Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230617

Spam mail classification using back propagation neural networks

Jingfeng chen

School of micro-electronics, Fudan University, Shanghai, 200011, China

19300750002@fudan.edu.cn

Abstract. Mail classification methods based on machine learning have been introduced to combat spams. However, few researches focus on the most powerful machine learning model that is neural networks. In this paper, the author trains BP neural networks to detect spams. The inputs of the neural networks are only information about words, punctures, signs, numbers and illegal words. Five neural networks which are different in number of neurons and number of layers are experimented on. All networks apply Rectified Linear Unit (ReLU) functions and Momentum learning technology. The results show that the network with four hidden layers enjoys the best classifying accuracy of 97.0%. In networks with two hidden layers, when the number of neurons in each layer is above 300, the accuracy is between 95.5% and 96.0%; and 100 neurons in each layer result in an accuracy of 93.8%. Although the training only captures information of words, punctures and signs, the networks have achieved high accuracy, and the author suggests that making the computer understand sentences as well as other kinds of improvements can lead to even higher performance.

Keywords. BP neuron networks, spam detection, classification.

1. Introduction

With the booming of technology, electronic mails have been becoming increasingly important in the daily lives. People send and receive electronic mails on a regular basis. Although this invention has made humans' lives far more convenient, malignant mails, or also called 'spam', can be annoying and even dangerous. A case in point is SMS mail attacks which involves spammers stealing information from victims by sending SMS primers to the link or through direct communication [1]. Tackling spams has also been a problem.

Nowadays programs based on machine learning models have been applied to combat spams. Among them there are some programs based on neural networks. Although in this field past researches have reached high accuracy in classifying mails that is usually over 97%, almost all of these researches actually combine neural networks with other machine learning methods. In one past research, the combination of Convolutional neural networks and long short-term memory technology not only enables the classification system to reach an accuracy between a fine 92.64-98.48%, but also trains the system to combat some tricks used by spammers [2]. And most of the researches capture not only word features but also the information of sending time and the information of senders. Nearly no papers have focused on the limit of the accuracy of neural networks when the program only capture words information and does not try to understand any sentence at all, and nearly no papers have researched on the limit of performance of the simplest kind of networks that is back propagation (BP)

DOI: 10.54254/2755-2721/5/20230617

neural network, and whether the number of layers and the number of neurons can influence the performance of mail classifying neuron networks. Thus, in this paper, the author wants to test the limit of the neuron networks' ability in deducing the intention of mails only by acquiring words' information and using BP neural networks.

Focusing on the link between neural networks and mail classification is meaningful because the development of chip designing technology in recent years has been making neural networks far more powerful than before. Chips like graphics processing units (GPU) and tensor processing units (TPU) and field programmable gate arrays (FPGA) are good at parallel calculations. In other words, they can conduct thousands of calculations of the same type at the same time. That is because unlike CPU, there are thousands of "working spaces" that are exactly the same on such chips and in a certain clock period of the chip, each calculation just occupies one of the working spaces until the clock period ends. Figures have shown that in floating-point processing speed, GPUs have outperformed CPUs greatly [3]. The power of parallel calculation can be best demonstrated in the dot multiplication of matrixes, and such calculation of matrixes is exactly the essence of neural networks. Nowadays, chip designing technology is still improving and new GPU and FPGA models emerge every year. Therefore, the technology of neural networks is of great potential.

Also, neural networks may obtain special ability in tackling the anti-detection technology used by spam senders. Spam senders try to make mails avoid detection systems mainly by the technology of Bayesian sneaking and poisoning, altering Internet protocol addresses, adding falsified header information, obfuscation, Hypertext Markup Language (HTML) manipulation, HTML encoding and so on [4]. These methods have threatened most of the traditional classifying models more or less, a case in point is 'Bayesian sneaking and poisoning', which makes the most traditional machine learning method that is Bayes classification out of order [5]. The reason why traditional systems are ruined is that these systems are not complex enough, and the functioning of them may be too 'human-like'. On the other hand, neural networks focus on the deepest essence of the human brain that is the system of neurons, and even the simple neural networks in this paper contain millions of links between neurons. Thus, neural networks are hard to ruin. Research which involves the testing of four mail datasets and the comparison between neural networks, decision trees, random forests and naïve Bayes shows that neural networks perform perfectly and steadily in combating spams, while the flaws of the other methods are exposed [6].

In this paper, the author uses back propagation (BP) neural networks to train the program of classifying electronic mails. When classifying, only the feature of words, the numbers of punctures and signs are acquired. The programs do not try to understand English sentences at all. The author uses datasets produced by Carnegie Mellon University. In the datasets, there are 3672 hams and 1500 spams. As for neural networks, the author builds five different BP neural networks which are different in number of layers and neurons. The goal is to test the limit of accuracy and to compare the power of different neural networks.

