Prediction of Net Bandwidth using Artificial neural Network

Tanushree.Selokar
S.B.Jain institute of technology research and management Nagpur

Sanjay. L. Badjate, Ph.D
S.B.Jain institute of technology research and management Nagpur

ABSTRACT
Multi step prediction is a complex task that has attracted increasing interest in recent years. The contribution in this work is the development of nonlinear neural network models for the purpose of building multi step Prediction of Internet Bandwidth i.e. bits per second transmission record of server. It is observed that such problems exhibit a rich chaotic behavior and also leads to strange attractor. . This paper compares the performance of four neural network configurations namely a Multilayer Perceptron (MLP) , generalized feed forward network(GFF) , Self organized feature map (SOFM), and the Jorden –Elmen network with regards to various performance measures Mean square error (M.S.E.),Normalized mean square error (N.M.S.E) and regression (r) . The standard back propagation algorithm with momentum term has been used for all the models.

There are various parameters like number of processing elements, step size, momentum value in hidden layer, in output layer the various transfer functions like tanh, sigmoid, linear-tan-h and linear sigmoid, different error norms L1,L2 ,Lp to L infinity, Epochs variations and different combination of training and testing samples are exhaustively experimented for obtaining the proposed robust model for long term as well as short step ahead prediction.

Keywords
Chaotic, multi step, prediction, cross validation.

1. INTRODUCTION
The main motivation for analysis and research of chaotic time series is to predict the future and understand the fundamental feature and processes in system, which are used in every sector of the human life. Recognizing chaotic dynamics is very important for understanding and managing real world problems. One of the primary reasons for employing neural network was to create a machine that was able to learn from experience. They have the capability to learn the complex nonlinear mappings from a set of observations and predict the future [1] .The modeling and analysis of chaotic time series has also attracted the attention of many researcher. The static MLP network has gained an immense popularity from numerous practical application published over the past decade, there seems to be substantial evidence that Multilayer perceptron indeed possesses an impressive ability .There have been some theoretical results that try to explain the reasons for the success [2] and [3]. There are many different methods that have been developed to deal with nonlinear time series prediction .Among them neural networks stands unique which is being able to adequately model the nonlinearity and non stationary while being simple to train and to implement . Since the initial works, neural network have been proved to be a powerful method in accuracy for time prediction exceeding conventional methods by order of magnitude [4]. Neural networks are the best method at discovering nonlinear relationships [5, 6] and perform well with missing or incomplete data [7]

In this paper there is an analysis and comparison between different neural network models with regards to Mean square Error (M.S.E.), Normalized mean square error (NMSE) and correlation coefficient (r) on real complex time series namely the bits per second transmission record of a server. This paper is organized as follows. First the static NN based model on MLP is used to model the given system to find out the optimal parameters for the network. Then for the same optimal parameters the different neural network models are experimented. Next the different neural network models are compared for all the multi step (1, 5, 10, 20, 50) ahead prediction. Then for the optimal neural network model the Epochs are varied from 2000 to 20000 to get more optimal values of performance measures. The various parameters like number of hidden layers, number of processing elements, step size, momentum value in hidden layer, in output layer the various transfer functions like tan-h, sigmoid, linear-tan-h and linear sigmoid, different error norms L1,L2 ,Lp to L∞ are exhaustively experimented.

2. Neural Network Models
2.1 Static NN based model.
Static NN s typically uses MLP as a backbone. They are layered feed forward networks typically trained with static back propagation .MLP solid based model has a solid foundation [8- 9]. The main reason for this is its ability to model simple as well as complex functional relationships. This has been proven through number of practical applications [3]. In [4] it is shown that all continuous functions can be approximated to any desired accuracy, in terms of the uniform norm, with a network of one hidden layer of sigmoidal or (hyperbolic tangent) hidden units and a layer of linear or tan h output unit to include in the hidden layer. .This is discussed in [3] and a significant result is derived approximation capabilities of two layer perceptron networks when the function to be approximated shows certain smoothness. The biggest advantage of using MLP NN for approximation of mapping from input to the output of the system resides in its simplicity and the fact that it is well suited for online implementation. The objective of training is then to determine a mapping from a set of training data to the set of possible weights so that the network will produce predictions y(t),which in some sense are close to the true outputs y(t). The prediction error approach is based on the introduction of measure of closeness in terms of mean square error (MSE) criteria:

