Server Load Prediction Based on Dynamic Neural Networks

Ghannam Aljabari* and Hashem Tamimi†
Palestine Polyetchnic University (PPU), Hebron, Palestine
*galjabari@ppu.edu, †htamimi@ppu.edu

***Abstract*—Predicting server load is involved in distributed system applications such as load balancing and load sharing. Applying machine learning based methods for load prediction in distributed system applications can improve the availability and performance of these applications. Many machine learning methods have been applied for load prediction. However, some researches show that applying Neural Networks (NN) technique is more efficient in predicting the load in future time. This paper is to investigate and compare different dynamic NN models in server load prediction such as Time-Delay Neural Network (TDNN) and Nonlinear Autoregressive Network with eXogenous inputs (NARX). Data used to forecast is acquired from Web-mail server of Palestine Polytechnic University (PPU). Results have shown that NARX model provide better performance in comparison to TDNN model in server load prediction.**

***Index Terms*—machine learning; load prediction; load balancing, neural networks**

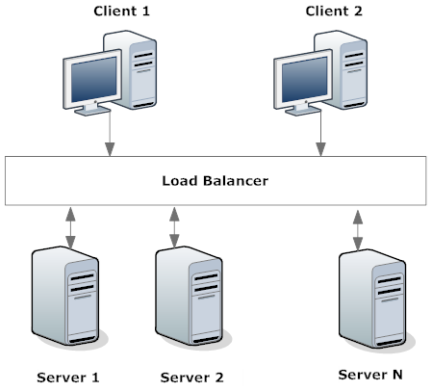

Fig. 1. Load Balancing

I. Introduction

Load balancing has been widely adopted in IT data centers for distributing workload across multiple servers or nodes, (Fig. 1). In order to provide IT services for a large number of online clients, different load balancing techniques have been applied to improve the performance and availability of distributed system applications.

In *network based load balancing* approach, load balancing is performed at the network layer (Layer 3) or transport layer (Layer 4). While such approach is often used in many distributed system applications such as web applications, it does not distribute the workload equally among web servers.

To balance workload more efficiently, load balancing is performed in *middleware*, often on a per-session or a per-request basis. In this type of load balancing, many of the existing approaches use the control theory method or *damping technology* to predict the server load. The damping method depends on a fixed factor in computing the load [1]. However, this factor should be adjusted dynamically according to the load state to avoid abnormal state of the server. As a result, a new prediction method is required that take into account the load condition dynamically to achieve *adaptive load balancing*.

Applying machine learning based method can solve this problem and predict the server load in future time [1]. Some research activities have been made in this area based on Support Vector Machine (SVM) method [2]. However, this paper is to investigate and compare different dynamic Neural Networks (NN) based methods in server load prediction. The reason of selecting dynamic NN for server load prediction is the ability of this approach to accurately forecast non-linear time series data. In addition, dynamic NN can be more compact and hence faster to evaluate than SVM approach.

II. Literature Review

The purpose of this section is to explain time series problem and time series prediction methods based on different dynamic neural networks models. These learning algorithms have been applied in many real-world forecasting applications such as power load prediction.

A. Time Series Prediction

The goal of time series prediction is to estimate some future values based on current and past sample data. Mathematically, *time series* (TS) is a sequence of vectors or scalars $y(t)$, where t represent elapsed time. The objective of TS prediction is to find a function $f(x)$ such that $y'(t)$ is the predict value of time series at future time point:

$$y'(t) = f(y(t-1), y(t-2), y(t-3), \ldots)$$

The estimation of a future value fall into two categories: *linear* and *non-linear*. Linear TS prediction depends on a linear combination of past and present values. However, most of the real-world TS prediction applications fall into the category of non-linear prediction [3].

