INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

FORECASTING OF DAILY NEED PRODUCT USING ARTIFICIAL NEURAL NETWORKS

MR. NITIN P. BHONE, MR. D. S. MANKAR, MR. A. J. MORE

Assistant Professor, AISSMS-IOIT, Pune, MH (India).

Accepted Date: 05/10/2015; Published Date: 01/11/2015

Abstract: - The performance of the artificial neural network (ANN) model, i.e. standard feed-forward neural network trained with Levenberg–Marquardt back propagation algorithm, was examined for forecasting daily need product. The 70% data is used to train the neural network, while the 30% of data is used to test the performance of the model. The model efficiency and accuracy were measured based on the mean square error (MSE) and test MSE. The model provided the best fit and the predicted trend followed the observed data closely. Thus, for precise and accurate forecasting, ANN appears to be a promising tool.

Keywords: ANN, Forecasting

Corresponding Author: MR. NITIN P. BHONE

Access Online On:

www.ijpret.com

How to Cite This Article:

Nitin P. Bhone, IJPRET, 2015; Volume 4 (3): 56-63

PAPER-QR CODE

INTRODUCTION

Prediction of daily need product is an important issue in collaborative planning & replenishment. Nowadays artificial neural networks (ANNs) have been popularly applied to supply demand chain problems such as prediction of collaborative supply chain planning, collaborative demand planning. An ANN model is a computer model whose architecture essentially mimics the learning capability of the human brain. The processing elements of an ANN resemble the biological structure of neurons and the internal operation of a human brain. Many simple interconnected linear or nonlinear computational elements are operating in parallel processing at multiple layers. In some applications it has been specified that ANNs have limitations for learning the data patterns. They may perform inconsistently and unpredictable because of the complex daily need data used. Sometimes data is so voluminous that learning patterns may not work. Continuous and large volume of data needs to be checked for redundancy and the data size should be decreased for the algorithm to work in a shorter time and give more generalized solutions.

This paper presents a case example of a supplier and a retailer collaborating to increase the retailer's forecasting accuracy for new product introductions. The approach is based on creating one forecast that is then shared within the supply chain. Three other streamlined approaches to planning and forecasting collaboration are also presented on a more general level. Finally, the paper discusses how the product life-cycle model can be used to select and combine the most suitable approaches to collaboration in different market situations

CHARACTERISTICS OF NEURAL NETWORKS

- The NNs exhibit mapping capabilities, that is, they can map input patterns to their associated output patterns.
- The NNs learn by examples. Thus, NN architectures can be 'trained' with known examples of a problem before they are tested for their 'inference' capability on unknown instances of the problem. They can, therefore, identify new objects previously untrained.
- The NNs possess the capability to generalize. Thus, they can predict new outcomes from past trends.
- The NNs are robust systems and are fault tolerant. They can, therefore, recall full patterns from incomplete, partial or noisy patterns.
- The NNs can process information in parallel, at high speed, and in a distributed manner.

ARTIFICIAL NEURAL NETWORK DESIGN

An Artificial intelligence (AI) system is widely accepted as a technology offering an alternative way to tackle complex and ill-defined problems. An artificial neural network (ANN) was used for the long-term performance prediction. Artificial Neural Network is a system loosely modeled on the human brain. A biological neuron is shown in fig.1.

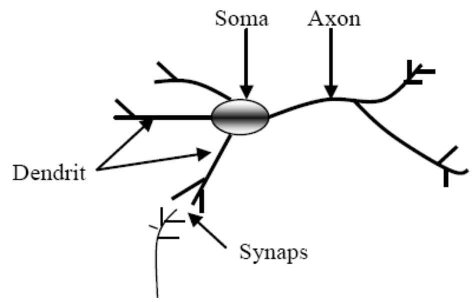


Figure1:A Simplified Model of a Biological Nuron

Machine learning approach is appealing for artificial intelligence since it is based on the principle of learning from training and experience. Connectionist models, such as ANNs, are well suited for machine learning where connection weights are adjusted to improve the performance of a network. An ANN is a network of nodes connected with directed arcs each with a numerical weight W_{ij}, specifying the strength of the connection (Figure-2).

These weights indicate the influence of previous node *x i* , on the next node, *yj*, where positive weights represent reinforcement; negative weights represent inhibition. Generally the initial connection weights are randomly selected.

Figure2. Connection weight between nodes.

First Feed-forward networks were studied. Input layer is composed of a set of inputs that feed input patterns to the network. Following the input layer there will be at least one or more intermediate layers, often called hidden layers. Hidden layers will then be followed by an output layer, where the results can be achieved (Figure-3). In feed forward networks all connections are unidirectional.

