Chinese spam filtering based on Stacked Denoising Autoencoders

Liuyan ZHANG, Yunfeng NIE, Shengyue DUAN

school of information Engineering, Nanchang HangKong University,
NanChang330063, China
2238720674@qq.com

Abstract. Aimed at the problem that the traditional feature selection method extracts the feature items and the filtering accuracy is degraded in the Chinese spam filtering process, this paper proposes a Chinese spam filtering method based on Stacked Denoising Autoencoder (SDA). Firstly, use the Continuous Bag-of-Words (CBOW) model training the Word2vec tool set for the processed corpus to transform word segments into vectors; the inputs are the word vectors; then apply the Stacked Denoising Autoencoder (SDA) to effectively extract the features in unsupervised learning. Finally, the improved softmax classifier is used for regression classification. The test was carried out on the TREC06C dataset, the experimental results show that compared with Bayesian model, KNN classification algorithm and traditional Stacked Denoising Autoencoder, the accuracy, precision, recall, and f1 score of the method reached 93.5%, 94.8%, 92% and 93.2%, and had better dichotomous effect and robustness in the application.

1. Introduction

According to the 2017 China Enterprise Mailbox Security Report, the average number of e-mail users sent and received by corporate e-mail users nationwide was about 1.61 billion per day. The proportion of normal mail was about 25.4%, and that of ordinary spam was 64.7%[1]. It can be seen that spam has become a serious disaster area of network security. Spam not only occupies network resources, but also damages users’ interests and poses a threat to network security because of its spread. Therefore, the research on anti-spam technology is very important.

Chinese spam filtering is based on content filtering, which is divided into four parts: Chinese word segmentation, text representation, feature learning and classification, in essence, it can be considered a special text Classification task. Therefore, text classification algorithms can be used to solve spam filtering problem, such as Bayesian classification algorithm[2,3] is one of the most commonly used methods of spam filtering. With the rise of machine learning, neural networks are used in spam classification because of their complex nonlinear fitting capabilities[4,5], such as k-Nearest Neighbor algorithm (KNN)[6]. Compared with the shallow neural network model, deep learning has more nonlinear structural units and can perform more complex tasks[7]. Convolutional Neural Networks (CNN) [8] in supervised learning have achieved good results in applications, but its performance depends on the number of labeled training samples, Huge data is very time consuming and labor intensive, Therefore, how to learn data characteristics in an unsupervised way has a very important research value.

Denoising Autoencoder is a modified structure of Autoencoder, which can extract more robust features[9]. Therefore, multiple Denoising Autoencoder can be stacked to form a deep neural network...
model. It pre-trains large amounts of unlabeled data in an unsupervised way and adapts the parameters to make the network model fit the data better. Compared with KNN, Bayesian model and traditional Denoising Autoencoder, the accuracy, precision, recall and f1 scores of the method reached 93.5%, 94.8%, 92% and 93.2%, with better classification performance and stability.

2. Stacked Denoising autoencoder (SDA)

2.1. Word2vec model
This article uses a NLP tool word2vec model launched by Google in 2013 to process text data, which can introduce semantic features not found in traditional text representation, the schematic diagram is shown in Fig.1, it can recognize the important relationship between words.

![CBOW and Skip_gram schematic](image)

Fig.1 CBOW and Skip_gram schematic

The Continuous Bag-of-Words (CBOW) model in Word2vec is able to predict the probability of the output of the central word based on n-1 words around the input. The structure diagram is shown on the left side of Fig.1. The network structure is divided into three layers: input layer, projection layer, and output layer.

Input layer: Use one-hot encoding, input the central word \( w_t \) context word, perform the calculation of formula (1), and obtain the word vector corresponding to the word \( w_t \).

\[
\begin{align*}
    w_t^T W &= v_t \\
    \text{Hidden layer: Through the formula (1), get the central word } w_t \text{ context words } W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2} \text{ word vector } v_{t-2}, v_{t-1}, v_{t+1}, v_{t+2} \text{ to calculate the probability of the occurrence of the word } w_t. \text{ based on these word vectors, it is necessary to sum the four word vectors, then take the average and perform the formula (2) to obtain the average calculation:}
    v_{\text{sum}} &= \frac{v_{t-2} + v_{t-1} + v_{t+1} + v_{t+2}}{4}\end{align*}
\]

Output layer: Calculate the probability of generating \( w_t \) from context words according to formula (3).

\[
\begin{align*}
P(W_t|W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2}) &= \frac{\exp[u_t^T(v_{t-2} + v_{t-1} + v_{t+1} + v_{t+2})]}{\sum_{j=1}^V \exp[u_t^T(v_{j-2} + v_{j-1} + v_{j+1} + v_{j+2})]}\end{align*}
\]

Where \( V \) represents the size of the dictionary, \( v_{t-2}, v_{t-1}, v_{t+1}, v_{t+2} \) represents the vector of the context word obtained by equation (1), \( u_t \) represents A scoring weight is a row in the output layer matrix.

