

Hourly Traffic Flow Predictions by Different ANN Models

Vedat TOPUZ
The Marmara University
Istanbul,
Turkey

1. Introduction

In this paper, different artificial neural network (ANN) models for predicting the hourly traffic flow in Second Talled Bridge of Bosphorus in Turkey is performed. Second Bosphorus Bridge has an important role in urban traffic networks of Istanbul. The prediction of the traffic flow data is a vital requirement for advanced traffic management and traffic information systems, which aim to influence the traveler behaviors, reducing the traffic congestion, improving the mobility and enhancing the air quality. Detecting the trends and patterns in transportation data is a popular area of research in the business world to support the decision-making processes. The primary means of detecting trends and patterns has involved statistical methods such as clustering and regression analysis most of which use linear models. The goal is to predict a future value of data by using the past data set. On the other hand, non-linear approaches offer more powerful models, since they give more possibilities in the choice of the input-output relation.

ANNs approaches have been applied to a large number of problems because of their non-linear system modeling capability by learning ability using collected data. They offer highly parallel, adaptive models that can be trained by the experience. During the last decade, ANN has been applied widely to prediction of the traffic data.

We compared the generalization performance of the different ANN models such as Multi Layer Perceptron (MLP), Radial Basis Function Network (RBF), Elman Recurrent Neural Networks (ERNN) and Non-linear Auto Regressive and eXogenous input (NARX) type.

2. Traffic flow prediction

Detecting the trends and patterns in transportation data is a popular area of research in the business world to support the decision-making processes (Topuz, 2007). According to the theory, the traffic flow forecasting research can be divided into two types. One type is determined based on the mathematical model and second type is knowledge-based intelligent model of forecasting methods. Short-term traffic flow prediction is more influenced by the stochastic interferential factors than the long-term one, the uncertainty is greater and the disciplinarian laws are less obvious. Thus using the short-term traffic prediction models based on the classical mathematical methods, the precision of forecast cannot satisfactorily meet the demand of real-time traffic control systems (Su & YU, 2007).

Source: Urban Transport and Hybrid Vehicles, Book edited by: Seref Soylu,
ISBN 978-953-307-100-8, pp. 192, September 2010, Sciyo, Croatia, downloaded from SCIYO.COM

Because the traffic system is a complex and variable system that involves a great deal of people, the traffic flow state has high randomness and uncertainty. The traditional traffic flow forecast methods such as Kalman filter, Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA) and etc had been unable to satisfy the demand of the forecast precision that was increasing in practice (CHEN & MA, 2009)

On the other hand, artificial neural networks (ANNs) have been applied to a large number of problems because of their non-linear system modeling capacity by learning ability using collected data. During the last decade ANNs have been applied widely to prediction of the traffic data (SADEK, 2007). Different ANN approach such as MLP (Smith & Demetsky, 1994), RBF (Celikoglu & Cigizoglu, 2006), ERNN (Li & Lu, 2009), Neuro-Genetic (Abdulhai et al. 2002) have been used to predict the traffic follow.

Such a prediction study including the NARX type ANN has been completed in this paper, to compare the effectiveness of different ANN approach.

3. Artificial Neural Network (ANN)

ANNs are designed to mimic the characteristics of the biological neurons in the human brain and nervous system. Given a sample vectors, ANNs is able to map the relationship between input and output; they "learn" this relationship, and store it into their parameters.. The training algorithm adjusts the connection weights (synapses) iteratively learning typically occurs through the training. When the network is adequately trained, it is able to generalize relevant output for a set of input data. They have been applied to a large number of problems because of their non-linear system modeling capacity.

3.1 Multi Layer Perceptron (MLP)

There are different types of ANN although the most commonly used architecture of ANN is the multilayer perceptron (MLP). MLP has been extensively used in many transportation applications due to its simplicity and ability to perform nonlinear pattern classification and function approximation. Therefore, it is considered the most commonly implemented network topology by many researchers (Transportation Research Board 2007).

MLP means a feedforward network with one or more layers of nodes between the input and output nodes and it is capable of approximating arbitrary functions. Two important characteristics of the MLP are: its nonlinear processing elements (activation function) which have a nonlinearity and their massive interconnectivity (weights). Typical MLP network is arranged in layers of neurons, where each neuron in a layer computes the sum of its inputs and passes this sum through an activation function (f). For this context designed MLP network is shown in figure 1.

The MLP network is trained with back-propagation, the first step is propagating the inputs towards the forward layers through the network. For a three-layer feed-forward network, training process is initiated from the input layer (Hagan 1996).

$$\begin{aligned} a^0 &= u \\ a^{m+1} &= f^{m+1}(W^{m+1}a^m + b^{m+1}), \quad m = 0, 1 \\ y &= a^3 \end{aligned} \tag{1}$$

Where, y is output vector, u is input vector, $f(\cdot)$ is the activation function, W is weighting coefficients matrices, b is bias factor vector and m is the layer index. These matrices defined as;

$$W^1 = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,S_0} \\ w_{2,1} & w_{2,2} & \dots & \cdot \\ \cdot & \cdot & \dots & w_{2,S_0} \\ w_{S_1,1} & w_{S_1,2} & \dots & w_{S_1,S_0} \end{bmatrix}, W^2 = [w_{1,1} \ w_{1,2} \dots w_{1,S_1}], b^1 = [b_1 \ b_2 \ \dots b_{S_1}]^T, b^2 = [b_1],$$

Here S_0 and S_1 are size of network input and hidden layer

In this study, sigmoid tangent activation functions are used in the hidden layer and linear activation function is used in the output layer respectively. These functions are defined as follows;

$$f^1 = \frac{(\exp^n - \exp^{-n})}{(\exp^n + \exp^{-n})}, f^2 = n \quad (2)$$

The total network output is;

$$n_i^m = \sum_{j=1}^{S_{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \quad (3)$$

