International Journal of Internet, Broadcasting and Communication Vol.13 No.1 100-106 (2021)
http://dx.doi.org/10.7236/IJIBC.2021.13.1.100

IJIBC 21-1-12

Developing Sentimental Analysis System Based on Various Optimizer

Seong Hoon Eom

Associate Professor, Department of Electrical and Electronic Engineering, Youngsan University, Korea
eomsh@ysu.ac.kr

Abstract

Over the past few decades, natural language processing research has not made much. However, the widespread use of deep learning and neural networks attracted attention for the application of neural networks in natural language processing. Sentiment analysis is one of the challenges of natural language processing. Emotions are things that a person thinks and feels. Therefore, sentiment analysis should be able to analyze the person's attitude, opinions, and inclinations in text or actual text. In the case of emotion analysis, it is a priority to simply classify two emotions: positive and negative. In this paper we propose the deep learning based sentimental analysis system according to various optimizer that is SGD, ADAM and RMSProp. Through experimental result RMSprop optimizer shows the best performance compared to others on IMDB data set. Future work is to find more best hyper parameter for sentimental analysis system.

Keywords: *Sentimental analysis, Natural language processing, Optimizer, Word embedding*

1. Introduction

Over the past few decades, natural language processing research has not made much progress, and human language has been an area that artificial intelligence cannot reach. However, the widespread use of deep learning and neural networks in image classification has also attracted attention for the application of neural networks in natural language processing. In the late 2000s, LSTM (Long Short-Term Memory) [1] neural networks were applied to natural language processing, achieving remarkable results. Sentimental analysis is also one of the natural language processing techniques that determine the positive degree of sentences in the field of natural language processing, which is attracting attention due to the revival of deep learning. With the development of deep learning, natural language processing technology has developed a lot, but it is not easy to analyze the sensitivity based on natural language processing. In this paper we propose a deep learning-based sentimental analysis system especially focus on selecting the best optimizer. Paper is organized as follows. Section 2 briefly explains the RNN (Recurrent Neural Network) model [2]. Section 3 outlines the LSTM model which overcomes the RNN model mentioned in section 2. Section 4 explains the sentimental analysis system. In section 5 represents the experimental results of the proposed system. Finally, the analysis of the experimental results and the future research will be described.

Manuscript Received: December. 2, 2020 / Revised: December. 7, 2020 / Accepted: December. 11, 2020
Corresponding Author: eomsh@ysu.ac.kr
Tel: +82-55-380-9434, Fax: +82-55-380-9249
Associate Professor, Department of Electrical and Electronic Engineering, Youngsan University, Korea

2. Time Series Prediction Model

Typical deep learning models receive a fixed size input vector as input and to produce a fixed form of output vector. However, when dealing with sequence, fixed-size input vectors can be problematic. To handle sequence data, System must read the data in the order in which it appeared and process it one by one. RNN is the deep learning model that makes this possible, and the architecture of the model is illustrated in Fig.1. As shown in Fig.1, RNN is a multi-layer neural network structure that receives data divided over time. In the case of natural language processing, a sentence can be divided into word or token and it entered into a model. In this case, the order of appearance of the word or token represents the order of time.

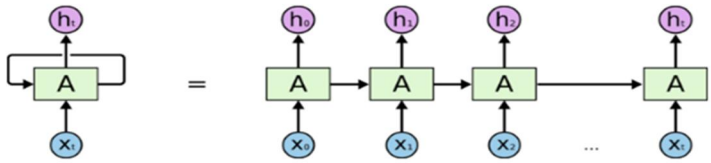

Figure 1. Architecture of RNN. Source : [2]

2.1 RNN structure and problems

The structure of RNN is as follows. The output of the layer is re-entered and the previous data together affects the result. RNN layer simply adds two inputs and apply a tangent function to output value and passes it to the next layer. This is formulated as follows.

$$S_t = \tan h(S_{t-1} + x_i) \tag{1}$$

The reasons for using the tangent function as an activation function are as follows: The first is that it ensures connection weight does not divergence quickly, and the second is that the derivate of tangent function is easily calculated. The output layer uses the sigmoid function as an activation function, and the formula is as follows.

$$O_n = \text{sigmoi d}(S_n) \tag{2}$$

RNN models are suitable for sequence data and are theoretically known to handle long term dependency problems well. However, it is said that is not well applied to natural language Processing. Bengio et al [1] experimented deeply and showed that RNN model can't overcome the long term dependency problem. Another problem of RNN is gradient vanishing. This means that if there are many layers in model, the derivate value of the loss function converges to zero. As a result, making the model difficult to learn, LSTM model has been proposed to solve these problems [2].