2. Preparation before the training the networks

2.1. The necessity to help the computer learn English words

To complete a task or to win a game, a human or a computer should have a clear vision of the rules. Take the Alphago as an example. Although the most sophisticated Alphago deep learning model called Alphazero learns the game without any interference of human, the rules of Go is planted in the Alphago at the beginning because the game is designed by human and the rules are written by human [7]. Similarly, the neural networks in this paper should 'know' something about English words before training.

In electronic mails, not all the words are correct words. Some contain spelling mistakes, and others are not English words at all. According the author's rough inspection, the 'words' that are not words at all are mainly in spams, and it is deduced that those strange "words" may be added to ruin the naive Bayes mail classification. And some hams contain spelling errors. Therefore, humans should tell computer what is an English word and what is not an English word.

DOI: 10.54254/2755-2721/5/20230617

2.2. To learn words from textbooks

To help the computer learn English words rapidly, electronic versions of English textbooks or English newspapers can be introduced. In this paper, the author introduces electronic versions of the New Concept English textbooks. They were obtained by the author himself. The purpose of learning words from textbooks is to acquire most of the frequently used English words. It does not matter if programmers do not have appropriate electronic English textbooks because copying articles from English newspaper websites can achieve similar effects.

Collecting all English words is not enough for the training of the classification network. That is because in English, the first letter of the first word is upper-cased; and some nouns have plural forms, verbs obtain past tense and present tense, and most adjectives can be turned into adverbs simply by adding "ly". The first question is simple. The author turns every letter in every word into lower case. (But this may cause some problem which will be mentioned in the discussion section) As for the second problem, because the author pays no attention to grammar errors, a word and a word plus an 's' or an 'es' are regarded as the same word by the program. Similar things happen when it comes to verbs and adjectives. In this way, the word dictionary in the computer can be more concise. What is not perfect is that irregular transformations are not interfered by the author. Words like 'cat' and 'cats' are regarded as the same word, while 'goose' and 'geese', 'go' and 'went', 'put' and 'putting' are judged as different words.

2.3. To learn words from mails in the datasets

Although 'reading' textbooks is an effective way for computers to learn English words, the textbooks cannot cover all words in daily lives. What's more, some 'words' may be the suffix of computer files, a part of the name of a website, or the name of an individual or a company. It is hardly possible to learn these special words via textbooks or news articles. Therefore, the author set up another program in which the computer 'asks' question about new English words. In this python program, the user inputs a number x. And then the computer read mails in the datasets randomly without knowing whether the mail it is reading is a ham or a spam. In the process, it picks out about 10 words that have lengths of x and display them on the screen. The user checks the contents of the screen one by one. If the 'word' is a correct word, he or she should press '1', and if the 'word' is not a word, the key '2' should be pressed. When the running of the program is finished, all the words on the screen will be classified as 'words' and 'not words' and added into two memory files. When the program is run again, words that have emerged in the two memory files and the word dictionary from textbooks will not be regarded as 'new words' by the computer.

2.4. The total number of English words the computer has learned

Up till the experiments are finished, the program above has learned 1281 words. And before learning these words, the computer has successfully learned 3579 words. Therefore, the computer has learned 4860 English words before the training of neural networks.

2.5. Numbers

It should be noticed that not all elements in a ham or a spam are words. There are numbers of prices, telephone numbers, dates and so on. When the program reads a word, it can examine whether all components are digit numbers. If all components are number digits, the program will regard the word as a number. The program will calculate the number of numbers in the mail.

2.6. Signs and punctures

Commas, full-stops, question marks and exclamation marks are also common elements in mails, no matter it is a spam or not. And there are other signs such as '#' and '$'. Collecting them is quite easy. Since all marks are separated by spaces in the datasets, all the program needs to do is to collect 'words' that have a length of one. Those 'words' can be slotted according to the ASCII code. As ASCII code has 128 elements, a list that have a length of 128 can be created to collect signs and punctures.

DOI: 10.54254/2755-2721/5/20230617

2.7. Illegal words

Some ‘words’ in mails do not belong to any of the following: correct English words, English names, names of companies, suffix of files, signs, punctures and numbers. They are called ‘illegal words’ in this paper and they usually emerge in spams. Before training, the total number of ‘illegal words’ is counted.

