Research on Text Classification Based on CNN and LSTM

Yuandong Luan
Faculty of Information Technology
Beijing University of Technology
Beijing, China
1183559690@qq.com

Shaofu Lin
Faculty of Information Technology
Beijing University of Technology
Beijing, China
linshaofu@bjut.edu.cn

Abstract—**With the rapid development of deep learning technology, CNN and LSTM have become two of the most popular neural networks. This paper combines CNN and LSTM or its variant and makes a slight change. It proposes a text classification model named NA-CNN-LSTM or NA-CNN-COIF-LSTM, which has no activation function in CNN. The experimental results on the subjective and objective text categorization dataset [1] show that the proposed model has better performance than the standard CNN or LSTM.**

Keywords—CNN, LSTM, text classification, deep learning

I. Introduction

Text categorization has always been a basic and popular research topic in the field of natural language processing. It has a wide range of applications in our life, such as film review classification, subjective and objective sentence classification, and text categorization technology has great significance in assisting people to make decisions in real life. Traditional text categorization methods mainly include dictionary-based method and machine learning method. Since the emergence of deep learning algorithm, the accuracy of text categorization has been greatly improved. In the task of text categorization using deep learning algorithm, convolutional neural networks(CNN) and long short term memory networks(LSTM) are widely used. Convolutional neural network is a kind of multi-layer neural network, which is an improvement of error back propagation network. It is good at dealing with related machine learning problems of images, especially large images. CNN was first proposed by Yann Lecun and applied to handwritten character recognition [2]. Recurrent neural network(RNN) is a kind of neural network structure which contains a loop. It has the ability to preserve information. Information is transmitted from layer to layer through the recurrent network module. The output of hidden layer depends on the information of past time at every moment. Chain attributes of recurrent neural networks indicate that the model is closely related to the problem of sequence annotation, and it has been widely used in natural language processing tasks such as text classification and machine translation. However, because the current output results of the recurrent neural network are related to a long input sequence, the gradient explosion or disappearance caused by the long-term dependence and the long sequence of the recurrent neural network is a problem. In order to avoid the long-term dependence of recurrent neural networks, researchers proposed long short term memory neural networks [3]. In this paper, we use CNN without activation function and LSTM with one of its variants coupled of input and forget gate, and test it on the subjective and objective text dataset provided by [1]. The validity of the model is proved by comparative experiments.

II. Related Work

Deep learning is one of the latest trends in machine learning and artificial intelligence research, and many significant breakthroughs have been made in this field all over the world [4]. Similarly, in the field of natural language processing, the use of deep learning has produced an effect that easily surpasses traditional methods. Ref. [5] briefly introduced the architecture and methods of deep learning and its application in natural language processing, and then discussed the current technical status in detail, and put forward some suggestions for future research in this field. Convolutional neural network is widely used in image field. [Kim, 2014] firstly proposed the application of convolutional neural network in text categorization task. He used a simple single-layer convolution neural network in many classification datasets while achieving excellent classification results, and made a detailed parameter adjustment [6]. This also inspires us to use deep learning methods in some tasks without the need for complex network structures. Ref. [7] introduced three text categorization methods based on multi-task learning of recurrent neural networks. These methods can improve the performance of the task with the help of other related tasks. Ref. [8] introduced the text classification method of standard CNN combined with standard LSTM. It used CNN to extract features of high-level phrase sequences and send them to LSTM to obtain sentence representation. This method has the ability to capture both local phrase features and global sentence representation. Ref. [9] summarized the differences between standard LSTM and its eight variants. Tests on three representative tasks showed that none of the variants can significantly improve the performance, and the forget gate and the output activation function is the key to the model. Ref. [10] summed up the effects of different parameter settings on the performance of CNN model in practical application. It also provided specific suggestions for practitioners.