2.2 LSTM

As mentioned in section 2, LSTM model is designed to overcome the long-term dependencies. Fig. 2 is a detailed internal structure of LSTM. The terms used in Fig. 2 are as follows. h_t and h_{t-1} are current and previous hidden cell state, C_t and C_{t-1} are current and previous cell state, x_t is input at time t, σ and tanh are activation function and finally + and X are addition and multiply operation respectively.

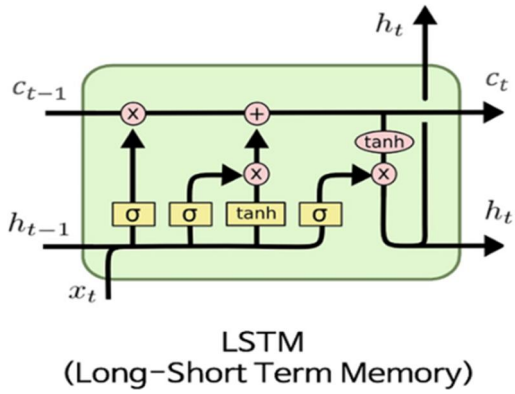

Figure 2. LSTM artitecture. Source : [4]

The internal structure of the LSTM is designed to selectively remember important inputs and discard the rest. LSTM model has additional cell state compared to RNN model. Cell state indicates current memory in LSTM. The role of this cell is to pass important information to be remembered at a certain point in time to the next state. The hidden state, on the other hand, is the total memory that is remembered during the training. It stores all information whether the information is important or not. The core part of the LSTM model is how to control the information flow in hidden state and cell state. The information flow in the hidden state and cell state is controlled by forget gate, input gate and output gate. Detailed operation in each gate is explained in [3].

3. Proposed model

In this research we propose a sentimental analysis system. We want to create a system that classifies movie review text as positive or negative. Overall model architecture is as follows.

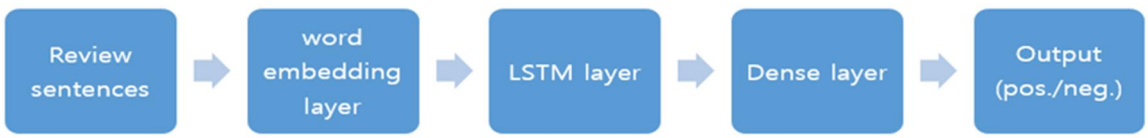

Figure 3. Proposed system architecture.

The details of each layer are as follows.

3.1 Training and Test data

We use input data as IMDB movie review dataset which is consisting of English sentences. IMDB dataset [5] contains 50,000 movie review text collected from the Internet Movie Database. IMDB dataset is composed of 50,000 learning data, 25,000 reviews for training and 25,000 for testing. This means that the number of positive and negative reviews is configured equally. The words or tokens in movie review dataset are already encored in integer array. Label has a value of integer 0 or 1. 0 is a negative review and 1 is a positive one. The last thing to consider is that each sentence in a movie review has a different length, so the size of the input vector is different. It is necessary to solve this problem because neural network can only

receive inputs of a fixed size. In this research we use zero-padding technique [6] to solve this problem.

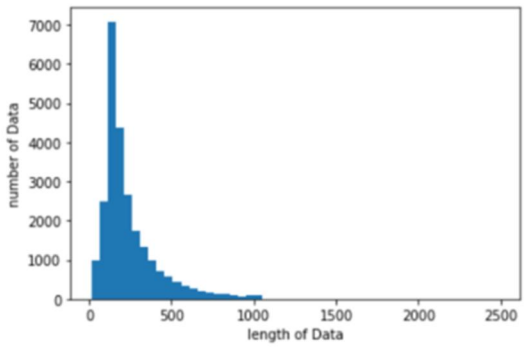

Figure 4. Distribution of length in IMDB.

