IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

Development of a Content Based Recommender Using Dynamic Artificial Neural Network

Md Zahidul Islam[1], Feroza Naznin[2], Asaduz Zaman[3]

Associate Professor, Dept. of Information & Communication Engineering, Islamic University, Kushtia, Bangladesh [1]

Research Asst., Dept. of Information & Communication Engineering, Islamic University, Kushtia, Bangladesh[2, 3]

Abstract: In recent years, we've been facing a boost in advancement of Artificial Intelligence (AI). It's continuously setting benchmark of our thought level of which AI is possible. Although using Artificial Neural Network (ANN) to achieve AI is somewhat less explored area primarily because of complexity of the training process, AI is moving towards ANN approach recently with many sophisticated learning algorithms. In this paper, we tried to develop a content based recommender which can be used as a personal assistant for categorizing and making recommendation on news articles. Our primary goal was to evaluate the possibility of such a recommender which can be dealt with a multiple layer network which has dynamic number of input nodes. We were also evaluating the performance of the system under various configurations. We are calling the system dynamic artificial neural network (DANN).

Keywords: recommender; artificial intelligence; artificial neural network; content based recommender; dynamic artificial neural network.

I. INTRODUCTION

The term Artificial Intelligence was first introduced by John McCarthy et el. [1] where they defined the term as "the science and engineering of making intelligent machines". This definition remains the most basic definition of Artificial Intelligence. Over the times passed, AI becomes more and more advanced.

The irony is that, Artificial Neural Network was introduced decades ago then that of Artificial Intelligence, but never got proper attention of mainstream researchers.

Artificial Intelligence is now becoming the digital personal assistant and the area where it can contribute is vast. With advanced AI, we are able to design such systems which can perform as true personal assistant in all means. The job of this personal assistant includes but not limited to making schedule, classifying and filtering mails, predicting stock price etc.

Automated textual content based classification has been used in many areas including document filtering, personalized news filtering, spam filter, web site categorization, consumer review etc. We are trying to create such a system which will be able to recommend web content (primarily news article in this case) based on previous behaviour pattern of the user of the system. The system is fully dynamic in nature. It will learn the behaviour pattern which in our case is news reading pattern of the user and will try to predict whether the article should be recommended to the user or not based on learnt user behaviour pattern. We are using multiple layer Dynamic Artificial Neural Network (DANN) for predicting and recommending articles.

In section II, we discussed about the background of our proposed model. Section III describes our proposed model. Section IV demonstrates performance of the model. And lastly, section V concludes everything.

II. BACKGROUND

Automated textual categorization problem has been researched for more than 50 years. In 1961, Melvin Earl. Maron first introduced the topic [2]. Later on, many researchers worked on textual categorization shading light on different aspect of the problem. Creecy et. el. first introduced machine learning to textual categorization problem. After that, several methods including Linear Discriminant Analysis (LDA), Latent Semantic Indexing (LSI), Linear Least Squares Fit (LLSF), k-Nearest Neighbours (kNN), Support Vector Machine (SVM) etc. all with their limitations and performance improvement over other methods in some specific performance measurement.

Dynamic Artificial Neural Network (DANN) is a new entity in the list. Ghiassi et. el. [3] [4] [5] used dynamic artificial neural network for predicting nonlinear event and time series events. They called the system DAN2, and the idea of dynamicity was confined in hidden layers. They achieved a good performance over traditional ANN approach. In traditional Feed Forward Back Propagation (FFBP) neural network, the system configuration remains constant where in DANN approach, the system configuration might change to make the system flexible with the nature of problem introduced. DAN2 offers flexibility in hidden layer number and nodes in hidden layer which makes it a good choice in text classification problems.

But our problem here is unique in nature. Our system model permits the use of dynamic number of input nodes which wasn't introduced in Ghiassi's methodology. Initially, the system doesn't know about the number of nodes in input layer. The number of nodes in input layer increases with time when the user of the system provide feedback on recommendation the system done. With each feedback the number of input node changes. To prevent

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

the system of growing infinitely, we only use certain number of recent feedbacks. For this reason, we are using modified DAN2 approach which we are calling DANN.

III.NETWORK MODEL

As we stated before, we are using DANN to represent our network model which is a modified version of DAN2 model. The overall architecture is shown in fig. (1).