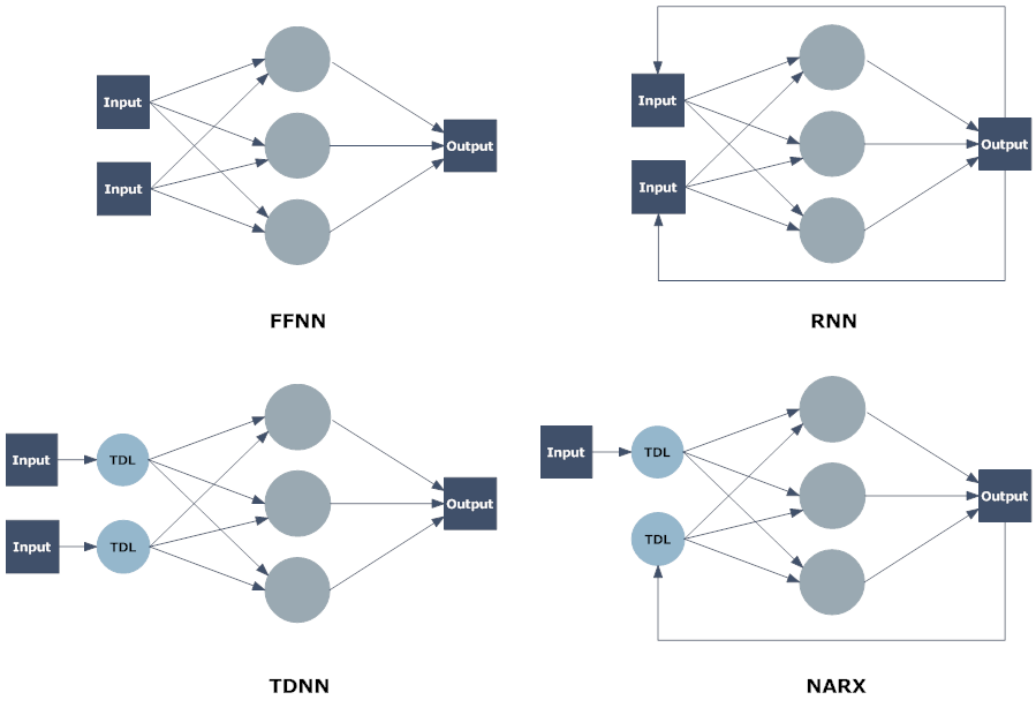

Fig. 2. NN modeling for time series prediction

B. Neural Networks for TS Prediction

Neural Networks approach is usually involved in TS prediction in which traditional TS prediction may not be able to capture the non-linear pattern in data [4].

Neural Networks can be classified into two categories: *static* and *dynamic*. Static (feedforward) networks have no *feedback* elements and contain no *time delay*. In another words, the output is calculated directly from the input through the feedfowrad connections. In dynamic network, the output depends not only on the current input, but also on the previous inputs to the network or estimated output of the network [5].

Many dynamic NN models have been proposed and applied for TS prediction including: Recurrent NN, Time Delay NN and Nonlinear AutoRegressive with eXogenous (external) inputs (NARX). Fig. 2 show different NN modeling for TS prediction [4].

Recurrent NN (RNN) is basically a feedforward NN (FFNN) with a recurrent loop, therefore the output signal is fed back to the input. This model is commonly used to perform multistep-ahead or *long-term prediction*. Time Delay NN (TDNN) integrates time delay lines (TDL) at the input of FFNN. This model is commonly used to predict future value based on past values of a time series $y(t)$. The defining equation for TDNN is:

$$y'(t) = f(y(t-1), y(t-2), \ldots, y(t-d))$$

Where $y'(t)$ is the predicted value of a time series, and d represent the time delay or memory [4], [5].

NARX is a combination of all above NN and well suited for TS prediction problem. This model can be used to predict future value of a time series $y(t)$ based on past values of that series and past values of a second time series $u(t)$. The defining equation for the NARX model is:

$$y'(t) = f(y(t-1), \ldots, y(t-d_{\mathrm{y}}), u(t-1), \ldots, u(t-d_{\mathrm{u}}))$$

Where $y'(t)$ is the predicted value of $y(t)$, d_{y} and d_{u} represent input and output time delay respectively. The NARX model provide better predictions than other NN models because it uses additional information contained in the previous values of $u(t)$ [4], [5].

III. Methodology

Neural Networks are composed of simple elements (neurons) operating in parallel. The values of the connections between these elements (weights) are adjusted or trained to perform a particular function. The network is adjusted based on a comparison of the network ouput and the target output.

