

## Investigation on the Chinese Text Sentiment Analysis Based on Convolutional Neural Networks in Deep Learning

Feng Xu<sup>1</sup>, Xuefen Zhang<sup>2,\*</sup>, Zhanhong Xin<sup>1</sup> and Alan Yang<sup>3</sup>

**Abstract:** Nowadays, the amount of web data is increasing at a rapid speed, which presents a serious challenge to the web monitoring. Text sentiment analysis, an important research topic in the area of natural language processing, is a crucial task in the web monitoring area. The accuracy of traditional text sentiment analysis methods might be degraded in dealing with mass data. Deep learning is a hot research topic of the artificial intelligence in the recent years. By now, several research groups have studied the sentiment analysis of English texts using deep learning methods. In contrary, relatively few works have so far considered the Chinese text sentiment analysis toward this direction. In this paper, a method for analyzing the Chinese text sentiment is proposed based on the convolutional neural network (CNN) in deep learning in order to improve the analysis accuracy. The feature values of the CNN after the training process are nonuniformly distributed. In order to overcome this problem, a method for normalizing the feature values is proposed. Moreover, the dimensions of the text features are optimized through simulations. Finally, a method for updating the learning rate in the training process of the CNN is presented in order to achieve better performances. Experiment results on the typical datasets indicate that the accuracy of the proposed method can be improved compared with that of the traditional supervised machine learning methods, e.g., the support vector machine method.

**Keywords:** Convolutional neural network (CNN), deep learning, learning rate, normalization, sentiment analysis.

### 1 Introduction

With the rapid development of information technologies, the internet has become one of the most important methods for information acquisition. According to the latest report published by the China Internet Network Information Center (CNNIC) [CNNIC(2018)], the cybercitizen number in China was about 802 million, and the internet popularity rate reached 57.7% in June 2018. As a public platform, the internet has become an

---

<sup>1</sup> School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing, 100876, China.

<sup>2</sup> Smart City College, Beijing Union University, Beijing, 100101, China.

<sup>3</sup> Amphenol Assemble Tech, Houston, TX 77070, USA.

\* Corresponding Author: Xuefen Zhang. Email: zhangxuefen@bnu.edu.cn.

indispensable information acquisition method and an important view exchange channel for people. This presents a serious challenge to the web monitoring.

Text sentiment analysis is a crucial task in the web monitoring area. The task of the text sentiment analysis is to decide the sentiment and opinion of the text through specific methods and classify the text according to the sentiment [Liu (2010)], which can help the government understand the public opinion. Text sentiment analysis has received a lot of research interest and become one of the most important topics in the area of natural language processing. Generally speaking, the text sentiment analysis methods in the literature can be divided two categories [Liu (2010)]: the sentiment dictionary methods and the feature classification methods. The first kinds of methods use the sentiment dictionary in various areas to analyze the web text and obtain the sentiment category of the text. Generally speaking, these methods have relatively high precision, but depend on the sentiment dictionary. Hence, it is necessary to establish the sentiment dictionaries for various areas. The second kinds of methods extract the features that can represent the text and then classify the text using the machine learning techniques. In these methods, the texts are converted into digital information that can be processed by computer, which can achieve better performances compared with the sentiment dictionary methods. With the continuous developments of the artificial intelligence techniques, the text sentiment analysis based on the feature classification has received more and more attention.

Although the text sentiment analysis using the traditional machine learning models, e.g., the support vector machine (SVM) model, has reasonably good results, the performances of these methods are degraded when dealing with mass data. The main reason is that the performances of these methods are dependent on the feature extraction. However, the mass web data and the emerging new words make it difficult for the traditional machine learning methods to extract and train the text feature. As a consequence, how to perform text sentiment analysis in a fast and accurate manner is one of the important research directions. Recently, deep learning has received a lot of research interest and successfully applied in several areas [LeCun, Bengio and Hinton (2015)]. By now, several research groups have studied the sentiment analysis of English texts using the deep learning methods [Joshi and Tekchandani (2016); Aires, Padilha, Quevedo et al. (2018); Lee, Kim and Cheong (2018)]. However, relatively few works have so far considered the Chinese text sentiment analysis. In this paper, a method for analyzing the Chinese text sentiment based on the convolutional neural network in deep learning is provided.

