EPiC Series in Computer Science

Volume 36, 2015, Pages 42–50

GCAI 2015. Global Conference on Artificial Intelligence

Application of Adaptive Neural Networks for the Filtration of Spam

G. Besiashvili, T. Bliadze and Z. Kochladze[1]
Ivane Javakhishvili Tbilisi State University, Tbilisi, Georgia
gela.besiashvili@tsu.ge, t.bliadze@gmail.com, zurab.kochladze@tsu.ge

Abstract

E-mail is one of the most widespread communication method in twenty first century. People daily get e-mails in their inboxes, including permanently increasing number of unwanted notifications (spam) too. The advanced e-mail service packs and antiviruses contain spam filtration software. Nevertheless, to classify e-mails as spam, fully depends on user, as information acceptable (or vital) for one consumer, may be a spam for another. Hence, it is crucial to have automatic spam filtration system for every individual user. The paper discuss the solution, a system that filters emails through consumers' parameters and gradually learns how to filter them accordingly. The system is based on multi-layer neural network, which uses logistic activation function, learns via tutor and counter-spread algorithm of mistakes (a.k.a. method of supervised learning, backward propagation of errors).

1 Introduction

Nowadays, e-mail is one of the most widespread methods for the communication. The number of e-mails increase permanently, but unfortunately, the amount of unwanted notifications (spam) increase on the same scale as well. According to the statistics, 18% of the traffic is spam [1]. The spam lags the internet traffic, consumes space on hard disc, and reduces the capacity of the network, not to mention the time spent by the people to classify them. Additionally, if various companies initially used the spam for advertising purposes (the name was coined from this), today spam represents one of the methods for hackers to assemble a cyber-attack the computers. Therefore, due to the arguments mentioned above, it is necessary to carry out some methods for defending oneself from the spam. Particularly, via designing such programs that are capable of spam filtration in an automatic regime.

This matter is being carried out for a long time. Methods for the spam filtration have been transformed and developed throughout the years. If the programs filtered the spam by using only titles and several keywords earlier [2], in recent years they practice analyzing the body of an e-mail by using different methods [3,4,5], which enables us to advance spam filters even more. While designing such program, the most crucial moment is to differentiate wanted and unwanted e-mails from one another.

[1] The authors of this paper are listed in alphabetical order.

G.Gottlob, G.Sutcliffe and A.Voronkov (eds.), GCAI 2015 (EPiC Series in Computer Science, vol. 36), pp. 42–50

It is clear, that the classification of e-mails as wanted and unwanted (spam) notifications is wholly depend on the consumers[2], because the information that is acceptable (sometimes vital) for one user, may be an unwanted, sometimes even irritating one for the other one, or be a dangerous spam for the computer. The most advanced service packs today include a spam filtration program designed for corporate consumers, mainly for the business sector, that are able to order systems, which envisages their needs and views about wanted and unwanted information. However, it is nearly unavailable or unaffordable for the individual users.

The paper proposes one of the possible solutions for the problem. That is to design a system, which is capable of learning, according to every individual user's viewpoints, to differentiate wanted information from spam and eventually, consumer will use this system to filter the spam accordingly.

2 Related work

Spam detection is very popular case nowadays. Detector systems may use machine learning for this purpose, which means system can learn from analyzed data how to react in particular situations. There are introduced several approaches for learning algorithms [6]: 1. Supervised learning: inputs and desired output is given by a teacher and system learns from this data for the future inputs. 2. Unsupervised learning: there are no outputs supplied, so learning algorithm figures out by itself the pattern of inputs. 3. Reinforcement learning: algorithm is supposed to find the best action in given situation by a process of trial and error [7] and it is mainly used in game theory, control theory, multi-agent systems etc. There are some widely used and famous learning methods, e.g. Naive Bayes classifier (Bayesian network) for classification problems, clustering, Decision Trees, Support vector machines, Artificial neural networks with Multi-Layer Perceptron [8]. The most widespread methods for spam filtering are artificial neural networks and Bayesian method, also support vector machines. Bayesian method relays on the theorem:

$$P(W|L) = \frac{P(L|W)P(W)}{P(L)} = \frac{P(L|W)P(W)}{P(L|W)P(W) + P(L|M)P(M)}$$

Whereas, *P(W|L)* is a conditional probability.

