Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews

Md. Rakibul Haque
Dept. of Computere Science & Engineering
Rajshahi University of Engineriing & Technology
rakibulhaq56@gmail.com
Rajshahi, Bangladesh

Salma Akter Lima
Dept. of Computer Science & Engineering
Rajshahi University of Engineriing & Technology
lima17055@gmail.com
Rajshahi, Bangladesh

Sadia Zaman Mishu
Dept. of Computer Science & Engineering
Rajshahi University of Engineriing & Technology
sadia.cse09@gmail.com
Rajshahi, Bangladesh

***Abstract*—With the huge expansion of text data sentiment analysis is playing a crucial role in analyzing the user's perspective about a particular product, company or any other physical or virtual entity. Sentiment analysis helps us to analyze user review about an entity and then drawing out a conclusion based on the sentiments it extracted from the reviews. Convolution Neural Network (CNN) and Long-Short-Term Memory Network (LSTM) are two well-known deep neural networks used for sentiment analysis. In this paper, we have compared between CNN, LSTM and LSTM-CNN architectures for sentiment classification on the IMDb movie reviews in order to find the best-suited architecture for the dataset. Experimental results have shown that CNN has achieved an F-Score of 91% which has outperformed LSTM, LSTM-CNN and other state-of-the-art approaches for sentiment classification on IMDb movie reviews.**

Keywords— Sentiment analysis, IMDb movie reviews, CNN, LSTM, LSTM-CNN, NLP.

I. INTRODUCTION

Sentiment Analysis is one of the most popular and trending research field in natural language processing (NLP) and text mining. It is becoming one of the most important and at a time one of most the interesting research area because most of the product's success depends on the review that it gets online. Sentiment analysis helps us to understand the relationship between natural text and humans emotions or judgment. It helps us to review a person's perspective about an entity which means a great deal to the producer of the entity. For instance, in this age, nobody goes to watch a movie unless they heard some good reviews of that film in social media or from some film critics The case is also the same in buying products. So reviews are taking on the marketing world. For this reason, an important and critical factor is to reduce the error and complexity for predicting the sentiment behind a review. In the past era In the past era different machine learning approach like Naïve Bayes, SVM [1] have been used for sentiment analysis which produced fare results.

Recently Deep Neural Network architectures have been performing significantly in NLP tasks because of the increasing amount of information. Convolutional Neural Network (CNN) is one of the most popular neural network architectures in image classification because of its compositionality and local invariance. CNN has also shown significant performance in natural language processing for its unique identification technique as it can easily identify a catchphrase in the natural text which is required in sentiment analysis. To predict negative or positive sentiment, we need to find the positive and negative sentiment and CNN does this for us. Ouyang, Zhou, and Li have applied a three Layer CNN on the movie review dataset from rottentomatoes website [2]. Kim [3] have also applied different variants of Convolution Neural Network for sentence classification. CNN has less connection for that it takes less time to train which is a great advantage.

Recurrent Neural Network (RNN) is another variant of neural network that can closely model the structural dependency of short text or sentences. However, it cannot model long term dependency because of its vanishing gradient problem [4]. This means it cannot store long term dependency and thus cannot fully represent the syntax of a sentence. For this reason, Hochreiter [4] has proposed LSTM which is an improved version of Vanilla RNN and provides long term-dependency. LSTM is largely used in different kind of NLP tasks that is mainly based on the structural dependency of the sentence such as prediction of next word in a sentence, question answering, translation task, image captioning, etc. Recently, Li [5] has shown that LSTM has performed significantly in text sentiment analysis than vanilla RNN.

Finally, we have added a layer of LSTM nodes before CNN architecture in order to create LSTM-CNN architecture. We have gone for the simplest LSTM-CNN architecture as training LSTM takes much more time compared to CNN architecture. We have used this LSTM-CNN in order to extract the features in sentiment classification.

In this paper, we have first used a word embedding method in order to create a vector space for the sentences in the review, then we have applied three neural network architectures CNN, LSTM, and LSTM-CNN architectures separately in order to extract the features in the sentences and after that the extracted features have been fed into a multi-layer perceptron network for classification of positive and negative sentiments. The main objective of the paper is to find the best suited deep neural network architectures that lead to better classification result on IMDb movie reviews dataset. We have implemented our architectures using a deep learning framework called Keras [6] which is a high-level API that runs on top of Tensor flow. Tensor flow has been developed by google in order to provide an efficient framework for deep learning.

The remainder of the paper is organized as follows. Section II reviews the architectures we have used to analyze IMDb movie reviews. In Section III, we have shown the performance result and competitive comparison of the architectures we have used with other state-of-art method for sentiment analysis. Finally, in Section V, we have drawn some conclusion based on our experimental analysis.

