Accuracy Analysis of Neural Networks in removal of unsolicited e-mails

P.Mohan Kumar
Assistant Professor (Senior)
SITE
VIT University Vellore.

P.Kumaresan
Assistant Professor
SITE
VIT University Vellore.

S.Yokesh Babu
Assistant Professor (Senior)
SCSE
VIT University Vellore.

ABSTRACT

Today communication has been revolutionized with email and other online communication systems. However, some computer users have abused the technology used to drive these communications, by sending out thousands and thousands of spam emails with little or no purpose other than to increase traffic or decrease bandwidth. With the electronic mail emerging as the primary means of communication, sorting of electronic mails is of prime importance. Most current sorting techniques are rule based, in which the user is supposed to give a set of rules, according to which mails are sorted. But configuring these rules is a tedious and often impossible task due to the variety of emails. In this paper a technique using neural network is deployed which automatically removes unwanted incoming mails, without the need for constant user intervention as well as its accuracy is analyzed in parallel.

Keywords:

neural network accuracy.

1. INTRODUCTION

Now a day's electronic mail has emerged as the primary means of communication. With the massive amount of information and speed the Internet is able to handle. The volume of unsolicited commercial e-mail messages transmitted by the Internet has reached to a large proportion of the total mail delivered every day. The spam messages raise a lot of problems for internet service providers and users also. Firstly, junk email occupies server storage space and consumes network bandwidth, for second, users are pushed to waste non-trivial amount of time for identifying and removing spam from own computers. The best solution for avoiding such discomfort would be to develop and refine automatic classifies that can distinguish legitimate e-mail from spam accurately and efficiently. The simplest and most common approaches are to use filters that screen messages based upon the presence of common words or phrases common to junk e-mail. Other simplistic approaches include blacklisting and white listing. In practice, effective spam filtering uses a combination of these three techniques. The primary flaw of the first two approaches is that they rely on spammers by assuming that they will not change their identities or alter the style and vocabulary of their sales pitches. White listing risks the possibility that the recipient will miss legitimate e-mail from a known or expected correspondent with a heretofore-unknown address, such as correspondence from a long-lost friend.

2. MOTIVATION

The idea is to use a neural network to classify spam (unsolicited emails) and ham (wanted, personal messages) emails. The spam filtering problem can be broken down into a simple classification problem and most of the time-tested networks and algorithms such as Back propagation can be used. This paper discuss about the evaluation of effectiveness of email classifiers based on the feed-forward back-propagation neural network. The results obtained in many papers show that the feed-forward back-propagation network algorithm classifier provides relatively high accuracy and sensitivity that makes it competitive to the best known classifiers.

3. OBJECTIVES

Spam is unsolicited email on the internet. A major problem facing internet computer users today is the deluge of unwanted and often rude email filling their email-boxes. Spam costs the sender very little to send. Most of the cost is absorbed by the recipient or by the carriers rather than by the sender. This is mostly in the form of lost productivity or network resources. In addition to the unnecessary strain it places on a corporate network, spam frequently contains viruses.

A text parser is used to calculate the statistical distribution of words within an email body. This information is used by feed-forward back-propagation system to determine the spam classification of the email. This design is exceptionally good as compared to present day filters based on its simplicity and limited scope of detection methods. This system could be further improved by incorporating other identifiers of email spam.

The primary challenge faced was determining the number of hidden nodes in the neural network architecture. From the literature survey and related experimental survey it was determined that if the number of hidden nodes was restricted to 20, the feed-forward back-propagation model optimally detected error. The second challenge faced was to determine how many times to iterate the assignment of weights in order to effectively train the neural network.

4. OVERVIEW OF THE PROPOSED SYSTEM

4.1 System Objectives

Based on the feasibility study of the requirements provided by the study, the Requirement Analysis Document is prepared to formally seek clarifications. After the needs (both implied and stated) are understood and a detailed analysis of the risk factors is made the following tasks are handled.[a].An outlay of general work schedule is formed.[b].An estimate of the time required is made.[c].Resources and manpower to be involved in the project are identified [d].System objectives are formulated.