2.2. Denoising Autoencoder
Denoising Autoencoder is a variant of the Autoencoder that takes the corrupted data as input and trains to predict data that was not corrupted, which can enhance the robustness of the algorithm and the generalization ability of the model. The network structure is shown in Fig.2.
Fig. 2 Structure diagram of Denoising Autoencoder

The encoding operation from input layer to hidden layer uses formula (4): 
\[ y = f_\theta(x) = s(Wx' + b) \]  
(4)

Where \( W \) is a weight matrix of \( d \times d' \) and \( b \) is an offset vector. \( s \) is the activation function of the encoder, such as sigmoid, tanh, relu, etc. This paper uses the sigmoid activation function, which is expressed as formula (5):
\[ f(z) = \frac{1}{1 + \exp(-z)} \]  
(5)

The operation of hiding the layer to the decoding layer uses formula (6):
\[ z = g_\theta(y) = s(W'y + b') \]  
(6)

The weight matrix \( W' \) of the inverse mapping is the transposed matrix of the weight matrix \( W \), and \( b' \) is an offset vector.

Each input data \( x^{(i)} \) is mapped to a \( y^{(i)} \) by formula (4), and \( y^{(i)} \) is reconstructed by formula (6) to obtain \( z^{(i)} \), then all the parameters of the model are obtained by continuous reverse finetuning to obtain the minimum average reconstruction error:
\[ L(x, y) = \|x - z\|^2 \]  
(7)

2.3. Stacked Denoising Autoencoder (SDA)

SDA\(^{17, 18}\) is a deep network model structure constructed by stacking multiple Denoising Autoencoder(DA), its purpose is to extract features by layer-by-layer DA. Fig. 3 is a Stacked DA model diagram, the stacking process is mainly divided into the following four steps:

Step1: Given the initial noise input \( x' \), the first layer of DA is trained in an unsupervised manner.

Step2: The output of the hidden layer of the first DA is used as the input of the second DA, and the same method is used to train the DA.

Step3: Repeat the second step until the training completes all DA.

Step4: The output of the hidden layer of the last SDA is used as the input to the classifier, and then the parameters of the classifier are trained using a supervised method.

Fig. 3 SDA model diagram

3. Improved stacked Denoising Autoencoder algorithm

3.1. Dropout regularization

In the training process, when the training error is gradually reduced, the fitting effect of the data is better. However, as the training error decreases and the test accuracy decreases gradually, overfitting occurs. In order to prevent overfitting, the Dropout technique is adopted in this paper, in the training process, the model discards a part of the neurons, which is to prevent over-fitting by modifying the network itself, the neural network model is shown below in Fig. 4.
Fig. 4 Dropout’s neural network model [18]

Probability formulas are added to each layer of the training network. When no dropout is added to the network, the output of the network is calculated as Equation (8):

$$y^{l+1} = f(w^{l+1}y^l + b^{l+1})$$  \hspace{1cm} (8)

The dropout regularization is to add a Bernoulli function to each input, randomly generating a vector of 0, 1, and the formula (8) is changed to (10):

$$r_j^{(l)} \sim \text{Bernoulli}(p)$$  \hspace{1cm} (9)

$$y^{l+1} = f(w^{l+1} \cdot r^{(l)} \cdot y^{(l)} + b^{l+1})$$  \hspace{1cm} (10)

3.2. L2 regularization

The softmax classifier is generally used to solve the classification problem, the softmax function is shown in formula (11). Each output is mapped to a range of 0 to 1 and the training is approximated to the optimum $\theta^T$, where $\theta_i^T x$ is determined to be a plurality of inputs.

$$P(i) = \frac{\exp(\theta_{ix})}{\sum_{k=1}^{K} \exp(\theta_{ix})}$$  \hspace{1cm} (11)

Add L2 regularization to the softmax classifier, after adding the regular term: $\frac{\lambda}{2n} \sum w^2$, get the formula (12):

$$J'(\theta) = J(\theta) + \frac{\lambda}{2n} \sum w^2$$

$$J'(\theta) = \frac{1}{m} \left[ \sum_{i=1}^{m} \sum_{j=1}^{K} \mathbb{1}[y^{(i)} = j] \cdot (\theta_j^T x^{(i)} - \log \left( \sum_{l=1}^{K} e^{\theta_l^T x^{(i)}} \right) \right] + \frac{\lambda}{2n} \sum w^2$$  \hspace{1cm} (12)

