British Journal of Mathematics & Computer Science
8(1): 57-71, 2015, Article no.BJMCS.2015.144
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

An Opinion Predictor Using Recurrent Neural Networks

Santhi Priya Polisetty[1*] and T. V. Rao[2]

[1]CSE Department, Sri Sairam College of Engineering, Anekal, Bangalore, Karnataka, India.
[2]CSE Department, PVP Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India.

Article Information

DOI: 10.9734/BJMCS/2015/16121
Editor(s):
(1) Qiankun Song, Department of Mathematics, Chongqing Jiaotong University, China.
(2) Xiaodi Li, School of Mathematical Sciences, Shandong Normal University, Ji'nan, 250014, Shandong, P.R. China.
(3) Tian-Xiao He, Department of Mathematics and Computer Science, Illinois Wesleyan University, USA.
Reviewers:
(1) Grienggrai Rajchakit, Department of Mathematics, Faculty of Science, Maejo University, Thailand.
(2) Anonymous, Italy.
(3) Anonymous, Malaysia.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=1030&id=6&aid=8452

Short Research Article

Received: 08 January 2015
Accepted: 18 February 2015
Published: 14 March 2015

Abstract

Aims: Netizens share their personal experiences, opinions at the review sites, discussion groups, blogs, forums and etc. With the rapid growth of technology, now-a-days almost everyone uses internet. Opinions are important because whenever a person needs to take a decision, helikes to hear others' opinions. Quotes of the attitude may be generally positive or negative. We propose a system for classifying text sentiment based on Neural Networks classifier. In this paper, we focus on classifying product reviews according to the opinion and the value judgment they posses, into two polarities, positive and negative, using the multilayer neural network.
We also address opinion prediction application for the products that are being launched in future. The product features, given as input to recursive neural network are used to predict the opinions, which are expected from customers. The opinion prediction is done using recurrent neural network with the help of back propagation with time (BPTT) algorithm.
Place and Duration of Study: Department of Computer Science and Engineering, Sri Sairam College of Engineering, Anekal, Bangalore between July 2014 and December 2014.
Methodology: We experimented on 500 opinions, among them 400 were used as training set, and 100 were taken to be testing set, for each type of mobile (Nokia Lumia 720, LG G3).
Results: For each mobile type we achieved up to 85% of correct classification of opinion reviews.
Conclusion: We presented a system for determining text sentiment of product reviews by classifying them using Neural Network. The method uses feed-forward Neural Network with ten

**Corresponding author: shantipriya.p@gmail.com;*

hidden layers. From the presented results, it can be seen that, a new approach is developed categorizing product reviews in 2 classes in the context of opinion mining. Experiments conducted on training sets show that with our approach we are able to extract relevant feedback from a specific domain of products. We compared our proposed opinion classification algorithm to standard algorithm BPNSO which showed the results are good between 60% to 80%.

Keywords: Opinion mining; neural networks; generalized delta rule; opinion prediction; recursive neural network; back propagation through time.

1 Introduction

Before the internet was commercialized and popularized, the medium relied upon word of mouth of individuals, with respect to suggestions about movies, electronics, books, products, etc. But, now-a-days it is simple to access reviews and opinions of a large number of people online, who express their opinions from every corner of the world. These opinions can be considered as fresh and recent, because these people have first-hand knowledge or experience regarding the particular product and they wish to share their reviews with others. Today, more and more strangers are posting their opinions by sharing their reviews available to strangers via the Internet through forums like twitter, Facebook, Wordpress.com, Blogspot.com, etc.

Data warehouses are used to store the textual data and data-mining techniques can be used to analyze them. Classification of documents according to the opinion expressed by the reviewer in a blog such as, positive or negative mood of a review, the favorable or unfavorable aspect by an expert, results into polarity of a document (positive, neutral, negative) and/or the intensity of each opinion (low, neutral, high), etc. Mining of unstructured data such as feedback surveys, e-mail complaints, bulletin boards, opinions, and product reviews that are available at websites is a challenging issue. In particular, online product reviews are often unstructured, subjective, and hard to digest within short timeframe. Product reviews exist in various forms across different websites [1]. This work concentrates on two tasks, one is opinion mining and second is opinion prediction.

An artificial neural network (ANN), often called a neural network (NN), is a mathematical model or a computational model based on biological neural networks. It consists of an interconnected group of artificial neurons which communicate by sending signals to each other over a large number of weighted connections and processes information using a connectionist approach to computation. In most cases, ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. In more practical terms, neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

Classification of a large number of user product reviews using neural network techniques into bipolar orientation (i.e. either positive or negative opinion) could help consumers in making their purchasing decisions. Research results along this classification can lead to users' reducing the time on reading numerous threads of text and focusing more on analyzing summarized information. Review mining can be potentially applied in constructing information presentation. For example, review classification could be integrated with search engines to provide statistics such as "350 hits found on Nokia Lumia 720 model review, 80% of which are positive and 20% are negative". Such kind of summarization of product reviews would be even more valuable to customers if the summaries were available in various forms on the web, such as review bulletin boards. Mobile reviews are available in many websites. We gathered our reviews on 2 models Nokia Lumia 720 and LG G3 from the 10 most popular review sites like gsmarena.com.