$$V_N(\theta, Z^N) = 1/2N)\sum_{t=1}^{N} [y(t) - y^\wedge(t \mid \theta]^2$$
$$= (1/2N)\sum_{t=1}^{N} \varepsilon^2 (t, \theta) - - - - - - (1)$$

The weights are then founds as: $\theta^\wedge$ = **arg** $\mathbf{min_0}$ $\mathbf{V_N(\theta, Z^N)}$**,** by some kind of iterative minimization scheme: $\theta^{(i+1)} = \theta^{(i)} + \mu^{(i)} f^{(i)}$,

Where $\theta^{(i)}$ specifies the current iterate (number "i"), $f^{(i)}$ is the search direction and $\mu^{(i)}$ the step size.

When NN has been trained, the next step is to evaluate it . This is done by standard method in statistics called independent validation. This method divides the available datasets into training and testing data sets. This method divides the available data sets into two sets namely training data set and testing data set .The training data set are nextly divided into two partitions: the first partition is used to update the weights in the network and the second partition is used to assess (or cross validate) the training performance .The testing data set are then used to assess how the network has generalized .The learning and generalizing ability of the estimated NN based model is assessed on the basis of certain performance measures such as MSE, NMSE and correlation coefficients and the regression ability of the NN by visual inspection of the regression characteristics for different outputs of system under study.

Since it is very likely that one ends up in a bad local minimum, the network is trained couple of times (typically least three times) starting from different initial weights. Neuro solution (version5) and MATLAB tool box (version7.0) are use for obtaining the results.

2.2 *Self* Organizing Feature Map.

Kohonen's SOFM (*Self Organizing Feature Map*) is a two layer network. The first layer of network layer is the input layer. Typically second competitive layer is organized as a two-dimensional grade. All the dimensions go from first layer to the second layer. The two layers are fully interconnected as each unit is connected to the entire unit in the competitive layer. When an input pattern is presented each unit in first layer takes on the values at the corresponding entry in the input pattern .The second layer unit then sums their input & competes to find a single winning unit the overall operation of SOFM (*Self Organizing Feature Map*) is similar to the competitive learning paradigm. Each inter connection in the Kohonen has an associated weight values for axon. The initial stage at network has randomized values for the weights. Typically the weights are sot by adding a small random number at average values for the entries in the input pattern [10-11].

2.3 Jordan –Elman Neural Network.

This network combines the past values of the context unit with the present input (x) to obtain the present net output. The Jordan context unit acts as a so-called low pass filter, which creates an output that is the weighted (average) value of some of its most recent past outputs. The output (y) of the network is obtained by summing the past values multiplied by the scalar parameter. The input to the context unit is copied from the network layer, but the outputs of the context unit are incorporated in the net through their adaptive weights.

$$y(n) = \sum_{i=0}^{n} x(n)\, \mu^{n-i}$$

In these networks, the weighting over time is inflexible since we can only control the time constant (i.e. the exponential decay). Moreover, a small change in time gives a large change in the weighting (due to the exponential relationship between the time constant and the amplitude). In general, we do not know how large the memory depth should be, so this makes the choice of problematic, without having a mechanism to adopt it.

3. Net flow series

There is increase in growth of internet users which result overloading on the popular websites. This might happen, for example, when a Web site is linked by some very popular news Web site such as Slashdot, or when its address is advertised to a wide public in the media. In both cases, the number of requests received by the Web site grows rapidly, causing the server's capacity to soon become exceeded. Overloaded Web sites service as many requests as possible (usually with very low performance), and simply drop the remaining ones. Such events are often referred to as Slashdot effects, hot spots, or flash crowds.