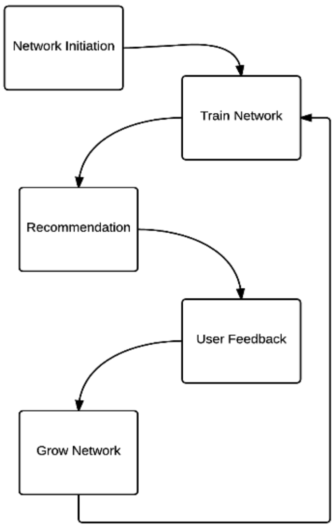

Fig.1. Network Model Overview

A. Network Initiation
Firstly, we'll initiate the network with random weights and biases. For our purpose, we've used a network with one hidden layer of specified number of nodes and one output layer of 1 node. As we're only expecting prediction of whether the article it is reading right now is recommended or not and which its confidence is on that recommendation, one output node is enough for us.
Network initiation is a very much important step as the architecture we define at initiation will have impact on the performance of the network. Using multiple hidden layers can help us providing some high level information which we could use to provide better accuracy. But as the system is dynamic in nature and the whole network is rebuilding based on user feedback, the network will perform very slowly. Its response rate will not be suitable for our intended application.

B. Train Network
Although we're using DANN as our network model, for training purpose, we'll use Feed Forward Back Propagation (FFBP) algorithm to train our network with Gradient Descent algorithm. Back Propagation algorithm works in two steps.

1) Phase 1- Propagation:
In this phase, the input pattern is passed through the network to determine the output. Then the output is back propagating through the network to determine the deltas.
2) Phase 2- Weight Update:
In this phase, deltas of next layer are used to determine the gradient of the weight and actual weight is updated using this gradient.

The backpropagation algorithm has some basic formulas. For computing activation, we are using sigmoid function as it shows non-linearity which is very important and also very easy to differentiate. Activation function can be written as (1).

$$a_j^l = \sigma\left(\sum_{k=1}^{n} w_{j\,k}^l a_k^{l-1} + b_j^l\right) (1)$$

Where a_j^l is the activation of j^{th} neuron of layer l, $b_j^l$is the bias of j^{th} neuron of layer l and $w_{j\,k}^l$ is the weight of neuron j to neuron k in layer l. one thing to note in this function is that, the activation of next layer depends on activation of previous layer which is input layer for second layer.
If we consider,

$$z^l = w^l a^{l-1} + b^l (2)$$

Which is basically a matrix operation, then (1) becomes,

$$a^l = \sigma(z^l) \qquad (3)$$

The activation function used here is sigmoid function which is non-linear in nature and very easy to differentiate. Sigmoid function is displayed in (4).

$$\sigma(x) = \frac{1}{1+e^{-x}} (4)$$

If we differentiate this function, we'll get,

$$\frac{d}{dx}\sigma(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \sigma(x) \times (1-\sigma(x)) \quad (5)$$

Cost function is another issue we have to consider. For this case, we'll use quadratic cost function which can be defined as (6).

$$C = \frac{1}{2}\sum_j (y_j - a_j^L)^2 \quad (6)$$

Where $a_j^L$is the activation value of last layer which is actually output layer.
According to Michael A. Nielsen [7], backpropagation algorithm has four fundamental equations which are as follows.
To compute the error at output layer, we'll use equation (7).

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L) \quad (7)$$

Where $\delta_j^L$is the error at output layer. $\frac{\partial C}{\partial a_j^L}$ is actually the cost at output layer which is simply the difference between the activation at output layer and expected outcome at output.
To compute error at hidden layer in terms of next layer, we have to use the following equation

$$\delta^l = \left((w^{l+1})^T \delta^{l+1}\right) \odot \sigma'(z^l) (8)$$

Where $(w^{l+1})^T$is the transpose of weight matrix of layer l, $\delta^{l+1}$is the output delta error of next layer and $\sigma'(z^l)$ is the differentiated value of weighted input for layer l.

Now to determine the rate of change of the cost with respect to any bias in the network, we will use the following equation.

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l (9)$$

And to determine the rate of change of the cost with respect to any weight in the network is,

$$\frac{\partial C}{\partial w_{j,k}^l} = a_k^{l-1}\delta_j^l \qquad (10)$$

C. Recommendation
In this phase, using the training, the network tries to predict the likelihood that the news article is actually recommendable to the user. We've used sigmoid function as activation function for prediction which is non-linear in nature. The recommendation task uses (11) to perform.

$$f(x) = \begin{cases} \text{yes}, & x \geq 0.5 \\ \text{not}, & x < 0.5 \end{cases} \qquad (11)$$

The system will use both function output to determine if the article in question is recommended or not and the value of x to provide its confidence about the decision.

