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

Figure-1

The rest of the paper is organised as follows: Section II provides the Literature review which is followed by Section III shows the Exploratory Data analysis. Section IV explains the system architecture and Section V talks about the results obtained. Section VI gives a conclusion which if followed by the references.

II. Literature Review

1D CNN is a modified version of traditional 2D CNNs which were designed to operate on an image or a video. The authors of [2] have shown that the 1D CNN model has achieved higher performance in terms of accuracy than the traditional methods. It has shown that 1D CNNs perform better for some applications than their 2D counterparts. The reasons for this are stated as follows: (1) Rather than matrix operations, FP and BP in 1D CNNs require simple array operations. This means that the computational complexity of 1D CNNs is significantly lower than 2D CNNs. (2) Recent studies show that 1D CNNs with relatively shallow architectures (i.e. small number of hidden layers and neurons) are able to learn challenging tasks involving 1D signals. On the other hand, 2D CNNs usually require deeper architectures to handle such tasks. Obviously, networks with shallow architectures are much easier to train and implement. (3) Usually, training deep 2D CNNs requires special hardware setup (e.g. Cloud computing or GPU farms). On the other hand, any CPU implementation over a standard computer is feasible and relatively fast for training compact 1D CNNs with few hidden layers. (4) Due to their low computational requirements, compact 1D CNNs are well-suited for real-time and low-cost applications especially on mobile or hand-held devices. The authors of [3] designed and trained a 1D CNN to locate and quantify structural damage in a five-story structure. Qianzi Shen *et.al*[4] have proposed an approach where thye combine convolutional neural networks (CNNs) and BLSTM (bidirectional Long Short-Term Memory) as a complex model to analyze the sentiment orientation of text. They have shown that this structure gives better results than a single CNN or an LSTM model. The authors of [5] have used CNN for Sentiment analysis of twitter data and have shown that it performs better than the traditional methods used for classification purpose. Akhtar Shad *et.al.*[6] proposed a method to learn sentiment embedded vectors from the Convolutional Neural Network (CNN). These are augmented to a set of optimized features selected through a multi-objective optimization (MOO) framework. The sentiment augmented optimized vector obtained at the end is used for the training of SVM for sentiment classification.

III. Exploratory Data Analysis

The distribution of different classes in the dataset is as shown in figure-3. The sentiments are divided into five classes and figure-3 shows the frequency associated with each sentiment class.

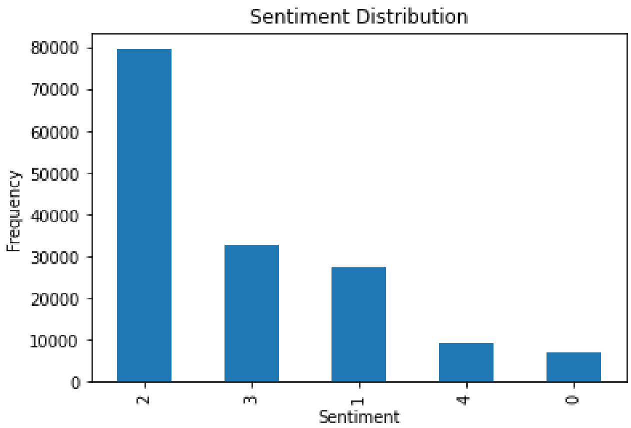

Figure-2

IV. System Architecture and Experimental Setup

The configuration of a CNN is formed by the following hyper-parameters: (1) Number of hidden CNN and MLP layers/neurons (2) Filter (kernel) size in each CNN layer(3) The choice of pooling and activation functions. The model used in this experiment has one Convolution layer along with a Max Pooling and a flatten layer. The kernel size for the convolution layer is set as four. Max-pooling is performed with a stride of one and the flatten layer flattens the output into a one dimensional vector. There are two dense layers in the network. The flattened vector is then fed as an input to the fully connected layers. The output of these dense layers is then passed through a softmax layer which calculates the probabilities for each of the class. The Sentiment Feature is the target variable in this experiment. The batch size used in this experiment is 512. The learning rate set for the experiment is 1e-5. The loss measure used here is the CrossEntropyLoss and the performance metric used are Accuracy, F1 score, Precision and Recall. The model summary is as show in figure(4).

```
CNN_Text(
  (embedding): Embedding(16768, 100, padding_idx=1)
  (conv): Conv1d(50, 100, kernel_size=(3,), stride=(2,))
  (conv2): Conv1d(100, 150, kernel_size=(4,), stride=(1,))
  (conv3): Conv1d(150, 200, kernel_size=(4,), stride=(1,))
  (conv4): Conv1d(200, 250, kernel_size=(4,), stride=(1,))
  (flatten): Flatten()
  (linear): Linear(in_features=10000, out_features=4096, bias=True)
  (bn1): BatchNorm1d(4096, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (linear2): Linear(in_features=4096, out_features=5, bias=True)
)
```

Figure-3

V. Results

The model was trained on a GPU based environment on pytorch framework. The parameters used in this experiment are described in the previous section. The CrossEntropy Loss obtained on the training set is 0.53 and the accuracy score obtained is 78.99. The CrossEntropy Loss and accuracy score obtained on the test set are 1.18 and 61.08 respectively. The accuracy plot for training and testing sets are shown in figure-5. The Loss plot for training and testing sets are shown in figure-6.