III. Text Classification Model Based on CNN and LSTM

The text categorization model based on CNN and LSTM or its variant can be divided into four layers: input layer, convolutional network layer, LSTM or its variants layer and softmax classifier layer. The model structure is shown in Figure 1.

978-1-7281-1223-7/19/$31.00 ©2019 IEEE

Dalian, China
March 29-31, 2019

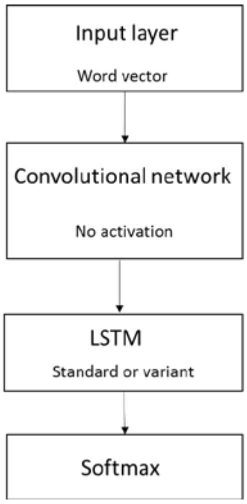

Fig.1. Model structure

A. Input Layer

This paper uses the subjective and objective text data in [7]. After reading into the data, the text is preprocessed first, and the required pure data can be obtained by removing redundant spaces, special characters except numbers or letters, and special expressions in English, such as tenses. Then, we use the learn module of TensorFlow library to generate word vectors, and get the word vectors of the text and input them into the convolutional network layer as features.

B. Convolutional Network Layer

As we all know, the convolutional neural network is a non-linear activation function applied to the results of convolutional operation, and then a full connection layer is used after the pooling operation to classification. The core of convolutional operation is called filter, also known as kernel function. It completes feature extraction by sliding from top to bottom and from left to right in the original matrix. In natural language processing, the width of the kernel function is generally equal to the width of the original matrix, and the kernel function only slides in the upper and lower directions, which guarantees the integrity of the word as the smallest granularity in the language [11]. In the sliding process of the kernel function, there are two kinds of padding strategies, zero-padding and valid-padding, decided by whether adding zero to the original matrix. Here we adopt valid-padding. On the other hand, the results of our convolutional layer are sent to the LSTM layer. LSTM needs the input of sequential relationship, and the pooling operation will destroy this relationship, so we remove the pooling operation. What's more, unlike the typical CNN, we need to apply the activation function to the convoluted results, instead we omit the activation function here.

C. LSTM Or Its Variant Layer

Long short term memory network (LSTM) is a special type of recurrent neural network (RNN), which has the ability to learn long-term dependence. Its structure is shown in Figure 2.

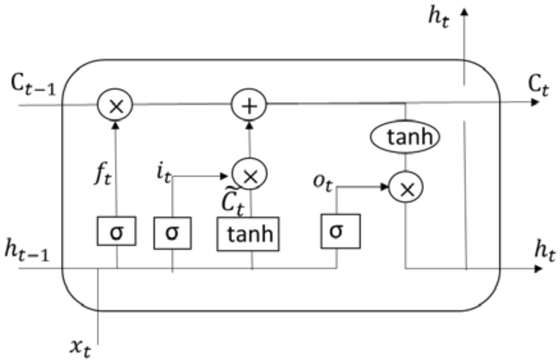

Fig.2. LSTM: standard

The core of LSTM is cell state, which can add or delete information to cells, and selectively let information flow through the door mechanism to achieve this purpose. LSTM consists of three gates: forget gate, input gate and output gate. First, the forget gate decides which information to delete from the cell state, and then the input gate decides what information to update to the cell state. After determining these two points, the cell state can be updated. Finally, the output gate decides the final output of the network. The state of each node in this process is determined by Equation (1)-(6).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)\#(1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)\#(2)$$

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t]) + b_C\#(3)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}\#(4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)\#(5)$$

$$h_t = o_t * \tanh(C_t)\,\#(6)$$

Among them, h_{t-1} denotes the hidden state of the previous layer, x_t denotes the current input, W and b denote the weight and bias, σ denotes the sigmoid function, f_t denotes the output of the forget gate, i_t denotes the output of the input gate, $\widetilde{C_t}$ denotes the intermediate temporary state, C_{t-1} denotes the cell state of the previous layer, C_t denotes the cell state of the next layer, o_t denotes the output of the output gate, and h_t denotes the hidden state of the next layer.