Opinion Prediction helps the product manufacturers to manufacture the perfect products which may not give any negative feedback in future. If the product is expected to receive negative feedback or reviews the manufacturer need not go for such type of product development. We have used recurrent neural networks to predict future reviews because it stores relavant data about past sales and generalize trends. The results of the experiment have shown that recurrent NNs are good in predicting the opinions.

The main contributions of the paper are

1. Opinion classification by multilayer perceptron NN using generalized delta rule algorithm.
2. Opinion prediction by recurrent NN using back propagation with time algorithm.

The rest of paper is organised as follows. Section 2 details all the related work. Section 3 details the opinion classification. Section 4 shows all the screen shots. Section 5 details opinion prediction domain and algorithm. Section 6 discusses the results.

2 Related Work

Jebaseeli et al. [2], built an opinion mining system for M-Learning reviews, to extract, determine whether the opinions and reviews, are positive or negative. Classification of opinion documents as blogs or news is addressed by the text mining community very well [3,4,5,6]. Several methods exist for determining the polarity of a document. Actually, the opinion polarities are often given stated by adjectives [4,5]. The use of adverbs attached to adjectives (for instance, the adverb ”very” attached to the adjective ”promising”) allows to determine the intensity of phrases (a group of words) [6]. For example, P. Turney, et al. [21] proposes an approach based on the polarity of words in the document. The main idea is to compute correlations between both adjectives in the documents and adjectives coming from a seed set. Two seed sets are considered: positive (e.g. good, fortunate, nice, ..) and negative (e.g. bad, inferior, poor, ..). From [7-12] different researchers proposed the different approaches for classifying the opinions into 6 polarities.

The computed output [13] is compared to the known output. If the computed output is correct, then nothing more is necessary. If the computed output is incorrect, then the weights are adjusted so as to make the computed output closer to the known output. This process is continued for a large number of cases, or time-series, until the net gives the correct output for a given input. The entire collection of cases learned is called a "training sample" [14]. In most real world problems, the neural network is never 100% correct. Neural networks are programmed to learn up to a given threshold of error. After the neural network learns up to the error threshold, the weight adaptation mechanism is turned off and the net is tested on known cases it has not seen before. The application of the neural network to unseen cases gives the true error rate [15]. HuiGao et al. [9] used clustering analysis for opinions extraction using the multilayer neural network.

In Kevin [16], has proposed an experiment for opinion expression extraction using deep RNNs, has shown the results outperform the shallow RNN with the same number of parameters. In Graulsel et al. [17] proposed a model for learning artificial earthquakes as an application for bioinformatics sequence classification with variable length instances. In Werbos et al. [18] has explained the key equations of back propagation and how it can be applied to various neural networks of different complexity. In Bengio et al. [19] proposes alternatives to the backpropagation algorithm, to predict sequences using long term context. In [20], Ahmad et al, have used fully connected recurrent NN with backpropagation through time algorithm for speech recognition for Arabic’s alphabet. From [21-25], researches proposed strategies to use neural networks for prediction, text mining.

Neural Network is a collection of natural or artificial neurons that uses for mathematical and computational model analysis. Popular algorithms in neural networks are Back-Propagated Delta Rule Networks (BP) (sometimes known and multi-layer perceptions (MLPs)) and Radial Basis Function Networks (RBF) are both well-known developments of Delta rule for single layer networks (itself a development of the Perception Learning Rule). Both can learn arbitrary mappings or classifications. Further, inputs (and outputs) can have real values. Kohonen clustering Algorithm is used for unsupervised neural networks. Long-Sheng Chen* and Hui-Ju Chiu proposed that [26] study proposed a Neural Network (NN) based index which combines the advantages of the machine learning techniques and information retrieval (semantic orientation indexes) to help companies detecting harmfully negative bloggers' comments quickly and effectively.

Long-Sheng Chen, proposed a new methodology for sentiment classification. He combined two efficient methodologies such as BPN and SO approaches [27]. This study proposed an NN based approach to classify sentiment in blogospheres by combining the advantages of the BPN and SO indexes. Compared with traditional techniques such as BPN and SO indexes, their algorithm performed well. We call that algorithm as BPNSO technique. In this paper we propose an algorithm for sentiment classification with neural networks using gradient decent algorithm. The proposed approach shows its superior performance, not only in classification accuracy, but also in least error rate in some specific range of training set. Experimental results indicated that our proposed NN based index outperforms traditional approach Back-Propagation neural network (BPN) and BPNSO.

3 Opinion Classifications

3.1 Preprocessing

Preprocessing is done same as stated in our previous paper [28]. Data preprocessing is done to eliminate the incomplete, noisy and inconsistent data.

1. *Filtration*:- Data is filtered by removing the urls, usernames, redundant characters in a particular word, questions, special characters.
2. *Tokenization*:- we segmented text by splitting it by spaces and punctuation marks ,and form a bag of words.
3. *Stopword Removal*:-Stopset words were removed from the data.
4. *Stemming*:- We used Porter stemming to stem the terms.
5. *Part-Of-Speech Tagging*:- POS tag for each word in the blog post. The online Part-Of-Speech tagger was used for POS tagging through http://nlpdotnet.com/Services/Tagger.aspx. Generally, adjectives in the text represent emotions such as good, superior etc for positive nature and poor, unfortunate for negative nature. From [29], use wordnet to cross check the adjectives.
6. *Detecting False Positives and False Negative Adjectives*:- The comment is passed through the negation detection stage.
7. Discard rare words by giving a lower limit to the frequency of accepted words equal 3.
8. Group all product reviews into D batches, where each batch size is m.(m is a constant)

3.2 Preparing Input to the Neural Network

The process to prepare input to the NN is depicted in Fig. 1. The steps are given on detail below.

