Research on mail classification algorithm based on improved convolutional neural network

Song dan¹, Lu Kui¹ and Dai Xufan²

¹ School of Computer Science and Engineering, Anhui University of Science & Technology, Huainan, Anhui, 232001, China
² School of Electrical and Information Engineering, Anhui University of Science & Technology, Huainan, Anhui, 232001, China
*Corresponding author’s e-mail: klu@aust.edu.cn

Abstract. Aiming at the problems of high dimension and sparse data in traditional text classification methods, this paper proposes an e-mail classification method based on i-cnn model by combining convolutional neural network (CNN) and inception V1 model. In the convolution and pooling operation, 1 * 1 convolution kernel is added to reduce the thickness of eigenvectors, reduce the parameters and improve the computational performance. Through data validation, i-cnn model for e-mail classification results as high as 92.18%, in the comparative experiment, i-cnn model compared with several machine learning classification models, achieved the highest classification accuracy, in the comparison with or without the emergence structure model, i-cnn model accuracy is higher than CNN model. It shows that the model has a good classification effect, and the integration of inception V1 model can improve the accuracy of text classification.

1. Introduction
With the development of social network media and the popularity of e-mail, more and more advertisements in social media are full of life, which brings a lot of trouble to people. Spam not only occupies the memory space, but also invades the privacy of the recipient, so the rapid classification of e-mail has become a hot topic. E-mail classification is a branch of natural language processing. Many scholars at home and abroad have done in-depth research on it, including traditional machine learning methods, such as naive Bayes, k-nearest neighbor algorithm, support vector machine, and deep learning methods, such as convolutional neural network, recurrent neural network.

In reference [1], five dictionaries are added based on the dictionary, and the semantic rules between texts are analyzed to realize text sentiment analysis. In reference [2], the word vector is mapped to a low dimension in skip-gram model, combined with CNN feature extraction, and a layer of highway network is added to optimize the overall feature, so as to improve the accuracy of the model. In reference [3], a KNN based method and a general graph based classification method are proposed to distinguish the spammers in Chinese comment websites. The classification effect of the two methods is obviously better than that of the pure index classifier. In reference [4], a series of experiments on classification tasks were carried out with CNN, and a simple modification of the architecture was made. Using both task specific vectors and static vectors, good results were achieved on multiple benchmarks. It is proved that a simple CNN with few super parameter adjustment and static vectors achieves good results on multiple benchmarks. In reference [5], a global maximum pooling model is proposed. By extracting multiple convolution layers and GMP layers, deeper semantic features are obtained, and emotional scores are
generated in the point accumulation layer to improve the efficiency of the model. On the basis of traditional convolutional neural network model, the pooling layer is removed and the GRU layer of gate valve cycle unit is added in reference [6], and a new model of series parallel convolutional gate valve cycle network is proposed, which improves the classification accuracy.

The structure of the traditional classification model is simple. Improving the model or combining with some other models will improve the accuracy of classification. Therefore, this paper optimizes its features through the emergence V1 model, clusters the sparse matrix into dense matrix, and finally classifies in the soft classifier. In this experiment, the data samples are iterated for many times. The results show that the accuracy gradually increases and tends to be stable. Through the comparative experiments of several classification models, i-cnn model has better classification effect than other machine learning models.

2. Related work

2.1. Convolutional neural network and text classification
Convolution neural network is a kind of deep neural network model, which is essentially multi-layer convolution operation. The output of the upper layer neuron is taken as the input of the next layer. Through multi-layer convolution calculation, the convolution operation results of each layer are converted nonlinearly. Compared with the traditional neural network, convolutional neural network has two advantages: parameter sharing and local connection, which greatly reduces the number of training parameters, but does not reduce the accuracy.

Parameter sharing can also be understood as "translation invariance", which refers to a group of neurons corresponding to a weight, rather than each neuron corresponding to a weight, thus reducing a lot of parameters. Local connection refers to the local connection with the neurons in the upper layer, rather than with all the neurons in the upper layer, which reduces many parameters and the amount of subsequent calculation.

When the traditional machine learning method is used to do text classification task, the tf-idf vector of the document is usually input as the feature of the model, which will make the tf-idf representation lose the order in the text sequence. When the convolution neural network is used to model the text data, it inputs the word vector, and then transforms the word vector with different length into the vector with fixed length through the sliding window, which can retain some local features and sequence information in the original text. Because the long-distance dependency between two words is difficult to capture, CNN based text classification is more suitable for short text.