4.2 DESIGN AND IMPLEMENTATION

4.2.1 Architecture:

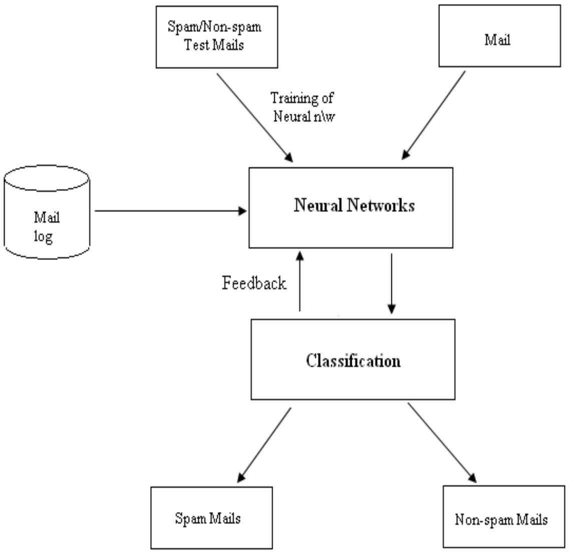

Fig1. Overall system architecture model

The above architecture shows the overall view of the system. First the neural network is trained using some pre-classified mails and the result is stored in the mail log. Then, the user mail is given to the neural network which after some processing classifies the mail to be spam or ham.

4.2.2 Detailed Design:

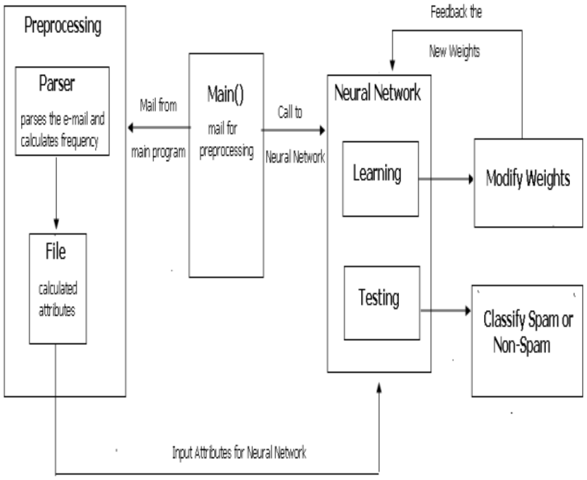

Fig: 2: Detailed Design

5. RELATED WORK

This paper evaluates the effectiveness of email classifiers based on the feed-forward back-propagation neural network. Results are evaluated using accuracy and sensitivity metrics. The results show that the feed-forward back-propagation network algorithm classifier provides relatively high accuracy and sensitivity that make it competitive over other methods.

The application of neural networks to detecting spam is definitely something that can and is being pursued as a viable option. However, to obtain optimum performance, we do have to do sufficient amount of data analysis. Also, this data analysis has to be general so as to block a wider variety of spam. 'The basic principal used in any spam filtering technique, whether heuristic or keyword based, is identical: spam messages generally look different than good messages and detecting these differences is a good way to identify and stop spam. The difference between these technologies really comes down to the problem of distinguishing between these two classes of email. The neural networks approach is more refined, more mathematical and potentially far more accurate and reliable in accomplishing this task. Although no single technology can achieve one hundred percent spam detection with zero false positives (despite vendor claims), machine-learned heuristics in general and neural networks in particular have proven extremely effective and reliable at accurately identifying spam and minimizing errors to an acceptable minimum'.

- It would definitely be interesting to conduct cross-validation between data sets used from different sources and one could develop a heuristic model to pick inputs to be used for the network.
- Fuzzy logic is another important content-based method to distinguish spam. A fuzzy logic approach to the same problem can bring some new insights into the problem.
- A combinational approach can be used to achieve higher classification rates (using header filters, content based filters and user specific information).

6. The Feed-Forward Back-propagation Neural Network:

Back-propagation, or propagation of error, is a common method of teaching artificial neural networks how to perform a given task. It is a supervised learning method. The back-propagation networks are necessarily multilayer neurons (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers. The first term, "feed forward" describes how this neural network processes and recalls patterns. In a feed forward neural network, neurons are only connected foreword. Each layer of the neural network contains connections to the next layer (for example, from the input to the hidden layer), but there are no connections back.

The term "back-propagation" describes how this type of neural network is trained. Back-propagation is a form of supervised training. When using a supervised training method, the network must be provided with both sample inputs and anticipated outputs. The anticipated outputs are compared against the actual outputs for given input. Using the anticipated outputs, the back-propagation training algorithm then takes a calculated error and adjusts the weights of the various layers backwards from the output layer to the input layer.

6.1 Summary of the back-propagation technique.

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.