Fig. 4 shows the distribution of length in IMDB datasets. In all, it has a length of 500 or less, in particular it can be confirmed that most of data with 100 to 300 lengths. On the other hand, the data with the longest length can also be found over 1,000 in length. It is reasonable to select the maximum length of words between from 100 to 300. In our research we set the maximum length of each sentence to 100.

3.2 Word embedding layer

Word embedding [7] is a technique that expresses words as vector in space in a different way than one-hot encoding. One-hot encoding [8] causes a catastrophic problem that increases the dimension as the set of words increases. Using one-hot encoding technique, only certain components are expressed as 1 and the rest are expressed as 0 and this kind of expression is called spares representation. Expressing words using one-hot encoding method is not realistic, and there is a disadvantage that does not contain the meaning of the word. Word embedding technique enables words with similar meanings appear close to each other, so it can include the meaning of words on vectors. In our proposed model, the embedding layer maps to 128 dimensions for input sentences.

3.3 LSTM layer

LSTM layer learns to classify the input review from the output of embedding layer whether it is positive or negative. An important factor to consider when designing LSTM layer is the selection of activation functions. LSTM layer has three gates : forget, input, and output gate and we can apply different activation functions. However, Farzad et al [9] have shown that applying activation functions differently does not have a significant impact on results. In our model we apply sigmoid function.

3.4 Dense layer

The role of dense layer is to convert the output from the embedding layer to a probability between 0 and 1. We also use the sigmoid function as activation function.

3.5 Optimizer

It is not known which optimizer is appropriate for LSTM model [10]. In general LSTM model may not be properly trained by certain optimizers due to gradient vanishing problems and gradient exploding problem [11]. Therefore, it is often difficult to know in advance which optimizer is suitable for data. The best approach is to apply various optimizers and to compare each performance. In this research, we compare SGD [12], RMSprop [13] and ADAM [14] method for selecting best optimizer for proposed system.

4. Experiment

The figure below shows the experimental result of various optimization techniques applying to the IMDB database.

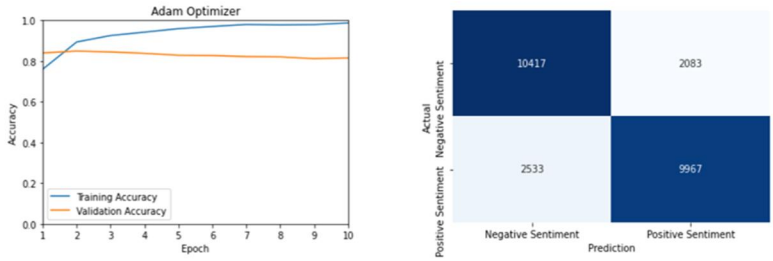

Figure 5. Training and test accuracy and confusion matrix using Adam optimizer.

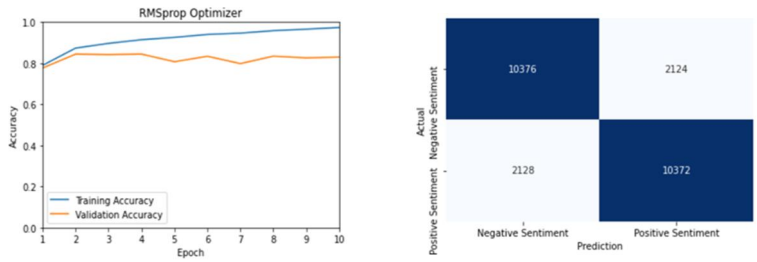

Figure 6. Training and test accuracy and confusion matrix using RMSprop optimizer.

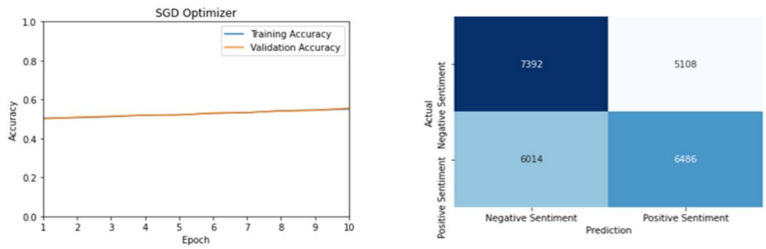

Figure 7. Training and test accuracy and confusion matrix using SGD optimizer.