The general idea is that every event is derived from previous events and the probability of occurring given event in the future can be determined from its previous occurrences. So, if particular fragment of text is considered to be more often in spam mails, than in ham (legitimate), we could assume that mail, containing this fragment, is spam too. The advantage of this method is its self-adaptivity (it could recognize word "f-r-e-e" in spammers' mail, according to previously learnt word "free"), easily construct multilingual filtering. It is necessary to make a word database which occurs in spam and ham. Then Bayesian filter will calculate the probabilities, if each word is in spam or ham.

Subset	All	CFS	Consistency	Logistic	Random Forest
Naïve Bayes	**0.8121**	0.5925	0.5974	0.4446	0.5185

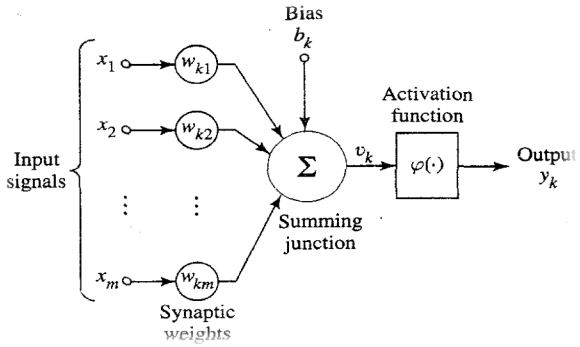
Subset	All	CFS	Consistency	Logistic	Random Forest
Hidden units	3	7	5	**5**	7
F_0.5	0.8994	0.8758	0.9015	**0.9051**	0.9021

[2] The system is supposed to be maintained by IT specialist, who can install and train neural network for the users.

These two tables are the results [9] of Naïve Bayes and Neural network testing on "SpamBase" [10] mail database and applied several different algorithms: Correlation-based Feature Selection (CFS), Consistency, Akaike Information Criterion (AIC) over a logistic regression, Random Forest. It shows that Naïve Bayes had not as good results as Neural Network. Naïve Bayes method is used in many server-side e-mail filters, such as: CRM114, DSPAM, SpamAssassin, SpamBayes, Bogofilter and ASSP.

3 Artificial Neural Network

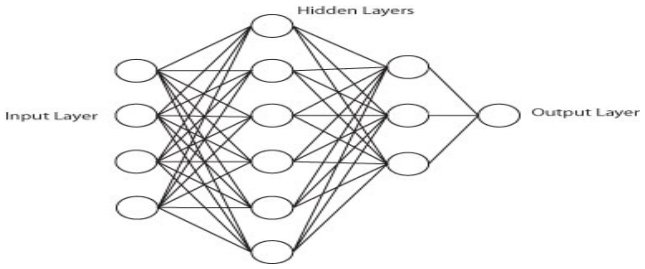
Artificial neural network is a system designed to simulate human's brain. Brain itself consists of specific cells – neurons, which can transfer synapses (a junction between two nerve cells). They have many inputs (dendrites) and one output (axon). Same is the model of the artificial neuron: inputs give data to neuron, which analyses it and then output section transfers it to another neuron.



The structure of artificial neuron is:

- Input signals ($x(1), x(2), .. x(n)$) ;
- Synaptic weights (w_{kj});
- Sum of synaptic weights + bias b_k (regulates the values of inputs);
- Activation function $f(u)$, which is minimizer of output amplitude.

These neurons form network with input, hidden and output neuron layers. There are multiple layers in hidden one, with different number of neurons.



For this project Sigmoid (logistic) activation function is chosen, because it allows output to change slightly, when the weights are changed slightly.