This paper compares the standard LSTM with one of its variants, called COIF-LSTM, which is coupled of input and forget gates. Its structure is shown in Figure 3.

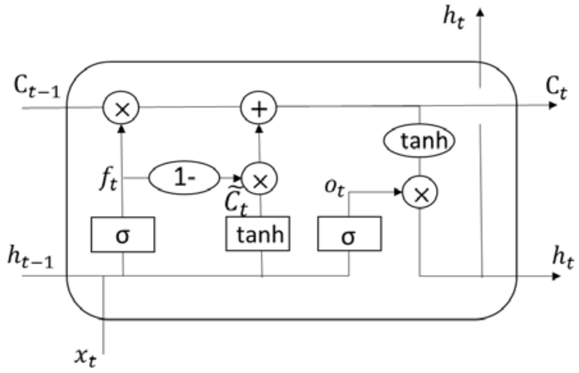

Fig.3. LSTM: coupled of input and forget gates

The only difference between them is that the calculation method of input gates is different. Instead of calculating the output of forget gate and input gate separately, in COIF-LSTM, the output of the input gates is determined by 1-f_t, so the updating mode of cell state of the next layer can be shown in Equation (7).

$$C_t = f_t * C_{t-1} + (1 - f_t) * \widetilde{C_t}\#(7)$$

D. Softmax Classifier Layer

After extracting high-level features from text by CNN without activation function combined with LSTM or its variants, the features are sent to the softmax classifier in a fully connected way for classification. Softmax is a special kind of function. It can map the output of neurons to (0,1) interval and select the class with the largest probability value as the result of prediction. The calculation of the softmax value is shown in Equation (8).

$$P_i = \frac{e^i}{\sum j\, e^j}\#(8)$$

Where P_i denotes the probability of the ith category, e^i denotes the corresponding value of the output of the ith category, and j denotes the total number of categories.

IV. Experiment

A. Data Set

The experimental data in this paper are derived from the subjective and objective text data used in [1]. The data set version number is subjectivity dataset v1.0, which includes 5000 subjective text and 5000 objective text data.

B. Experimental settings

In this experiment, for the convolutional network layer, we use the word embedding dimension of 256, the size of the filter is 3,4,5, the number of filters is 128, the sliding step is 1, and the valid padding method is used. For LSTM layer, we use two-tier stacked LSTM and set the number of hidden units to 128.

C. Evaluating indicator

In order to evaluate the performance of our model, we use the precision, recall and f1-score as the evaluation criteria of this experiment. To illustrate the meanings of these indicators, confusion matrix is introduced first [12], shown in TableI.

TABLE I. Confusion Matrix

	Negative Pre	Positive Pre
Negative Act	TN	FP
Positive Act	FN	TP

TN(True Negative) represents the number of true negative classes, that is, the number of samples predicted as objective text and actually as objective text.

FN(False Negative) represents the number of false negative classes, that is, the number of samples predicted as objective text and actually as subjective text.

FP(False Positive) represents the number of false positive classes, that is, the number of samples predicted as subjective text and actually as objective text.

TP(True Positive) represents the number of true positive classes, that is, the number of samples predicted as subjective text and actually as subjective text.

Precision, which represents the percentage of the amount of relevant information retrieved to the total amount of information retrieved, is an index to measure the signal-to-noise ratio of a retrieval system. It can be expressed as Equation (9).

$$Precision = \frac{\mathrm{TP}}{TP + FP} \times 100\%\ \#(9)$$

Recall, which represents the percentage of the amount of relevant information retrieved to the total amount of relevant information in the system, is an index to measure the success of a retrieval system in detecting relevant documents from a collection of documents. It can be expressed as Equation (10).