The convolutional neural network (CNN) [LeCun, Bottou, Bengio et al. (1998); Hinton, Osindero and Teh (2006); Hinton and Salakhutdinov (2006)] is one of the hot research topics of the artificial intelligence in the recent years. CNN is a type of deep learning model for processing data, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features from various levels. As described later, CNN is typically composed of three types of layers: convolution layer, pooling layer, and fully connected layer. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a full connection layer, is to map the extracted features into final output, which is used to perform the object tasks, such as classification. In general, CNN contains several convolution and pooling layers, which are connected alternatively. As one layer feeds its output into the next layer,

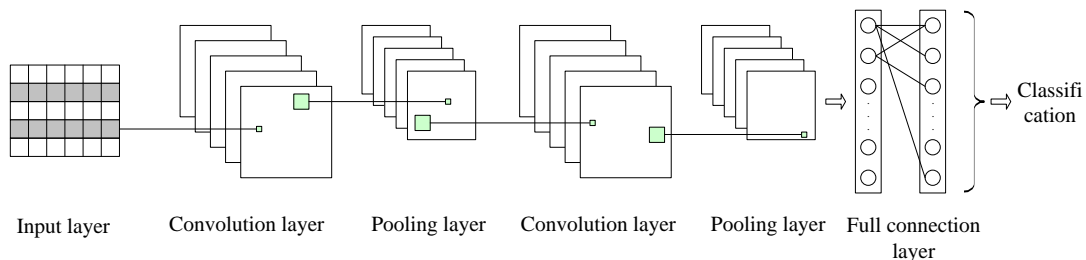
the feature extraction can be hierarchical, which reduces to a better description of the hidden features of the input data. The step where input data are transformed into output through these layers is called forward propagation. The CNN training is to optimize the parameters in the network, which is performed so as to minimize the difference between outputs and ground truth labels. The process is called back propagation, in which optimization algorithms, e.g., gradient descent, are applied. By now, the CNN has been applied successfully in the image and video classification areas [Dahl, Yu, Deng et al. (2012); Abdel-Hamid, Mohamed, Jianget et al. (2014); Sak, Senior and Beaufays (2014); Sak, Senior, Rao et al. (2015); Bi, Qian and Yu (2015); Sercu, Puhersch, Kingsbury et al. (2016); Jin, McCann, Froustey et al. (2017); Ren, He, Girshick et al. (2015); Cui, McIntosh and Sun (2018)].

Due to the wonderful learning ability, the CNN is applied in the Chinese text sentiment analysis in this paper. A regularization method for the feature values of each layer is designed in order to tackle the nonuniform distribution of the feature values due to the text lengths. Moreover, the performances caused by the text features dimensions are evaluated and the optimal dimension is found through simulations. In addition, a method for updating the learning rate of the training process of the CNN is proposed in order to improve the performance of the CNN. Experimental results on the typical dataset indicate that the performances of the proposed method are better than the traditional machine learning methods. The rest of this paper is organized as follows. In Section 2, the basic knowledge on the CNN is reviewed. Section 3 describes the details of the proposed method and presents the experimental results and related analysis. Finally, Section 4 concludes the paper.

## 2 Overview of CNN

### 2.1 CNN architecture

In the recent years, a tremendous interest in deep learning has emerged. Among deep learning models, the most established algorithm is CNN, a class of artificial neural networks that has been a dominant method in various areas. Initially it was used for object recognition tasks but now it is being applied in many areas, such as detection, classification, etc. Fig. 1 illustrates the architecture of a typical CNN model [LeCun, Bottou, Bengio et al. (1998); Hinton, Osindero and Teh (2006); Hinton and Salakhutdinov (2006)].