6.2. Back-propagation Algorithm:

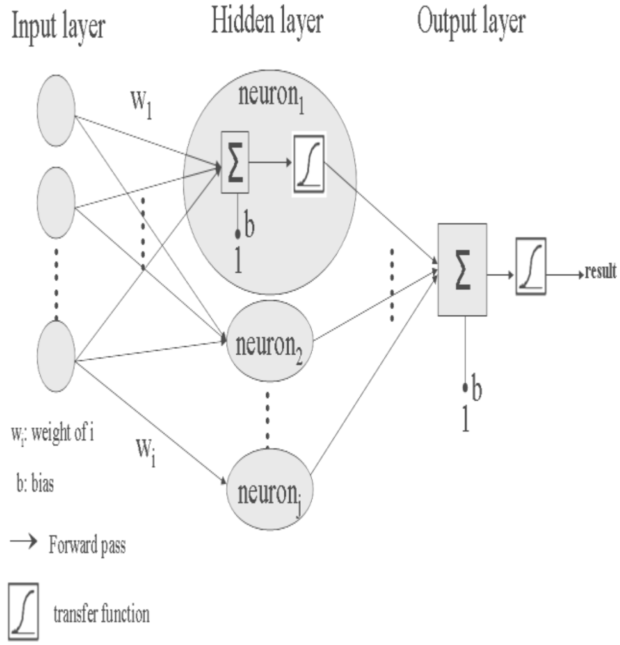

Fig: 3 Back propagation concept.

Actual algorithm for a 3-layer network (only one hidden layer):

```
Initialize the weights in the network (often randomly)
Do
    For each example e in the training set
    O = neural-net-output(network, e) ; forward pass
    T = teacher output for e
     Calculate error (T - O) at the output units
     Compute delta_wi for all weights from hidden layer to output layer ; backward pass
      Compute delta_wi for all weights from input layer to hidden layer ; backward pass continued
      Update the weights in the network
Until all examples classified correctly or stopping criterion satisfied
Return the network
```

7. DETAILED IMPLEMENTATION

7.1 Implementation:

In a nutshell the implementation procedure is described as follows:

1. Building a word list 2.Creating a parser 3.Creating the neural network
4.Train the network 5.Test the network

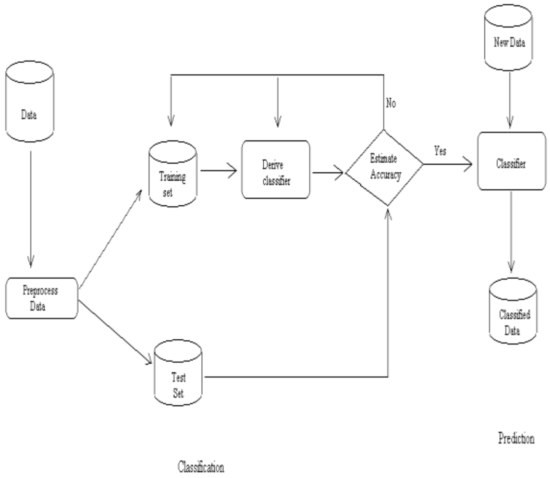

Fig 4. Email classification process model

The overall implementation procedure can be viewed in three phases

1. PARSING
2. LEARNING
3. TESTING

Phase1 (PARSING):

- In this module parsing is done. Take the document from the mail server and then calculate the number of spam words and total number of words. Then the attribute and frequency is calculated from the file containing predefined spam words. This value of the frequency is then given to the neural network for learning purposes.

Phase 2 (LEARNING):

- Now, after the parsing learning of the network has to take place. Input cases are provided by the parser and the input mails
- (both spam and non-spam mails). In the Learning phase, we have to check the obtained output with the target output, if both have significant differences then we

have to modify the weights and again train the network. This is iterative process.

Phase 3 (TESTING)

- In this module, after the learning is complete we start taking the mails from the mail log, then we parse them and calculate the number of spam words and their frequency. Then input is given to the input layer and net output for that layer is calculated. Output from Input Layer becomes the input for hidden layer and again net output is calculated. Same is done for output layer. The value which we get from the output layer specifies or determines whether the mail is a spam mail or not. The formula for the calculation of net

Output is:

$$\text{Net} = \sum w(i,j)*p(i)$$

First the neural network is trained using some pre-classified mails and the result is stored in the mail log. Then, the user mail is given to the neural network which after some processing classifies the mail to be spam or non-spam.