Second step is propagating the sensibilities (d) from the last layer to the first layer through the network: d^3, d^2, d^1 . The error (e) calculated for output neurons is propagated to the backward through the weighting factors of the network. It can be expressed in matrix form as follows:

$$d^3 = -2\dot{F}^3(n^3)(e) \quad (4)$$

$$d^m = \dot{F}^m(n^m)(W^{m+1})^T d^{m+1}, \text{ for } m = 2, 1$$

Here $\dot{F}^m(n^m)$ is Jacobian matrix defined as follow;

$$\dot{F}^m(n^m) = \begin{bmatrix} \frac{\partial f^m(n_1^m)}{\partial n_1^m} & 0 & 0 \\ 0 & \frac{\partial f^m(n_2^m)}{\partial n_2^m} & 0 \\ 0 & 0 & \frac{\partial f^m(n_{S_3}^m)}{\partial n_{S_3}^m} \end{bmatrix} \quad (5)$$

e is mean square error,

$$e = \frac{1}{2} \sum_{\gamma=1}^q (y^\gamma - \hat{y}^\gamma)^2 \quad (6)$$

Where γ is the sample in dimension q .

The last step in back-propagation is updating the weighting coefficients. The state of the network always changes in such a way that the output follows the error curve of the network towards down.

$$\begin{aligned} W^m(k+1) &= W^m(k) - \alpha d^m (a^{m-1})^T \\ b^m(k+1) &= b^m(k) - \alpha d^m \end{aligned} \quad (7)$$

where α represents the training rate, k represents the epoch number. By the algorithmic approach known as gradient descent algorithm using approximate steepest descent rule, the error is decreased repeatedly

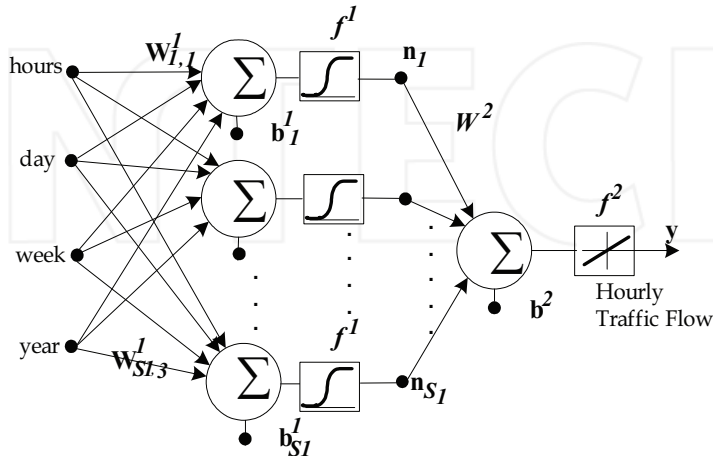


Fig. 1. Designed MLP Network Schematic Diagram.

3.2 Elman Recurrent Neural Networks (ERNN)

ERNN is also known partially recurrent neural network which is subclass of recurrent networks. It is MLP network augmented with additional context layers (W^0), storing output values (y), of one of the layers delayed (z^{-1}) by one-step and used for activating this other layer in the next time (t) step. The self-connections of the context nodes make it also sensitive to the history of input data, which is very useful in dynamic system modeling (Elman, 1990).

$$y_{(t+1)} = f^1(W^1x + b^1) + y_{(t)}W^0 \quad (8)$$

While ERNN use identical training algorithm as MLP, context layer weight (W^0) is not updated as in equation 8. Schematic diagram of designed ERNN network is given in figure 2.

The ERNN network can be trained with any learning algorithm that is applicable to MLP such as backpropagation that is given above.

3.3 Radial Basis Function Network (RBF)

RBF network has a feed-forward structure consisting of two layers, nonlinear hidden layer and linear output which is given in figure in 3. RBF networks are being used for function approximation, pattern recognition, and time series prediction problems. Their simple structure enables learning in stages, and gives a reduction in the training time.

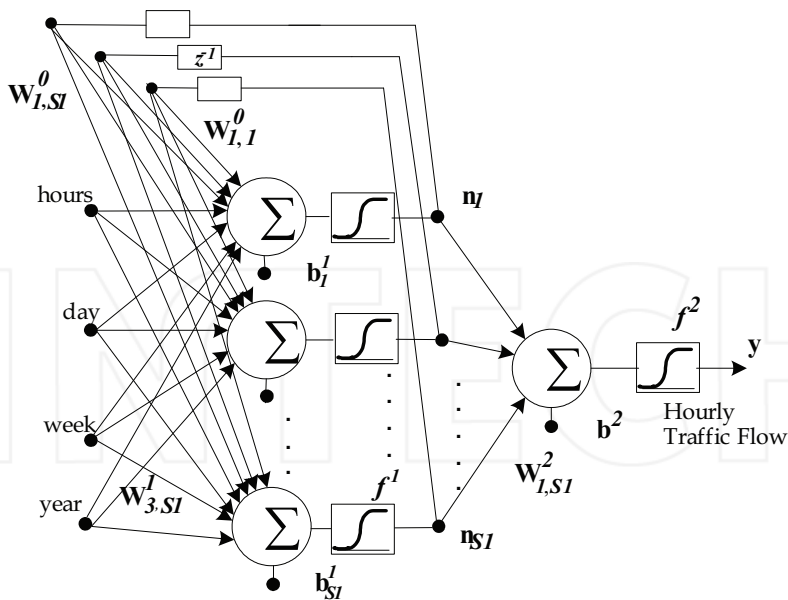


Fig. 2. Designed ERNN Network Schematic Diagram

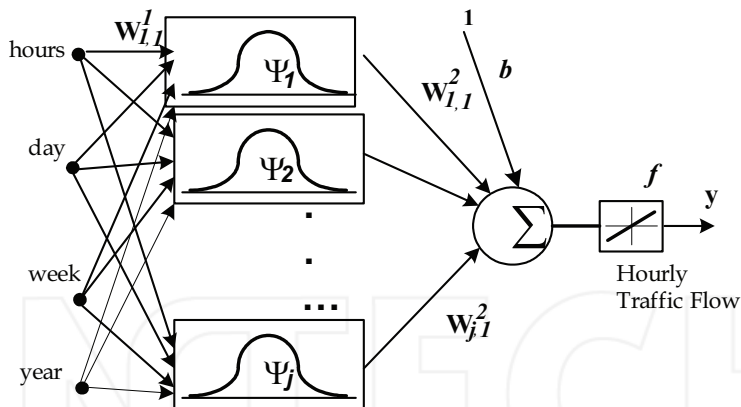


Fig. 3. Designed RBF Network Schematic Diagram.