**Figure 1:** Schematic of the basic structure of a convolutional neural network (CNN)

As shown in the figure, a CNN contains the input layer, the convolution layers, the pooling layers, the full connection layer. A typical architecture consists of several convolution layers and a pooling layer, which are connected alternatively, followed by one full connection layer.

### 2.1.1 Convolution layer

A convolution layer is a basic component of the CNN that performs feature extraction, which typically consists of a combination of linear and nonlinear operations, i.e., convolution operation and activation function. The layer has a set of feature maps with neurons arranged in it. Each neuron in a feature map has a receptive field, which is connected to a neighborhood of neurons in the previous layer through a set of trainable weights, also referred to as filter banks or kernels. Inputs are convolved with the learned weights in order to compute a new feature map, and the convolved results are sent through a nonlinear activation function. All neurons within a feature map have weights that are constrained to be equal in order to reduce the complexity. More formally, suppose  $H_i$  is the feature of the  $i$ th layer of a CNN. We have

$$H_i = f(H_{i-1} \otimes W_i + b_i) \quad (1)$$

where  $W_i$  is the weight vector of the convolution kernel in the  $i$ th layer,  $\otimes$  is the convolution operator,  $b_i$  is the offset vector, and  $f(x)$  is the activation function. Several kinds of nonlinear functions can be used as activation functions, which are introduced as follows.

#### (1) Sigmoid function

The expression of Sigmoid function is given by

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

It is evident that the input of Sigmoid function is a real-value number and the output value is in the range of 0 and 1. It is the most widely used function in the area of machine learning. However, it has two major drawbacks when used in CNN. First, it is saturated and the gradient is vanishing, i.e., the derivative values of the function become very small and tend to zero, as the input increases. Secondly, the output of the function is not zero-centered, which may cause gradients to oscillate between positive and negative values.

#### (2) Tanh function

The expression of tanh function is given by

$$\tanh(x) = 2f(2x)-1 \quad (3)$$

where  $f(x)$  is the Sigmoid function given by (2), It is known that the output of the tanh function is zero-centered. However, the saturating gradient problem still exists.

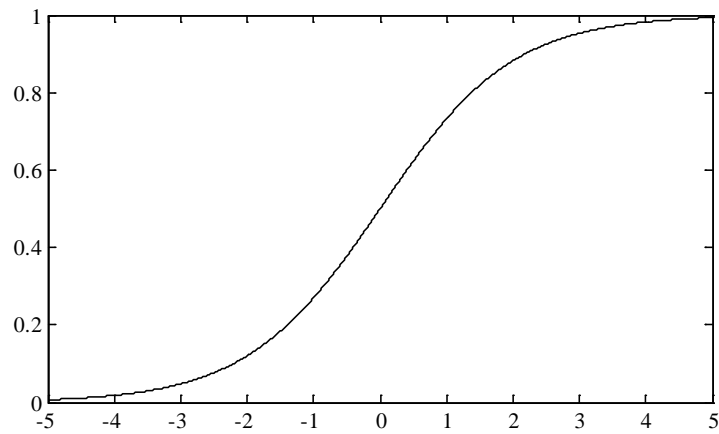
#### (3) ReLU function

ReLU function is the most typical unsaturated nonlinear function, whose expression is given by