Flash crowds present a significant problem to Web site owners. The research community has investigated flash crowds for several years. A widely adopted solution consists of offloading a portion of the Web site load to some distributed infrastructure such as a content delivery network (CDN). In that case, the Web site replicates its content to a number of CDN servers, and starts redirecting the clients to these servers when the load becomes too high. This effectively increases the client-serving capacity of the Web site by that of the CDN, which enables the Web site to service all the clients with a good performance.

So we have used India Chennai Offshore Dev Center server's bandwidth utilization traffic report of 21st February 2008, from 10:38 a.m. for 24 hours in a time samples of 1 minute with the use of Net flow Analyzer software.

4. Performance measures:

The mean square error is given by

$$\text{MSE} = \frac{\sum_{j=0}^{P} \sum_{i=0}^{N} (d_{ij} - y_{ij})^2}{NP} \quad \text{-----(2)}$$

Where P = number of output PEs (processing elements), N=number of exemplars in the data set ,y_{ij} =network output for exemplar i at PEj , d_{ij}=desired output for exemplar i at PEj.

NMSE (Normalized Mean square Error) the normalized mean square error is defined by the following formula:

Where P=Number of output PEs,

N= Number of exemplars in data set

$$NMSE = \frac{P.N.MSE}{\sum_{j=0}^{P}\left[\frac{N\sum_{-i=0}^{N} dij^{2} \sum_{i=0}^{2} dij}{N}\right]} \qquad \text{---(3)}$$

[c] The size of the mean square error (MSE) can be used to determine how well the network output fits the desired output, but it doesn't necessarily reflect whether the two sets of data move in the same direction. For instance by simply scaling the network output, we can change the MSE without changing the directionality of the data. The correlation coefficient ®solves this problem. By definition, the correlation coefficient between a network output x and a desired output d is.

$$r = \frac{\frac{\sum_{-i}(Xi - \overline{X})(di - \overline{d})}{N}}{\sqrt{\frac{\sum_{i}(d - \overline{d})}{N}}\sqrt{\frac{\sum_{i}(xi - x)2}{N}}}$$

Where,

$$\overline{X} = \frac{1}{N}\sum_{i}^{N} Xi \quad \text{and} \quad \overline{d} = \frac{1}{N}\sum_{i=1}^{N} di \qquad \text{----(4)}$$

The correlation coefficient is confined to the range [-1,1]

5. Experimental Results

An exhaustive and careful experimentations has been carried to determine the configuration of the static MLP Model. All the possible variations for the model such as number of hidden layers, number of processing elements in each hidden layer, different transfer functions like tan h, linear tanh, sigmoid, linear sigmoid in output layer, different supervised learning rules like momentum ,step, conjugant gradient and quick propagation are investigated in simulation. The step size and momentum are gradually varied from 0.1 to 1 for static back propagation rule. After meticulous examination of the performance measures like MSE, NMSE, Correlation Coefficient and the regression ability. The optimal parameter are found and mentioned in the table 1.

Table 1: Parameters for the Neural network models

Sr. no.	Parameters	Hidden Layer	Output Layer
1	Processing elements	11	1
2	Transfer function	tanh	Lintanh
3	Learning rule	Momentum	momentum
4	Step Size	1	0.1
5	Momentum	0.8	0.8

Next for the same parameters the neural network models feed forward neural network model, self organizing feature map , and Jordan Elman are trained and the results are obtained and plotted on the graphs as shown in figure 1 to 4.

