An Improved Convolutional Neural Network for Text Classification

Xiaojing Fan, Runa A, Zhili Pei, Mingyang Jiang*

College of Mechanical Engineering, Inner Mongolia University for the Nationalities, Tongliao, China

*Corresponding author: mingyangjiang@imun.edu.cn

Abstract. This paper studies the text classification based on deep learning. Aiming at the problem of over fitting and training time consuming of CNN text classification model, a SDCNN model is constructed based on sparse dropout convolutional neural network. Experimental results show that, compared with CNN, SDCNN further improves the classification performance of the model, and its classification accuracy and precision can reach 98.96% and 85.61%, respectively, indicating that SDCNN has more advantages in text classification problems.

Keywords: Classification, CNN, Dropout

1. Introduction

In recent years, the development of Internet and related technology is getting better and better. With the rapid development of mobile app, the utilization rate is higher and higher, which makes the Internet information more complex. Some online social apps are increasingly deep into people's lives, providing people with a lot of information resources. The Internet has become an inseparable part of people's life. With it, the amount of data that people accept on the network increases in geometric order of magnitude. How to obtain valuable information from these massive data has become the research direction.

Text classification (TC) [1] can obtain valuable information from text data. Its main work is to analyze text features and determine which category text belongs to. With the continuous development of text classification research, the application of text classification has become very extensive, such as text information filtering [2], social public opinion [2], search engine [3], [4].

Deep Learning (DL) is an important branch of machine learning. It was proposed by Hinton in 2006 [5], and it is widely used in classification field [6]. Convolutional neural network (Convolution Neural Network, CNN) is a commonly used deep learning model. It can classify large-scale text data with its own advantages, which shows the potential of CNN [7]. In recent years, there have been many research results of text classification based on CNN [8]. In 2014, Kim proposed a method to complete sentence classification using CNN [9]. He adopted a relatively simple CNN model. The first layer is composed of word vector matrix, followed by convolution layer and Max pooling, and the last layer is fully connected softmax layer. Many researchers also use the experimental results of this model to compare with the results of other CNN text classification models to verify the effectiveness of the proposed method. In the same year, Kalchbrenner et al. proposed a...
dynamic convolutional neural network (DCNN) [[10]], in which the convolution layer uses wide convolution, and the pooling layer uses K-MAX pooling and dynamic K-MAX pooling. Hu B et al. published an article on CNN model suitable for two sentence matching [[11], [12]].

This paper mainly studies the text classification based on CNN. A text classification model based on CNN is constructed, and experiments are carried out on the Reuters-21578 data set. The experimental results show that the model constructed has good classification performance.

2. Method

2.1. CNN Model Structure

CNN is composed of multi-layer perceptrons. It uses its own advantages and deep models to learn multi-level abstract features in text, which can effectively improve the performance of text classification and has great applications. CNN has two characteristics: one is the local receptive field, that is, the nodes in the convolution layer are connected with some nodes in the upper layer, rather than fully connected. Secondly, the weight parameters of convolution layer nodes are shared. The characteristics of local link and weight sharing greatly reduce the number of model training parameters. CNN mainly includes input layer, convolutional layer, pooling layer and fully connected layer. In Figure 1, C is the convolutional layer, S is the pooling layer, and finally the classification result is output.

![Fig 1. CNN model structure](image)

The first step of CNN model training is the forward calculation process. The data is input to the CNN model for convolution and pooling operations, and finally the class label is output. The second is the reverse calculation process, which calculates the error between the output and the actual value of the input, and adjusts the weights according to the principle of minimizing the error. These two steps are repeated until the CNN model parameters converge.

The error function of CNN in text classification is as follows.

\[
E^X = \frac{1}{2m} \sum_{i=1}^{m} [x^{(i)} - \hat{y}^{(i)}]^2
\]

(1)

Where m is the number of samples, \( \hat{y}^{(i)} \) is the output tag, and \( x^{(i)} \) is the actual class tag of the i-th sample.