3. The BP neural networks in this paper

3.1. The input layer

There are 3579 words learned from the textbooks and they contribute 3579 neurons in the input layer. The values of the neurons are the counting numbers of each word. In this paper, 1500 neurons are prepared for the words learned from mails. In fact, only 1281 words are collected in this section. Therefore, the values of the first 1281 neurons are the counting number of each word from the ‘words learned from mails’ section. The rest 219 neurons are permanently zero; in other words, they do not have any effect in the neural network. The numbers of signs and punctures are also contributed to the input layer. As there are 128 ascii codes, there are 128 neurons in this section. The values of the neurons are the counting numbers of each sign or puncture. The other two neurons are the counting number of numbers and the counting number of illegal words. Totally, there are 3579 + 1500 + 128 + 1 + 1 = 5209 neurons in the input layer.

3.2. The hidden layers

Hidden layers act as bridges in BP neural networks. To enable neural networks, express non-linear models, there should be non-linear elements in hidden layers. The non-linear elements in the hidden layers in this paper are all Rectified Linear Unit (ReLU) functions except the one connected to the output. The ReLU function is ruled as: f(x)=x if x>=0; f(x)=0.05x if x<0; df(x)/dx at x=0 is 1. This function is used in recent years in the field of machine learning and deep learning [8]. Although the performance of ReLU may be inferior to the sigmoid function, the author uses it to save the training time and test the potential. The hidden layer connected to the output is another function is ruled as: g(x)=x if -10<x<10; g(x)=0.02(x+10)-10 if x<=-10; g(x)=0.02(x-10) +10 if x>=10; df(x)/dx at x=10 or x=-10 is 0.02. The hidden layers, the input layer and the output layer are connected in the same way as regular BP networks do. One of the neural networks in this paper has two hidden layers and each layer contains 100 neurons. The structure is shown in Figure 1.

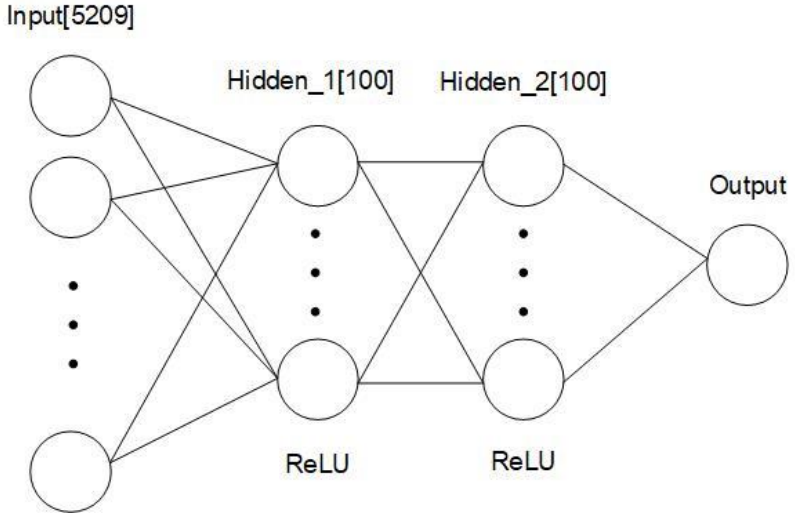

Figure 1. The structure of one BP neural network in the paper.
(Photo credit: Original)

DOI: 10.54254/2755-2721/5/20230617

3.3. The output layers

The output is a single real number. Values which are equal to or greater than zero indicate that the input is a 'ham', while values smaller than zero is related to 'spams.

3.4. Training

80% of the mails are used as the training set, and the rest 20% is the test set. In the training process, the computer chooses a mail randomly from the training set. And then it turns the mail into input that has 5209 dimensions and after that the output is calculated. The ideal value of the output for a spam is -10, and for a ham, it is 10. The error is the gap between the ideal value and the output, and then back propagation is conducted.

Five different BP neuron networks are trained in this paper. Four of them are networks with two hidden layers, the two hidden layers have the same number of neurons and the numbers are 100, 300, 500 and 900 in the four networks. The other network contains four hidden layers and each hidden layer contains 100 neurons. The structure of network 1 is shown in Figure 1, and the structures of the other four networks are shown in Figure 2 to Figure 5. In all neural networks, the learning rate is 0.00048, and the technology of Momentum is applied in the back propagation. All networks are trained 40000 times in the training procedure.