Artificial neuron network is an interesting method for spam filtering and researchers continue to develop such systems.

MLP Architecture	Learning Rate	Momentum	Average Classification Rate
20 -10 - 10 -10 - 5	0.1	0.8	62.4 %
20 -10 - 10 -10 - 5	0.1	0.95	63.2 %
20 - 10 - 10 - 5	0.1	0.85	60.61 %
15 - 15 - 15 - 5	0.1	0.85	60.59%

This chart is an example [2] of results of neural network analysis (4025 Feature vectors with 57 input attributes) and it was mentioned, that system was quite unstable and was not able to detect spam, however, there was also presented another variation - the reduced data set and it had much better performance levels.

4 System AntiDote

Spam automatic filtration system "AntiDote" presented in the paper consists of two sub-systems: Sub-system of e-mails pre-analysis and learnable sub-system, which learns how to differentiate wanted information from spam and then provides an automatic filtration for the spam.

4.1 Sub-system of pre-analysis

In order to differentiate wanted e-mail from the spam, the system at first should conduct the analysis of the incoming notifications via certain parameters. These parameters are selected on basis of existing text analysis methods and their number in system is up to thirty nine. We will put down some of them: the total number of letters (C), ratio of alphabetical characters (frequency of characters / C), white space ratio (frequency of white space / C), the number of words (M), frequency of punctuations (. ‘ ; ? ! : ( ) – " « » < > [ ] { }), the average length of words, the average length of sentences according to letters, the average length of sentences according to words, frequency of special characters (*, _ , +, =, %, $, @ ,- , \/), etc.

The system enables the user to select (Figure 1) desired parameters from this list and add those parameters, which according to his needs are necessary for appropriate filtration and are not foreseen in the system (by all means indicating possible meanings of the parameter).