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\%\#(10)$$

F1 score, which takes both recall and precision into account, is the harmonic mean of precision and recall. The higher recall and precision are, the higher F1 score will be gotten. It can be expressed as Equation (11).

$$F1 = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{\mathrm{FN} + \mathrm{FP}}{2}} \times 100\%\#(11)$$

D. Experimental comparison

In this paper, a single CNN and LSTM model is used as the contrast model, and produce four kinds of subjective and objective text data classification models by combining CNN and LSTM and their variants. They are standard CNN combined with standard LSTM called CNN-LSTM model, non-activation function CNN combined with standard LSTM called NA-CNN-LSTM model, standard CNN combined with variant LSTM called CNN-COIF-LSTM model, and non-activation function CNN combined with variant LSTM called NA-CNN-COIF-LSTM model.

V. Results Analysis

In this paper, the above model is used to experiment on a given data set. The final experimental results are shown in Table II. *comparison result*

Model	Precision	Recall	F1 score
CNN	98.9353%	98.5197%	98.7270%

LSTM	98.9816%	99.1598%	99.0706%
CNN-LSTM	99.4769%	98.9197%	99.1975%
NA-CNN-LSTM	99.2201%	99.2598%	99.2400%
CNN-COIF-LSTM	98.9816%	99.1598%	99.0706%
NA-CNN-COIF-LSTM	99.1415	99.3398%	99.2406%

From the above results, we can draw some interesting findings.

In terms of precision, CNN-LSTM performs best, followed by NA-CNN-LSTM. As far as recall is concerned, NA-CNN-COIF-LSTM performs best, followed by NA-CNN-LSTM. Generally speaking, NA-CNN-LSTM and NA-CNN-COIF-LSTM are the best performers in terms of F1 score. This is in line with our expectation that CNN without activation function combined with LSTM or its variant will have better performance, which proves the validity of the model in this paper.

The model performance of CNN combined with LSTM variants is not necessarily improved, such as the performance of CNN-COIF-LSTM is the same as that of LSTM. However, the performance of CNN without activation combined with LSTM variants is obviously improved, which proves the validity of this model again.

VI. Conclusion

Unlike the typical CNN, which contains convolution operation and activation function, this paper constructs two text classification models called NA-CNN-LSTM and NA-CNN-COIF-LSTM by combining CNN without activation function and LSTM, and one of its variants COIF-LSTM. Through comparative experiments, it is proved that the combination of CNN without activation function and LSTM or its variant has better performance. Ref. [9] proposed eight variant models of LSTM. The next step of this paper is to explore the performance of the combination of CNN and other variants of LSTM.

References

[1] Bo Pang, Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of the ACL, 2004.

[2] Lecun, Y. L. , et al. "Gradient-Based Learning Applied to Document Recognition." Proceedings of the IEEE 86.11(1998):2278-2324.

[3] Hochreiter, Sepp , and Schmidhuber, Jürgen. "Long Short-Term Memory." Neural Computation 9.8(1997):1735-1780.

[4] Minar, Matiur Rahman , and J. Naher . "Recent Advances in Deep Learning: An Overview." (2018).

[5] Otter, Daniel W. , J. R. Medina , and J. K. Kalita . "A Survey of the Usages of Deep Learning in Natural Language Processing." (2018).

[6] Kim, Yoon . "Convolutional Neural Networks for Sentence Classification." Eprint Arxiv (2014).

[7] Liu, Pengfei , X. Qiu , and X. Huang . "Recurrent Neural Network for Text Classification with Multi-Task Learning." (2016).

[8] Zhou, Chunting , et al. "A C-LSTM Neural Network for Text Classification." Computer Science (2015).

[9] Greff, Klaus , et al. "LSTM: A Search Space Odyssey." IEEE Transactions on Neural Networks & Learning Systems 28.10(2015):2222-2232.

[10] Ye Zhang, Byron C. Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.(2016).

[11] Yan Fang. An Analysis of the Internal Structure of Words.(2014).

[12] Ting, Kai Ming . "Confusion Matrix." (2011).