The proposed model uses Gaussian kernel (Ψ) as the hidden layer activation function. The output layer implements a linear combiner of the basis function responses defined as; (Haykin, 1994)

$$y = b + \sum_{j=1}^q w_{i,j} \Psi_j \quad (9)$$

Where, q is the sample size, Ψ_j is response of the j th hidden neuron described as;

$$\Psi_j = \exp \left[-\frac{\|\mathbf{x} - c_j\|^2}{2\sigma_j^2} \right] \quad (10)$$

Where, c_j is Gaussian function center value, and σ_j is its variance.

RBF network training has two-stage procedure. In the first stage, the input data set is used to determine the center locations (c_j) using unsupervised clustering algorithm such as the K-means algorithm and choose the radii (σ_j) by the k -nearest neighbor rule. The second step is to update the weights (\mathbf{W}) of the output layer, while keeping the (c_j) and (σ_j) are fixed.

3.4 Non-linear Auto Regressive and eXogenous Input type ANN (NARX)

A simple way to introduce dynamics into MLP network consists of using an input vector composed of past values of the system inputs and outputs. This the way by which the MLP can be interpreted as a NARX model of the system. This way of introducing dynamics into a static network has the advantage of being simple to implement. To deduce the dynamic model of realized system, NARX type ANN model can be represented as follows (Maria & Barreto, 2008);

$$y(k+1) = f_{ANN}(y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)) + \varepsilon(k) \quad (11)$$

where $y(k+1)$ is model predicted output, f_{ANN} is a non-linear function describing the system behavior, $u(k)$, $y(k)$, $\varepsilon(k)$ are input, output and approximation error vectors at the time instances k , n and m the orders of $y(k)$ and $u(k)$ respectively. Order of the process can be estimated from experience. Modeling by ANN relies on the consideration of an approximate function of f_{ANN} . Approximate dynamic model is constructed by adjusting a set of connection weight (\mathbf{W}) and biases (\mathbf{b}) via training function defined as MLP network. The NARX network, it can be carried out in one out of two modes:

Series-Parallel (SP) Mode: In this case, the output's regressor is formed only by actual values of the system's output defined as;

$$y(\hat{k}+1) = f_{ANN}(y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)) + \varepsilon(k) \quad (12)$$

Figure 4 shows the topology of one-hidden-layer NARX network when trained in the SP-mode.

Parallel (P) Mode: In this case, estimated outputs are fed back and included in the output's regressor defined as follows;

$$y(\hat{k}+1) = f_{ANN}(y(\hat{k}), y(\hat{k}-1), \dots, y(\hat{k}-n+1), u(k), u(k-1), \dots, u(k-m+1)) + \varepsilon(k) \quad (13)$$

While NARX-SP type network is used in the training phase, NARX -P type network is used in the testing phase, which is given in figure 5.