978-1-7281-6410-6/19/$31.00 ©2019 IEEE

II. METHODOLOGY

A. Dataset Description

This dataset is one of the largest IMDb movie review dataset [7]. The dataset contains 50,000 movie reviews belonging to two categories either positive or negative. We have obtained this dataset from Keras [6] where each review is encoded as a sequence of word indexes. Then we have split the dataset into a 70:30 training and testing data. 20% of the training samples has been used as validation data. We have made the reviews same in length by zero-padding the shorter reviews so that is easy to train the architectures.

B. Word Embedding

Word Embedding is one of the main prerequisites for most of the NLP tasks where we deal with natural text. While working with any kind of text. Each word must be converted into an n-dimensional vector for fitting to any architecture. Now, there is two way of doing this. One is one-hot key encoding which includes Bag of words model and another one is word embedding. Bag of words is a very sparse representation which results in a waste of memory whereas Word embedding provides a compact representation of each word. Word embedding is done in a way that similar kinds of words are gathered together in n'th dimensional space. Here n represents the vector dimension of each embedded word. There are two pre-trained word embedding models. One is word2vec [2] and another one is Glove [8]. However, we used an embedding layer that has been provided by Keas, where we used a vocabulary of 8000 unique words and each word is embedded into a 100 dimension vector space. Instead of using a pre-trained embedding word model, we have trained our embedding layer by using the training samples from IMDb movie review dataset.

C. Convolutional Neural Network

The Convolutional neural network mainly consists of mainly three Layers [2]. One is the convolution layer in which the input matrix is being convoluted with the filters. Filter which is also known as the kernel is used to determine specific feature. Many filters are applied in a CNN. The weights of the filter are initialized from glorot-uniform distribution. Later it is adjusted to the weight for detecting a specific feature through training the Network. The second one is pooling layer in which the pooling process happens which combines the output of neuron cluster at one layer into a single neuron next layer. There are different types of pooling such as Max Pooling, Average Pooling. In most of the cases, Max pooling is used. Pooling provides a form of translation invariance. Pooling layer is used between two convolution layer. The third one is a fully connected layer which is used after convolution and pooling layers. The function of the layer is high-level reasoning from the low-level feature detected in convolution layer.

The input of the CNN consists of 500 words per review which are fed into embedded layer to create a 100-dimensional vector for each word. We have used two convolutional layers in order to extract features and two pooling layers to provide translation invariance. We have used ReLU [9] activation function in convolutional layer. The output of convolutional and pooling layer are fed into a fully connected layer which in turns feed the extracted features into a hidden layer for classification. The output layer consists of one node with sigmoid activation function as it is a binary classification problem.

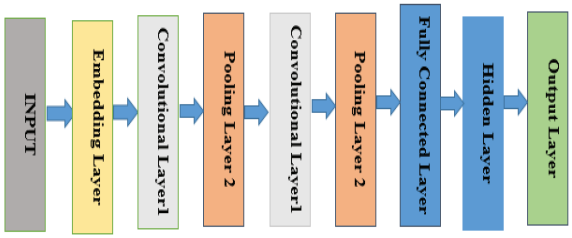

Fig. 1. Higher level representation of CNN architecture used for sentiment classification on IMDb movie reviews.

D. Long Short Term Memory

LSTM is an improvised version of RNN which supports long term dependency. Similar to the RNN, LSTM has a temporal loop inside its Layer but the basic difference between LSTM and RNN is the memory cell which can store or update information based upon the provided input sentences.

There are mainly three Gates in an LSTM cell. One is Forget Gate layer which is a Sigmoid layer that decides the information that needed to throw away from the memory cell. Second One is Input Gate Layer which determines what new information is going to be stored in the memory cell. It is further divided into two-Layer, one is Sigmoid layer and another one is tanh layer. The third one is Output Gate Layer which determines the output of the corresponding LSTM Cell. Mathematically, the gate can be defined by

$$F_t = \sigma(W_f . [h_{t-1} , x_t] + b_f) \quad (1)$$

$$I_t = \sigma(W_i . [h_{t-1} , x_t] + b_f) \quad (2)$$

$$O_t = \sigma(W_o . [h_{t-1} , x_t] + b_o) \quad (3)$$

$$h_t = O_t * \tanh(C_t) \quad (4)$$

Here F_t,I_t and O_t represent the forget, input and output gate Layer. b_f, b_o represent the bias factor of those layers. x_t and h_t are the input and output of the current unit and h_{t-1} is the output of the previous input $x_{t-1.}$]. σ]. σ represents sigmoid layer with sigmoid activation function and tanh represents tanh layer with a tanh activation function[10] [11]..

The input to our LSTM network is the same as CNN. In our LSTM network, we have used only one LSTM layer after the Embedding Layer in order to avoid the complexity of the model. We have used 200 LSTM units in the LSTM layer. The features extracted using LSTM has been similarly fed into an MLP network for classification. The output layer is also the same as in CNN.