D. User Feedback
As the system evolves, user feedback is a very important feature for this system to work. Using user feedback, the system can evolve its configuration to match user behavior pattern.
The user is normally asked to like or dislike an article he/she has been accessed which are tracked by the system. Conceptually, the system generates the list of articles which are likely to be liked by user. So, when a user likes and article, it effects the system a very little. But when a user dislikes an article, it means a lot to the system. The system is supposed to configure itself so that this types of articles does not appear in the recommended article list.

E. Grow Network
Based on user recommendation, the network grows. It doesn't matter what it the response of the user, the system is designed to grow after gaining a positive or negative feedback from the user.
For each new feedback article, the N most frequent words appeared in the article is chosen. With this N frequent word, we've considered two options,

1. To generate N hidden nodes and N input nodes.
2. To generate N input nodes.

As we developed the system to be fully connected, no matter how promising first approach looks like, it generates a lot of neural connection. We soon find that the growth rate of our network is exponential. So we opted for second option where growth of the network is only linear.
When the user finds it worthy to re-train the network after providing enough feedback, user can instantiate re-training schedule for the system.

IV. PERFORMANCE

Measuring performance of this system is a hard job. Because the system is dynamic and most of the configuration is intervened by the user. But as we are only evaluating the possibility of designing such dynamic artificial neural network which could be used as a recommender system, the measure of performance is simplified enough.
First, we'll be evaluating the performance of the system by the means of positive feedback using Table 1.

TABLE 1 FEEDBACK PERFORMANCE

Training	Input nodes	Total Articles	Negative Feedback	Performance
T#1	10	18	12	33.33%
T#2	370	42	14	66.67%
T#3	976	32	7	78.12%
T#4	1543	56	14	75.00%
T#5	1786	36	6	83.33%

There are some interesting trends in Table.1 data. T#1 data is totally random which we generated at network initiation. This has nothing to do with user input. So the performance is very poor. But we can see that, as user retrains the network, the performance increases accordingly.

Another important factor to be noted that, number of nodes doesn't increased compared to increase in number of training articles. It's because, we eliminated words that are already in our list and before elimination we boosted up their connection weight according to their frequency so that they can contribute more than that of they could previously. In same manner, when a user provide negative feedback, we decrease their connection weight.

Table 2 depicts the performance measurement of our network in both with hidden layer nodes fixed and hidden layer nodes dynamic configuration. This data can provide the insight of what is the cost benefit relationship of using both ways. From the table, we can see that the number of connection of the network grows exponentially when we use both input and hidden layer dynamic which we can see from Network Connection column of the table. As the number of connection weights become very large after only a few user feedback, the processing time needed to recommend article become much higher over time. In C1 configuration, the N_C becomes near 30,000 for only 16 articles where in C2, 18 articles generate only 7421 N_C which is 4 times smaller than that f C1 Configuration.

Another important feature to note from Table 2 is that, in configuration C1 where we set both input and hidden layer nodes dynamic, the performance is higher in compared to that of C2 configuration where we set only input layer nodes to be dynamic.

Measuring the trade-off, we decided to use only input layer nodes to be dynamic. Because, response in time is the key feature when we are trying to recommend something to end-user. If the recommendation becomes wrong, there are always ways to reconfigure the system by training it.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

TABLE 2 TRAINING AND PREDICTION PERFORMANCE

	C1: Input and Hidden Layer Nodes Dynamic							C2: Input Layer Nodes Dynamic, Hidden Layer Fixed						
	N_A	N_{IN}	N_{HN}	N_C	T_T (Minute)	T_{AT} (Seconds)	P (%)	N_A	N_{IN}	N_{HN}	N_C	T_T (Minute)	T_{AT} (Seconds)	P (%)
T#1	3	24	24	601	2.3	0.12	45.65	18	370	20	7421	13.80	2.577	66.67
T#2	12	108	108	11,773	22.472	4.532	74.54	42	976	20	19,541	64.65	6.854	78.12
T#3	8	52	52	2,575	12.43	0.65	69.23	32	1543	20	30,861	145.80	11.227	75.00
T#4	16	169	169	28,731	48.87	9.32	82.34	56	1786	20	35,741	196.24	15.432	83.33