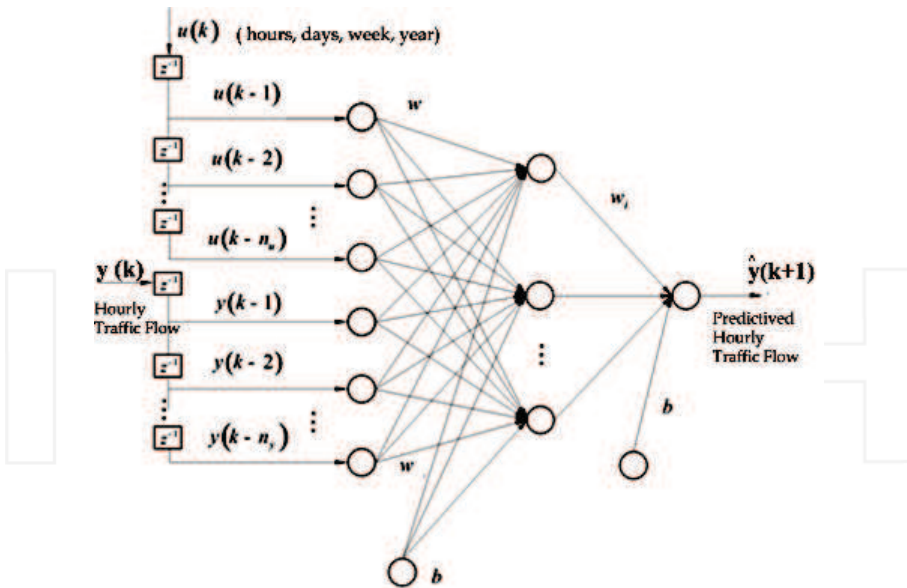


Fig. 4. Architecture of the NARX network during training in the SP-mode

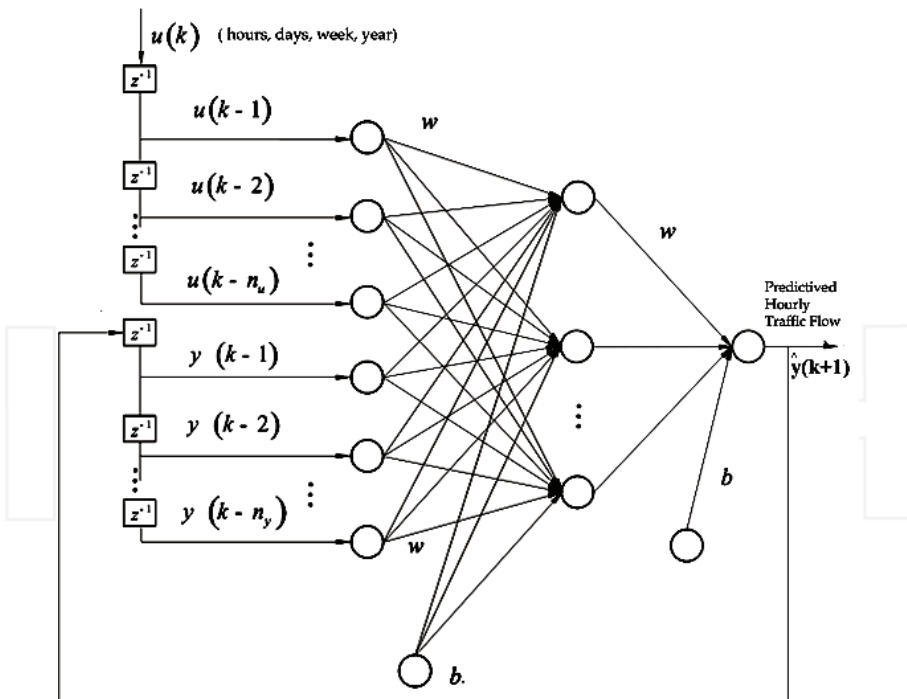


Fig. 5. Architecture of the NARX network during testing in the P-mode.

4. Designing process

ANN designing process involves four steps. These are gathering the traffic data, selecting the ANN architecture, training the network, and testing the network. In the training, four input variables have been used. These are, hours of the day: 0-23; days of the month: 1-30; days of the week: 1-7; and years: 2001-2003. Selecting the daily traffic demands as distinct variables both for days of the month and days of the week are mandatory situation, because of the weekly and monthly variance of the daily traffic demands. In the training phase, the target variables have been used as 2160 data related to the volumes, or hourly passing number of the vehicles (demand) of 4th months (April) of the years 2001-2003. In the testing phase, 720 data related to the hourly demand of 4th month (April) of the year 2004, have been used to compare the actual and neural output of the system performance.

Gathering the Traffic Data: Data recorded by the tolls has been used for training, testing and analysis. Only 2880 number of traffic demand data related to the hours of the days of 4th months of the years 2001-2004 has been taken into consideration (Transportation Survey 2004).

Selecting the Best Network Architecture: The number of hidden layer and neurons in the hidden layer(s) play very important roles for ANNs and choice of these numbers depends on the application. Influenced by theoretical works proved that single hidden layer is sufficient for ANNs to approximate any complex nonlinear function with any desired accuracy. In addition, determining the optimal number of hidden neurons is still a question to deal with. Although there is no theoretical basis for selecting these parameters, a few systematic approaches are also reported but the most common way of determining the number of hidden neurons is still trial and error approach. The number of input nodes that depend on the number of the time lags is also very important for the performance of the NARX type ANN model. The value of the time lags has to be determined by trial and error effort. Radius of the center point in nonlinear activation function is great effect the performance of the RBF network. Therefore, in addition to hidden layer neuron number, radiuses of the center point in neurons are also determined.

Training and testing the network: Designed MLP, ERNN and NARX network are trained with Levenberg-Marquardt backpropagation learning algorithms, which described in section 3. MATLAB with Neural network toolbox is used for all ANN application (Demuth & Beale 2002).

The optimum learning, η and the momentum, α , rates are determined after trials observing the RMSE and R obtained at the end of the testing stage. When a smaller η is selected, the changes to the synaptic weights (W) in the network had been smaller. The momentum rates α prevented the learning process from terminating in a local minimum of the error surface. It is seen that η and α should be decreased if the number of the input and output layers are increased. On the other hand, the iteration number increases by decreasing η and α value.

Once trained, the network can be used for predicting (test phase) the output for any input vector from the input space. This is called the "generalization property" of the network. To show this property of trained networks, the same experiment is done with testing data set that are not involved in training data set. It is also used the early stopping strategy (validation phase) to improve the generalization properties of designed network. In this strategy, the validation data set is used and the error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations, which we choose 10, the training process is stopped. Root mean square

error (RMSE) values and correlations coefficients (R) are selected to compare the performance of designed network, which is defined as follows;

Features may be correlated, that is, a change in one feature Y is associated with a similar change in another feature \hat{Y} . A correlation coefficient can be calculated as;

$$R = \frac{\sum_{i=1}^K (x_i - m_1)(y_i - m_2)}{\sqrt{\sum_{i=1}^K (x_i - m_1)^2 \sum_{i=1}^K (y_i - m_2)^2}} \quad (12)$$

Here m_1 is the mean value of all the values x_i of feature Y , and m_2 is the mean of all y_i values of the \hat{Y} feature. The correlation coefficient assumes values in the interval $[-1,1]$. When $R = -1$ there is a strong negative correlation between Y and \hat{Y} , when $R=1$ there is a strong positive correlation, and when $R=0$ there is no correlation at all.

Root mean squared error is also defined as follows;

$$RMSE = \sqrt{\frac{1}{q} \sum_{i=1}^q (y_i - \hat{y}_i)^2} \quad (13)$$

Where q is sample size, y_i , \hat{y}_i are network output and predicted output respectively.

5. Experiments and results

In this section, firstly the performance of all MLP, ERNN, RBF and NARX type ANN model are investigated. Then scattering diagram and traffic flow graphics are drawn for the best ANNs models.

5.1 MLP network experiments

As a mentioned above, the performance of MLP network is highly dependent on the hidden layer neuron number. Therefore, performances of MLP network is tested using 5, 10,15,25,50 and 100 neurons in the hidden layer. End of the training and testing procedure s obtained correlations coefficients (R) and MSE values are given in table 1.

Model	Training		Test	
	RMSE	R	RMSE	R
MLP-5*	0.0186	0.95	0.0190	0.944
MLP-10*	0.0093	0.975	0.0099	0.971
MLP-15*	0.0094	0.9750	0.0102	0.970
MLP-25*	0.0063	0.983	0.0077	0.978
MLP-50*	0.0032	0.991	0.0054	0.984
MLP-100*	0.0030	0.992	0.0070	0.98

*The numbers show the hidden neuron

Table 1. Different MLP Network Models RMSE and Correlation Coefficients (R) Values at the Training and Testing Phase.