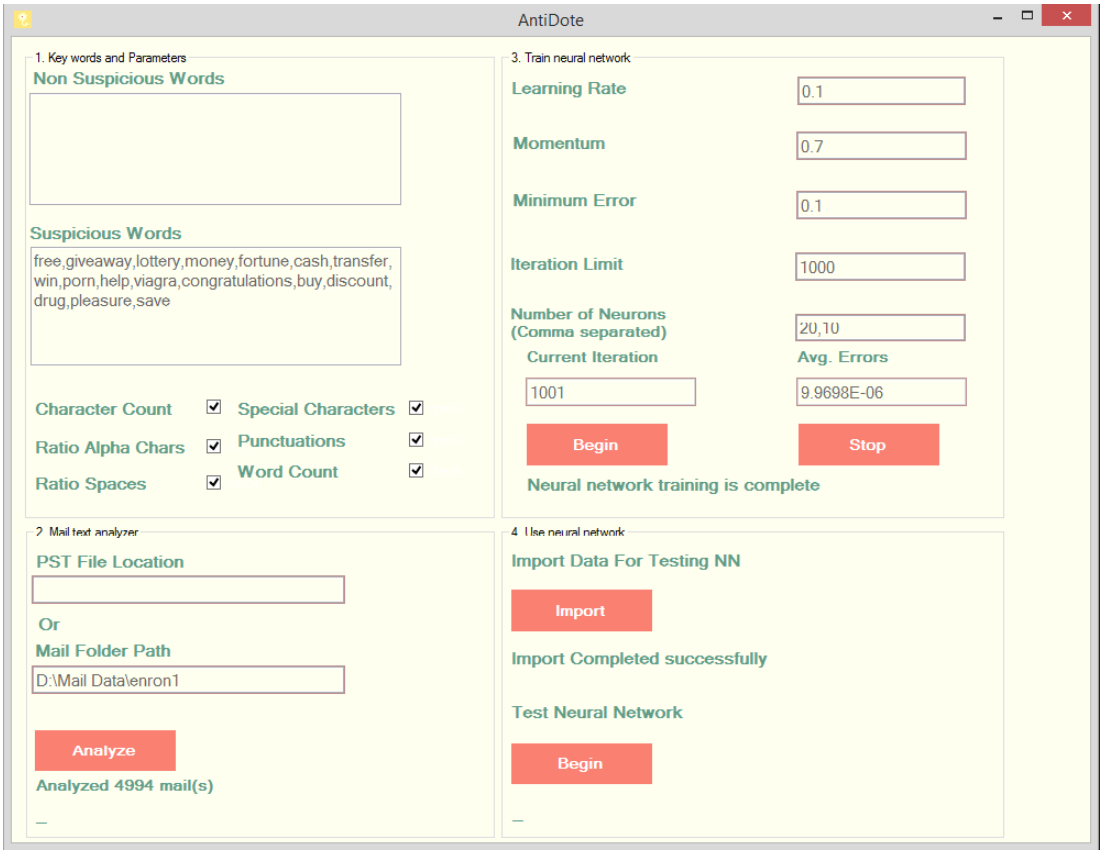


Figure 1: System "AntiDote"

In the first section can be supplied suspicious and non-suspicious words, as well as additional parameters can be checked/unchecked as desired.

After selecting the parameters mail text analyzer should be started. Enron (bankrupted company) mail database is used for learning and testing [11] in this system. After importing e-mails, the system generates a file with analyzed data (Figure 2).

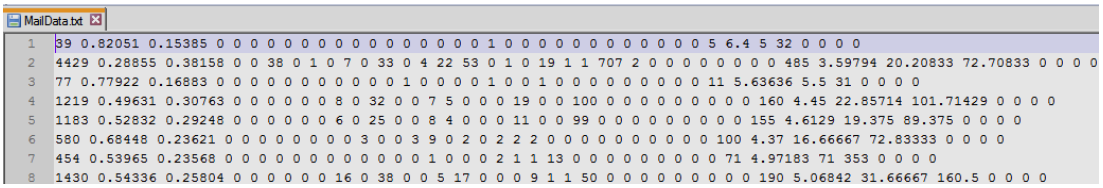


Figure 2: A file of vectors made by pre-analysis system

We have selected 39 attributes (some of them are described above in this section: character count, ratio of alphabetical chars etc.) for the reviewed example in the paper. The pre-analysis system analyses the certain e-mails via these attributes and represents them as vectors. File (Figure 2) with forty-attribute (39 input variables + 1 target output) vectors is disposed to the learnable sub-system.

4.2 Learnable sub-system

The vectors worked out in pre-analysis sub-system are disposed to the system's main, learnable sub-system. This system works in two regimes: learning and guessing regime. In the first regime, the system gets the certain amount of vectors generated by pre-analysis.

The learnable sub-system is constructed by using multi-layer neural network (**MLN**) [12,13 ,14]. It is stipulated from fact, that in comparison with the other systems, **MLN**-based systems are capable of modeling the learning process in easiest and fastest ways. There are three **MLN** architectures discussed in the paper: Double-layer neural network 20-10 neurons and three-layer networks 20-20-20 and 30-20-10 neurons. In all three cases sigmoid activation function is used.

$$f(u_k) = \frac{1}{1 + \exp(-\sum_{j=1}^{m} w_{jk} x_j - b_k)},$$

Whereas, $u_k = \sum_{j=1}^{m} w_{kj} x_j - b_k$.

For realizing this method, necessary data and sequence of actions are:

1. Incoming E signal and corresponding output C.
2. Calculate E Feed Forward; Mass quantities S_I "are defined in network and activators for each neuron too.
3. For resolving the meanings of the mistakes δ_0, calculations start from the outgoing layer directed towards the incoming layer via following rule:
   For output neuron

$$\delta_{0=(C_{I-X(i)})X(5)(1-X(i))}$$

   For hidden neuron

$$\delta_{i=(\sum_{min>i} W_{MJ}\delta_0)X(i)(1-x(i))}$$

4. The masses in network are as follows:

$$w^*_{IJ} = W_{IJ} + \rho\delta_0 X(i),$$

   ρ – learning speed.