To prepare the input to the neural network, the process uses the following steps.

1. Generally, adjectives in the text represent the opinion of the reviewer. So, in this paper we treat only the adjectives of the review post to determine the reviewer's opinion. As stated in [30], we used TF*PDF algorithm for counting the significance or weights of the terms in the reviews. This helps us now to recognize the most significant adjectives, which explain the nature of review in blog posts. For each batch of reviews calculate the weight of the terms separately using the formula, where the total weight of a term (W_j) is equal to the summation of the weight of the term in each batch of product reviews respectively using the Eq.1. and where $|F_{jb}|$ is calculated using the Eq.2.

$$W_j = \sum_{b=1}^{b=D} |F_{jb}| \exp\left(\frac{n_{jb}}{N_b}\right) \quad (1)$$

$$|F_{jb}| = \frac{F_{jb}}{\sqrt{\sum_{k=1}^{k=K} F_{kb}^2}} \quad (2)$$

where, W_j=Weight of term j; F_{jb}=Frequency of term j in batch b; n_{jb}=Number of reviews in batch b where term j occurs; N_b=Total number of reviews in batch b; k=Total number of terms in a batch; D=number of batches.

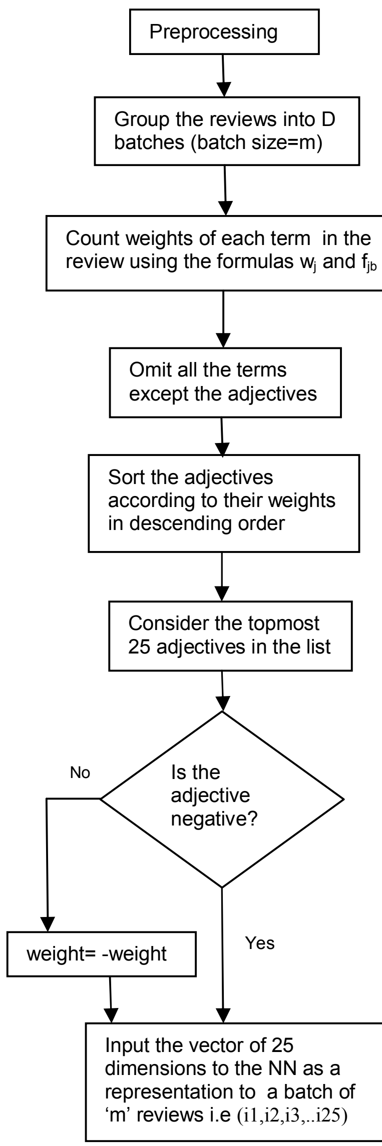


Fig. 1. Process of preparation of input to neural network

2. Consider the batch of reviews, one by one. Each batch is a set of reviews, where each word in the review is tagged by a part of speech. Omit all other parts of speech words, except the ones with adjectives. List all adjectives in the particular batch of review with their corresponding weights calculated in step 1. Sort the values in descending order according to their weights. We expected the topmost weighted adjectives in the list may better represent the opinion of all the reviews in the particular batch. Now consider the 25 adjectives in the topmost order with the topmost weight. If it is a negative adjective make its weight negative.
3. In the pre-processing step, we mentioned that the number of batches are D. Represent each batch of review as a vector of 25 dimensions, where the values of all the dimensions are the weights of the top weighted 25 adjective terms calculated in the step 2(in case of the negative adjective terms, dimension value is also negated). So, this process gives us D number of vectors each of 25 dimensions, where the value of each dimension is an integer.
4. This D vectors are given as an input to the neural network finally as the training data. Each input vector pattern is of form (i_1,i_2,i_3,..i_{25}).

3.3 Designing a Network for an Opinion Classification System

In the *Opinion Miningsystem* we are going to use *Generalized Delta Rule (GDR)* algorithm for training a multilayer perceptron neural network with feed forward interlayer connections. The Weight vectors v and w are randomly assigned. The threshold for i/p layer is 0 and for hidden and output layers it is taken as 1. We used sigmoid activation function. Learning rate (η) and Momentum Coefficient are taken as 0.9 and 0.7 respectively. NN design is shown in Fig. 2.

