Software Engineering
2016; 4(2): 9-12
http://www.sciencepublishinggroup.com/j/se
doi: 10.11648/j.se.20160402.11
ISSN: 2376-8029 (Print); ISSN: 2376-8037 (Online)

SciencePG
Science Publishing Group

Chinese Spam Filtering Based On Back-Propagation Neural Networks

Peiguo Li[1], Yan Ye[2]

[1]Department of Mathematics, Jinan University, Guangzhou, China

[2]Department of Computer Science, Guangzhou College of Commerce, Guangzhou, China

Email address:

peiguo.li@163.com (Peiguo Li), bfredleaf@163.com (Yan Ye)

To cite this article:

Peiguo Li, Yan Ye. Chinese Spam Filtering Based On Back-Propagation Neural Networks. *Software Engineering*. Vol. 4, No. 2, 2016, pp. 9-12. doi: 10.11648/j.se.20160402.11

Received: March 30, 2016; **Accepted:** April 8, 2016; **Published:** April 16, 2016

Abstract: As the email service is becoming an important communication way on the Network, the spam is increasing every day. This paper describes a new filtering model based on email content by using Back-Propagation Neural Networks (BPNN). And for the Chinese email, it uses Natural Language Processing & Information Retrieval Sharing Platform (NLPIR) system to perform Chinese word segmentation. The simulation results show that this model can precisely filter the Chinese spam.

Keywords: Spam, BPNN, NLPIR

1. Introduction

Spam, also known as junk email, is a subset of electronic spam involving nearly identical messages sent to numerous recipients by email [1]. The spams take up huge Internet resources and users' time. And the cost of spam in terms of lost productivity has reached about $20 billion annually, according to the National Technology Readiness Survey [2].

Nowadays, two general measures have been used in anti-spam system: filter-based and content-based [3]. In filter-based way, sets of rules have to be set up by user or some filter system [4], such as white/black list, specific words in email address or email title, etc. In this way, the user or filter needs to frequently update the rules to adapt to the changing spamming. In content-based way, the email server uses some classification algorithms based on email content, to determine if an email is spam or not. Most of content-based measures use an artificial intelligent algorithm as the classification, which means they don't require complex rules to be maintained.

At present, several algorithms have been used for spam classification, include support vector machines [5], Bayesian classifiers [6], boosting decision trees [7], rough sets [8], neural networks [9], fuzzy logic [10], etc. In this paper, we introduce the use of BPNN for content-based spam filtering, and NLPIR system to perform Chinese word segmentation. During the training stage of the BPNN, we use GA to optimize the architecture of the BPNN to get an optimal result.

2. Related Studies

2.1. The Standard BPNN

The back-propagation neural network is a widely used supervised learning algorithm. A typical BPNN, see Figure 1, there is an input layer, an output layer, a hidden layer among them. During the training stage, the current output is compared with the desired output of the training sample, and then to adjust the weights of the network.

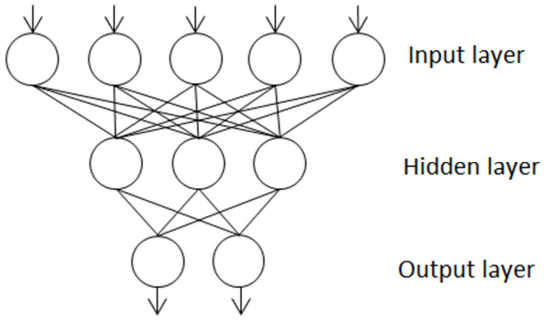

Figure 1. *A typical BPNN.*

There are two main problems of the standard BPNN, its slow training speed and easy to fall into a local optimal solution.

2.2. GA-BPNN

To resolve these problems, we introduce Genetic Algorithm (GA) to speed up the training speed and avoid the local optimal solution. In this study, we use five steps to combine the GA and BPNN:

1. Encode: design an encoding scheme for all the weight of the network, such as:

$$\overbrace{\omega_{11},\omega_{12},\dots,\omega_{1m}}^{\textit{weights of input layer}},\overbrace{b_{11},b_{12},\dots,b_{1n}}^{\textit{thresholds of hidden layer}},\overbrace{\omega_{21},\omega_{22},\dots,\omega_{2j}}^{\textit{weights of}\text{ hidden layer}},\overbrace{b_{21},b_{22},\dots,b_{2k}}^{\textit{thresholds of}\text{ output layer}} \quad (1)$$

2. Initialization: initialize a random population of N chromosomes (encoded weights of the network);
3. Fitness: construct a BPNN by decoding every chromosome in the population, and calculate the mean square error of the BPNN as the fitness of current chromosome;
4. GA loops;
5. Test: if the end condition is satisfied, stop the loop and get the best structure of the BPNN;

Figure 2 shows the whole picture of this process.