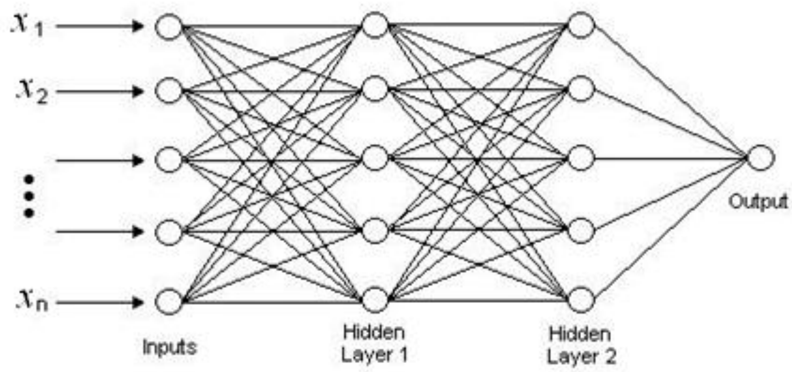


Figure 3: Hidden Layers Network with N Inputs &one output.

Multi Layer Perceptron (MLP) networks are layered feed-forward networks typically trained with static backpropagation. These networks, also known as back propagation networks, are mainly used for applications requiring static pattern classification. The back propagation algorithm selects a training example, makes a forward and a backward pass, and then repeats until algorithm converges satisfying a pre-specified mean squared error value. The main advantage of MLP networks is their ease of use and approximation of any input/output map. The main disadvantage is that they train slowly and require lots of training data.

Generalized feed-forward (GFF) networks are a generalization of the MLP networks where connections can jump over one or more layers, but these networks often solve problems much more efficiently.

A) TRAINING ALGORITHM

Training is the process by which the free parameters of the networks (i.e. the weights) get optimal values. Supervised learning models, that are used for MLP and GFF networks, train certain output nodes to respond to certain input patterns and the changes in connection weights, due to learning, cause those same nodes to respond to more general classes of patterns. In these models input layer units distribute input signals to the network. Connection weights modify the signals that pass through it. Hidden layers and output layer contain a vector of processing elements with an activation function. Usually the Sigmoid function is used as the activation function.

Every unit *ui* computes its new activation *ui* as a function of the weighted sum of the inputs to unit *ui* (*u j*) from directly connected cells. Therefore, the output of each processing unit for the forward pass will be defined as:

$$v(n) = \sum_{i=0}^{K} W_{ji} \times x_i$$

$y_j = f(v(n))$ Where $f(x) = \frac{1}{1+e^{-x}}$

The backward pass is the error back-propagation and adjustment of weights. Gradient descent approach with a constant step length, also referred to as learning rate, is used to train the network. This method minimizes the sum of squared errors of the system until a given minimum or stop at a given number of epochs, where epoch is the term specifying the number of iterations to be done over the training set. The error is multi-dimensional and may contain many local minima. A momentum term may be added to avoid getting stuck in local minima or slow convergence. The output of each processing unit for the backward pass is defined as:

$$f'^{(v(n))} = y_j \times (1 - y_j)$$

Weights are then updated by the formula where ��n��is the mean squared error and ��is the step size:

$$\delta_j(n) = -\frac{\partial \varepsilon(n)}{\partial v(n)}$$

$$W_{ji}(n+1) = W_{ji}(n) + \Delta W_{ji}(n)$$

After the training process is completed, the network with specified weights can be used for testing a set of data different than those used for training. The results achieved can then be used for generalization of the approximation of the network

TRAINING & TESTING OF NEURAL NETWORK

Back propagation is a systematic method of training multilayer artificial neural networks. It is built on high mathematical foundation and has very good application potential and is applied to a wide range of practical problems and has demonstrated power. Back propagation neural network is a three-layered feed forward architecture. The three layers are input layer, hidden layer and output layer. Functioning of back propagation proceeds in three stages, namely learning or training, testing or inference and validation shows the 1-m-n (1 input neurons, m hidden neurons, and n output neurons) architecture of a back propagation neural network model. Input layer receives information from the external sources and passes this information to the network for processing. Hidden layer receives information from the input layer, and does all the information processing, and output layer receives proceeds information from the network and sends the results out to an external receptor. The input signals are modified by interconnections weights, known as weight factor w_{ij}, which represent the interconnections of i^{th} node of the first layer to the j^{th} node of second layer. The sum of modified normally decrease during initial phase of training, as does the training set error.

However, the network begins to over fit the data; the error on the testing will typically begin to rise. When the testing error starts increasing for a specified number of iterations, the training is stopped, and the weight at the minimum value of the testing error are returned.

PREPARING DATA

The input data to neural network is not always hard physical measurements. Scaling the data sometimes is needed so that the neural network can learn better. 4Cast XL use asigmoidal activation function as its output. Therefore, user needs to scale the input data first so that the target activations to values can be comfortably learned. Please take note that 4Cast XL only process numeric data. All the data need to be converted to the between 0 to 1. Neural network will learn better if there is uniformity in the data. Here, a real life example is use. This data is taken from the demand of daily need product, from period January 2007-08 to December 2007-08 as shown in table 1.