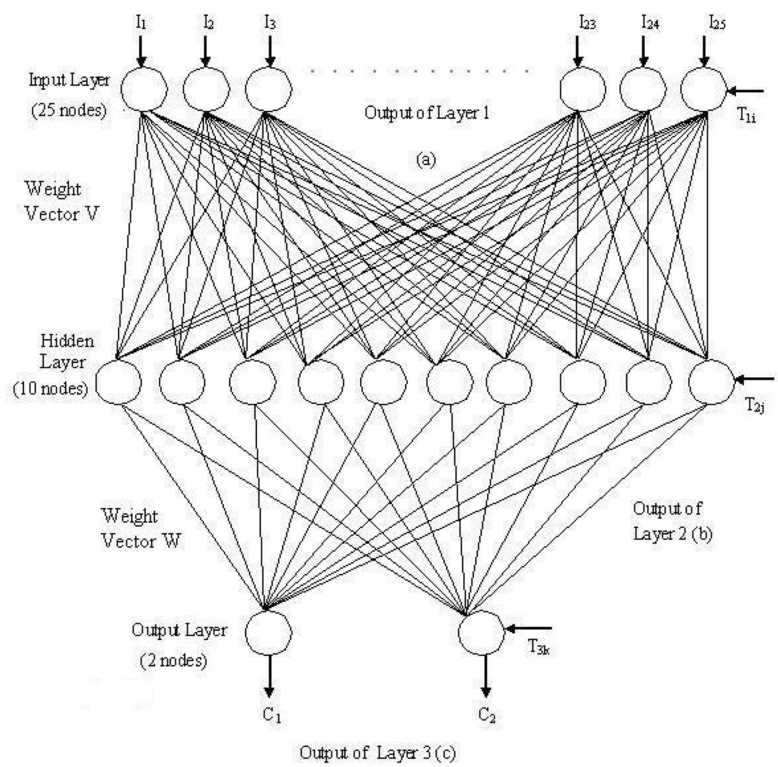


Fig. 2. Design of the neural network

4 Results

Experiments are done in jdk1.1.3

In Table 1 we presented the no: of reviews we considered in the dataset for training as well as for testing.

Table 1. Dataset

	Training		Testing	
	Positive opinions	**Negative opinions**	**Positive opinions**	**Negative opinions**
LG G3	250	150	50	50
Nokia Lumia 720	200	200	50	50

The neural network is with 25 inputs and 2 outputs c1 and c2

The results were interpreted as shown in Table 2.

Table 2. Interpretation of output of the Neural network

C1	C2	Result
1	0	Positive feedback
0	1	Negative feedback

The accuracy was considered as the ratio of number of opinions correctly classified to total number of input opinions.

$$Accuracy = \frac{\textit{No. of Opinions currently classified}}{\textit{No. of input opinions}}$$

Here in Table 3, we present the result as a comparison of previous study BPNSO'NN and the proposed NN. The observation is that the proposed NN performed well for the training set % between 60% and 70%.

Table 3. Performance table

Performance	BPNSO's NN		Proposed NN	
Training set	**Accuracy (Y)**	**Error rate(Y)**	**Accuracy (Y)**	**Error rate (Y)**
50%	69	31	65	35
60%	70	30	74.4	25.6
70%	71	29	72.5	27.5
80%	70	30	63	27
90%	71	29	58	42

We have presented the plotted accuracy of the classification in Fig. 3 and we plotted the error rate for the classification in Fig. 4.

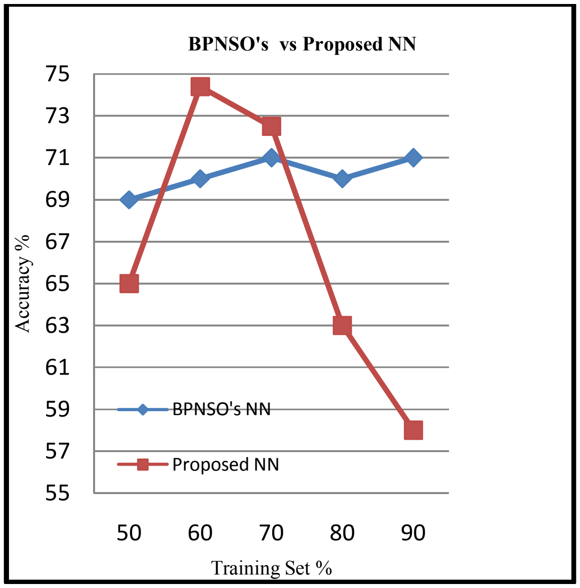


Fig. 3. Accuracy

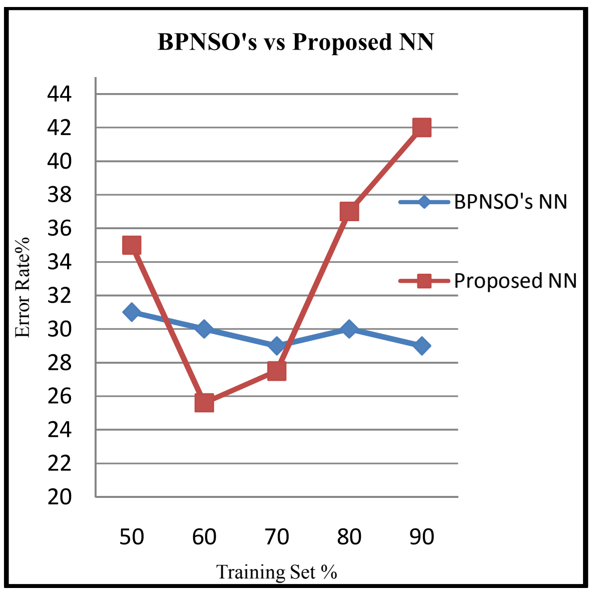


Fig. 4. Error rate

5 Opinion Prediction Problem Domain

Generally the new products are launched basing on the feedback of old products. If one of the feature of product is absolutely bad, then it is obvious it gets bad opinions / reviews from customers. Before manufacturing any product, depending on the current trend, the customer's feelings are assumed by the company people. But, this may not be successful always. There may be certain deviations of the manufactured product from the current trend. So, it is very useful to the manufacturer if a opinion predictor is built.