2.2. Activation Function

Sigmoid, tanh, and ReLU are commonly used activation functions, and we use ReLU in CNN. It is a piecewise function, when the input is less than 0, the output is equal to 0, when the input is greater than 0, the output is the input value. Compared with sigmoid and tanh, the advantage of ReLU is that when the input is greater than 0, the gradient is always 1, which effectively alleviates the problem of gradient disappearance. Moreover, ReLU does not include division and power operation, and only one assignment is needed to get the activation, which is faster. Therefore, we use ReLU in both the convolutional and the fully connected layer.
2.3. Dropout Optimization
Dropout can effectively prevent the model from overfitting. In the training phase, dropout randomly discards some nodes with probability $p$, but retains its weight. In the next training, these weights are restored, and again some nodes are randomly discarded with probability $p$ and this process is repeated, which is equivalent to training a different local network each time. The dependence between nodes is reduced, the model is prevented from overfitting, and training only the local network each time reduces the training time. In the test, all discarded nodes are combined with local networks to improve generalization ability. The calculation of the dropout node is as follows.

$$h = m \cdot f(WC + b) \quad m_j \sim Bernoulli(p)$$  \hfill (2)

Where $C$ is the output of the previous layer, $W$ is the weight, $f$ is the activation function, and $m$ is the binary mask matrix. Each element $m_j$ in $m$ takes a random value of 1 or 0 with Bernoulli distribution probability $p$. After calculating the activation value of each node, multiply the corresponding element in $m$ to get the output $h$. When $m_j$ is 0, the output of the corresponding node is 0, and when $m_j$ is 1, the output of the corresponding node is retained. In order to keep the consistency of the total output in the training and the testing, the node activation in the training should be divided by the probability of $1-p$.

2.4. Sparsity Constraint
In order to improve the classification performance, L1 norm is added to the parameters of convolution layer to increase the sparsity constraint and select effective features. Compared with L2 norm, L1 norm has better sparsity effect. The weight adjustment formula is as follows.

$$(k_{i,j}(t+1))_{\mu} = (k_{i,j}(t))_{\mu} - \alpha [\frac{\partial e}{\partial (k_{i,j}(t))_{\mu}} + \text{sign}((k_{i,j}(t))_{\mu})]$$  \hfill (3)

$$\text{sign}((k_{i,j}(t))_{\mu}) = \begin{cases} 1 & (k_{i,j}(t))_{\mu} > 0 \\ -1 & (k_{i,j}(t))_{\mu} < 0 \\ 0 & (k_{i,j}(t))_{\mu} = 0 \end{cases}$$  \hfill (4)

$$b_j(t+1) = b_j(t) - \alpha (t+1) \sum_{\mu} (\delta_j)_{\mu}$$  \hfill (5)

Where $\alpha$ is the learning rate, $(k_{i,j}(t))_{\mu}$ is the $(\mu)$-th element in the weight $k_{i,j}$, $\text{sign}((k_{i,j}(t))_{\mu})$ is the subgradient of L1 norm to $(k_{i,j}(t))_{\mu}$. As shown in the formula (3), the function sign($\cdot$) changes with $k$, and $q$ is the sparse parameter. In this paper, $q$ is 0.00001, and when $q = 0$, it is the original CNN model.

3. Results
The experiment uses the sparse dropout constrained convolutional neural network (SDCNN) proposed in this paper to complete text classification. During training, dropout is added to the fully connected layer, and L1 norm is added to the learning process of the convolutional layer. Experimental results show that SDCNN effectively solves the problems of overfitting and time-consuming, and improves the accuracy of text classification.

The experimental data set is Reuters-21578, and the SDCNN training process is as follows.

Step 1: The word vector matrix of training set is input into SDCNN, and the label of the text is vector $\hat{x}$.

Step 2: The output of convolution layer is 384, the output of pooling layer has 384 local features, the number of nodes in full connection layer is 128, and the number of nodes in output layer is 10. The activation function is ReLU.
Step 3: Calculate the output of the convolutional layer and the pooling layer.
Step 4: The output of the fully connected layer is calculated by dropout.
Step 5: The output vector $\hat{y}$ calculated by the softmax function.
Step 6: The error is calculated by the formula $E = \frac{1}{2} \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$.
Step 7: The optimized error is calculated by the formula $\hat{E} = E + q \sum_{\mu\nu}|(k_{ij})_{\mu\nu}|$, where $q \sum_{\mu\nu}|(k_{ij})_{\mu\nu}|$ is the L1 norm.
Step 8: Adjust the parameters of the fully connected layer, calculate the derivative of the error $E$ and $\hat{E}$ with respect to the weight $W$ and the bias $b$, and adjust it according to the back propagation algorithm.
Step 9: When the training reaches the maximum number of times, the process ends, otherwise, it returns to Step 3.

In the experiment, the value of L1 is 0.00001, and dropout is 0.6. We have completed the comparison experiments on the train error rate, test error rate and train time of SDCNN and CNN. The experimental results is as follow.