8. RESULTS ANALYSIS.

With larger number of inputs, the network complexity increases, but so does the performance. Utilizing a large input data set, without identifying the most important inputs, does not necessarily improve performance. We have shown in our case, that the network fails (classifies all email as spam) when we use the complete original data set. So rather than pick a lot of words from an email, we'd do better to pick fewer words but which occur in very different amounts in spam and non-spam emails. In this case, some of the inputs which varied a lot between spam and non-spam emails were words like 'meeting', 'hp' etc. In a university context, one could use the occurrences of words like 'university', 'research' etc, which occur very rarely in spam emails. By applying varying thresholds while restricting the number of inputs we can have a performance trade-off between the complexity of the network (number of inputs) and the accuracy of classification. For a higher classification rate, it is important to utilize a combination of spam filtering methods, rather than just the neural network based spam blocking. Commercial email software such as Eudora utilize more adaptive means to change even the inputs given to the neural network, thereby making the spam blocking highly personalized and optimized on a person-to-person basis.

9. CONCLUSION AND FUTURE ENHANCEMENT

The application of neural networks to detecting spam is definitely something that can and is being pursued as a viable option. However to obtain optimum performance, we do have to do sufficient amount of data analysis. Also, this data analysis has to be general so as to block a wider variety of spam. 'The basic principal used in any spam filtering technique, whether heuristic or keyword-based, is identical: spam messages generally look different than good messages and detecting these differences is a good way to identify and stop spam. The difference between these technologies really comes down to the problem of distinguishing between these two classes of email. The neural networks approach is more refined, more mathematical and potentially far more accurate and reliable in accomplishing this task. Although no single technology can achieve one hundred percent spam detection with zero false positives (despite vendor claims), machine-learned heuristics in general and neural networks in particular have proven extremely effective and reliable at accurately identifying spam and minimizing errors to an acceptable minimum'.

10. Future work:

It would definitely be interesting to conduct cross-validation between data sets used from different sources and one could develop a heuristic model to pick inputs to be used for the network. Fuzzy logic is another important content-based method to distinguish spam. A fuzzy logic approach to the same problem can bring some new insights into the problem. A combinational approach can be used to achieve higher classification rates(using header filters, content based filters and user specific information).

Reference:

[1] Sivanadyan, Thiagarajan, Detecting Spam emails using neural networks, www.cae.wisc.edu/~ece539/project/f03/sivanandyan.pdf, June-July 1999

[2] D. Puniškis, R. Laurutis, R. Dirmeikis, An Artificial Neural Nets for Spam e-mail Recognition, Kaunas University of Technology, 2006

[3] Yue Yang and Sherif Elfayoumy, Anti-Spam Filtering Using Neural Networks and Baysian Classifiers, www.student.cse.buffalo.edu/~pejusdas/document/p2report.pdf

[4]. Ponnapalli A formal selection and pruning algorithm for feed forward ANN optimization.IEEE Conference transaction on NN 1999 vol 10.

[5]. Amit Ramesh and M.J.Matrix.Learning moment sequences from demonstration IEEE proceedings ICDL-2000 MIT.

[6] F.L Chung and T.LEE network growth approach to design feed forward NN.IEE

[7]. Md.Yosuf, javeria iqubal Punjab University Pakistan. Hash table based feed forward NN 2009 International conference on emerging trends.

[8]. M.Shami. S.Dumars 1998 A Bayesian Network approach for filtering Junk e-mail. In learning for text categorization papers from AAAI workshop.

[9] M.Shami Application of machine learning to information access In AAAI 1997-Proceedings of 14th national conference on AI.

[10] .M.Gori and A.Tesi on the problem of local minima in back probagation .IEEE Transaction pattern analysis and machine learning VOL 14 Jan-1992.

[11]. M.Bianchini,P.Frasconi and M.Gori Learning in Multilayered NN used as auto associaters. IEEE transactions on NN Vol 6 March 1995.

[12]. H.Dundar and k rose .the effect of quantization on multilayr NN.IEEE transaction on NN Nov-1995

11. Authors Details

[1]Mr. P.Mohankumar is working as Assistant Professor., (senior) in School of Information Technology and Engineering, VIT University, Vellore.
His area of Research includes Advanced Database Management Systems and Neural networks having more than ten years teaching and academic activities.

[2] Mr. P.Kumaresan is working as Assistant Professor in School of Information Technology and Engineering,VIT University, Vellore. He has vast teaching experience. His research interests include Data Mining – Semantic Web Mining