It could be easily figure out in this table that, MLP-50 network shows the best generalization performance. It is also realized that to use more than 50 neurons in the hidden layer are not

increase the generalization ability of designed networks. The best MLP network model scattering diagram is shown in figure 6.

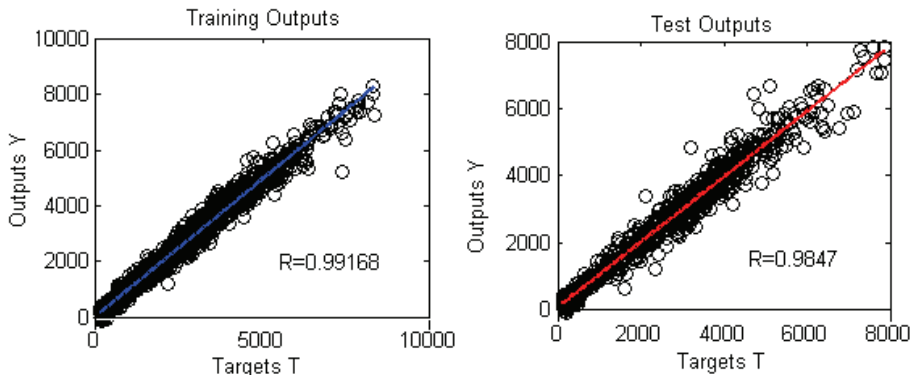


Fig. 6. MLP-50 Network Model Scattering Diagrams for Train and Test Output.

To show the obtained results, for only last week's predicted values with real traffic flow are given in figure 7. These figures with scattering diagrams show us that prediction errors are small enough to be accepted

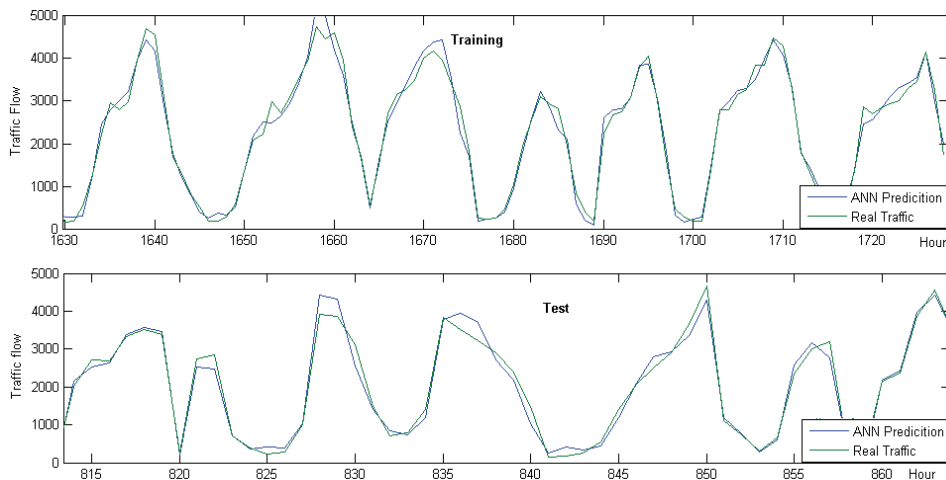


Fig. 7. The Best MLP Network Model Traffic Flow Graphics at the Train and Testing Phase.

5.2 ERN network experiments

The performance of ERNN is also dependent on the hidden layer neuron number. Hence, performances of ERNN is tested using 5, 10,15,25,50,100 neurons and obtained correlations coefficients (R) and MSE values are given in table 2. It could be easily figure out in this table that, ERNN -10 network shows the best generalization performance. The best ERNN model scattering diagram is shown in figure 8. Achieved the best ERNN results, with real traffic flow are given in figure 7.

Model	Training		Test	
	RMSE	R	RMSE	R
ERNN -5*	0.0590	0.832	0.0527	0.839
ERNN -10	0.0556	0.844	0.0507	0.846
ERNN -15	0.0754	0.778	0.0667	0.790
ERNN -25	0.0924	0.730	0.0827	0.741
ERNN -50	0.0872	0.740	0.0789	0.749
ERNN -100	0.0936	0.715	0.0820	0.734

*The numbers show the hidden neuron.

Table 2. Different ERNN Models RMSE and Correlation Coefficients (R) Values at the Training and Testing Phase.

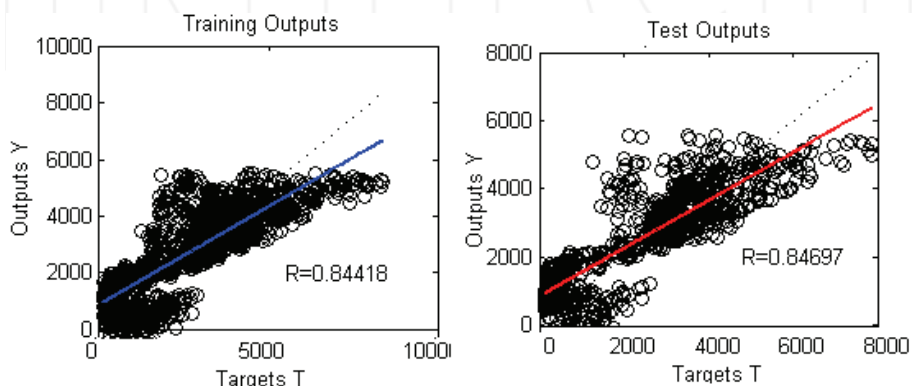


Fig. 8. ERNN -10 Network Scattering Diagrams for Train and Test Output.