E. LSTM-CNN

In LSTM-CNN architecture, we have combined the above two models. Firstly we have embedded our input sentence by using our trained embedding layer. Then we have fed the input

into LSTM layer for analyzing the text order. The LSTM unit performs a great role in analyzing the syntax structure of a sentence. Then the output of the LSTM has been fed into the five layer convolution network for identifying positive and negative catchphrases in other words features in the sentence. The final output layer is the Sigmoid layer which is same as CNN and LSTM.

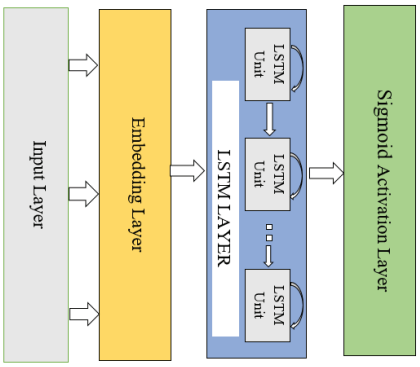

Fig. 2. Higher level representation of LSTM Network.

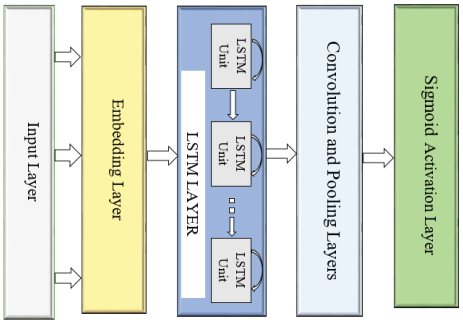

Fig. 3. Higher level Representation of LSTM-CNN network.

III. Experimental Analysis

A. Experimental Setup

We have implemented all the three architecture in Google Colab which provides faster computing environment. For the CNN architecture, we have trained the model for 8 epochs with a batch size of 128 as after that there was no more decrease in loss during the training phase of the architecture. For the same reason, we have trained LSTM network for 5 epochs with a batch size of 128. For the LSTM-CNN network, we have trained the network for 6 epoch with the same batch size as LSTM and CNN. We have used Adam optimizer [12] in order to minimize the loss function which was computed using Binary Cross-Entropy [13]. We have used Dropout [14] technique to avoid overfitting in the network. The training loss curves of the models are shown in figure 4.

B. Results and Performance Analysis

The dataset of IMDb movie reviews deals with binary classification problem because the reviews can be either positive or negative. We have used several accuracy metrics based on the confusion matrix [15] in order to evaluate and compare the performance of three architectures. We have also used the ROC [16] curve in order to evaluate the performance of our classifier. As the problem is a binary classification problem the roc curve gives a much better interpretation of the results.

TABLE I. Accuracy metric of the architectures model using precision recall F-score and accuracy derived from F-score

Evaluation Measure	CNN	LSTM	LSTM-CNN
Accuracy	0.90	0.88	0.89
Recall	0.95	0.82	0.90
Specificity	0.84	0.90	0.87
Precision	0.87	0.90	0.87
F-Score	**0.91**	**0.86**	**0.88**

From the above table, we can observe that CNN has outperformed LSTM and LSTM-CNN. The reason behind that is LSTM performs well in NLP task where the syntactic and semantic structure both is important. In the case of sentiment analysis, finding the positive and negative catchphrases are more important the syntax or semantic structure of the sentence. In fact, exploring the syntax of sentences sometimes results in a degradation in classification performance for sentimental analysis. So that is the main reason CNN has outperformed the other two methods.

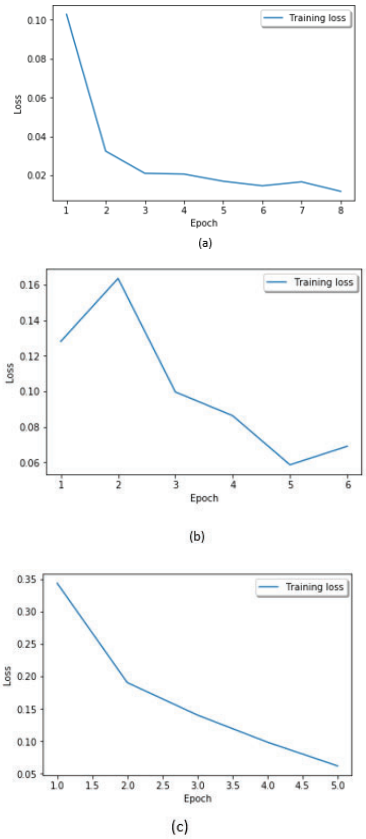

Fig. 4. Graphical Representation of loss during training phase (a) CNN, (b) LSTM, (c) LSTM-CNN.

