Multilayer Perceptron Based Spam Detection with Regularization

Yujie Qiu

Abstract. Spam email is a kind of junk email. Recent years, with the rapid growth of the internet users, especially emails users, the spam emails have been regarded as a severe problem. there are many classification methods that can be used to detect spam, Naïve Bayesian and Decision tree for example. These methods all gains good performance in most of cases, but the true positive rate and the false positive rate of them are not good enough. In this paper we designed a neural network based spam classification algorithm to filter spam. Our model is a classical multi-player perceptron composed of two hidden layer, one input layer and one output layer. By applying different threshold to plot the roc curve, we demonstrate that our method outperforms most of existed method. We also demonstrate that ensemble learning will boost the whole method.

Keywords: Multi-layer perceptron, spam, ADAM, ROC

1 Introduction

Spam email is a kind of junk email. Most of spam emails are commercial advertisements or contain disgusting contents like violence or porn. Some of them may contains viruses that might hurt receivers' computer. Recent years, with the rapid growth of the internet users, especially emails users, the spam emails have been regarded as a severe problem. Internet users' normal life has been effected. Under that cases, developing a technique to filter the spam has been a more and more important topic. In most of cases, spam can be filtered by a black list. By rejecting to receive email from specific addresses that have been annotated as the main source of spam, the number of spam in our mail box will decrease. This method only works under the condition that the source of spam is limited. However, when the spam sender builds a huge group of computers from allover the world that being controlled by trojans horse, it is no longer realistic to ban all of them. Another method being applied to solve the spam problem is to filter email based on the content of it. The content of spam has its own pattern that separate it from normal email, by extracting feature of email's contents, and reject to receive emails that has these features has been the main technique to filter spam. The main algorithm of extracting features are the bag of word algorithm and the TF-IDF algorithm [1], that utilized the frequency of terms to represent one email. With these features extracted, there are many classification methods that can be used to detect spam. Naïve Bayes algorithm [2, 3] is one of the most popular method, by applying Bayesian formula to calculate the posterior of that being spam for one email, this algorithm has gain good performance in the past years.

Another method is the decision tree method [4, 5] that classify spam with a tree structure. These methods all gains good performance in most of cases, but the true positive rate and the false positive rate of them are not good enough. In this paper we designed a neural-network based spam classification algorithm to filter spam. Our model is a classical multi-player perceptron [6] composed of two hidden layer, one input layer and one output layer. The output layer of our model will give the probability of an email to be spam. But applying different threshold just as [7] proposed, we demonstrate that our method outperforms most of existed method (classical actually). In section 1, we will illustrate our method, and in the next section we will show the performance of our model on a spam data set, and comparing it to several main techniques and boost the model performance with ensemble learning method, in the last session we will discuss further work of our model.

2 Method

2.1 Multi-layer Perceptron

Decision tree and naïve Bayesian methods has shown promising performance since the last several decades. But their performance is not good enough. On the other hand, in order to gain good performance, we need some hand crafted features. Recent years, Neural network has been proposed as a promising model to learn a classifier or a regressor for its ability to fit a very complex model. Here we design a neural network with two hidden layer, and the architecture of our model is shown in figure 1.

2.1.1 Feed Forward Process

Multi-layer perceptron will transform the input into a complex feature space by the linear combination of input vector. The core point for the neural network to lean ingenious is the activation function:

$$f(x) = \frac{1}{1 + e(-x)} \quad ,(1)$$

in which x is the linear combination of this layer.
But by applying this activation function, the gradient vanishing problem [10] might occurs, so in our model, we adopt the RELU function [9]:

$$f(x) = \max(0, x), \quad ,(2)$$

This function punishes all the negative value to be zero, and this activation function will give us linear gradient during the training process.
With the activation function shown above, we get the feed forward process as:

$$p = f(w^{(2)} f(w^{(1)} x + b^{(1)}) + b^{(2)}). \quad ,(3)$$

2.1.2 Regularization

In most of cases, the training process of neural network will make the model fit training data perfectly, but the variance of model will be very high, which means the model was over fitted to the data, so here we adopt the L2 norm [11] of parameters to the neural network, that will make our cost function as:

$$J(\theta; x, y) = \frac{1}{2} \sum_{i=1}^{N} | y_i - \bar{y}_i |^2 + \frac{1}{2} w^{(1)T} w^{(1)} + \frac{1}{2} w^{(2)T} w^{(2)}. \quad ,(4)$$

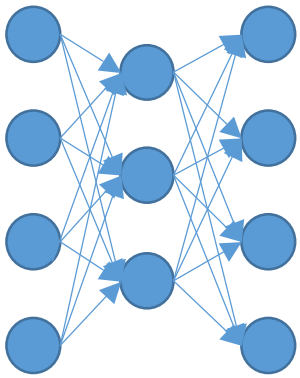

Fig. 1. Our model has two hidden layers, one of them has 90 neurons and one of them has 40 neurons. All of the neuron in the adjacent layer was fully connected just as this figure illustrated.