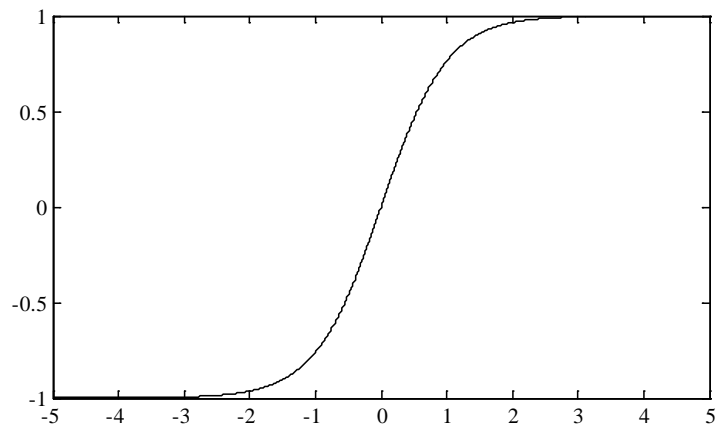
$$f(x) = \max(x, 0) \quad (4)$$

It is known that the function has thresholding at zero. Moreover, the values of the ReLU

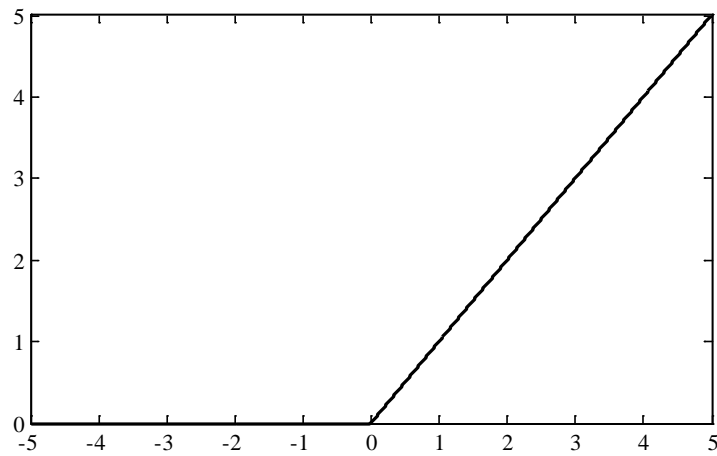
function are zero in the certain region, which make the CNN have the sparse property and the overfitting problem can be addressed to certain extent. The recent studies suggest that the use of unsaturated nonlinear functions may significantly improve the performances of the CNN [Nair and Hinton (2010)] , In our work, the ReLU function is applied. Figs. 3(a)-3(c) illustrate the curves of these three activation functions, respectively.



(a) Sigmoid function



(b) Tanh function



(c) ReLU function

**Figure 2:** Curves of different activation functions in the CNN

### 2.1.2 Pooling layer

The pooling layer is connected after the convolution layer. The purpose of the pooling layer is to perform downsampling to the features according to some rules. As a consequence, the feature dimension is reduced and the scale invariant property of the feature remains hold. Suppose  $H_i$  and  $H_{i+1}$  are the input and output of a pooling layer, respectively. We have

$$H_{i+1} = s(H_i) \quad (5)$$

where  $s(x)$  is the pooling function used. The typical pooling functions include the maximum pooling and the mean pooling [Rawat and Wang (2017)].

#### (1) Maximum pooling

Max pooling is the most popular form of pooling function, which extracts the values within a receptive region of the input feature maps, outputs the maximum value and discards all the other values.

#### (2) Mean pooling

Mean pooling, also known as average pooling, takes the average of all the values in each feature map, whereas the depth of feature maps is retained

### 2.1.3 Full connection layer

After a series of convolution layers and pooling layers, the full connection layer is connected to the CNN. The output information of the convolution layers and pooling layers is further proposed in the full connection layer, whose output is used for the task such as classification. The layer typically has the same number of output nodes as the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU, as previously discussed.