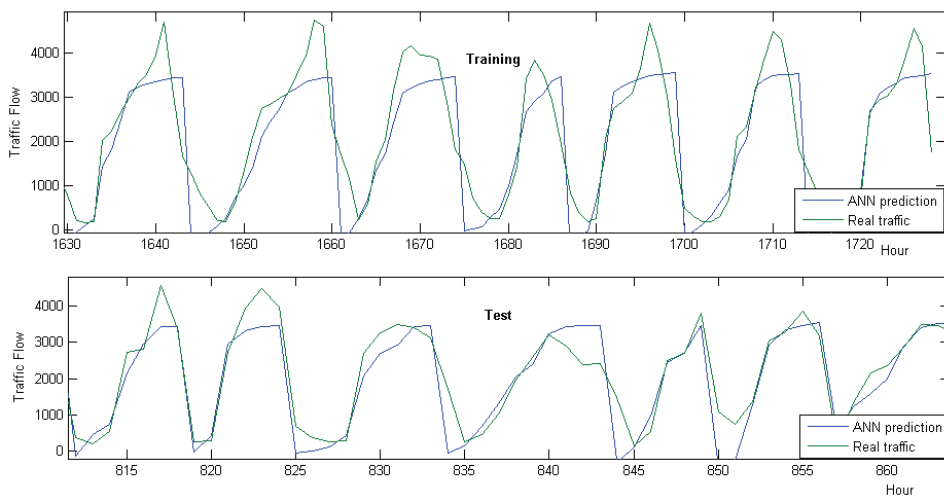


Fig. 9. The Best ERNN Model Traffic Flow Graphics at the Train and Testing Phase

5.3 RBF network experiments

Because of the performance of RBF network is depending on the radius of the center points and hidden layer neuron number, different RBF network models are tested. Obtained correlations coefficients (R) and MSE values are given in table 3. It could be easily figure out in this table that, RBF N250/S2 network shows the best generalization performance. The best RBF network model scattering diagram is shown in figure 10. Related traffic flow graphics on the test and training data is also given in figure 11.

Model	Training		Test	
	RMSE	R	RMSE	R
RBF N50/S2*	0.0148	0.96	0.0155	0.957
RBF N50/S3	0.0149	0.96	0.0172	0.953
RBF N50/S3.5	0.0161	0.956	0.0183	0.949
RBF N100/S2	0.0096	0.974	0.0123	0.966
RBF N100/S3	0.0118	0.968	0.0152	0.958
RBF N100/S3.5	0.0115	0.969	0.0148	0.959
RBF N250/S2	0.0066	0.982	0.0108	0.971
RBF N250/S3	0.0096	0.974	0.0144	0.961
RBF N250/S3.5	0.0102	0.973	0.0150	0.959
RBF N1000/S2	0.0042	0.989	0.0145	0.961
RBF N1000/S3	0.0074	0.980	0.0153	0.959

*N show the hidden neuron, S show the radius of the center points

Table 3. Different RBF Network Models RMSE and Correlation Coefficients (R) Values at the Training and Testing Phase.

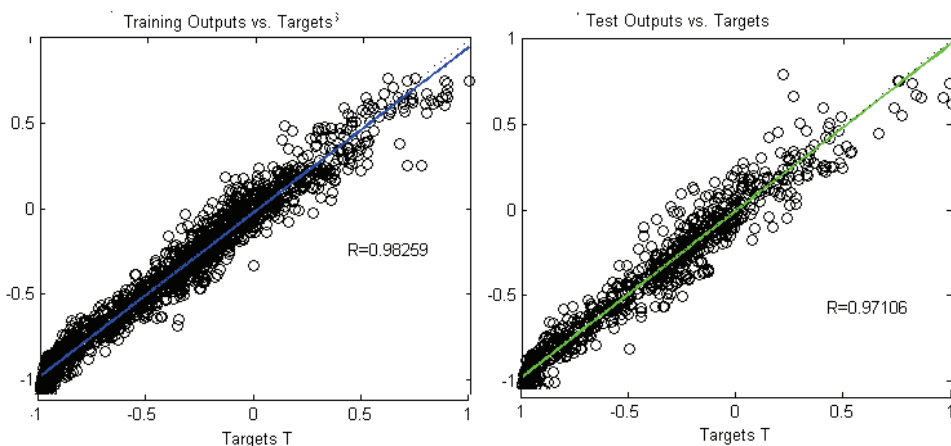


Fig. 10. RBF N250/S2 -10 Network Scattering Diagrams for Train and Test Output.

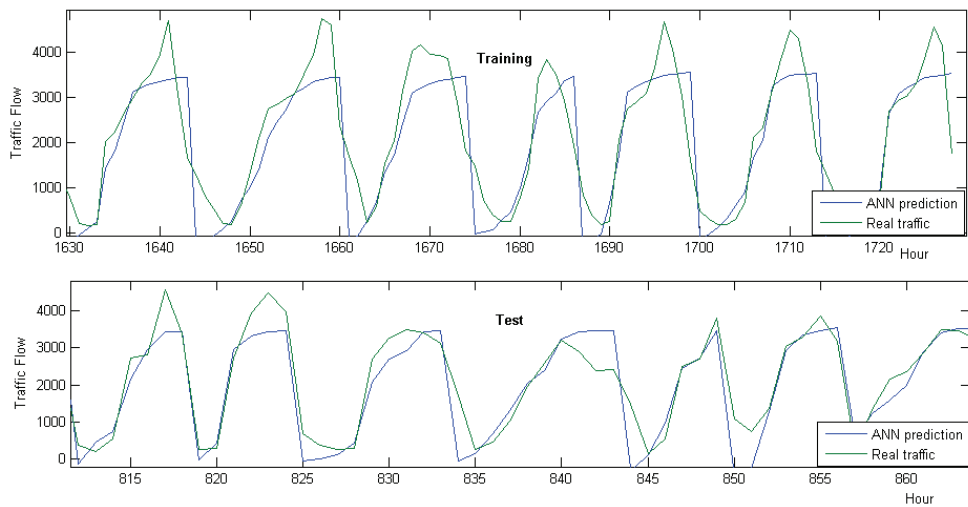


Fig. 11. Best RBF Network Model Traffic Flow Graphics at the Train and Testing Phase

5.4 NARX network experiments

The number of input nodes to the NARX type ANN model is dependent on the number of the time lags that is determined by the analysis of the auto-correlation function belonging to examined time series data set. Four time lags are found sufficient, hence; the input layer consisted of four nodes. Because the second step is largely a trial-and-error process and experiments involving neural networks with more than 4 input nodes and more than 100 neurons in the hidden layer didn't show any improvement in prediction accuracy. Obtained correlations coefficients (R) and MSE values are given in table 4, The best NARX network model scattering diagram is shown in figure 12.