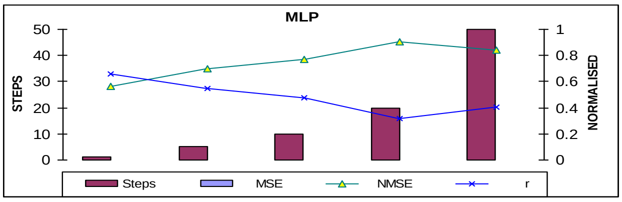

Fig1 plot of performance parameter against the steps ahead for MLP Network.

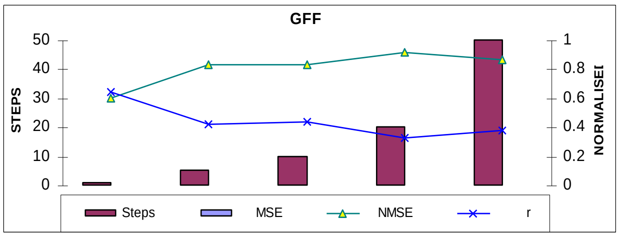

Fig2 plot of performance parameter against the steps ahead for GFF Network.

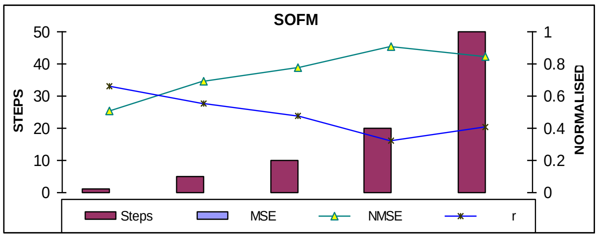

Fig3 plot of performance parameter against the steps ahead for SOFM Network.

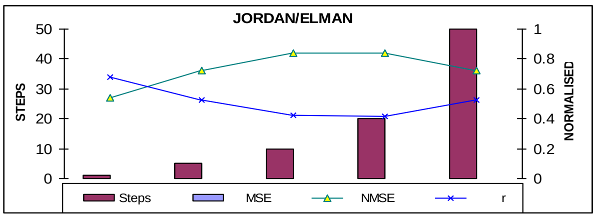

Fig4 plot of performance parameter against the steps ahead for Jordan Elman Network.

It is seen that JORDAN/ELMAN neural network is able to predict Net flow time series elegantly so far as distant predictions are concerned. Then epochs are varied from 2000 to 20000 to observe the variation in the performance measures results for the Jordan Elman network for 1, 5, 10, 20 and 50 step ahead prediction and the results are plotted in the figures 5 to fig 9. Also the plot for desired output and actual output are plotted for all the steps shown in the figure 10 to 14.