2.2. Text segmentation
Different from the English text, there is a separator between words, so there is no word segmentation processing. There is no separator between Chinese text words, and a single Chinese character can not fully express the meaning, so Chinese text should be word segmentation processing. There are many problems in Chinese word segmentation, the most important one is ambiguity. Different word segmentation of sentences will have different meanings. For this problem, many different convolution kernels can be used in convolution. Secondly, word segmentation dictionary should be comprehensive and updated in time. Word2vec word segmentation machine is selected for Chinese word segmentation. It is a Chinese data processing toolkit based on gensim library. In the dictionary, every word has a unique vector correspondence, and the text can be transformed into vector representation.

2.3. Stop word processing
Stop words are words with high frequency and no practical significance in the text, such as "de", "ah", "Le", and so on. They also include punctuation marks and network expression packs. They do not express any emotional tendency and have no influence on the emotional division of the text. E-mails are generally short texts, and the existence of stop words has a greater impact on the results.
The common way to remove stop words is to select a comprehensive vocabulary, filter and remove duplicate words, and remove stop words one by one. The de-duplicated text not only reduces the dimension of feature vector, but also reduces the complexity of subsequent calculation, and improves the accuracy of classification to a certain extent.

2.4. Word embedding
When the image is processed by computer, the input is a two-dimensional matrix, so when processing Chinese e-mail, the text should also be transformed into a matrix form, and the words in the text should be transformed into word vectors through the open source tool word2vec. CBOW and skip-gram are the two models of word2vec. CBOW is more suitable for databases with less data samples, while skip-gram has higher classification accuracy in large corpora. So skip-gram model is used to construct word embedding model.

Skip-gram model predicts the context $S_{w_t} = (w_{t-k}, ..., w_{t-1}, w_{t+1}, ..., w_{t+k})$ through the input word $w_t$, where $k$ is the size of $W_t$ convolution window, that is, the number of predicted word vectors in the context, and CBOW model predicts a word vector $w_t$ according to the context $S_{w_t}$. The training objective optimization functions of skip-gram model and CBOW model are shown in formula (1) and formula (2) respectively:

$$L_{\text{Skip-Gram}} = \sum_{w_t \in C \setminus S_k} \sum_{j \neq 0} \log P(w_{t+j} | w_t)$$

(1)

$$L_{\text{CBOW}} = \sum_{w_t \in C} \log P(w_t | S_{w_t})$$

(2)

Where $C$ is all the word vectors in the data sample and $K$ is the size of convolution window. Word2vec gets the parameters of hidden layer in the trained network, which are the word vectors learned by word2vec.

3. i-CNN model
Although the traditional CNN model has reduced the parameters in the convolution and pooling layer through partial connection and down sampling, with the increase of sample data, there will be data sparsity, resulting in a waste of space. At the same time, it will also bring the problem of too high dimension, which will increase the computational complexity. In order to solve these two problems, the concept of intrusion is proposed. Combining V1 model with CNN model, i-cnn model is proposed. $1 \times 1$ convolution kernel is added after convolution layer. By reducing the channel dimension of eigenvector, the parameters are reduced, and the calculation efficiency and classification accuracy are improved.

The framework flow of i-cnn model is shown in Figure 1: input the processed word vectors into the model, use the filter to obtain the feature vectors, change the window size to obtain multiple feature vectors, pool layer to filter the strongest feature vectors, and finally output the probability of categories to the soft classifier, calculate the loss function by comparing the classification results and dataset labels, and then pass it back through the gradient descent algorithm. The model parameters are adjusted to reduce the loss until the network converges, and the training process is completed.
3.1. Input layer
The input layer processes the Chinese E-mail as a word vector, which is generally a matrix of $n \times k$, where $n$ is the total number of words in the text. The dimension of the word vector of the sample data should be consistent. If the length is not enough, padding is used to fill in the zero. If the length is too long, it will be discarded. The Open Source Toolkit word2vec is used to transform each word into a $k$-dimensional vector, which is obtained by training as an unknown parameter.

Our company has some general invoices, sales invoices, VAT invoices and other service industry invoices. After processing, it becomes \{the company, some, ordinary invoice, commodity, sales, invoice, VAT, invoice, and other, service, industry, invoice\}.

3.2. Optimization of convolution layer

3.2.1 Convolution layer
It is the most important step to extract the feature of sample data in convolution layer. On the input $n \times k$ matrix, the convolution operation is defined as formula (3):

$$c_i = f(w \cdot x_{i:i+h-1} + b)$$

(3)

Where $X_{i:i+h-1}$ represents the sliding matrix window with the size of $h \times k$ composed of the $i$-th to $i+h-1$ rows of the input matrix, $W$ is the same as the dimension of $X_{i:i+h-1}$, which is also $h \times k$, and $b$ is the bias parameter. $F$ is a nonlinear activation function and $C_i$ is a scalar.