Let us suppose it takes one year to manufacture a product which includes design, development, and test, release etc. So, the requirement is to predict the opinion one year in advance. So, the recurrent neural network considers one year as one time unit. At each time instance t, the features for the product t were given to the network and the opinion for day t + 1 were obtained as the output of the network.

5.1 Preprocessing

1. List out all product features.
2. Pre-processing and Preparing input to Recurrent Neural Network

Do same as described in section 3.1 and section 3.2.

5.2 Designing a Network for an Opinion Prediction System

Opinion Prediction is done with the recurrent neural network using the back propagation through time (BPTT) algorithm.

5.2.1 Back propagation through time algorithm

This algorithm is executed in 2 steps.

Step 1: Unfold the network through time.

The recurrent neural network is a combination of 2 feed forward neural networks f and g. To train any recurrent network it should be unfolded according to time till n units. For our experimentation, the recurrent neural network's depth was considered as n=5. So we unfolded 'f' as f_1, f2, f3, f4, f5 . For neural network 'f', if the input is a_i then the output is a_{i+1}. The recurrent neural network is presented in Fig. 5. The Fig. 6 shows the unfolded RNN.

Step 2: Apply typical back propagation algorithm to the unfolded network.

One initial vector is required called a_0. The vector values are randomly assigned.

The training data for Back-propagation through time algorithm should be always in pairs, (x_0,y_0), (x_1,y_1), (x_2,y_2), .. (x_{n-1},y_{n-1}). Where x_i, y_i are inputs and outputs respectively. The figure shows the unfolded RNN.

Each training pattern consists of 7-tuple (a_t, x_t, x_{t+1}, x_{t+2}, x_{t+3}, x_{t+4} , y_{t+4}) . Then training is similar to training done with back-propagation algorithm, but only difference is each epoch runs for y_t times.

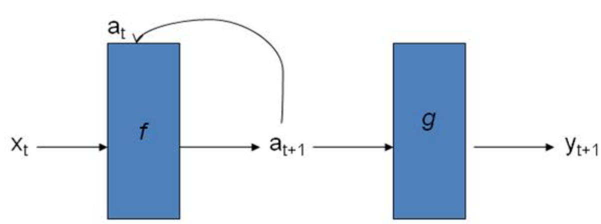


Fig. 5. Recurrent neural network

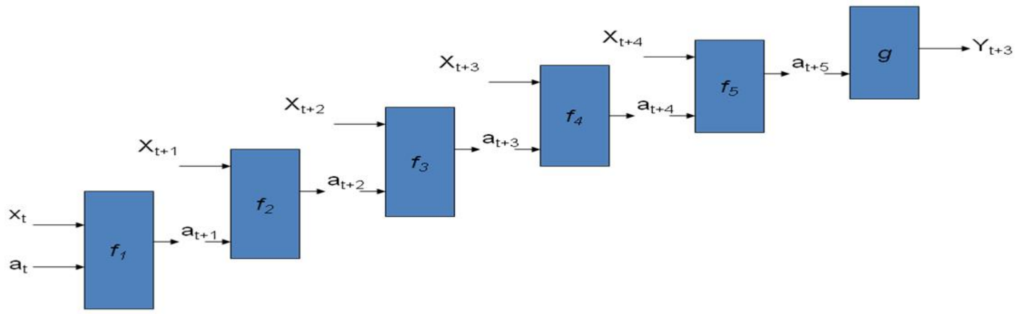


Fig. 6. The unfolded recurrent neural network with depth=5

From section.5 in the preprocessing step we obtained each input vector pattern of the form (i_1,i_2,i_3,..i_{25}). Now at x_t consider i_1 to i_5 , at x_{t+1} consider i_6 to i_{10}, at x_{t+2} consider i_{11} to i_{15}, at x_{t+3} consider i_{16} to i_{20}, at x_{t+4} consider i_{21} to i_{25}.

The output is whether the feedback is positive or negative .i.e output close to 1 was considered as positive feedback and the outputs closer to 0 were considered as negative feedback.

The network parameters are

a) The number of nodes in the recurrent neural network If the number is veryl ess, it could not properly generalize the problem, if the numberis more than enough, it could over fit the problem.
(b) The learning rate of the neural network – Initially, we couldn’t determine the learning rate, we tried with various learning rates and finally decided at the learning rate of 0.001 for positive opinions and 0.01 for negative opinions.
(c) The momentum rate 0.7 is used to remove high frequency variations in the error surface.

6 Screen Shots

Experiments are done in jdk1.1.3. The training weights, vector v and vector w are shown in the following subsections6.1, 6.2 and 6.3 of screenshots. The testing results for negative and positive feedback in screenshots are also presented in the following subsections 6.4 and 6.5.