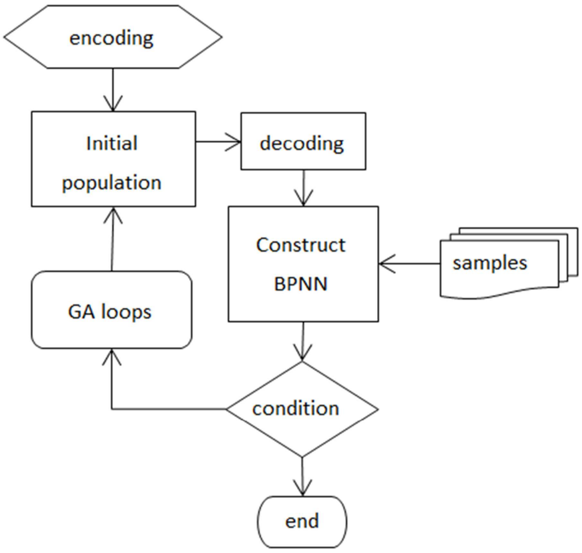

Figure 2. *Workflow of GA-BPNN.*

2.3. NLPIR System

The NLPIR segmentation system is developed from ICTCLAS system. Its main functions include Chinese segmentation, POS tagging, Named entity recognition, user dictionary and Keywords identification [11]. The system can work with GBK, UTF-8 and Big5 encoding content. And there are a lot of APIs/Interfaces for C/C#, Java, etc.

In this paper we use this system to do segmentation of the Chinese email content. After the segmentation, we get a set of keywords from the email content, which construct the input value of the BPNN.

3. Chinese Spam Filter Model

There are four models in our spam filter, segmentation model, keyword extraction model, spam filtering model and user feedback model.

3.1. Segmentation Model

In this model, the main job is do segmentation of the email content. Before that, we need to preprocess the email content.

1. Identify the encoding of the email;
2. Remove the head data of the email;
3. Remove some specific characters, like space, slash; some spammers intentionally add these characters to interfere the spam filter.

After above processing, the email contents are input into the segmentation system, and then get the result words. Figure 3 shows the workflow of this part.

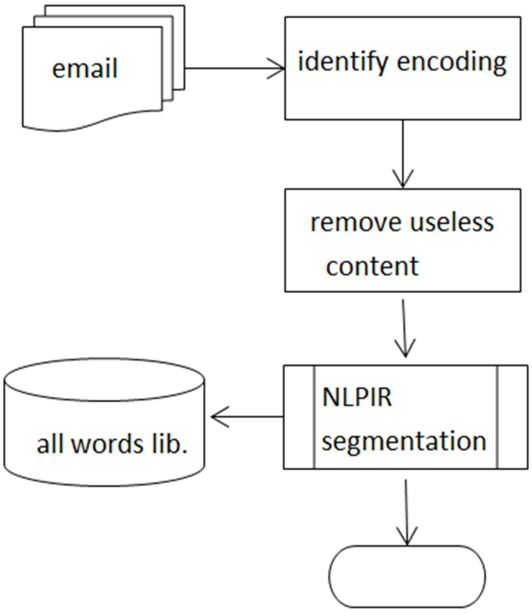

Figure 3. *Workflow of segmentation model.*

3.2. Keyword Extraction Model

Among the massive words generated by segmentation, there are only some keywords which will be helpful for spam filtering. So we need to extract the keywords from all words library. In this study, we select keywords according to the probability of these words appear in the spam, i.e.

$$\frac{\textit{times appeared in spam}}{\textit{times appeared in all email}} \quad (2)$$

At the same time, we need to avoid selecting some meaningless words as keywords, such as yes, no. These words frequently appear in all emails, but they are not helpful for filtering the spam.

According to the table of keywords, every email is converted to a feature vector. In the vector, every column is 0 or 1, which means the corresponding keyword appearing or not, see Table 1 and Table 2.