## 2.2 CNN training

CNN training is a process of finding the parameters ( $W_i$  and  $b_i$ ) of each layer) that minimize the error between the real output and the predicted output based on the labels given in the training dataset, which contains two stages: the forward propagating stage and the backward propagating stage [Rawat and Wang (2017)]. In the forward propagation stage, the CNN calculates the cost function according to the error between the real and the predicted outputs. After that, the error is backward propagated and the parameters are updated. Assume the parameters that need to update form the vector  $(\mathbf{W}, \mathbf{b})$ . The cost function can be expressed as

$$E(\mathbf{W}, \mathbf{b}) = L(\mathbf{W}, \mathbf{b}) + \lambda \mathbf{W}^T \mathbf{W} \quad (6)$$

The first term in (6) is the error between the real and the predicted outputs. Two commonly-used functions are the cross entropy function and the mean squared error function. The second term in (6) is the norm of the weight vector  $\mathbf{W}$ . The purpose of the term is to control the overfitting effect. The parameter  $\lambda$  is a proportional factor. The gradient decent method is used to update the parameters  $W_i$  and  $b_i$  in the CNN training process. The method iteratively updates the parameters so as to minimize the objective function. The gradient of the objective function provides us the direction in which the function has the steepest rate of increase, and each parameter is updated in the negative direction of the gradient with a step size determined based on a parameter, called learning rate. The parameter is used to control the error propagation intensity in the backward propagation stage. The calculation details can be expressed as

$$W_i \leftarrow W_i - \eta \frac{\partial E(\mathbf{W}, \mathbf{b})}{\partial W_i} \quad (7)$$

$$b_i \leftarrow b_i - \eta \frac{\partial E(\mathbf{W}, \mathbf{b})}{\partial b_i} \quad (8)$$

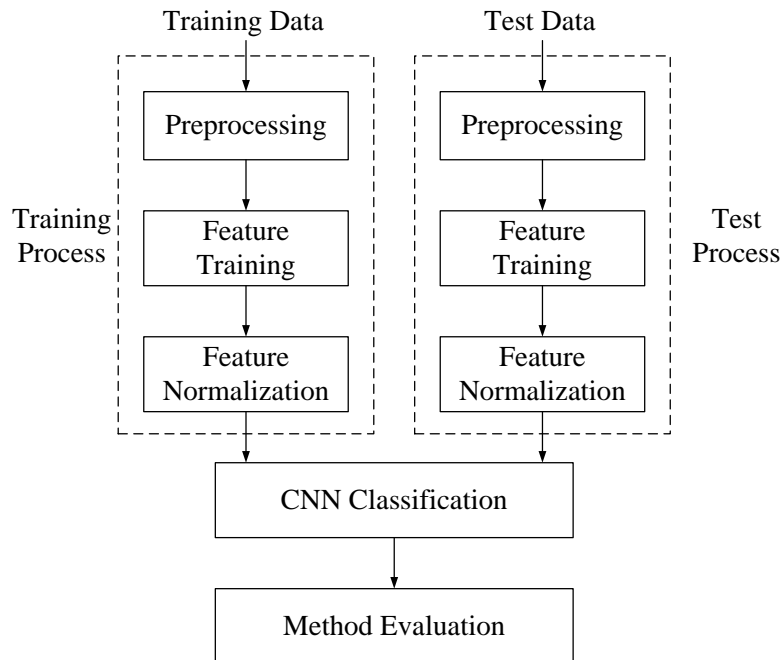
$\eta$  is the learning rate. In the training process, the parameter  $\eta$  can be fixed or changed according to certain rules

## 3 Experiment results and analysis

### 3.1 Dataset and experiment flowchart

In order to evaluate the performance of the proposed method in this paper, we performed experiments on the dataset ChnSentiCorp, which is a typical sentiment analysis dataset collected and published by the research group of Dr. Songbo Tan [ChnSentiCorp (2010)]. There are a total of 10000 comments in the dataset. Among them, 7000 comments are positive, and the remaining 3000 comments are negative. The word number of each comment is from 20 to 2000. The dataset is frequently used for the Chinese text sentiment analysis in the literature [Zhang, Qian and Fanetal (2014); Chen, Huang, Wangetal (2017)]. The numbers of positive comments and negative comments are not equal in the dataset, which may have influence on the performances [He and Garcia (2008)]. In order to tackle the problem, we randomly choose 3000 positive comments to construct our dataset, which contains a total of 6000 comments. Among them, 4800 comments (80%) are used to train the CNN, and the remaining 1200 comments (20%) are used for test. The experiments are performed using Python and Visual C++ programs.