6.1 Training Weights are shown in Fig. 7.

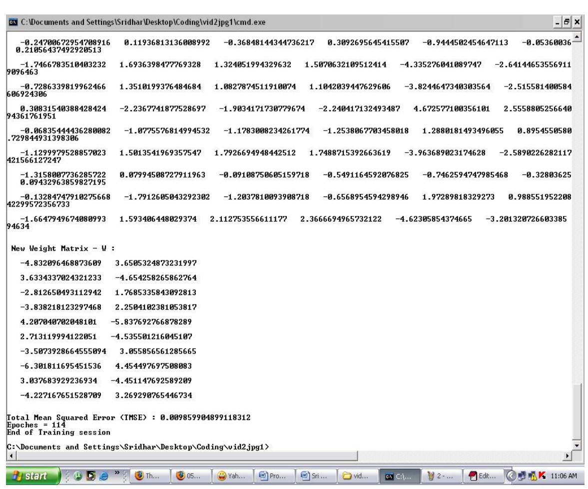
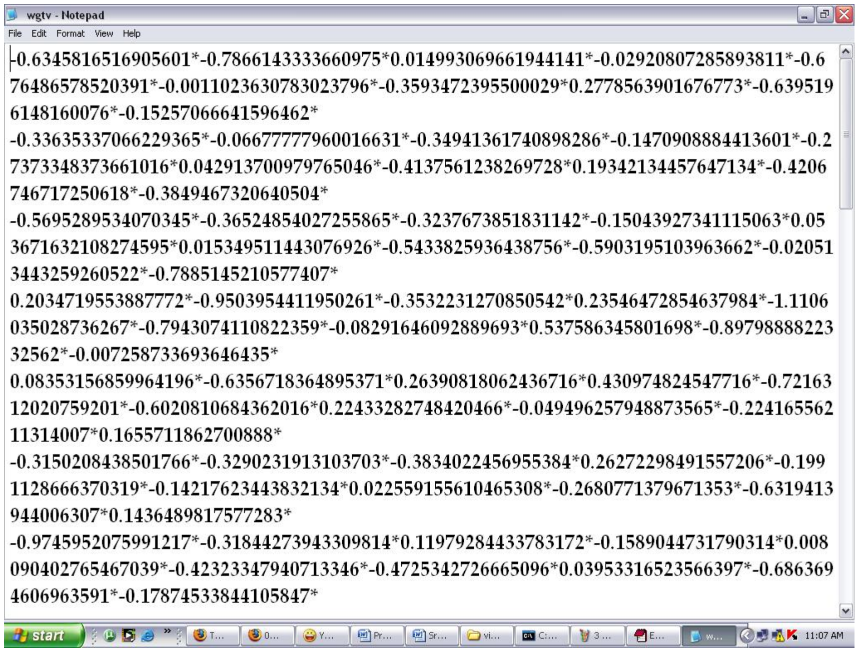


Fig. 7. Trained weights

6.2 Weight Vector V is shown in Fig. 8.

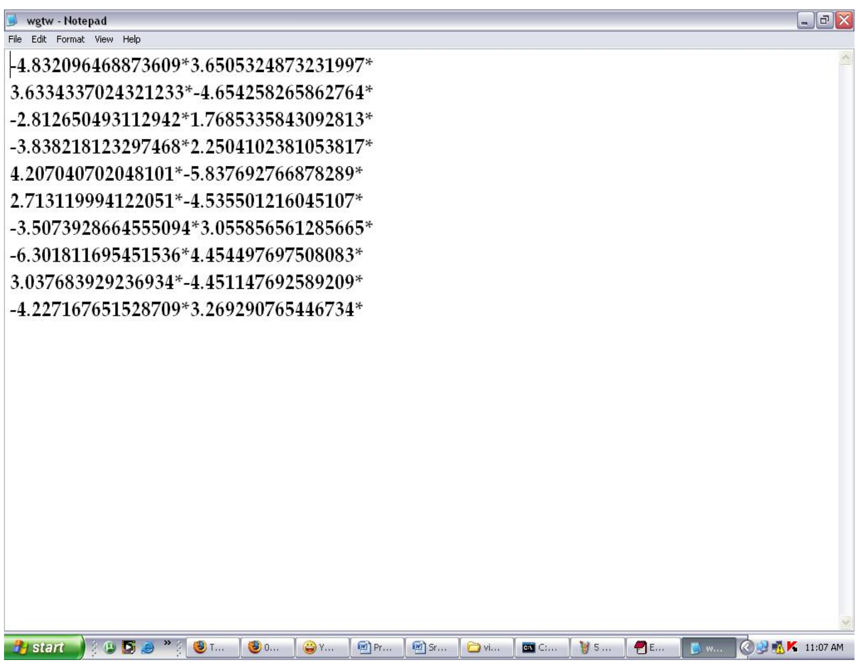

wgtv - Notepad

File Edit Format View Help

-0.6345816516905601*-0.7866143333660975*0.014993069661944141*-0.02920807285893811*-0.6
76486578520391*-0.0011023630783023796*-0.3593472395500029*0.2778563901676773*-0.639519
6148160076*-0.15257066641596462*
-0.33635337066229365*-0.06677777960016631*-0.34941361740898286*-0.1470908884413601*-0.2
7373348373661016*0.042913700979765046*-0.4137561238269728*0.19342134457647134*-0.4206
746717250618*-0.3849467320640504*
-0.5695289534070345*-0.36524854027255865*-0.3237673851831142*-0.15043927341115063*0.05
3671632108274595*0.015349511443076926*-0.5433825936438756*-0.5903195103963662*-0.02051
3443259260522*-0.7885145210577407*
0.2034719553887772*-0.9503954411950261*-0.3532231270850542*0.23546472854637984*-1.1106
035028736267*-0.7943074110822359*-0.08291646092889693*0.537586345801698*-0.89798888223
32562*-0.007258733693646435*
0.08353156859964196*-0.6356718364895371*0.26390818062436716*0.430974824547716*-0.72163
12020759201*-0.6020810684362016*0.22433282748420466*-0.049496257948873565*-0.224165562
11314007*0.1655711862700888*
-0.3150208438501766*-0.3290231913103703*-0.3834022456955384*0.26272298491557206*-0.199
1128666370319*-0.14217623443832134*0.022559155610465308*-0.2680771379671353*-0.6319413
944006307*0.1436489817577283*
-0.9745952075991217*-0.31844273943309814*0.11979284433783172*-0.1589044731790314*0.008
090402765467039*-0.42323347940713346*-0.4725342726665096*0.03953316523566397*-0.686369
4606963591*-0.17874533844105847*