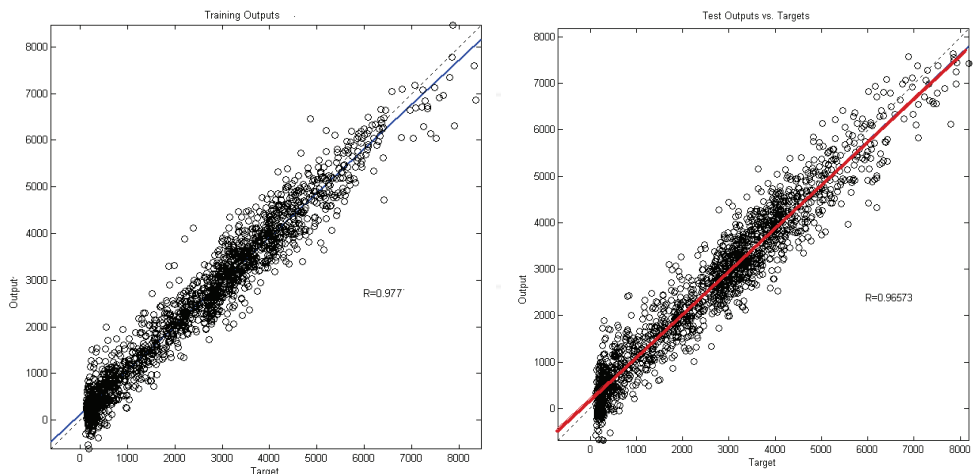


Fig. 12. NARX-H100D4 Network Scattering Diagrams for Train and Test.

Model	Training		Test	
	RMSE	R	RMSE	R
NARX-H5D5	0.0693	0.791	0.0750	0.749
NARX-H10D5	0.0632	0.814	0.0627	0.839
NARX-H15D5	0.0591	0.833	0.0554	0.849
NARX-H25D5	0.0457	0.874	0.0527	0.867
NARX-H50D5	0.0310	0.916	0.0327	0.911
NARX-H100D5	0.0090	0.975	0.0109	0.960
NARX-H5D10	0.0998	0.696	0.0872	0.713
NARX-H10D10	0.0897	0.718	0.0820	0.734
NARX-H15D10	0.0816	0.747	0.0789	0.749
NARX-H25D10	0.0646	0.813	0.0527	0.839
NARX-H50D10	0.0360	0.898	0.0507	0.846
NARX-H100D10	0.0118	0.968	0.0212	0.958
NARX-H100D20	0.0184	0.948	0.0190	0.944
NARX-H100D2	0.0083	0.977	0.0153	0.959
NARX-H100D3	0.0088	0.975	0.0144	0.961
NARX-H100D4	0.0083	0.977	0.0127	0.965

Table 4. Different NARX Network Models RMSE and Correlation Coefficients (R) Values at the Training and Testing Phase. (H hidden neuron number D delay time)

Achieved predicted values with real traffic flow are given in figure13. These figures with scattering diagrams show us that prediction errors are small enough to be accepted.

6. Conclusion

In this paper, different ANN models have predicted daily traffic demand in Second Tolloed Bridge of Bosphorus, which has an important role in urban traffic networks of Istanbul. To find the best-predicted model feed forward type (MLP, RBF), recurrent type (ERNN) and NARX type ANNs are used. Previously every model parameters are investigated for each network type. In this stage, root mean square errors (RMSE) and correlation coefficient (R) values are used as a performance criterion.

Concerning neural networks that are created and simulated there have been excluded the below conclusions:

- The most efficient training algorithm proved to be the Levenberg-Marquardt Backpropagation, as it surpassed the others not only in quality of results but also in speed.
- For the MLP, ERNN and NARX type network, the best type of normalization proved to be the one in region -1 -1 in combination with sigmoid tangent activation function in the hidden layer nodes. For the RBF network, the best radius of the center points is found 2. For a NARX network, suitable choice of input delay is 4
- The number of nodes used in the hidden layer depends from the type and the complexity of the time series, and from the number of inputs.
- The results have shown that feed forward type network models have advantages over recurrent type network.
- It is also concluded that, many other transportation data prediction studies can be implemented easily and successfully by using the different ANN architectures.