The work flow to solve problems with NN approach has the following steps:

1) Data collection
2) Data preprocessing
3) NN configuration
4) NN training
5) NN testing

Considering these steps for load server prediction problem, we have obtained the load data from the Webmail server of Palestine Polytechnic University (PPU) for 7 days and with interval of 5 minutes. The load data include load average, memory usage and total number of processes running on the server. The samples of 6 days have been taken as training data

TABLE I
TDNN PREDICTION ERROR

TD	RMSE
2	0.0280
3	0.0268
4	0.0280
5	0.0273
6	0.0257
7	0.0290

TABLE II
NARX PREDICTION ERROR

TD	RMSE
2	0.0245
3	0.0247
4	0.0249
5	0.0245
6	0.0244
7	0.0273

and the 7th day as testing data. So, the training set contains 1728 sample and the testing set has 288 sample.

To use NN for predicting a time series, the first step is load the data, normalize it and convert it to a time sequence. Then we need to configure network inputs and targets for training. Because our objective is to perform one step-ahead prediction, we do not need to feed the output back to the input. Also, to achieve optimal result, we need to train the NN several times. This is because network training is more likely to be trapped in local minima.

To compute the prediction error, Root Mean Squared Error (RMSE) is the most commonly used method. RMSE is defined as the root of MSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i' - y_i)^2}$$

IV. RESULTS

We have conducted two set of experiments to investigate forecasting with NN. The first set of experiments was conducted with TDNN. In TDNN, the load average TS is used only to predict server load. The load average measure the trend in CPU utilization and include all demand for the CPU. The objective of the experiment was to perform one-step prediction based on past and current values of load average. The data was normalized to take values between zero and one. Before training, initial inputs and targets need to be remove from original data. This is because predicting the next value of the time series begin after filling the time delay line with initial values. For these experiments, ten neurons is used in the hidden layer, and the default Levenberg-Marquardt (LM) function is used for training.

Table I display the average prediction error generated with 7-fold cross validation. From this table, best result is obtained with maximum time delay of 6. The predicted values of the server load in the last day with the actual values is shown in Fig. 3. This result show that NN is able to accurately forecast the next value of the server load.

The second set of experiments was conducted with NARX model. In NARX model another TS is used in addition to load average to predict the next value. The objective of the experiment was to perform one-step ahead prediction based on the history of the input sequence of load average and the history of another input sequence, in our case total number of processes and memory usage. For this reason, the network does not have feedback connection and the amount of history will only affect the response.

The configuration and training of NARX model is similar to TDNN. However, NARX model has two inputs that need to be prepared before training and simulating. The result is displayed in Table II. Best result is obtained with maximum time delay of 6. The prediction error represent the difference between the predicted value and the actual value. Fig. 4 show the predicted and actual values of the last day.

V. CONCLUSION

In this paper, we introduced a load balancing technique based on machine learning method. Also, we proposed a solution using NN for server load prediction. We have investigated two model of NN for time-varying data, TDNN and NARX. Best result is obtained using NARX model in comparison to TDNN.

In the future, it is planned to collect more data sequences from the server such as traffic to refine prediction performance. Also, we will investigate additional forecasting methods for server load prediction.

REFERENCES

[1] J. Wang, Y. Ren, D. Zheng, and Q. Wu, "A machine-learning based load prediction approach for distributed service-oriented applications," in *International Conference on Computational Science (ICCS)*, 2007.

[2] Y. Yu, X.Zhan, and J. Song, "Server load prediction based on improved support vector machines," in *IEEE International Symposium on IT in Medicine and Education*, 2008.

[3] N. Sapankevych and R. Sankar, "Time series prediction using support vector machine: A survey," in *IEEE Computational Intelligence Magazine*, 2009.

[4] C. A. Mitrea, C. K. M. Lee, and Z. Wu, "A comparison between neural networks and traditional forecasting methods: A case study," in *International Journal of Engineering Business Management*, vol. 1, no. 2, 2009, pp. 19–24.

[5] M. Beale, M. Hagan, and H.Demuth, "Matlab neural network toolbox user's guide," The Math Works Inc., 2010, http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf.