Where $\lambda$ is the coefficient of the regular term, and the partial derivative is obtained for $J'(\theta)$:

$$\frac{\partial J'(\theta)}{\partial w} = \frac{\theta_j^T x^{(i)} \cdot \mathbb{1}[y^{(i)} = j]}{n} - \frac{\lambda}{n} w$$

$$\frac{\partial J'(\theta)}{\partial b} = \frac{\partial J'(\theta)}{\partial w}$$  \hspace{1cm} (13)

It can be found that the L2 regularization has no effect on the update of the offset $b$, but has an effect on the update of the weight $w$:

$$w \rightarrow w - \eta \frac{\partial C_\theta}{\partial w} - \frac{\eta \lambda}{n} w = \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_\theta}{\partial w}$$  \hspace{1cm} (15)

3.3. Chinese spam filtering process based on stacked Denoising Autoencoder

The process of processing the spam by the Stacked Denoising Autoencoder is shown in Fig 5. The steps of implementing Chinese spam filtering based on the Stacked Denoising Autoencoder are as follows:

Step1: Filtering characters. Remove all special characters in the text, a large number of stop words,
invalid characters, etc.

Step 2: Chinese Word Segmentation. In order to train the word2vec model, Chinese word segmentation processing of text data is required. This paper directly separates each Chinese character, that is, obtains the word vector of the corresponding word.

Step 3: Alignment. The input training samples need to be aligned so that their dimensions are consistent.

Step 4: Start training the word2vec model according to the formula in section 1.1.

Step 5: SDA. Train the deep network according to the training process of the SDA in Section 1.3, that is, use the Chinese word vector obtained in the training word2vec model as the input of the Denoising Autoencoder, taking the Denoising Autoencoder as the basic unit, each layer is pre-trained as a Denoising Autoencoder, and then stacked, and finally the overall reverse tuning training is performed to obtain the parameter weight, the bias and Output.

Step 6: In the last layer of the deep network, add the improved softmax classifier in section 3.2 above, minimize the classification error in supervised learning, reverse finetune the network, obtain the most classified model, and classify the output.

Step 7: Optimize the model based on classification effects.

4. Experiment and analysis

4.1. Experimental data and evaluation indicators
The experimental environment is: hardware configuration CPU i7-7700K, GPU Gtx1060, running memory 16G; software configuration CUDA9.0, cudnn7.1; operating system Ubuntu16.04; language python3.6; framework tensorflow.

The experimental corpus uses the Chinese spam data set (TREC06C), which is from the public spam corpus provided by the International Text Search Conference. The experimental data distribution is shown in Table 1.

| Total number of samples | Positive sample | Negative sample |
|------------------------|-----------------|-----------------|
| 11360                  | 5596            | 5764            |

Commonly used spam filtering system evaluation indicators have precision, recall, accuracy, and f1 scores, the calculation is based on the confusion matrix in Table 2, as shown in formula (16), (17), (18), and (19).

| Total number of samples | Positive example | Negative example |
|-------------------------|------------------|------------------|
| Positive sample rate    | TN               | FP               |
| Negative sample rate    | FN               | TP               |

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

4.2. Experimental model improvement and result analysis
The traditional SDA training effect is shown on the left side of the figure. The fitting effect of the data was not very good, and the accuracy, precision and f1 scores were slightly oscillated. To solve this problem, add dropout regularization to the network. The dropout value is set to 0.7, and the loss rate, accuracy, precision, recall rate, and f1 scores change are shown in the figure, the Dropout regularization
is added to the network, the data fitting effect is better, and the value is relatively stable.

(a) Original SDA loss value  
(b) Loss value after adding dropout  

Fig. 6 cost

(a) Original SDA accuracy  
(b) Accuracy after adding dropout  

Fig. 7 accuracy

(a) Original SDA precision  
(b) Precision after adding dropout  

Fig. 8 precision
4.3. Experimental model comparison and result analysis

This experiment was performed on the TREC06C dataset, the accuracy, precision, recall and f1 scores of Chinese spam filtering are compared with the naive Bayesian Gaussian model, the naive Bayesian Bernoulli model, KNN, and the improved SDA. The rate and the f1 score value, the test results are shown in Table 3.