A convolution kernel is performing convolution operation. Assuming that the size of convolution window is 2, convolution is performed on a $2 \times k$ sliding window, $n-1$ results can be obtained, and then the $n-1$ dimensional eigenvectors are assembled, as shown in formula (4)

$$C = (C_1, C_2, C_3, \cdots, C_{n-1})$$

(4)

Different convolution window $h$ can be set in convolution layer to extract different features. A single convolution operation may cause the deviation of sentence meaning understanding. Different convolution kernels with different sizes can be set, and each convolution operation of convolution kernel

![Figure 1 Flow chart of network framework](image)

Figure 1 Flow chart of network framework
can extract a feature, which can extract different features and reduce the classification error due to semantics.

3.2.2 Inception V1 model

The concept V1 model mainly improves the convolution layer in the network. For the problems of too many network parameters, easy over fitting, and deep network gradient descent stagnation, 1 * 1 convolution kernel can reduce the channel dimension of eigenvector. In the convolution layer, different size windows are used for feature extraction. Because Max pooling itself is also a feature extraction, it can also be used as a branch, which is the inception V1 model. The structure of the model is shown in Figure 2. Before 3 * 3, 5 * 5 and after Max pooling, 1 * 1 convolution kernel is added to reduce the thickness of the eigenvector.

![Figure 2 Structure diagram of inception V1 model](image)

According to the above figure for comparative calculation, assuming that the upper input eigenvector dimension is 256 and the output eigenvector dimension is 256, the 256 dimension eigenvector is directly output with 3 * 3 * 256 convolution kernel, and the parameters are 256 * 3 * 3 * 256 = 589824. If we first pass through a 1 * 1 * 64 convolution kernel, then pass through a 3 * 3 * 64 convolution kernel, and finally pass through a 1 * 1 * 256 convolution kernel, the parameters are 256 * 1 * 1 * 64 + 64 * 3 * 3 * 64 + 64 * 1 * 1 * 256 = 69632, which reduces most of the parameters.

3.3. Pooling layer

The pooling layer compresses the upper feature vectors to extract the main features, and reduces the network parameters and subsequent computational complexity. There are many pooling methods, mainly using max pooling, that is, taking the maximum eigenvector generated by each convolution kernel in the upper layer as the eigenvalue, as shown in formula (5):

$$c = \max(c_1, c_2, \ldots, c_{n-b+1})$$  \hspace{1cm} (5)

The feature extracted by each convolution kernel in the upper layer generates a feature value, and then all the values constitute the feature vector of pooling layer, which is sent to the full connection layer, and the softmax function is used to output the probability of each text category.

3.4. Fully connected layer

The full connection layer integrates the previous highly abstract features, transforms the dimensions, adds the nonlinear mapping of the activation function, and outputs the classification situation through softmax function. The softmax expression formula (6) is as follows:

$$y_i = \frac{e^i}{\sum_{j=1}^{n} e^j}$$  \hspace{1cm} (6)
i is the i-th value of the n-dimensional vector. For the input \( n \) is a vector, and each value of the vector represents the probability of the input for each class.

4. Case analysis and Application

4.1. Experimental data
The data of this experiment is the Chinese E-mail obtained and sorted from the network. According to the semantics, the e-mail is divided into normal e-mail and spam. Normal e-mail is what we need, while spam is some advertisements, tweets and so on. According to the semantics, the e-mails are put in two folders, 5000 normal e-mails are put in the positive examples folder for comments, and 5000 spam e-mails are put in the negative examples folder for negative comments. Experiments are carried out on two different datasets to evaluate the proposed model. For each data set, we randomly select 80% as the training set and 20% as the test set, and carry out several iterations until the loss function converges.

4.2. Experimental environment
The experiment uses Intel Core i5 processor, 8g memory, windows system, the development language is Python 3.7 in Anaconda integrated environment to write the experimental code, and the development tool is pychar Community Edition. The main environment configurations are shown in Table 1:

| Table 1. Experimental environment and configuration |
|-----------------------------------------------|
| experimental environment | Environment configuration |
| Operating system | Windows10 |
| Programing language | Python3.7 |
| Word segmentation tool | Word2vec |
| Learning framework | Tensorflow |

4.3. Experimental design
In this experiment, word2vec Chinese toolkit in gensim Library of anaconda is used for word segmentation and part of speech tagging, and the words in e-mail are transformed into word vectors, and then the word vectors are trained. The main experimental parameters are: the dimension of word vector, the size of convolution window, pooling method and so on. The dimension of the word vector is set to 128, the convolution window size is set to (3,4,5), the convolution core number is set to 128, the ratio is 0.5, the batch size is 64, and the evaluation interval is 50. The parameter setting of the network model is shown in Table 2:

| Table 2. Network model parameters |
|----------------------------------|
| Hyperparameter | Parameter value |
| Word vector dimension | 128 |
| Convolution window size | 3,4,5 |
| Number of convolution kernels | 128 |
| ratio | 0.5 |
| Batch size | 64 |
| Evaluation interval | 50 |
| Pooling method | max-pooling |
| inception Convolution kernel | 1*1 |