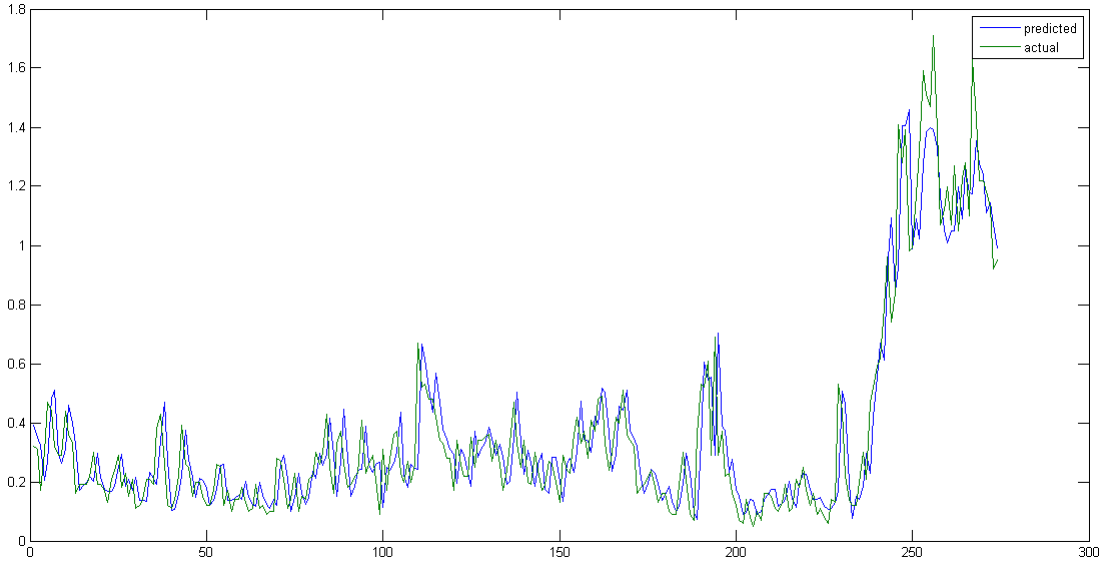

Fig. 3. TDNN predicted and actual data

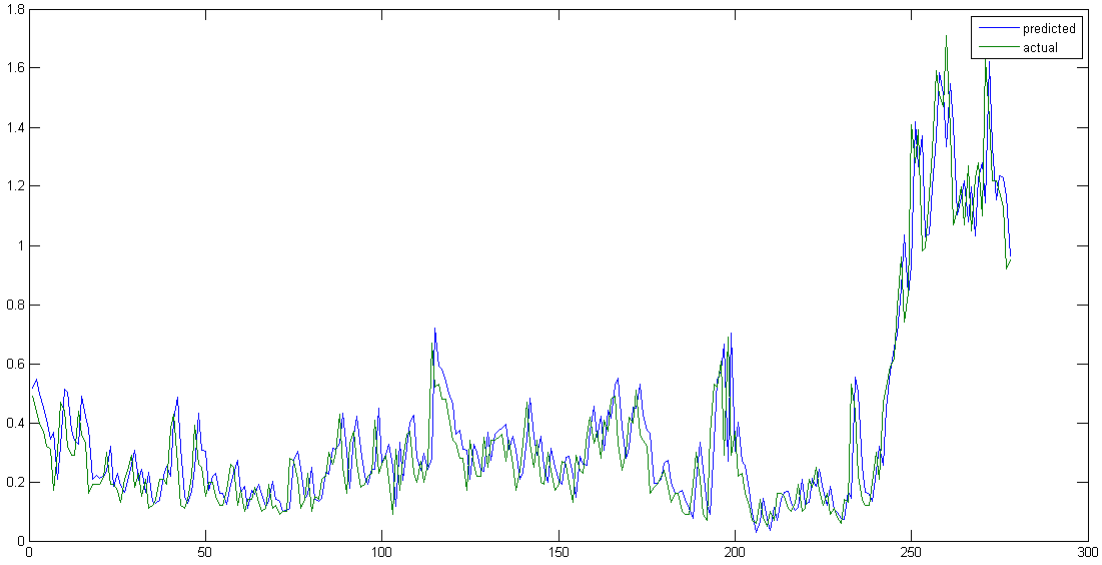

Fig. 4. NARX predicted and actual data