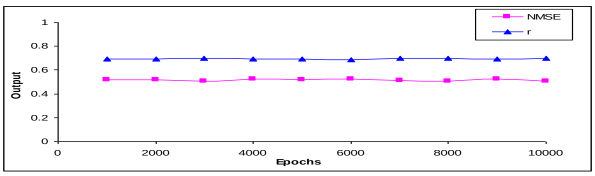

Fig 5 Plot of NMSE and correlation coefficient against Epochs for One step ahead

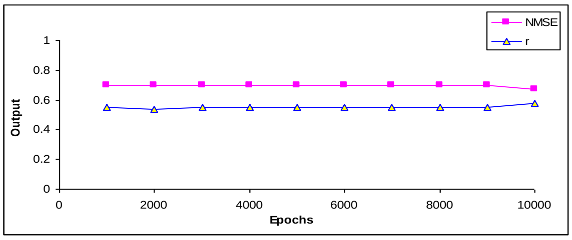

Fig 6 Plot of NMSE and correlation coefficient against Epochs for five step

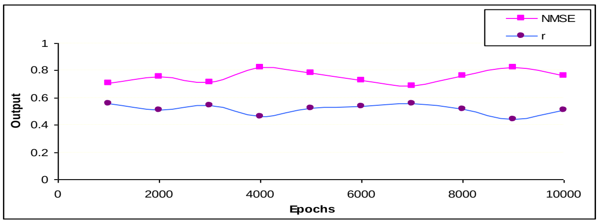

Fig 7 Plot of NMSE and correlation coefficient against Epochs for five step

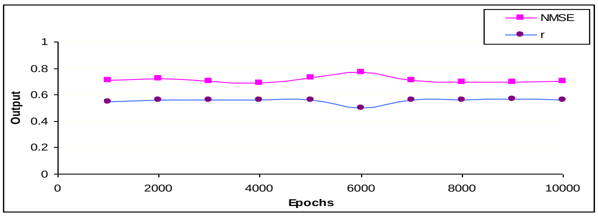

Fig 8 Plot of NMSE and correlation coefficient against Epochs for 20 step.

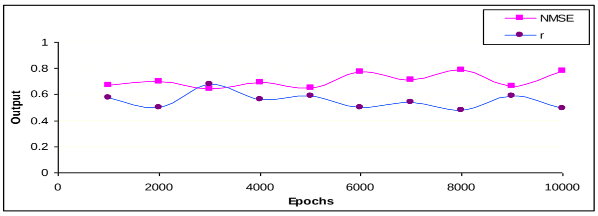

Fig 9 Plot of NMSE and correlation coefficient against Epochs for 50 step.

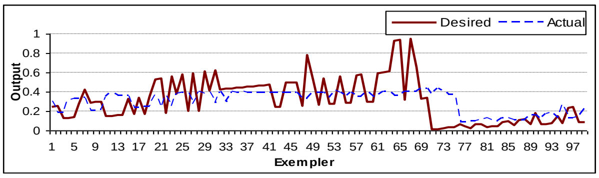

Fig 10 Plot between Desired Output Vs Actual Network Output for single step ahead

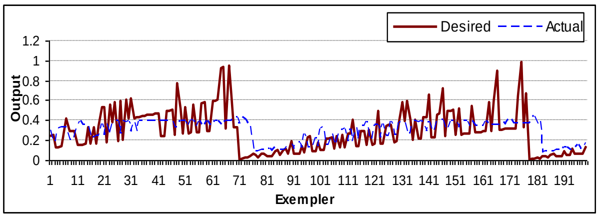

Fig 11 Plot between Desired Output Vs Actual Network Output for single five step ahead

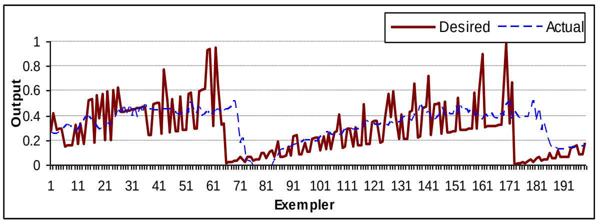

Fig 12 Plot between Desired Output Vs Actual Network Output for single ten step ahead

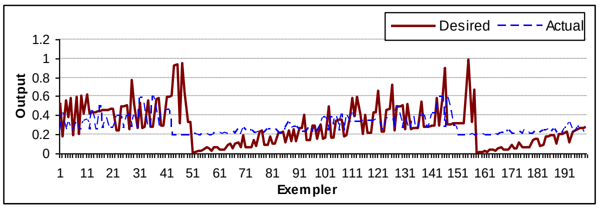

Fig 13 Plot between Desired Output Vs Actual Network Output for single twenty step ahead

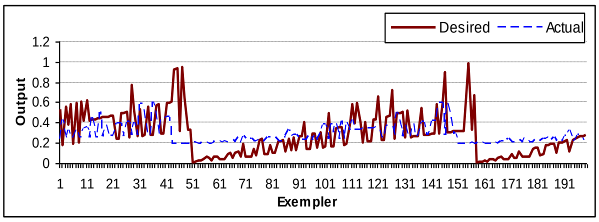

Fig 14 Plot between Desired Output Vs Actual Network Output for single fifty step ahead.