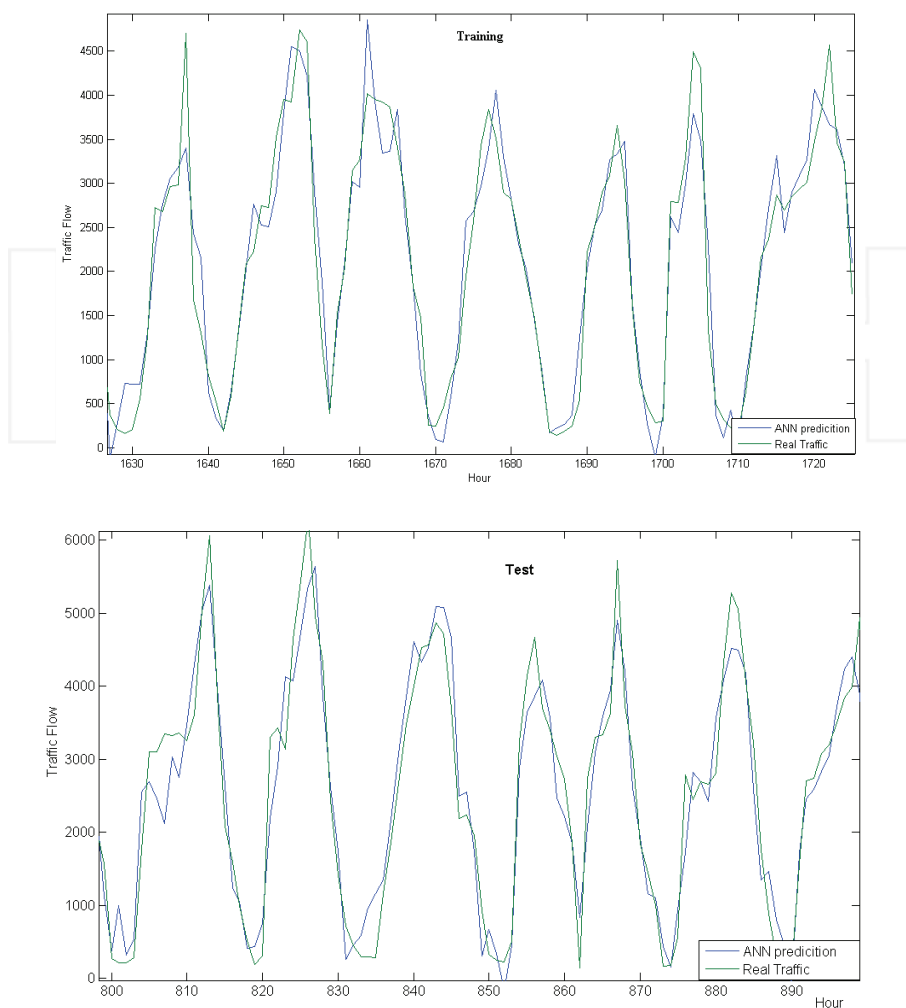
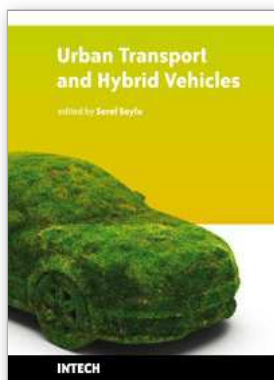


Fig. 13. NARX-H100D4 Network I Traffic Flow Graphics at the Train and Testing Phase

7. References

- TOPUZ , V. (2007). Traffic Demand Prediction Using ANN Simulator, In: KES 2007/WIRN 2007, Part I, LNAI 4692, B. Apolloni et al. (Ed.), pp. (864-870), Springer-Verlag, Berlin Heidelberg
- [Su, H & Yu, S. (2007). Hybrid GA Based Online Support Vector Machine Model for Short-Term Traffic Flow Forecasting In: APPT 2007, LNCS 4847, M. Xu et al. (Ed.), pp.(743-752), Springer-Verlag Berlin Heidelberg.

- CHEN, B.P. & MA, Z. (2009), Short-term Traffic Flow Prediction Based on ANFIS, International Conference on Communication Software and Networks ICCSN. 2009, pp. 791-793, ISBN 978-0-7695-3522-7, IEEE.
- SADEK, A. W. (2007), Artificial Intelligence Applications in Transportation, In: Transportation Research E-C113. Artificial Intelligence in Transportation Information for Application, pp.(1-6). Transportation Research Board, ISSN 0097-8515
- Smith, B.L.& Demetsky, M.J. (1994): Short-term Traffic Flow Prediction: Neural Network Approach. Transp. Res. Rec. 1453, 98-104 .
- Celikoglu, H.B & Cigizoglu, H.K (2007) Modelling public transport trips by radial basis function neural Networks In: Mathematical and Computer Modelling 45 (2007) 480-489 Elsevier.
- Ruimin Li, R & Huapu Lu, H. Combined Neural Network Approach for Short-Term Urban Freeway Traffic Flow Prediction. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part III, LNCS 5553, pp. 1017-1025, 2009. Springer-Verlag Berlin Heidelberg 2009
- Abdulhai, B.& Himanshu, P & Will, R. (2002) 'Short-Term Traffic Flow Prediction Using Neuro-Genetic Algorithms', Journal of Intelligent Transportation Systems, 7: 1, 3 – 41
- Transportation Research Board, Artificial Intelligence and Advanced Computing Applications Committee (January 2007) Artificial Intelligence in Transportation Information for Application Transportation Research Circular number E-C113 Washington, DC ISSN 0097-8515
- Hagan, T. M. & Demuth, H.B & Beale, M., Neural Network Design, PWS Publishing Company, 1996, 2-44.
- Elman, J. L. (1990) Finding Structure in Time; Cognitive Science, 14, 179-221.
- Haykin, S. (1994) Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, New York, USA. haykin p 300-305
- Jose Maria P. Junior & Guilherme A. Barreto (2008) Narx reference: Long-Term Time Series Prediction with the NARX Network: An Empirical Evaluation In: Neurocomputing Volume 71, Issue 16-18 Pages: 3335-3343 ISSN:0925-2312
- General Directorate Of Highways. Traffic and Transportation Survey 2000-2004. GDOH, Ankara, Turkey (2004)
- Howard Demuth & Mark Beale (2002) Neural Network Toolbox User's Guide .The MathWorks, Inc. MA



Urban Transport and Hybrid Vehicles

Edited by Seref Soylu

ISBN 978-953-307-100-8

Hard cover, 192 pages

Publisher Sciyo

Published online 18, August, 2010

Published in print edition August, 2010

This book is the result of valuable contributions from many researchers who work on both technical and nontechnical sides of the field to be remedy for typical road transport problems. Many research results are merged together to make this book a guide for industry, academia and policy makers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vedat Topuz (2010). Hourly Traffic Flow Prediction Using Different ANN Models, Urban Transport and Hybrid Vehicles, Seref Soylu (Ed.), ISBN: 978-953-307-100-8, InTech, Available from:

<http://www.intechopen.com/books/urban-transport-and-hybrid-vehicles/hourly-traffic-flow-prediction-using-different-ann-models>

INTech

open science | open minds

InTech Europe

University Campus STeP Ri

Slavka Krautzeka 83/A

51000 Rijeka, Croatia

Phone: +385 (51) 770 447

Fax: +385 (51) 686 166

www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai

No.65, Yan An Road (West), Shanghai, 200040, China

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元

Phone: +86-21-62489820

Fax: +86-21-62489821