This operation was implemented with weight decay parameters in pytorch optimizer. And the weight decay parameter is used to tuning the importance of the regularization part and model bias part.
On the other hand, we also use dropout operation, preserving activation value the fixed probability, was proved to be equal to the L2 regularization [17,18].

2.2 Auto-encoder

In traditional machine learning, the depth of neural network can't be too deep for the effect of gradient vanishing [4], which strongly constraint the performance of multi-layer perceptron. Hinton proposed a deep belief net [15], and Vincent proposed a stacked autoencoder [16] to automatically extract the features from the data, and enhance the performance of classifier.

The formation of autoencoder is very simple, and it was composed of a encoder and a decoder. The decoder plays as an inverse operation of decoder.

$$\text{code}=f_1 \quad f_2 \quad f_3(X)$$

$$\bar{\text{X}}=f_3^{-1} \quad f_2^{-1} \quad f_1^{-1}(\text{code})$$

When we can recover the X from its code with small error, we can claim that the code contains most of information of the original data, on the other hand, the code is a good representation for the original feature. When we concatenate this autoencoder to another classifier, like SVM, we will gain a deeper model, which will show good performance.

2.3 Model Training

The learning process of the neural network is back propagation [8], based on the classical batch gradient descent algorithm. But the batch gradient descent algorithm has its own obstacle. For the saddle point problem, the batch gradient descent method will be fail to find the global optimal value. So in our method we adopt the ADAM algorithm, the update rules for the ADAM algorithm [12] will record two new parameters that serve as averaged mean of past gradient and variance of past gradient:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \end{aligned} \quad ,(\mathbf{5})$$

where β is the hyper parameters that portion between current gradient and past gradient. Without considering decay, the parameter update will be:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t} + \delta} m_t \quad ,(\mathbf{6})$$

where η is learning rate, δ is a hyper parameter that works in case variance v to be zero.

For the training loss, I use cross entropy to measure.

$$H(p,q) = -\sum_{x} p(x) \log q(x)$$

2.4 ROC Curve for Model Metric

In [7], the author set different threshold to find the best one, that will enhance the model, in our method, we use the AUROC (area under receiver operating characteristic curve) to evaluate the performance of a model. The ROC curve [13] will use True positive rate (TPR) and false positive rate (FPR) as:

$$TPR = \frac{TP}{TP + FN} \quad , \quad FPR = \frac{FP}{FP + TN} \qquad (7)$$

3 Result and Discussion

In order to measure the performance of our model, we compared it with naïve Bayesian and decision tree method on a classical SPAM dataset.

3.1 Data Set Description

The spam data set is created by Hewlett-Packard Labs, consist of 4601 samples that we downloaded from the UCI machine learning database. The number of attributes is 56, and most of attributes show that how frequent for a word or character to occur in a given email. And the remaining attributes measure the length of sequences of consecutive capital letters.

In order to measure the performance, we split the whole data set into two parts: 80 percent of it will be used to train model, 20 percent of it will be utilized to measure the performance of model.

Before make further analysis, we make a simple analysis to the feature covariance, and the result is shown in Fig. 2. This figure inspires us that the effective variable is not that much, and we can digest lots of insight from it.

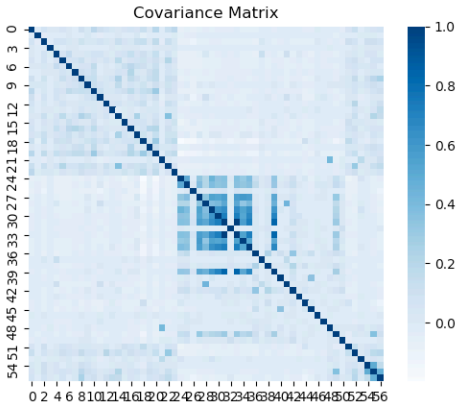

Fig. 2. Interaction between different features. Variables 24-39 are strongly co-vary.

3.2 Comparing Result

We comparing three different models, Naïve Bayesian, decision tree and Neural network machine, we also implement a neural network without regularization, the ROC curve of these models is shown below:

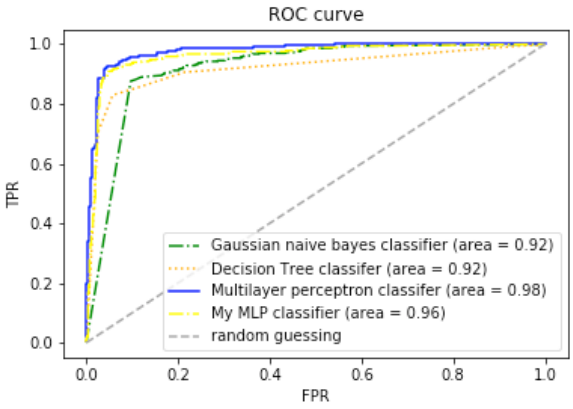

Fig. 3. Area under ROC curve of different model. This curve is plotted by applying different threshold to assign label. The MLP with regularization perform best.