| Evaluation index | the naive Bayesian Gaussian model | the naive Bayesian Bernoulli model | KNN | The improved SDA |
|------------------|----------------------------------|-----------------------------------|-----|------------------|
| precision        | 0.86                             | 0.881                             | 0.901| 0.948            |
| recall           | 0.724                            | 0.837                             | 0.907| 0.92             |
| accuracy         | 0.99                             | 0.916                             | 0.947| 0.935            |
| F1               | 0.837                            | 0.875                             | 0.924| 0.932            |

From the experimental results, it can be analyzed that, on the same dataset, compared with the naive Bayesian model and KNN, the performance of the improved SDA model is obviously due to the naive Bayesian model. The improved SDA has a recall rate of 92% in spam filtering, while the KNN model has a recall rate of 90.7% for Chinese spam, which indicates that the improved SDA recognizes 1.3% more spam than KNN, and because it has better accuracy and f1 score than KNN, the Chinese spam filtering performance of the improved SDA model is improved by more than 1% compared to KNN. In practice, these differences are very important for Chinese spam filtering.
5. conclusion
In this paper, based on the characteristics of Chinese text in Chinese spam, this paper proposes to apply SDA to Chinese spam filtering. In the model training, in order to avoid over-fitting, add Dropout regularization in the base Denoising Autoencoder, and add L2 regularization constraint in the softmax classifier to improve the classification performance, with the experiment to verify the validity of the model in Chinese spam filtering. Compared with the KNN model, the naive Bayesian model and the traditional Denoising Autoencoder on the public dataset TREC06C, the accuracy, precision, recall and f1 scores of the model reached 93.5%, 94.8%, 92% and 93.2%. Compared with KNN and Naive Bayesian model, this method has the best detection efficiency and has a good classification in spam detection.

Acknowledgments
This work was funded by a research project on geospatial sensor network deployment issues (41101426) under complex terrain conditions.

References
[1] 2017 China Enterprise Mailbox Security Research Report. 2018-06-08. https://ti.360.net/blog/articles/2017-report-of-mail-security/.
[2] ALMEID A T A. YAMAKAMI A. Advances in spam filtering techniques [M]. Berlin: Springer, 2012: 199 — 214.
[3] Graham Paul(2002). Better Bayesian Filtering[OL]. 2013-11-13. http://www.paulgraham.com/spam.html.
[4] PUNISKIS D., LAURTIS R. DIRMEIKIS R. An artificial neural nets for spam E-mail recognition[J]. Electronics and electrical engineering, 2006, 69(5): 73-76.
[5] GUO S T, XU Z G. Spam filtering based on bp neural network[J]. Computer security, 2009, 10(12): 20-23.
[6] Tian Ze, Yan SongYuan, Xu JingDong. Spam filtering technology based on improved K-nearest neighbor[J].Computer Engineering and Applications, 2002-09-19.
[7] BENGIO Y. Learning deep architectures for AI[J]. Foundations and Trends in Machine Learning, 2009, 2(1): 1-127.
[8] KIRZHEVSKY A, SUTSKEVER I, and HINTON G E. Imagenet classification with deep convolutional neural networks[C]. Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, 2012: 1097-1105.
[9] Vincent, Pascal Larocheille, Hugo Bengio, Yoshua Manzagol, Pierre-Antoine. Extracting and Composing Robust Features with Denoising Autoencoders[C]. Proceedings of the 25th international conference on Machine learning, 2008: 1096-1103.
[10] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]/NIPS 2012: 1106-1114.
[11] COATES A, NG A Y, LEE H. An analysis of single-layer networks in unsupervised feature learning[J]. Journal of Machine Learning Research. 2011, 15: 215-223.
[12] Tomas Mikolov, Kai chen , Greg Corrado, et al. Efficient estimation of world representations in vector space][EB/OL]. [2014-09-19]. http://arxiv.org/abs/1301.3781v3.
[13] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [C]/NIPS, Lake Tahoe, USA, December, 05-08, 2013.
[14] BENGIO Y. Learning deep architectures for AI[J]. Foundations and Trends in Machine Learning, 2009, 2(1): 1-127.
[15] Neural Networks and Deep Learning [EB/OL]. [2015-12-23]. http://neuralnetworksanddeeplearning.com/chap3.html.
[16] VINCENT P, LAROCHELLE H, LAJOIE I, et al. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine
Learning Research, 2010, 11(6): 3371-3408.

[17] Minmin C, Eddie X, Killian Q, Fei Sha. Marginalized denoising autoencoders for domain adaptation[ C]. Proceedings of the 28th international conference on Machine learning, 2012.

[18] Hinto G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science, 2006,313: 504–507.

[19] Qin Shengjun, Lu Zhiping. Research of Unbalance Sentiment Classification Based on Denoising Autoencoders [J]. Science Technology and Engineering, 2014, 51(12): 232-235.