Besides, the final results are obtained by averaging the results of several independent training processes, since the CNN training is stochastic. The experiment flowchart is illustrated in Fig. 3.



**Figure 3:** The experiment flowchart

### 3.2 Preprocessing

First, we perform the word segmentation on each comment using the Jieba module in the Python program. Jieba is a powerful open-source Chinese word segmentation module, in which probabilistic language models are used.

Jieba contains a built-in dictionary. In the segmentation process, the directed acyclic graph (DAG) is constructed for each sentence, and the hidden Markov model and Viterbi algorithm is applied for the words that are not in the built-in dictionary. The maximum probability path in the DAG is output using the dynamic programming after the segmentation process.

### 3.3 Feature training and regularization

After the segmentation, the feature vectors of the comments in the dataset are trained using the word2vec program in the Python. There are two training models in the word2vec program: CBOW and Skipgram [Lian (2015)]. The main difference between the two models is that the former uses the context to predict the current word, whereas the latter uses the current word to predict the context. In our experiment, the Skipgram model is used.



Due to the text lengths, the distribution of the feature values may be nonuniform. In order to tackle the problem, a regularization method is designed. Suppose  $\{F_i^{(n)}\}$  is the set that contains the  $i$ -th components of the features, the regularization is given by

$$F_i^{(n)} \leftarrow \frac{F_i^{(n)} - \mu_i}{\sigma_i^2} \quad (n = 1, 2, 3, \dots) \quad (9)$$

where  $\mu_i$  and  $\sigma_i^2$  are the mean value and the covariance, respectively.

The dimensions of the trained features have certain influence on the performances of the CNN. In our work, we evaluate the performances under different dimensions through simulations. For each chosen dimension, the regularization method given by (9) is used.

### 3.4 Indicators

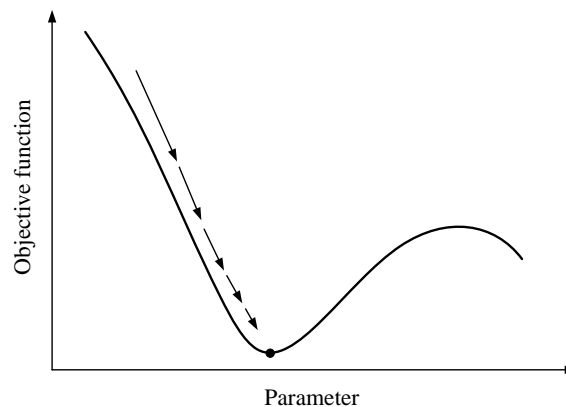
In order to evaluate the performance of the proposed method, three indicators, namely, precision, recall, and  $F$  value are used. The definition of these three indicators can be found in Powers [Powers (2011)]. The parameter  $\alpha$  in  $F$  value is set as 0.5.

### 3.5 Results and analysis

From the previous analysis, we know that the parameter of learning rate plays an important role in the training process of the CNN. The parameter can be fixed or changed according to some rules. In our experiment, the exponential attenuation rule is applied, which is given as