From the figure we know that our neural network model outperforms all the existed method. By comparing the performance of MLP with and without regularization, we found that model with regularization has better performance on the test set. The reason is that regularization can be treated as a model selection, and the low variance model was selected, which makes the model has a better generalization ability.

We also compare our model with the existed method, like Maximum likelihood in [19] and [20], the result in shown in Tab 1. From the result we find that by employing the power of multi-layer perceptron, we can enhance the performance of optimal probability prediction.

Method	Gaussian Naïve Bayes	Decision Tree	Maximum Likelihood	**MLP**
Acc	81.7%	89.8%	88.7%	**93.9%**

Tab 1. Accuracy comparison between different method, we found that MLP has the better performance.

We also compares the autoencoder method and ensemble learning method [20], the result in shown in Tab 2.

Method	Autoencoder+SVM	**MLP with 32 experts**	Decision tree with 32 experts
Acc	90%	**94.5%**	91.6%

Tab 2. Accuracy comparison between autoencoder and ensemble learning.

3.3 Discussion

In this paper, we use multilayer perceptron to train a classifier for spam detection, instead of using different train threshold we use ROC curve to measure the performance of model. From the result shown below, we can draw the conclusion that among different model, multiplayer perceptron gives us a accurate and efficient model for spam detection, and for different threshold, considering the process of drawing ROC curve, model still performs well.
Though autoencoder has been proved strong performance, but in our experiment, the accuracy is not very high, the main reason is that the training of autoencoder is very hard, and we can't find a perfect model that compress the original data into low dimension, and we believe that this will be a good direction for further investigation.

4 Conclusion and Future Work

Conclusion: With the help of multi-layer perceptron, we gains performance enhancement on the spam base dataset. And with given features, we can predict spam with over 90% accuracy, that beats lots of existed method. What's more, we found that with the help of ensemble learning method, the performance of classifier can be further boosted.
Future work: Though we can use the bag of word model of IF-IDF model to extract feature of any given emails, but this method can not be utilized to represent a documents perfectly, which means these feature is not perfect for the classification task. We believe that by integrating the feature extracting task and classification task, the model can be trained in a whole, and gains better performance. These emerging deep learning method like CNN and LSTM will be good tools to solve this.

References

1. Srividhya, V., and R. Anitha. "Evaluating preprocessing techniques in text categorization." International journal of computer science and application 47, no. 11 (2010): 49-51
2. Androutsopoulos, Ion, John Koutsias, Konstantinos V. Chandrinos, George Paliouras, and Constantine D. Spyropoulos. "An evaluation of naive bayesian anti-spam filtering." arXiv preprint cs/0006013 (2000)
3. Androutsopoulos, Ion, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach." arXiv preprint cs/0009009 (2000)
4. Zhang, Yudong, Shuihua Wang, Preetha Phillips, and Genlin Ji. "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection." Knowledge-Based Systems 64 (2014): 22-31
5. Carreras, Xavier, and Lluis Marquez. "Boosting trees for anti-spam email filtering." arXiv preprint cs/0109015 (2001)
6. Riedmiller, Martin, and AG Maschinelles Lernen. "Multi Layer Perceptron." Machine Learning Lab Special Lecture, University of Freiburg (2014)
7. Milne, L. K., T. D. Gedeon, and A. K. Skidmore. "Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood." In Proceedings Australian Conference on Neural Networks, pp. 160-163. 1995
8. LeCun, Yann, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. "Handwritten digit recognition with a back-propagation network." In Advances in neural information processing systems, pp. 396-404. 1990
9. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012
10. Hochreiter, Sepp. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6, no. 02 (1998): 107-116
11. Yang, Yi, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. "l2, 1-norm regularized discriminative feature selection for unsupervised learning." In IJCAI proceedings-international joint conference on artificial intelligence, vol. 22, no. 1, p. 1589. 2011
12. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014)
13. Hanley, James A., and Barbara J. McNeil. "The meaning and use of the area under a receiver operating characteristic (ROC) curve." Radiology 143, no. 1 (1982): 29-36
14. Hochreiter, Sepp. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6, no. 02 (1998): 107-116.
15. Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18, no. 7 (2006): 1527-1554.
16. Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313, no. 5786 (2006): 504-507.
17. Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15, no. 1 (2014): 1929-1958.
18. Wager, Stefan, Sida Wang, and Percy S. Liang. "Dropout training as adaptive regularization." In Advances in neural information processing systems, pp. 351-359. 2013.

19. Wang, Yong, and Ian H. Witten. "Modeling for optimal probability prediction." (2002): 650-657.