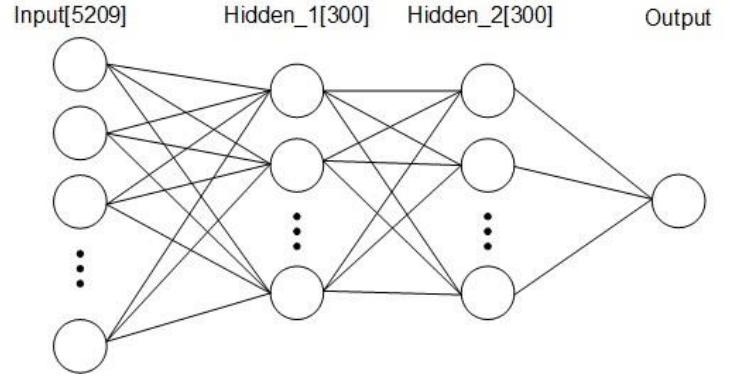

Figure 2. The structure of network 2.
(Photo credit: Original)

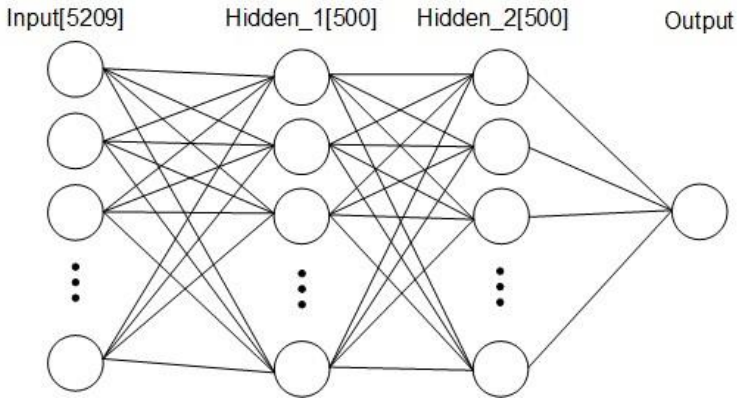

Figure 3. The structure of network 3.
(Photo credit: Original)

DOI: 10.54254/2755-2721/5/20230617

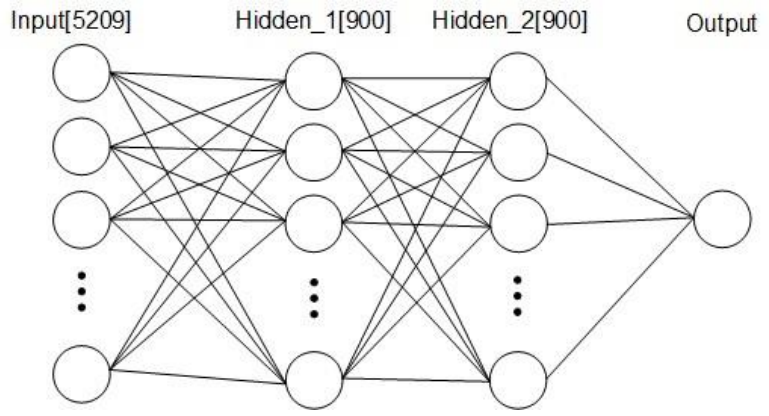

Figure 4. The structure of network 4.
(Photo credit: Original)

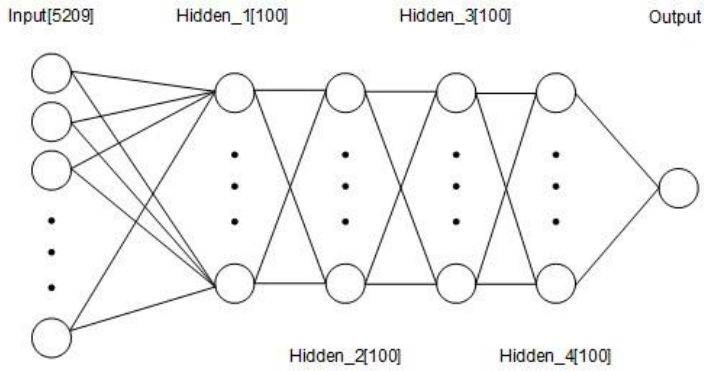

Figure 5. The structure of network 5.
(Photo credit: Original)

3.5. The evaluation of the BP neuron networks

In this paper, 'a mail is a spam' is defined as positive, and 'a mail is not a spam' is defined as negative. True positive (TP) means that a spam is regarded as a spam by the neural network, and false positive (FP) means the result of the neural network shows a mail is a spam but the mail actually is not. Similarly, true negative (TN) and false negative (FN) can be defined.

Two indexes determine the score of the network. One of them is called 'spamPt' in the related python program. It equals to 100*TP/(TP+FN) and it shows the ability of the neural network to detect spams. A score of 0 means the network fails to detect any spam, while a score of 100 means all spams are correctly regarded as spams. The other index is called 'hamPt' in the related python program. It equals to 100*TN/(TN+FP) and it shows the ability to make hams pass the spam detecting system. A score of 100 means that all hams are regarded as hams and an individual can receive all hams sent to him or her, while a score of 0 means that all hams are regarded as spams; in other words, all hams sent to the individual are blocked. These two indexes are calculated by checking all mails in the test set, and the final score is the minimum of the two indexes.