It should be noted that the amount of incoming data vectors and the number of these vectors' attributes has an impact on errors made during the networks operation and its speed. If parameters are a lot more than the totality of incoming data, over-fitting will occur with higher possibility. Incoming vectors ten times more than parameters are an acceptable proportion for the so-called noiseless target and approximately two times more - this is minimum acceptable requirement.

So-called Incremental Learning is selected as the type of learning and it means a correction of mass and the update of the network after passing every vector, while in case of the other type - batch learning the network is being updated only after passing the learnable tests.

Before the program starts functioning in learning mode, it is provided by another parameters: *Learning rate* - value which defines size of weight and bias changes in learning in network (the input value is from 0.1 to 0.3).

Momentum - value which is added to the weights and is the fraction of previous weight update result. This is to avoid reaching local minimum during learning. However, its high value causes

instability of the system and the small value - it will not avoid the occasion of reaching local minimum well and the speed will be low (the default value is from 0.7 to 0.9).

The minimum error limit and *iteration limit* are two methods for stopping network, which may be used separately or combined; the network stops functioning if minimum errors limit (0.001 - 0.1) or the number of iterations for every incoming vector in network will exceed the indicated limit (number of iterations when every vector go through the network again is often called Epoch).

Number of neurons is filled in separated by comma; e.g. 20,10 means two layer network of which first layer contains 20 neurons and the other one - 10.

The dynamic of errors is saved in errors_(DateTime).csv file, and the weights in weights_(DateTime).csv file. The information regarding the meaning of these parameters and its application rule is provided to the consumer as an instruction. After this, user will fill in the parameters of "AntiDote" system (Figure 1).

5 Results

The analyzed initial data consists of 5975 e-mails, from which 3672 e-mails represent legitimate mail, while 1500 is spam. After disposing the similar e-mails, 4994 unique e-mails remain. Whereas the outgoing answers are in [0-1] interval, it was recommended to scale incoming data and converting it to the same interval (in certain cases the diapason for the parameters in vector varies from 31000 to 0), but this process didn't show any better results.

The test was conducted on the following architectures:

NN Architecture	Learning Rate	Momentum	Minimum error	Max Iteration	Avg Test Output
20 - 10	0.1	0.7	0.0001	365	0.00004825
20-10	0.3	0.7	0.00001	48	0.001201
20-20-20	0.1	0.7	0.0001	36	0.002777
20-20-20	0.4	0.7	0.0001	124	0.0006
20-20-20	0.5	0.8	0,0001	56	0,000948
30-20-10	0,4	0,7	0.0001	7	0,00296
20-20-20	0,1	0,91	0,0001	253	0,001923

The time for training is quite short, less than a minute for 2 hidden layer architecture. The table shows, selecting the same amount of neurons in layers and the increase of Learning Rate resulted in fewer errors, than the increase of Momentum in the same architecture. Also, in future works, could be added other outputs too. Figure 3 shows the graphic of the result of test for each e-mail. X-axis represents mails, y-axis the result - approximate guess of mail being spam.