Table 1. The results of different models

| Model   | Training error rate (%) | Testing error rate (%) | Train time(S) |
|---------|-------------------------|------------------------|---------------|
| CNN     | 0                       | 1.46                   | 4218          |
| SDCNN   | 0.59                    | 1.04                   | 2801          |

As shown in Table 1, the training error rate of CNN is 0, which fully fits the training data. But the testing error rate is relatively large, resulting in overfitting. The SDCNN training error rate is 0.59%, and the testing error rate is 1.04%, which reduces the testing error rate and effectively alleviates the overfitting. We also recorded the classification accuracy and precision, and the results are shown in Table 2.

Table 2. Classification test results

| Model | Accuracy (%) | Precision(%) |
|-------|--------------|--------------|
| BP    | 96.92        | 84.01        |
| KNN   | 97.03        | 84.70        |
| SVM   | 97.28        | 84.79        |
| AE    | 97.55        | 84.90        |
| CNN   | 98.54        | 85.37        |
| SDCNN | 98.96        | 85.61        |

As shown in Table 2, comparing the classification accuracy and precision, SDCNN is better than KNN, BP, SVM and AE. Moreover, compared with CNN, SDCNN further improves the classification performance of the model, and its classification accuracy and precision can reach 98.96% and 85.61%, respectively, indicating that SDCNN has more advantages in text classification problems.

4. Conclusions
This paper studies the text classification based on deep learning. Aiming at the problem of over fitting and training time consuming of CNN text classification model, SDCNN is constructed based on sparse dropout convolutional neural network. Experimental results show that, compared with CNN, SDCNN can prevent the model from overfitting while reducing training time, and further improve the classification performance of the model. Although the SDCNN model has a better classification effect on text classification problems than other models, the SDCNN model can also design more convolution kernels of different sizes, add more layers, and continue to improve the performance of the model. We use SDCNN to do experiments on English data sets, and we will do the same classification experiments on Chinese data sets to to verify the method.

Acknowledgements
This work was supported by Research Program of science and technology at Universities of Inner
Mongolia Autonomous Region (NJZY20112), Science and technology projects of Inner Mongolia Autonomous Region (2020GG0190), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (NJYT-19-B18), the Natural Science Foundation of Inner Mongolia Autonomous Region of China (2019MS08036), Industry-University-Research Innovation Fund of Ministry of Education Science and Technology Development Center—“Zhi Rong Xing Jiao” Fund (2018A01027), the science research project of Inner Mongolia University for the Nationalities (NMDYB19060), Inner Mongolia University for Nationalities doctoral research start fund project (BS543).

References
[1] Cunha W, Canuto S, Viegas F, et al. Extended pre-processing pipeline for text classification: On the role of meta-feature representations, sparsification and selective sampling. Information Processing & Management, 57(4):102263. (2020)
[2] Alfaro C, Cano-Montero J, J Gómez, et al. A multi-stage method for content classification and opinion mining on weblog comments. Annals of Operations Research, 236(1):197-213. (2016)
[3] Mathirajan M, Sivakumar A I. A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. International Journal of Advanced Manufacturing Technology, 29(9-10):990-1001. (2006)
[4] Ning Y H, Zhang L, Ju Y R, et al. Using Semantic Correlation of HowNet for Short Text Classification. Applied Mechanics and Materials, 513-517:1931-1934. (2014)
[5] Hinton G E, Salakhutdinov R R. Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786): 504-507.(2006)
[6] Cao F, Chen B. New architecture of deep recursive convolution networks for super-resolution. Knowledge-Based Systems, 178:98-110. (2019)
[7] Fan H, Xia G S, Hu J, et al. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. Remote Sensing, 7(11):14680-14707. (2015)
[8] Jiang M, Liang Y, Feng X, et al. Text classification based on deep belief network and softmax regression. Neural Computing & Applications, 29, 61–70. (2018)
[9] Kim Y. Convolutional Neural Networks for Sentence Classification. Computer Science,1408.5882. (2014)
[10] Kalchbrenner N, Grefenstette E,Blunsom P. A convolutional neural network for modelling sentences. Computer science, 655-665. (2014)
[11] Hu B, Lu Z, Li H,et al. Convolutional neural network architectures for matching natural language sentences. Advances in Neural Information Processing Systems, 2042-2050. (2014)
[12] Grelu P, Akhmediev N . Dissipative solitons for mode-locked lasers. Nature Photonics, 6(2):84-92. (2010)