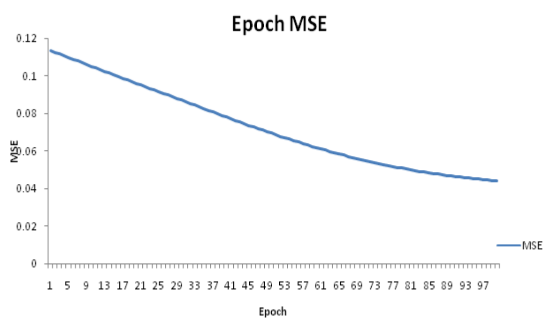
The first part is used to train the neural network, while the second part of data is used to test the performance of the model. A model is considered good if the error of out-of sample testing is the lowest compared with the other models. If the trained model is the best one for validation and also the best one for testing, one can assume that it is a good model for future forecasting. The general partition rule for training and testing set is 70% and 30%, respectively.

Table 1: Demand of Daily Need Product (April2007to March 2008)

Year(2007-08)	50Ps.	1Rs.	3Rs.	5Rs.	20gm	50gm	100gm	200gm
April	29874	21351	5500	3675	7020	3740	1940	1160
May	28370	22200	5200	2546	4800	5700	2500	1320
June	23472	21500	3540	1290	2560	4900	2000	1160
July	24540	22250	3780	2100	5220	6180	2100	720
August	33600	26160	3100	1950	3240	5660	1950	380
September	25940	23290	4400	5720	2000	2320	760	700
October	30820	21100	3100	4400	2510	3560	1340	680
November	35840	21300	3276	1840	1160	3700	1280	680
December	32486	21150	5220	1440	1320	4400	1420	1400
January	31290	22800	3240	1680	1160	7020	2620	840
February	27400	16560	2000	1600	785	4800	1160	840
March	34856	26200	2510	1600	453	2560	1200	520

Table 2: Predicted result of Daily Need Product (April2008 to March 2009)

Year(2008-09)	50Ps.	1Rs.	3Rs.	5Rs.	20gm	50gm	100gm	200gm
April	30041	24851	5100	3200	7320	3940	2040	1420
May	30527	21200	5300	2000	4900	5900	2600	2620
June	29814	31500	1350	1200	2120	1500	3200	1160
July	30640	15250	3458	2500	5000	1580	2200	1200
August	30607	21350	4100	1130	3130	1660	1650	1160
September	30832	45290	5400	5100	1000	2320	730	1320
October	30730	25000	2100	4415	2510	1360	1450	1160
November	30035	11300	2276	1845	3160	3900	1150	785
December	30036	26330	4220	1300	1520	1400	1910	453
January	29897	29230	3220	1680	1660	9020	21390	530
February	29948	11250	2633	1800	965	4900	1160	450
March	30291	32200	2150	1900	133	2430	1215	630



MSE	**0.044349**
AE	4.25755
Test MSE	0.039257
No. Of Layer	4
No. Of HL1	4
No. Of HL2	1
Input Nodes	8
Output Nodes	8
Mean	0.198978
Std Dev	0.216824
Iterations:	98

CONCLUSION

The aim of this paper has been to show the possibility of using the neural networks for predictions of daily need product. Results show that, in most of the cases, the network produces results parallel to the market demands therefore this can be used as an alternative way in these systems. The test mean square error value is 0.03925 which is also display on the network output sheet. The MSE is 0.04435. The general rule here is that the test error and the actual MSE should not have a big difference. For example MSE=0.00067 and the test error is 0.00089, this can be consider good modeling and suitable to be used as a forecasting model in a real life scenario.

REFERENCES

1. Johanna Småros, “Collaborative Forecasting in Practice” Logistics Research Network 2002 Conference, Birmingham, UK.

2. Tonya Boone, Ram Ganeshan, Alan J. Stenger, “The Impact of CPFR on Supply Chain Performance: A Simulation Study[1]”

3. P. D. Sreekanth,N. Geethanjali, P. D. Sreedevi, Shakeel Ahmed,N. Ravi Kuma and P. D. Kamala Jayanthi, “Forecasting groundwater level using artificial neural networks” CURRENT SCIENCE, VOL. 96, NO. 7, 10 APRIL 2009

4. Mehmet Kurban And Ummuhan Basaran Filik, “Next Day Load Forecasting Using Artificial Neural Network Models With Autoregression And Weighted Frequency Bin Blocks” International Journal Of Innovative Computing, Information And Control Icic International °C 2009 Issn 1349-4198 Volume 5, Number 4, April 2009 Pp. 889—898

5. C. Fernández, E. Soria, P. Sánchez-Seiquer, L. Gómez-Chova, R. Magdalena, J. D. Martín, M. J. Navarro, and A. J. Serrano “Weekly Milk Prediction on Dairy Goats using Neural Networks”

6. Neural Networks & Learning Machines by simon haykin,PHI Pvt.Ltd. User’s Guide For 4Cast XL