$$\eta^{(t)} = \eta_0 \cdot a^{\lfloor t/T \rfloor} \quad (10)$$

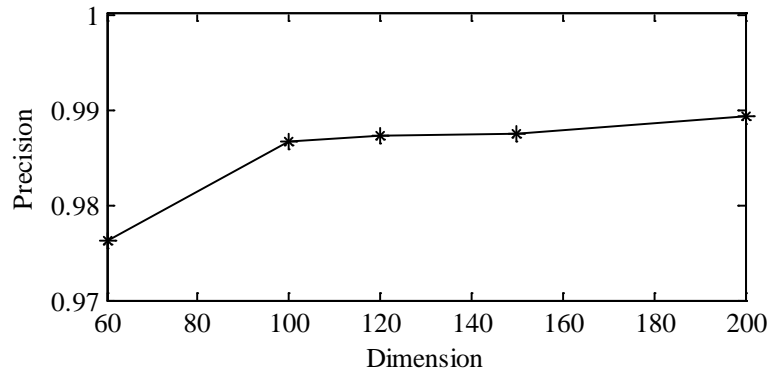
where the notation  $\lfloor x \rfloor$  is the largest integer that are less than or equal to  $x$ ,  $t$  is the iteration number. We know from the above equation that the learning rate becomes  $a$  times of its original value after every  $T$  iteration.



**Figure 4:** The learning rates in the optimization process. The step size is dynamically adjusted from a large step size to a small step size

In general, the parameter  $a$  is set to be a positive value that is less than 1. As discussed in the previous section, the learning rate is used to control the step size in the gradient

descent. It is desirable to adjust the step size dynamically from a large step size to a small step size in the optimization process, as shown in Fig. 4. The main reason is that a large step size can accelerate the convergence speed at the initial of the optimization process, whereas a small step size can avoid the convergence to a local minimum [Boyd and Vandenberghe (2004)]. Now we perform the CNN classification using the above-mentioned dataset. The simulation parameters are selected as follows:  $\eta_0 = 0.001$ ,  $\alpha = \frac{2}{3}$ ,  $T = 500$ , and  $T_{max} = 50000$ .



**Figure 5:** The precision performances of the CNN classification using different feature dimensions

Fig. 5 illustrates the precision performances of the CNN classification using different feature dimensions. It is known from the figure that the precision is significantly improved when the feature dimension is increased from 60 to 100. However, the precision improvement is little if the feature dimension is larger than 100. Hence, the feature dimension is set as 100 in our experiment.

Tab. 1 compares the performances of the proposed method with two other supervised learning methods [Nasrabadi (2007)], i.e., SVM and adaboost. We know from the table that our method outperforms the other two methods. For example, the F measure of the proposed method increases about 4.5% and 1.7% compares with the SVM method and the adaboost method, respectively. The increase has important significances in current web monitoring.

**Table 1:** Performance comparisons of different methods

| Method   | Precision | Recall | F measure |
|----------|-----------|--------|-----------|
| SVM      | 0.9085    | 0.9949 | 0.9498    |
| adaboost | 0.9845    | 0.9677 | 0.9760    |
| Proposed | 0.9867    | 0.9983 | 0.9925    |

Besides, it is found from the experiment that the performances of the designed attenuation rule are very close in various tests, which suggests that the designed rule has good stability.

#### **4 Conclusion and future work**

The text sentiment analysis is an important topic in the web monitoring area. In order to improve the performance of Chinese text sentiment analysis, a method based on CNN is proposed in the paper. A regularization method for the feature values of each layer is designed in order to tackle the nonuniform distribution of the feature values due to the text lengths. Moreover, the feature dimensions are optimized through simulations. The learning rate is an important parameter in the CNN training process. An exponential attenuation rule is proposed for updating the learning rate of the training process of the CNN, whose parameters are fixed according to the experiment. Experimental results on the typical dataset indicate that the performances of the proposed CNN-based method are better than the traditional supervised learning methods, such as SVM and adaboost. In addition, the designed updating rule has good stability, which indicates it is a good choice in practice. As a future work, we will investigate the performances of the CNN-based method using other text features.