The ROC curves of the architecture is shown below that also shows CNN has performed better than LSTM and LSTM -CNN

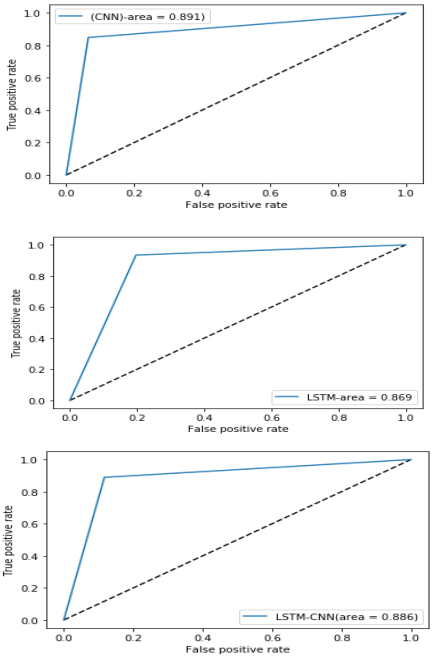

Fig. 5. ROC curves for CNN, LSTM and LSTM-CNN architectures.

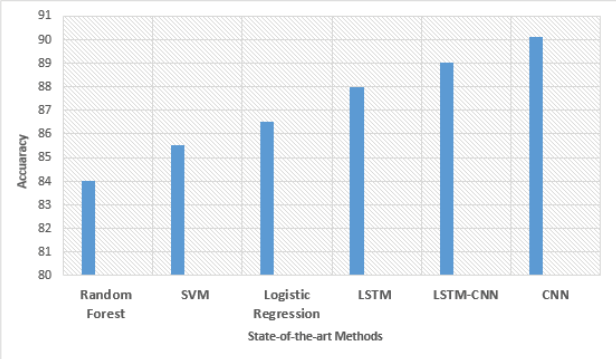

Fig. 6. Comparison with state-of-the-art method for sentiment classification on IMDb movie reviews

It can be observed that, CNN has outperformed all other approaches for sentiment classification for its unique characteristics to find pattern or catchphrase in sentences. LSTM and LSTM-CNN have also performed better than the traditional method [17] used for sentiment classification on IMDb movie review dataset. From the above discussions, we can conclude that CNN is the best-suited architecture for performing sentiment classification on IMDb movie reviews.

IV. Conclusion

Sentiment analysis is becoming very important as the amount of online data increasing at a huge rate. For this reason, we need sentiment analysis on social media or online reviews for predicting and forecasting public opinion. We have found that CNN has performed better than LSTM and LSTM-CNN because of the reason that syntax is not as important as positive or negative in sentiment classification. CNN has performed 2% better than LSTM and 1% better than LSTM-CNN in terms of accuracy. CNN has also outperformed other state-of-the-art method on IMDb dataset. For future work, we have decided to use the convolutional neural network in other fields of natural language processing and evaluate the performance of used methods in those fields.

References

[1] Rana, S., & Singh, A., "Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques." *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2016, pp. 106-111.

[2] Ouyang, X., Zhou, P., Li, C. H., & Liu, L. "Sentiment analysis using convolutional neural network." *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*. IEEE, 2015, pp. 2359-2364.

[3] Kim, Y., "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882*, 2014.

[4] Hochreiter, S., "The vanishing gradient problem during learning recurrent neural nets and problem solutions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02, 1998, pp. 107-116.

[5] Li, D., & Qian, J., "Text sentiment analysis based on long short-term memory." *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*. IEEE, 2016, pp. 471-475.

[6] F.Chollet, "Keras." (2015).

[7] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. , "Learning word vectors for sentiment analysis." *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 2011, pp. 142-150.

[8] Pennington, J., Socher, R., & Manning, C., "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.

[9] Xu, B., Wang, N., Chen, T., & Li, M., "Empirical evaluation of rectified activations in convolutional network." *arXiv preprint arXiv:1505.00853*, 2015.

[10] Hochreiter, S., & Schmidhuber, J., "Long short-term memory." *Neural computation* 9.8, 1997, pp. 1735-1780.

[11] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. , "LSTM: A search space odyssey." *IEEE transactions on neural networks and learning systems* 28.10, 2016, pp. 2222-2232.

[12] Kingma, D. P., & Ba, J. , "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.

[13] Rubinstein, R., "The cross-entropy method for combinatorial and continuous optimization." *Methodology and computing in applied probability* 1.2, 1999, pp. 127-190.

[14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1, 2014, pp. 1929-1958.

[15] Sokolova, M., & Lapalme, G., "A systematic analysis of performance measures for classification tasks." *Information processing & management* 45.4, 2009, pp. 427-437.

[16] Bradley, A. P., "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern recognition* 30.7, 1997, 1145-1159.

[17] H. Pouransari, and S. Ghili, "Deep learning for sentiment analysis of movie reviews." Technical report, Stanford University, 2014.