4. Assumptions

Among neural networks with two hidden layers, the more neurons each layer obtains, the better the total score will be. That is because in neural networks, adding neurons enables the network to form more complex functions and to deal with more complicated situations. The last model obtains more layers. Considering that the most advanced AI models like the Alphago feature deep learning, the author makes an assumption that the network with 4 hidden layers can perform the best because the former model is 'deeper'. The estimation of the highest accuracy of the five networks is 92%. It is

DOI: 10.54254/2755-2721/5/20230617

about 6% lower than those top classification systems in the world because the system in this paper does not capture all useful information of mails and does not understand English sentences [3].

5. Results

The experiment is conducted on an CPU of i7-8565u using python programs. Every time the training is conducted 4%, the training pauses and the neural network is tested and scored. In each training procedure, there are 25 scores which are related to 4% of the training, 8%, 12%, and up to 100%. Figure 6 to Figure 9 show the result of the experiments conducted on network 1 to network 4. And table 1 shows the relationship between the number of neurons in each layer and the highest score of each neuron network.

The results show that in neural networks which have two hidden layers of the same number of neurons, the enhance of neurons in each layer can lead to the improvement of classifying accuracy. Such improvement is fairly obvious when the number of neurons is small, but not significant at all when the number of neurons in each layer has reached 300.

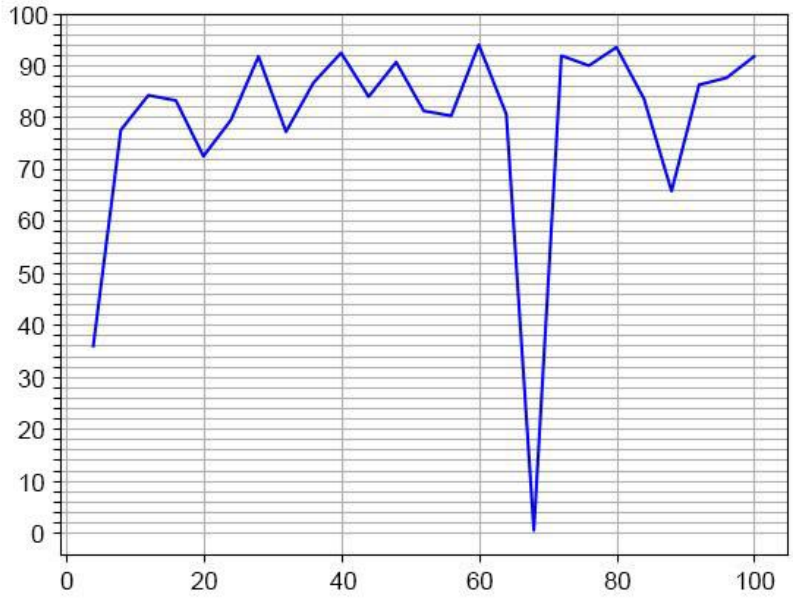

Figure 6. The testing performance of network 1. (100 neurons in each layer, 2 layers)
(Photo credit: Original)

DOI: 10.54254/2755-2721/5/20230617

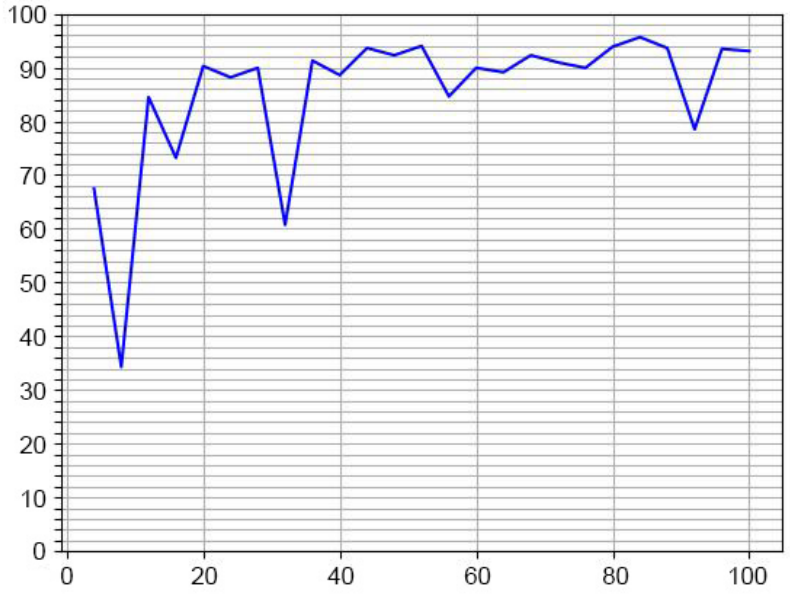

Figure 7. The testing performance of network 2. (300 neurons in each layer, 2 layers)
(Photo credit: Original)