Table 1. *Keywords table.*

word	probability
free	0.23
credit	0.22
order	0.20
discount	0.18
…	…

Table 2. *Feature Vector table.*

word	email 1	email 2	email 3
free	1	0	0
credit	0	0	0
order	1	1	0
discount	0	0	0
…	…	…	…

3.3. BPNN Filtering Model

As described in 2.2, we introduce a GA-BPNN as our spam filtering model. We use three layers network, with 30 input nodes, 10 hidden nodes, and 1 output node.

At the training stage of the BPNN, we use GA to optimize the weights of the BPNN, as shown in Figure 4.

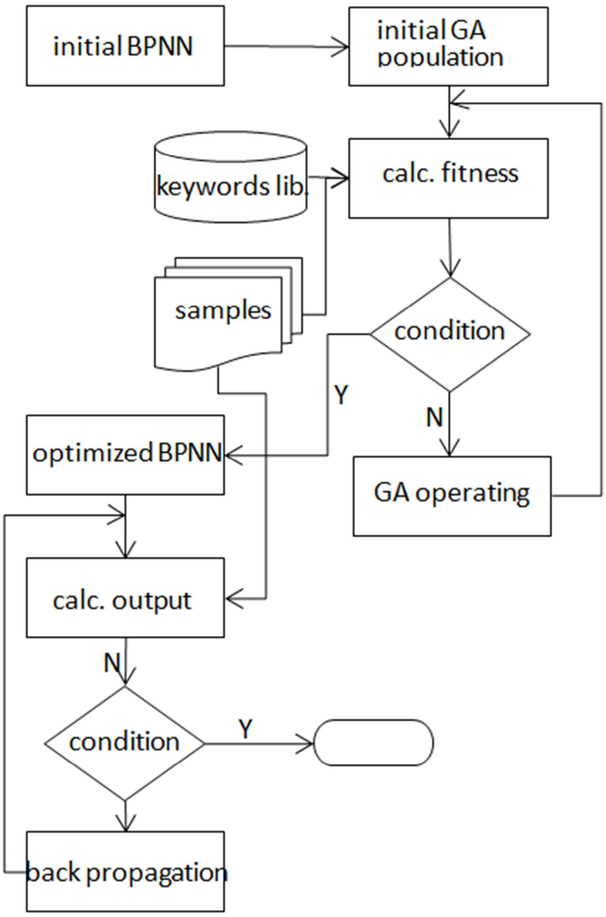

Figure 4. *Workflow of the training stage.*

3.4. User Feedback

In practical application, we should add user feedback model, to allow user to reconfirm whether an email is spam or not. And the filter could readjust the weights of the BPNN, i.e. learning in practice.

Figure 5 shows the framework of the whole system.

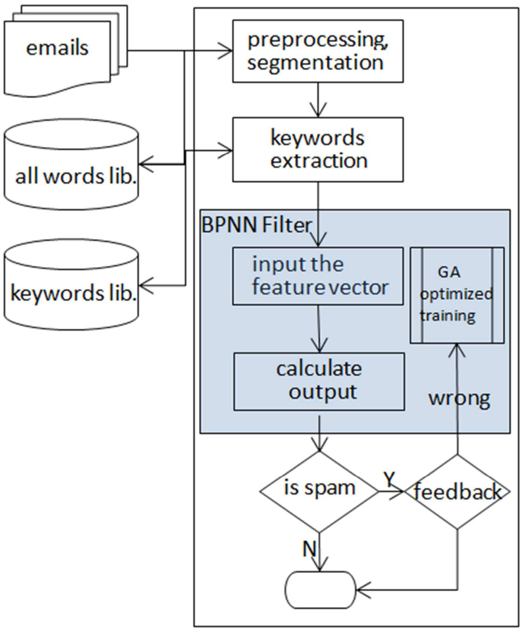

Figure 5. *Framework of the system.*

4. Experiments

In this section, we implement our GA-BPNN spam filter in Java language, and evaluate the effectiveness of our proposal.

4.1. Experimental Settings

We use the open source project JOONE (Java Object Oriented Neural Engine) [12] as the implementation of the BPNN. The parameters of the JOONE are:

1. For input layer, using LinearLayer, no transformation function; for hidden layer and output layer, using Sigmoid function as the transformation function, i.e. SigmoidLayer;
2. Using three layers network, with 30 input nodes, 10 hidden nodes, and 1 output node;
3. For links of three layers, using FullSynapse.