#### **References**

- Abdel-Hamid, O.; Mohamed, A. R.; Jiang, H.; Deng, L.; Penn, G. et al.** (2014): Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533-1545.
- Aires, J. P.; Padilha, C.; Quevedo, C.; Meneguzzi, F.** (2018): A deep learning approach to classify aspect-level sentiment using small data sets. *IEEE International Joint Conference on Neural Networks*, pp. 1-8.
- Bi, M.; Qian, Y.; Yu, K.** (2015): Very deep convolutional neural networks for lvcsr. *Sixteenth Annual Conference of the International Speech Communication Association*.
- Boyd, S.; Vandenberghe, L.** (2004): *Convex Optimization*. Cambridge University Press.
- Chen, Z.; Huang, Y.; Wang, Y.; Fu, X.; Fu, K.** (2017): A fine-grained sentiment analysis method using semantic similarity feature. *Computer Applications & Software*, vol. 34, no. 3, pp. 27-30.
- CNNIC** (2018): *The 42nd China Statistical Report on Internet Development*.
- Cui, Q.; McIntosh, S.; Sun, H.** (2018): Identifying materials of photographic images and photorealistic computer generated graphics based on deep cnns. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 1-16.
- Dahl, G. E.; Yu, D.; Deng, L.; Acero, A.** (2012): Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30-42.
- He, H.; Garcia, E. A.** (2008): Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, vol. 21, no. 9, pp. 1263-1284.
- Hinton, G. E.; Osindero, S.; Teh, Y. W.** (2006): A fast learning algorithm for deep

belief nets. *Neural Computation*, vol. 18, no. 7, pp. 1527-1554.

**Hinton, G. E.; Salakhutdinov, R. R.** (2006): Reducing the dimensionality of data with neural networks. *Science*, vol. 313, no. 5786, pp. 504-507.

**Jin, K. H.; McCann, M. T.; Froustey, E.; Unser, M.** (2017): Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509-4522.

**Joshi, R.; Tekchandani, R.** (2016): Comparative analysis of twitter data using supervised classifiers. *International Conference on IEEE Inventive Computation Technologies*, vol. 3, pp. 1-6.

**LeCun, Y.; Bengio, Y.; Hinton, G.** (2015): Deep learning. *Nature*, vol. 521, no. 7553, pp. 436.

**LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.** (1998): Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324.

**Lee, S. H.; Kim, D. M.; Cheong, Y. G.** (2018): Predicting emotion in movie scripts using deep learning. *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 530-532.

**Lian, Z.** (2015): Exploration of the working principle and application of word2vec. *Sci Tech Information Development & Economy*, vol. 2, pp. 145-148.

**Liu, B.** (2010): Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, vol. 2, pp. 627-666.

**Nair, V.; Hinton, G. E.** (2010): Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning*, pp. 807-814.

**Nasrabadi, N. M.** (2007): Pattern recognition and machine learning. *Journal of Electronic Imaging*, vol. 16, no. 4, pp. 049901.

**Powers, D. M.** (2011): Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37-63.

**Rawat, W.; Wang, Z.** (2017): Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, vol. 29, no. 9, pp. 2352-2449.

**Ren, S.; He, K.; Girshick, R.; Sun, J.** (2015): Faster r-cnn: towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, pp. 91-99.

**Sak, H.; Senior, A.; Beaufays, F.** (2014): Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth Annual Conference of the International Speech Communication Association*.

**Sak, H.; Senior, A.; Rao, K.; Irsoy, O.; Graves, A. et al.** (2015): Learning acoustic frame labeling for speech recognition with recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4280-4284.

**Sercu, T.; Puhersch, C.; Kingsbury, B.; LeCun, Y.** (2016): Very deep multilingual convolutional neural networks for LVCSR. *IEEE International Conference on Acoustics,*

*Speech and Signal Processing*, pp. 4955-4959.

**Tan, S.** (2010): Chnsenticorp. <http://www.searchforum.org.ch/tansongbo/corpus-senti.html>.

**Zhang, L.; Qian, G.; Fan, W.; Hua, K.; Zhang, L.** (2014): Sentiment analysis based on light reviews. *Journal of Software*, vol. 25, no. 12, pp. 2790-2807.