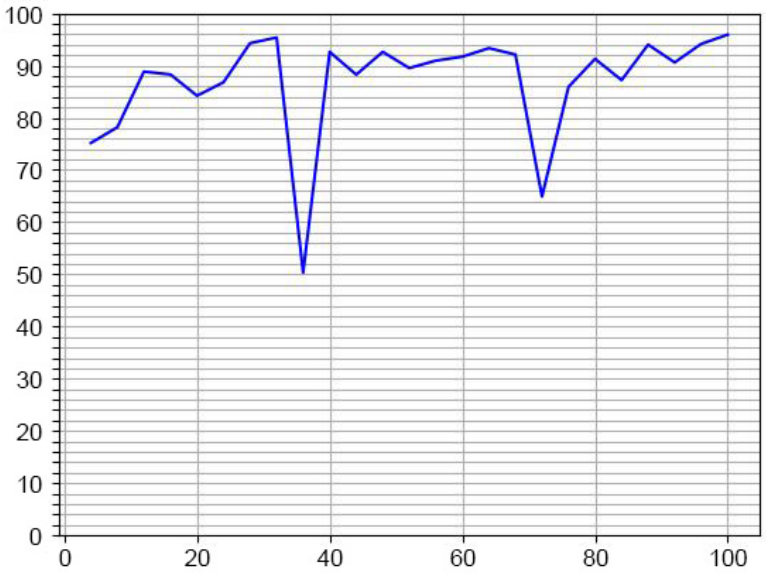

Figure 8. The testing performance of network 3. (500 neurons in each layer, 2 layers)
(Photo credit: Original)

DOI: 10.54254/2755-2721/5/20230617

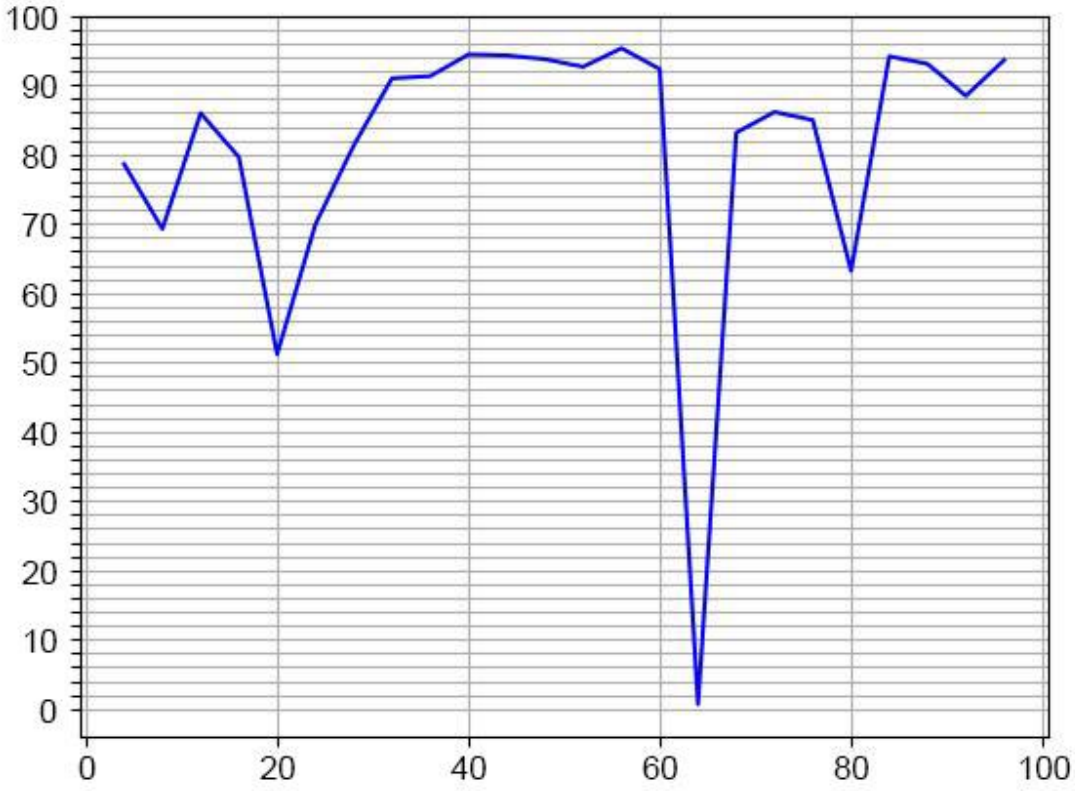

Figure 9. The testing performance of network_4. (900 neurons in each layer, 2 layers)
(Photo credit: Original)

Table 1. The testing performance of 4 networks with 2 hidden layers.
(Table credit: Original)

Number of neurons in each layer	Highest score
100	93.8
300	95.8
500	95.9
900	95.8

Figure 10 shows the performance of the only network that obtains 4 hidden layers. In this neural network, a score of 97.0 is reached, and this neural network enjoys the highest accuracy. Interestingly, almost all lines in the testing graphs have undergone one or two plummets. The accuracy falls greatly to 50% or so, and sometimes, about 20%.