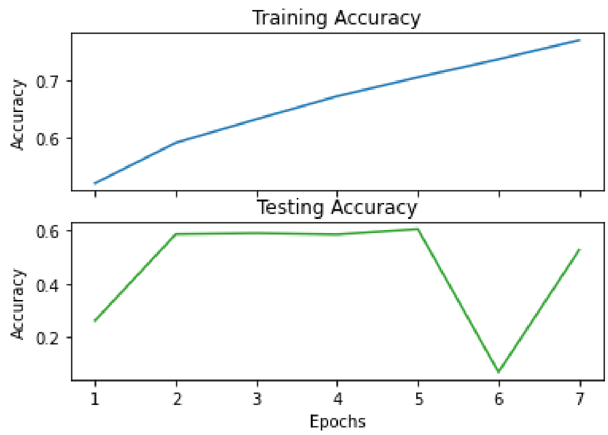

Figure-4

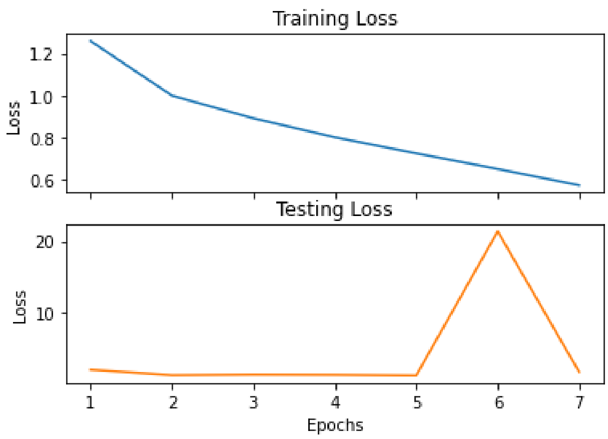

Figure-5

Table-1 gives us details of different accuracy measures to measure the performance of the model.

Performance Measure	Score
F1	0.51
Precision	0.48
Recall	0.49
Accuracy	61.08%

Table-1

A comparison of the 1D Convolution Neural Network with other deep learning methods is shown in table-2:

Classification Model	Accuracy(%)
1D CNN	61.08
SVM	54.27
RNN	66.97
LSTM	72.81

Table-2

VI. CONCLUSION

It can be said based on this experiment that 1D convolution does not perform as compared to other complex Recurrent state models. The results obtained are not as up to the mark as obtained from other methods. However, CNN is an effective model that can provide better performance than most of the state of the art model architectures. The accuracy of the current model can also be increased by further fine-tuning of the hyper-parameters and further pre-processing of the dataset. Compact 1D CNNs can promise a sole advantage of being applicable to those applications where the labeled data for training is scarce and a low-cost, real-time implementation is desired.

VII. REFERENCES

[1] Pace RK, Barry R. Sparse spatial autoregressions. Statistics & Probability Letters. 1997 May 5;33(3):291-7.

[2] Kiranyaz S, Ince T, Hamila R, Gabbouj M. Convolutional neural networks for patient-specific ECG classification. In2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2015 Aug 25 (pp. 2608-2611). IEEE.

[3] Avci O, Abdeljaber O, Kiranyaz S, Inman D. Structural damage detection in real time: implementation of 1D convolutional neural networks for SHM applications. InStructural Health Monitoring & Damage Detection, Volume 7 2017 (pp. 49-54). Springer, Cham.Aslam N, Xia K, Ali A, Ullah S. Adaptive TCP-ICCW congestion control mechanism for QoS in renewable wireless sensor networks. IEEE sensors letters. 2017 Oct 2;1(6):1-4.

[4] Shen Q, Wang Z, Sun Y. Sentiment analysis of movie reviews based on cnn-blstm. InInternational Conference on Intelligence Science 2017 Oct 25 (pp. 164-171). Springer, Cham.

[5] Liao S, Wang J, Yu R, Sato K, Cheng Z. CNN for situations understanding based on sentiment analysis of twitter data. Procedia computer science. 2017 Jan 1;111:376-81.

[6] Akhtar MS, Kumar A, Ekbal A, Bhattacharyya P. A hybrid deep learning architecture for sentiment analysis. InProceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers 2016 Dec (pp. 482-493).