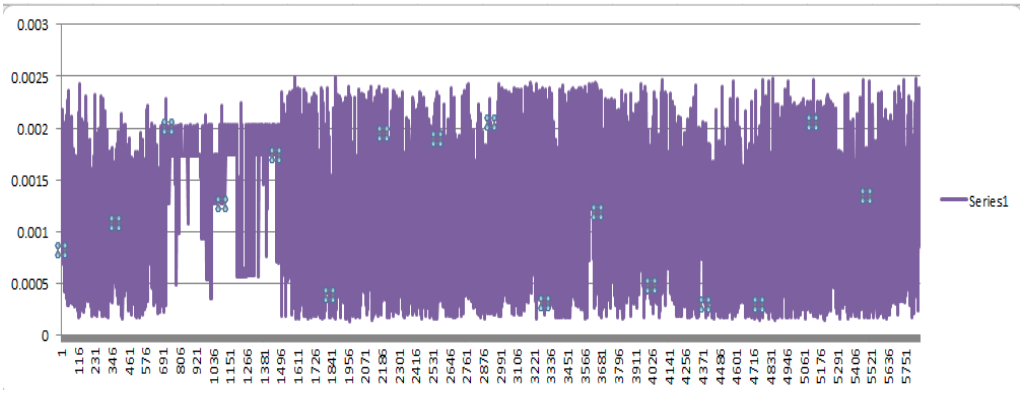


Figure 3: The graphic of tests results

The dynamic of errors were in some cases bumpy and sometimes smooth, according to error output file (Figure. 4)

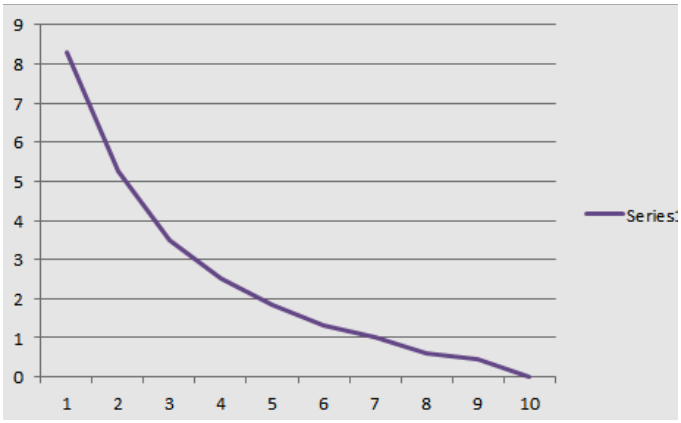


Figure 4: The graphic of errors dynamic

The examined data clarifies that created system is able to successfully filter the spam, however this system can be further improved, both in pre-analysis sub-system of e-mails as well as in learnable sub-system. In particular, deep learning approach may be used, which will improve the operation of the system.

6 Conclusion

The service program “antidote” is designed for filtering the spam in automatic regime according to provided parameters. The consumer of this program can, according to his needs, select parameters from the given list (he can also add other parameters) and make multilayer neural network learn to differentiate wanted and unwanted e-mails from one another.

Today the program is being improved in order to be more easily applicable for the users.

References

1. Email Statistical Report, http://www.radicati.com/wp/wp-content/uploads/2010/04/Email-Statistics-Report-2010-2014-Executive-Summary2.pdf.
2. Thiagarajan Sivandayan. Spam? Not Any More! Detecting Spam emails using neural networks. ECS/CS/ME 539 Project.
3. Why Bayesian filtering is the most effective anti-spam technology; GFI White Paper, (http://www.gfi.com/whitepapers/why-bayesian-filtering.pdf).
4. Priyanka Chhabra, Rajesh Wadhvani, Sanyam Shukla, Spam Filtering using Support Vector Machine; Special Issue of IJCCT Vol.1 Issue 2, 3, 4; 2010 for International Conference [ACCTA-2010], 3-5 August 2010.
5. V.Kanaka Durga, M.R. Raja Ramesh, Accurate Spam Mail Detection using Bayesian Algorithm; ISSN: 2278 – 1323; International Journal of Advanced Research in Computer Engineering & Technology; Volume 1, Issue 4, June 2012.
6. Stuart J. Russell and Peter Norvig; Artificial Intelligence A Modern Approach, 2nd edition; 2002.
7. Christopher M. Bishop; Pattern Recognition and Machine Learning; 2006.
8. Ethem Alpaydin; Introduction to Machine Learning; Second edition, 2009.
9. Alessandro Zanni and Israel Saeta Pérez, UPC; Spam filtering; 2012.
10. Spambase Data Set https://archive.ics.uci.edu/ml/datasets/Spambase.
11. Enron mail datatbase http://www.csmining.org/index.php/enron-spam-datasets.html.
12. Michael A. Nielsen ; Neural Networks and Deep Learning; Determination Press, 2014.