DOI: 10.54254/2755-2721/5/20230617

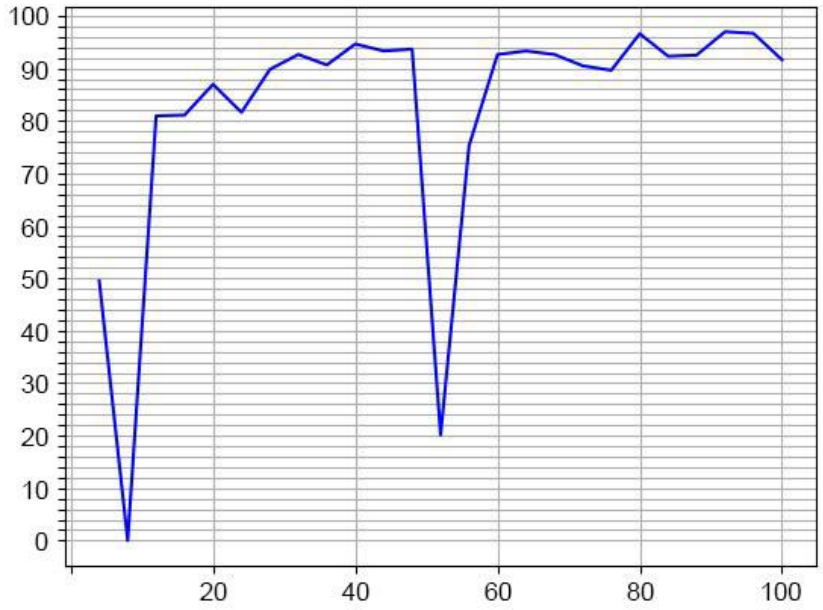

Figure 10. The testing performance of network_5. (100 neurons in each layer, 4 layers)
(Photo credit: Original)

6. Discussions

The assumption that the increase of number of neurons in each layer can lead to better performance is correct. The fact that the enhancement in performance is not significant when the number of neurons in each layer reaches implies that the complexity of the neuron network has reached a limit. Only by taking other measures can the classifier reach a higher level. It is also true that deeper networks work better, and surprisingly, the network_5, which has only 400 neurons in the hidden layers can win the network_4, which has 1800 neurons in the hidden layers. It seems that increasing the number of layers is more effective in enhancing the accuracy than adding the number of neurons in each layer. Nevertheless, it should be mentioned that the increase in the number of layers has hardened the training process. When the network gets deeper, the author has undergone more failures in the training process. All the weights become 'nan' (not any number) because of overflow in matrix multiplication and the neural network breaks out permanently. This is a common situation because when errors go through a great number of layers, usually 8 to 9 layers, they may diverge and the network will not be able to reach a steady state any longer [9].

The plummets in the accuracy graph in testing origin from the training process. In each training, the computer selects a mail in the training set randomly, and in some occasions, although the network can classify most of the mails correctly, the computer chooses some strange mails in a row, and the neural network judges almost all of them wrong. This leads to major changes in the weights of the networks, and the change of the weights make the network misjudge mails that has been judged correctly before. Nevertheless, because the selection of training mail is at random, the accuracy will recover and it might be possible that the neural network has got rid of a local minimum and has successfully entered a better local minimum. As long as the training times is great enough, it is quite likely that the network moves among various local minimums.

In this paper, the accuracy of the network has not reached the top level in all mail classifying programs. This is because the neural networks only capture words information and do not care about the meaning of the sentences. To tackle this flaw, a machine learning program which tries to understand English sentences roughly can be introduced. This program may try to turn complex English sentences to key words, and it may also try to judge if there are significant syntax errors in the

DOI: 10.54254/2755-2721/5/20230617

sentences. Another method is to use convolutional neural networks (CNN). Such networks are most commonly used in image classification because two-dimension convolution functions are of great power. Nevertheless, CNN can also be of one dimension and some researches have focused on the CNN networks in mail classification [10]. CNN is of great potential because this kind of function not only focuses on a single element but also lays emphasis on the elements beside it as well as the links between elements.