Fig. 8. Weight vector V

6.3 Weight Vector W is shown in Fig. 9.


wgtw - Notepad

File Edit Format View Help

-4.832096468873609*3.6505324873231997*
3.6334337024321233*-4.654258265862764*
-2.812650493112942*1.7685335843092813*
-3.838218123297468*2.2504102381053817*
4.207040702048101*-5.837692766878289*
2.713119994122051*-4.535501216045107*
-3.5073928664555094*3.055856561285665*
-6.301811695451536*4.454497697508083*
3.037683929236934*-4.451147692589209*
-4.227167651528709*3.269290765446734*

Fig. 9. Weight vector W

6.4 Testing for a negative feedback of a new batch of reviews is shown in Fig. 10.

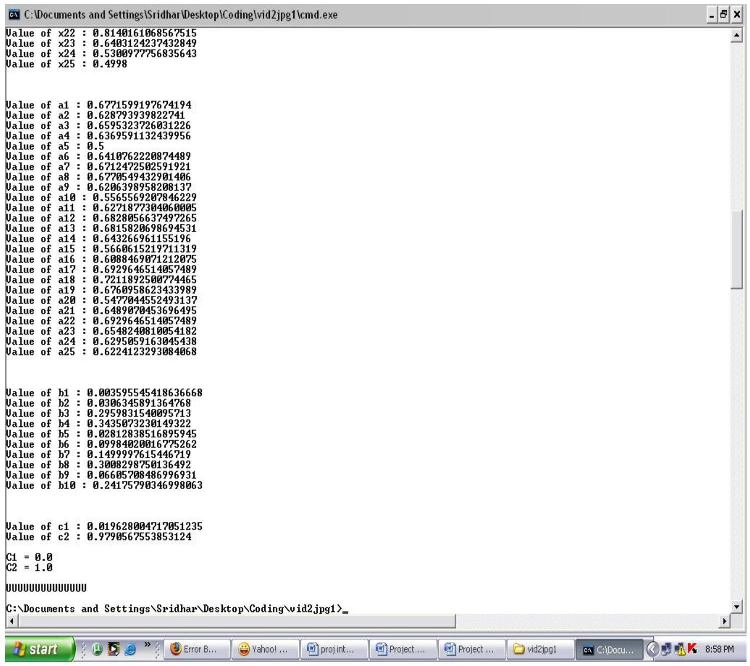


Fig. 10. Testing for a negative result of a new batch of reviews

6.5 Testing for a positive feedback of a new pattern is shown in Fig. 11.

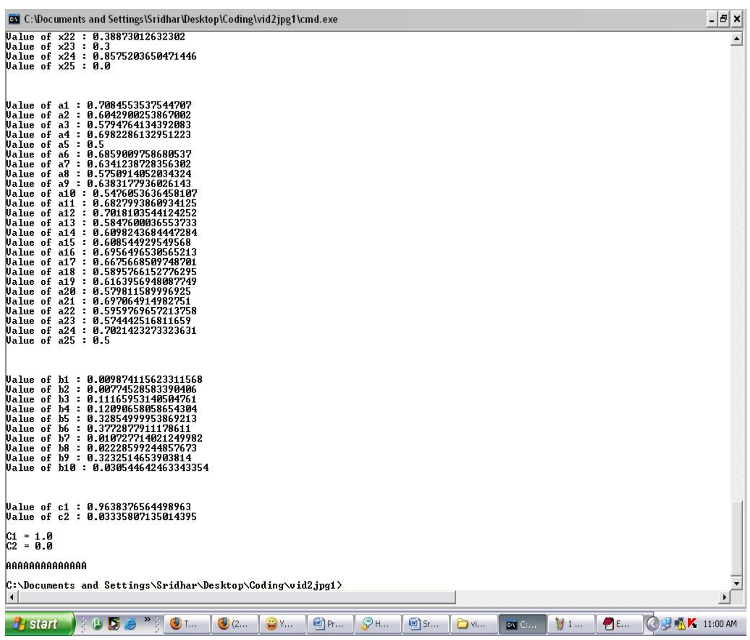


Fig. 11. Testing for positive result of a new batch of reviews

7 Conclusion

We proposed an algorithm for opinion prediction to predict if the product is going to give positive feedback or negative. This is helpful for the product designer/manufacturers such that they will develop only products of good feedback. Results showed there was 65% accuracy in prediction.

Future works may be manifold. First, our method depends on good quality of reviews extracted from blogs. We want to extend our training corpora method by applying text mining approaches on collected reviews in order to minimize lower noisy texts. Second, in this work we focused on adjectives, we plan to extend the extraction task to other categories. Third would be extending these results by using Deep Multilayer Neural Networks with more than two hidden layers for determining text sentiment.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Kushal Dave, Steve Lawrence, David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. Presented at the 12th International Conference on World Wide Web, Budapest, Hungary; 2003.