In this experiment, the classification effect of i-cnn model is verified by the sorted Chinese e-mails
(1) The training set data is preprocessed, word segmentation, stop word, word vector transformation input network model, training, every 50 times to do evaluation detection, retain the accuracy value, use the accuracy to measure the performance of the model.
(2) In order to compare the performance of i-cnn model with other classification methods in Chinese e-mail classification, three traditional classification models, support vector machine (SVM), naive Bayes (NB) and k-nearest neighbor algorithm (KN), are used to do a comparative experiment, using the same set of data set, 5000 positive comments and 5000 negative comments each.

(3) In order to compare the optimization effect of the emergence V1 model on the traditional CNN model, all parameters are consistent in the same data set.

4.4. Analysis of experimental results

The training results of the dataset in the model are shown in Figure 3. After 600 times of iterative training, the accuracy rate has steadily increased to more than 90%, and the highest is 93.7%, which verifies that the i-cnn model has a good classification effect on spam.

![Figure 3 Accuracy of training data](image)

As can be seen from table 3, the classification accuracy of i-cnn model reaches 92.18%. In addition to the classification effect of support vector machine, the classification performance of i-cnn model is much higher than that of naive Bayes and k-nearest neighbor algorithm, which indicates that i-cnn model has better classification performance.

| Model      | accuracy /% |
|------------|-------------|
| SVM        | 90.11       |
| Naive Bayes| 86.5        |
| KNeighbors | 68.34       |
| i—CNN      | 92.18       |

Table 3. Accuracy of data classification

As can be seen from table 4, in the same data sample, the accuracy rate of CNN model is improved by 1.45% by adding the emergence model. The effect is not very good. The main reason is that the depth of the network is not enough, and the emergence V1 model is better for deep network optimization.

| Model         | accuracy /% |
|---------------|-------------|
| CNN           | 90.73       |
| i-CNN         | 92.18       |

Table 4. Comparison of results with and without induction structure
5. Conclusion
Through the e-mail classification experiment, we can see that i-cnn model has a good effect in text classification, and the emergence V1 model can indeed optimize the network and improve the classification effect. However, judging from the results, there is still a lot of room for improvement. For this experiment, the data samples are not large enough, the types are not comprehensive enough, the network depth is not enough, and the current network model is lack of versatility and flexibility. More exploration is needed to improve the classification effect. We can try to combine with other models or improve the convolution neural network model for further research.

Acknowledgments
Thanks for the help of my classmates in the past six months, and thanks for the guidance of Mr. Lu, I can successfully complete a series of tasks such as experiment, thesis writing, and revision. Mr. Lu not only gives us academic guidance, but also cares and encourages us in our life. Our classmates help each other and study together in a friendly environment. It's your concern and encouragement that gives me the message to stick to it. Thank you from the bottom of my heart!

References
[1] Wu J S, Lu K. (2019) Research on sentiment analysis of Chinese microblog based on multiple sentiment dictionaries and rule sets. Computer applications and software , 36 (9): 93-99.
[2] Huang H, Jing X Y, Dong X W, et al.(2019) CNNs text mail classification model based on skip gram. Computer technology and development, 29 (6): 143-147.
[3] C. Xu, J. Zhang, K. Chang, et al.(2013) Uncovering collusive spammers in chinese review websites, in: 22nd ACM international Conference on information & knowledge management. ACM. San Francisco.pp.979–988.
[4] Kim Y. (2014) Convolutional neural networks for sentence classification, in: Conference on empirical Methods In Natural Language Processing. Doha. pp. 1532–1543.
[5] Zhou J F, ye S R, Wang H.(2019) Text sentiment classification based on deep convolution neural network model [J]. Computer Engineering, 45 (3): 300-308.
[6] Tang X L, Lin W X, Du Y M, et al.(2019) Feature extraction and classification of short text based on series parallel convolution gate valve recurrent neural network, engineering science and technology, 51 (4): p93-99.
[7] Wang Feng,Cheng Jian,Liu Weiyang,et al.(2018) Additive margin softmax for face verification[J].IEEE Signal Processing Letters, 25(7):926–930.
[8] Lu Chi,Huang Heyan,Jian Ping, et al.(2017) A P–LSTM neural network for sentiment classification[M]//Advances in Knowledge Discovery and Data Mining. Cham::Springer,: 524–533.