Here, N_A=Number of Articles, N_{IN}=Number of Input Nodes, N_{HN}=Number of Hidden Nodes, N_C=Number of connection created for fully connected network, T_T=Training Time Required in Minutes, T_{AT}=Average Prediction Time Required in Seconds and P=Feedback Performance

V. CONCLUSION

Content based recommender is very difficult to achieve with traditional Neural Network architecture. Feature list for this problem becomes very large which produces difficulty in normal network flow. To overcome the challenge, we had to think differently. Our goal was to evaluate the possibility and if possible feasibility of using DANN as content recommender.

As we were only focusing on some special agenda, we did not focus on improving performance by using some extra measure. Some well-known data mining techniques can clean the data which could produce significant improvement on performance. To become a true personal recommender, it has to go a very long way which includes both technical and algorithmic measures to be taken. But as final words, we can say that it's possible to use DANN effectively as a personal recommender which can recommend based on user behavior.

ACKNOWLEDGMENT

We acknowledge the support of CVIIL in completing the research work. CVIIL stands for Computer Vision and Intelligent Interfacing lab (CVIIL) which is one of the prominent Research Labs of Dept of Information and Communication Engineering, Islamic University, Kushtia.

REFERENCES

[1] McCarthy, John, and Patrick J. Hayes. "Some philosophical problems from the standpoint of artificial intelligence." Readings in artificial intelligence (1969): 431-450.
[2] Maron, Melvin Earl. "Automatic indexing: an experimental inquiry." Journal of the ACM (JACM) 8.3 (1961): 404-417.
[3] Creecy, R. H., Masand, B. M., Smith, S. J., & Waltz, D. L. (1992). Trading MIPS and memory for knowledge engineering. Communications of the ACM, 35(8), 48-64.
[4] Ghiassi, M., H. Saidane, and D. K. Zimbra. "A dynamic artificial neural network model for forecasting time series events." International Journal of Forecasting 21.2 (2005): 341-362.
[5] Ghiassi, M., J. Skinner, and D. Zimbra. "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network." Expert Systems with applications 40.16 (2013): 6266-6282.
[6] Ghiassi, M., Olschimke, M., Moon, B., & Arnaudo, P. (2012). Automated text classification using a dynamic artificial neural network model. Expert Systems with Applications, 39(12), 10967-10976.
[7] Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015

BIOGRAPHIES

Asaduz Zaman has completed his M.Sc. in 2015 from the Department of Information & Communication Engg, Islamic University, Kushtia, Bangladesh. He completed his B.Sc. from the same university at 2013. He is currently working in Artificial Neural Network. He also has research interest in Visual Object Tracking, Gesture Recognition and Medical Imaging. Currently, he is a research assistant in Computer Vision & Intelligent Interfacing Lab (CVIIL) of Department of Information & Communication Engineering, Islamic University, Kushtia, Bangladesh.

Feroza Naznin is currently doing her M.Sc. in the Dept of Information and Communication Engineering, Islamic University, Kushtia. She has completed her B.Sc. from the same university at 2014. As present she is a research assistant in Computer Vision & Intelligent Interfacing Lab (CVIIL) of Department of Information & Communication Engineering, Islamic University, Kushtia, Bangladesh. Her research interests are artificial neural network, Computer Vision, Image processing etc.

Md Zahidul Islam has received his B.Sc. and M.Sc. degree from the Department of Applied Physics and Electronic Engineering, University of Rajshahi (RU), Bangladesh in 2000 and 2002 respectively. In 2003, he has joined as a Lecturer in the Department of Information & Communication Engineering, Islamic University (IU),Kushtia, Bangladesh. He has done his Ph.D research on Visual Object Tracking System from the Department of Computer Engineering at Intelligent Image Media and Interface Lab, Chonnam National University (CNU), South Korea. In August 2011, Dr. Islam has been successfully awarded his PhD from the same department. Besides, he has done his research internship in 3D Vision Lab in Samsung Advanced Institute of Technology (SAIT), Suwon, South Korea. Dr. Islam has also other research interests like computer vision, 3D object, human and motion tracking and tracking articulated body and genetic algorithm etc. Currently he is an Associate Professor and head of the Department of Information & Communication Engineering, Islamic University (IU), Kushtia, Bangladesh.