Training accuracy of RMSprop is 0.9772. In the meantime, the accuracy of Adam and SGD was 0.9883 and 0.5516 respectively. While testing accuracy of RMSprop, Adam and SGD is 0.8299, 0.8154 and 0.5551 respectively. According to above experimental results, RMSprop optimizer shows the best performance in training and test data compared to SGD and Adam. Another measurement of the algorithm's performance evaluation in machine learning is confusion matrix. Confusion matrix, also known as an error matrix [15], is

a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class or vice versa.

As you can see in figure 6, the confusion matrix of using RMSprop optimizer showed the 82% accuracy with 10376 sentences classified as negative for negative sentences with 10372 positive sentences out of 25,000 data. In case of Adam optimizer 9967 sentences classified as negative for negative sentences with 10417 positive sentences out of 25,000 data. The worst performance SGD optimizer only 55% were correctly classified.

5. Conclusion and Future works

In this research we propose the sentimental analysis system based on various optimizer. Training accuracy of RMSprop is 0.9772. In the meantime, the accuracy of Adam and SGD was 0.9883 and 0.5516 respectively. While testing accuracy of RMSprop, Adam and SGD is 0.8299, 0.8154 and 0.5551 respectively. In IDMB dataset RMSprop optimizer shows the best performance in training and testing phase. Future work is to find more best hyper parameter for sentimental analysis system and we are considering the combination of convolution layer between input and LSTM layer for extracting more useful meaning. Finally we will apply the system to HCNC company customer bulletin board. The company run a customer bulletin board for the reputation for the products what they sold. The product review from customers is an important part of the company because it is important information. Proposed sentimental analysis system based on the bulletin board we will evaluate and develop our proposed model.

Acknowledgement

This work was supported by Youngsan University Research Fund of 2020.

References

[1] Olah, Chris, and Shan Carter. "Attention and augmented recurrent neural networks." *Distill* 1.9: e1, 2016.
[2] http://www.comp.hkbu.edu.hk/~markus/teaching/comp7650/tnn-94-gradient.pdf
[3] Hochreiter, S. and J. Schmidhuber. “Long Short-Term Memory,” *Neural Computation* 9: 1735-1780, 1997.
[4] LSTM Figure (source: https://upload.wikimedia.org/wikipedia/commons/9/98/LSTM.png)
[5] Andrew L. Maas, Raymond E. Daly, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL),* 2011.
[6] Hashemi, M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. *J Big Data* **6,** 98, 2019.
DOI: https://doi.org/10.1186/s40537-019-0263-7
[7] Kusner, Matt, et al. "From word embeddings to document distances," *International conference on machine learning*. 2015.
[8] one-hot encoding: RODRÍGUEZ, Pau, et al. Beyond one-hot encoding: Lower dimensional target embeddi ng. Image and Vision Computing, 75: 21-31, 2018.
[9] Farzad, Amir, Hoda Mashayekhi, and Hamid Hassanpour. "A comparative performance analysis of differen t activation functions in LSTM networks for classification," *Neural Computing and Applications* 31.7: 2507 -2521, 2019.
[10] Jiang, Siyu, and Yimin Chen. "Hand gesture recognition by using 3DCNN and LSTM with adam optimiz er," *Pacific Rim Conference on Multimedia*. Springer, Cham, 2017.
[11] ARPIT, Devansh, et al. h-detach: Modifying the lstm gradient towards better optimization. *arXiv preprint*

arXiv:1810.03023, 2018.

[12] Bottou, Léon. "Stochastic gradient descent tricks." *Neural networks: Tricks of the trade*. Springer, Berlin, Heidelberg, 421-436, 2012.

[13] Kurbiel, Thomas, and Shahrzad Khaleghian. "Training of deep neural networks based on distance measure s using RMSProp," *arXiv preprint arXiv:1708.01911,* 2017.

[14] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization," *arXiv preprint arXiv: 1412.6980* , 2014.

[15] Stehman, Stephen V. "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*. **62** (1): 77–89. Bibcode:1997RSEnv..62...77S, 1997.
DOI: https://doi.org/10.1016/S0034-4257(97)00083-7