6. Conclusion

It is seen that Jordan Elman neural network recurrent network modal is able to predict Internet Bandwidth i.e. bits per second transmission record of server quite well in comparison with the Multilayer preceptron (MLP), generalized feed forward network (GFF), Self organized feature map (SOFM), and the Jorden –Elmen network Static NN configuration such as MLP NN based model is failed to cope up with the underlying nonlinear dynamics. It is seen that MSE, NMSE of the proposed dynamic model for testing data set as well as for training data set are significant than those of static MLP NN model. . In addition it is also observed that the correlation coefficient of this model for testing and training exemplars are much higher for the multi step (k=1,5,10,20,50) ahead prediction.. Then numbers of epochs are varied from 2000 to 20000 and again the neural network is trained in the steps of 2000 and performance measure parameters are plotted in figure 5 to 9. Hence the Jordan Elman recurrent neural network model has outperformed the static MLP based neural network significantly for short step (K=1, 5, 10) and for long step (K=20, 50) ahead prediction as well as the output of neural network model closely follows to the actual output for the multi step ahead prediction for all the steps as shown in the graph 10 to 14.

7. REFERENCES

[1] Barron, A R, Universal approximation bounds for superposition's of a sigmoid function, IEEET Transactions on information *Theory,39,pp930-945,1993*.

[2] Juditsky , A Hjalmarson, H, Benveniste, A, Delyon , B,Ljung , L, Sjoberg, J & Zang, Q,Nonlinear black-box models in system identification: Mathematical foundation, *Automatica,vol 31 , no 12,pp 1725,1995*.

[3] De Varies B & J C Principe, The gamma model-A new neural model for temporal processing, Neural Network, *vol5,pp565,1992*.

[4] Sanjay V Dudul Identification of a Liquied of a Liquid Saturated Steam Heat Exchanger Using Focused Time Lagged Recurrent Neural Network Model, IETE Journal of Research Vol 53 No1,January-February2007,pp69-83.

[5] K.V. Rajaram, M.A. Faruqi, Development of intelligent system, for predicting sticker induced breakouts in continuous casting based on artificial neural networks, Final Report ME/1998/ TISCO/2, Indian Institute of Technology, Kharagpur, India (September 1998).

[6] Naraendra K S & Parthasarathy , K, Identification and control of dynamic systems using neural networks,IEEE Trans on Neural Networks,vol 1 , no 1 , pp4-27,1992.

[7] Quin Waibel ,A, T Hanazana, G Hinton K Shikano & K J Lang, phoneme recognition using time delay neural Transactions on Acoustics , speech , and signal processing *vol ASSP-37,pp328,1989.*

[8] Jose C Principe , Neil R Euliano & W Curt Lefebvre, Neural and Adaptive Systems –Fundamentals Through Simulation, *John-Wiley & Sons,Inc,2000*.

[9] S Z, Su H T & Mc Avoy T J, Comparison of Four Neural Net Learning Methods for Dynamic System Identification , IEEE Transactions on Neural Networks, vol 3 ,no 1, pp 122-30, 1992.

[10] Kohonen Springer Verlag, "*self organizing feature map*" Berlin, New York 1995.

[11] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.

[12]S.K. Chand, Modeling of complex systems from time series data: a computational intelligent approach, Ph.D. thesis, Indian Institute of Technology, Kharagpur, India (January 2000).

[13]J.C. Sprott, Chaos and Time-series Analysis, Oxford University.

[14]Press, NY, 2003[11] K.Funahashi: On the Approximation Realization of Continuous Mapping by Neural Networks, Neural *Networks.vol.2, No.3, 1989*

[15] K.Doya, S.Yoshizawa: Memorizing oscillatory patterns in the analog neural network, proceedings of the *3rd IJCNN I, 27-32, 1989*

[16] M.Casdagli:- Non-linear predictions in chaotic time series physical D,35 335_359,1989.