Another flaw in this word classifying system is about the processing of words information. As is mentioned before, the classifying system have no command of irregular plural transformations and tense transformations. It does not know that 'geese' and 'goose', 'went' and 'go' have the same meanings. Even worse, due to the flaws in the python programs written by the author, some easy transformations such as 'wolves' and 'wolf', 'putting' and 'put' are not known by the system either. To solve this problem, extra codes should be written. In addition, the vocabulary dictionary still has some potentials; it can be enlarged through efforts of humans. Moreover, there may be too many English names in the words dictionary. Actually, changing all 'Jack's into 'John's or changing all 'Alice's into 'Anna's has hardly any effect in mail classification. To improve the system in this aspect, English names can be put into a group and when the system reads a mail, the system can regard English names as English names instead of daily English words.

Also, it is not certain whether skills such as mixing bad contents into normal texts can spoil the classifying system. A paper that focuses on detecting spams based on reading part of a mail sheds light on me [11]. In this paper, the classifying system tries to find some critical parts that can determine the mail is malignant. If such parts are found, the system will be more likely to conclude that the mail is a spam.

7. Conclusion

In this paper, BP neural networks are used to tell spams from hams. Only by collecting information of the words, 'wrong words', signs, punctures and numbers, the classifying system reaches an accuracy of a satisfying 97.0% with the help of a neural network which has four hidden layers. When the classifying system use neural networks that obtain only two hidden layers instead of the one with four hidden layers, it enjoys an accuracy of 95.8%. By comparing five different networks, the author discovers that increasing the number of layers is more effective in enhancing the accuracy than adding neurons in a single layer, but the increase in layers make the training more possible to fail. Increasing the number of neurons in each player can also lift the accuracy up more or less. With the help of Momentum technology and random mail selection in the training process, the BP neural networks can get rid of local minimums and adjust themselves to new local minimums although the performance may be very low temporally.

Through this research, a creative method to classify mails that only involves capturing information of words, 'wrong words', punctures, signs and numbers is introduced. It lays great emphasis on the essence of the English language that is words and may produce a new idea to combat spams.

It is true that the classifying in this paper enjoys fine accuracy and great potential, but the programs in this system still have some flaws. To achieve further improvement, the author may try to set up another AI program, enlarge the vocabulary dictionary in the computer and make it more concise, increase the number of layers and neurons of neural networks, and try to build CNNs.

References

[1] Syed Md. Minhaz Hossain, Jayed Akbar Sumon, Anik Sen, Md. Iftaker Alam, Khaleque Md. Aashiq Kamal, Hamed Alqahtani, Iqbal H. Sarker (2022). Spam Filtering of Mobile SMS Using CNN–LSTM Based Deep Learning Model. Hybrid Intelligent Systems (pp.106-116).

[2] Hong Yang, Qihe Liu, Shijie Zhou, Yang Luo (2019). A Spam Filtering Method Based on Multi-Modal Fusion. Applied Sciences, Volume 9, Issue 6.

[3] Jim X. Chen (2016). The Evolution of Computing: AlphaGo. Computing in Science & Engineering, Volume 18, Issue 4, Pages 4-7.

DOI: 10.54254/2755-2721/5/20230617

[4] Thamarai Subramaniam, Hamid A. Jalab, Alaa Y. Taqa (2010). Overview of textual anti-spam filtering techniques. International Journal of the Physical Sciences, Vol. 5(12), pp. 1869-1882.
[5] Pradeep Kumar Roy, Jyoti Prakash Singh, Snehasish Banerjee (2019). Deep learning to filter SMS Spam. Future Generation Computer Systems, Volume 102, January 2020, Pages 524-533.
[6] Tingmin Wu, Shigang Liu, Jun Zhang, Yang Xiang (2017). Twitter spam detection based on deep learning, ACSW '17: Proceedings of the Australasian Computer Science Week Multiconference, Article No. 3, January 2017, Pages 1-8.
[7] Ivan Bratko (2018). Alphazero – What's Missing, Informatica, Vol 42, No 1 (2018), Bratko
[8] Koki Saitoh (2016). Deep Learning from Scratch, 2016 O'Reilly Japan, Inc, ISBN 978-7-115-48558-8.
[9] Zhihua Zhou (2016). Machine Learning, ISBN 978-7-302-42328-7
[10] Sunita Dhavale (2020). C-ASFT: Convolutional Neural Networks-Based Anti-spam Filtering Technique, Proceeding of International Conference on Computational Science and Applications, pp 49-55.
[11] Chao Wang, Qun Li, Tian-yu Ren, Xiao-hu Wang, Guang-xin Guo (2021), High Efficiency Spam Filtering: A Manifold Learning-Based Approach, Mathematical Problems in Engineering, Volume 2021, Article ID 2993877