[2] A. Nisha Jebaseeli, Kirubakaran. Neural network classification algorithm with M-Learning reviews to improve the classification accuracy. International Journal of Computer Applications (0975 – 8887). 2013;71(23).

[3] Turney PD, Littman M. Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems. 2003;21(4):315–346.

[4] Yang H, Si L, Callan J. Knowledge transfer and opinion detection in the trec 2006 blog track. In Notebook of Text REtrieval Conference; 2006.

[5] Esuli A, Sebastiani F. Page Ranking wordnet synsets: An application to opinion mining. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), Prague, CZ. 2007;424–431.

[6] Benamara F, Cesarano C, Picariello A, Reforgiato D, Subrahmanian VS. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In Proceedings of ICWSM conference; 2007.

[7] Michel Planti´e, Mathieu Roche, G´erard Dray, Pascal Poncelet. Is a voting approach accurate for opinion mining? Centre de Recherche LGI2P, Site EERIE Nˆımes, ´Ecole des Mines d'Al`es – France.

[8] Nan Li, Desheng Dash Wu. Using text mining and sentiment analysis for online forums hotspot detection and forecast. Decision Support Systems, Elsevier. 2010;48:354–368.

[9] HuiGao, Xu Chen, Yan Fu, Identifying web public sentiment of opinions, proceedings of the 2009. International Symposium on Web Information Systems and Applications (WISA'09)Nanchang, P. R. China. 2009;270-272.

[10] Wei Jin, Hung Hay Ho, Rohini K. Srihari. Opinion miner: A novel machine learning system for web opinion mining and extraction, KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009;1195-1204.

[11] Anitha N, Anitha B, Pradeepa S. Sentiment classification approaches – A review. International Journal of Innovations in Engineering and Technology (IJIET). 2013;3(1):22-31. ISSN: 2319 – 1058.

[12] Yaakub MR, Li Y, Algarni A, Peng B. Integration of opinion into customer analysis model. In Boissier, Olivier & Benatallah, Boualem (Eds.) The 2012 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshops, IEEE Computer Society Conference Publishing Services (CPS), Macau, China. 2012;164-168.

[13] Cover TM, Thomas JA. Elements of Information Theory. John Wiley; 1991.

[14] Connor Jerome T, Douglas Martin R, Les E. Atlas. Recurrent neural networks and robust time series prediction. Neural Networks, IEEE Transactions. 1994;5(2):240-254.

[15] Venugopal V, Baets W. Neural networks and their applications in marketing management. Journal of Systems Management. 1994;45:16-16.

[16] Duh Kevin. Deep Learning for Natural Language Processing and Machine Translation; 2014.

[17] Grau Isel, et al. Backpropagation through time algorithm for training recurrent neural networks using variable length instances. Computación y Sistemas. 2013;17(1):15-24.

[18] Werbos Paul J. Backpropagation through time: What it does and how to do it. Proceedings of the IEEE. 1990;78(10):1550-1560.

[19] Bengio Yoshua, Paolo Frasconi. Credit assignment through time: Alternatives to backpropagation. Advances in Neural Information Processing Systems. 1994;75.

[20] Ahmad Abdul Manan, Saliza Ismail, Samaon DF. Recurrent neural network with backpropagation through time for speech recognition. Communications and Information Technology. ISCIT 2004. IEEE International Symposium on. 2004;1. IEEE, 2004.

[21] Yochanan Shachmurove. Department of Economics, The City College of the City, University of New York and The University of Pennsylvania, Dorota Witkowska, Department of Management, Technical University of Lodz CARESS Working Paper #00-11Utilizing Artificial Neural Network Model to Predict Stock Markets; 2000.

[22] Hassan AKI, Abraham A. Modeling consumer loan default prediction using ensemble neural networks. Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on. 2013;719,724.

[23] Govindarajan M, Chandrasekaran RM. Classifier based text mining for neural network, AIKED'08 Proceedings of the 7th WSEAS International Conference on Artificial intelligence, knowledge engineering and data bases. World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA ©. 2008;476-481.

[24] Christopher W. Sherry. A regression-based training algorithm for multilayer neural networks, thesis. RIT Scholar Works. 2012;7.

[25] Furferi R, Gelli M. Yarn strength prediction: A practical model based on artificial neural networks (2010) Advances in Mechanical Engineering; 2010. Art. no. 640103.

[26] Anuj Sharma, Shubhamoy Dey. Document level sentiment analysis approach using artificial network and sentiment lexicons, in ACM SIGAPP Applied Computing Review. 2012;12(4): 67-75.

[27] Long-Sheng Chen, Hui-Ju Chiu. Developing a Neural Network based Index for Sentiment Classification", in Proceedings of the International Multi Conference of Engineers and Computer Scientists. 2009;1. IMECS 2009, March 18 - 20, 2009, Hong Kong.

[28] Santhi Priya Polisetty, Madhuri Polisetty, Rao TV. An approach for emotion identification from weblog corpora. International Journal of Computer Technology and Applications. 2014;5(1) .

[29] Miller GA, Beckwith R, Fellbaum C, Gross D, Miller KJ. Introduction to WordNet: An on-line lexical database. International Journal of Lexicography. 1990;3(4):235–244.

[30] KhooKhyou Bun, Mitsuru Ishizuka. Topic extraction from news archive using TF*PDF algorithm, proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002, 0-7695-1766-8. 2002;73-82.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=1030&id=6&aid=8452