When applying the Genetic Algorithm, we use another open source project JGAP (Java Genetic Algorithm Programming package) [13]. The settings of the JGAP are:

1. Using the vector of the BPNN's weights as the chromosome, see Formula 1;
2. Generating an initial population by random;
3. Using the mean square error of the BPNN output as the fitness;

For the segmentation, we get the NLPIR (ICTCLAS 2013 version) system. With JNI framework, we can call the APIs of the NLPIR system to do the segmentation.

The testing sample emails, we use the email set collected in 2005 by the CCERT. We choose about 6,000 emails, include spams and normal emails.

4.2. Results and Analysis

At first, we used all the samples in the email set, about 30,000 emails, to train our GA-BPNN filter. The segmentation system completed its work and got the keywords table quickly. But the convergence speed was very slow; it took about 16 hours, and didn't complete the training. So we decreased the number of samples from 30,000 to 10,000, the training work was finished well, and the training time was acceptable.

And we did the test in two ways: one with GA optimized filter, another with only BPNN filter. Table 3 shows the testing results with GA optimized filter:

Table 3. *Testing results with GA optimized.*

training samples	Training time	Testing samples	Recall rate	Precision rate
10,000	120mins	200	93%	100%
6,000	50mins	200	87%	100%

And Table 4 shows the testing results with only BPNN filter:

Table 4. *Testing results with only BPNN.*

training samples	Training time	Testing samples	Recall rate	Precision rate
10,000	210mins	200	88%	100%
6,000	80mins	200	78%	100%

As we can see from the testing results, the filter is able to precisely detect the spam after trained by an appropriate sample sets. With the GA optimized, the filter was trained quickly, and it had a higher precision rate. But its recall rate isn't ideal in both; we think the reason is the limitation of the training samples. This limitation result in the narrow scope of the keywords, and then affects the recall rate of the filter.

5. Conclusions

In this study, we propose a GA-BPNN-based spam filter model for Chinese emails, and introduce NLPIR system for the segmentation. Then we implement our model in Java, by using some open source project, and get some experimental results. The experiments demonstrate the effectiveness of our model, also some need improved aspects. In the future, we will optimize our BPNN structure, and Genetic Algorithm, to improve the performance of our model.

In practice, the task of spam filtering needs to combine multiple techniques, include content-based and filter-based. Only a comprehensive spam filtering system can fight off the spammer.

References

[1] https://en.wikipedia.org/wiki/Email_spam.

[2] http://www.informationweek.com/spam-costs-billions/d/d-id/1030111.

[3] Ismaila Idris, Ali Selamat and Sigeru Omatu, "Hybrid email spam detection model with negative selection algorithm and differential evolution", Engineering Applications of Artificial Intelligence, Volume 28, February 2014, pp. 97–110.

[4] Ismaila Idrisa, Ali Selamat, "A combined negative selection algorithm–particle swarm optimization for an email spam detection system", Engineering Applications of Artificial Intelligence, Volume 39, March 2015, Pages 33–44.

[5] Atefeh Heydaria, Mohammad ali Tavakolia,, "Detection of review spam: A survey", Expert Systems with Applications, Volume 42, Issue 7, 1 May 2015, Pages 3634–3642.

[6] M. Sahami, S. Dumais, D. Heckerman and E.A. Horvitz, "Bayesian approach to filtering junk email", Proc. of AAAI'98 Workshop on Learning for Text Categorization, Madison, WI, July (1998), pp. 55–62.

[7] X. Carreras and L. Marquez, "Boosting trees for anti-spam email filtering", Proc. of Fourth Int. Conf. on Recent Advances in Natural Language Processing, Tzigov Chark, Bulgaria, September (2001).

[8] Zhao Wenqing and Zhang Zili, "An email classification model based on rough set theory", (AMT 2005). Proceedings of the 2005 International Conference on Active Media Technology.

[9] J. Clark, I. Koprinska, J. Poon, "A neural network based approach to automated email classification", Proc. of the IEEE/WIC Int. Conf. on Web Intelligence (WI'03) (2003).

[10] M. M. Fuad, D. Deb, M. S. Hossain, "A trainable fuzzy spam detection system", Proc. of the 7th Int. Conf. on Computer and Information Technology (2004).

[11] http://ictclas.nlpir.org/docs

[12] https://sourceforge.net/p/joone/wiki/Home/

[13] https://sourceforge.net/p/jgap/wiki/Home/