

Short-Term Chaotic Time Series Forecast

Jiin-Po Yeh
I-Shou University
Taiwan

1. Introduction

The neural network was initiated by McCulloch and Pitts (1943), who claimed that neurons with binary inputs and a step-threshold activation function were analogous to first order systems. Hebb (1949) revolutionized the perception of artificial neurons. Rosenblatt (1958), using the McCulloch-Pitts neuron and the findings of Hebb, developed the first perception model of the neuron, which is still widely accepted today. Hopfield (1982) and Hopfield et al. (1983) demonstrated from work on the neuronal structure of the common garden slug that ANNs (artificial neural networks) can solve non-separable problems by placing a hidden layer between the input and output layers. Rumelhart and McClelland (1986) developed the most famous learning algorithm in ANN-backpropagation, which used a gradient descent technique to propagate error through a network to adjust the weights in an attempt to find the global error minimum, marking a milestone in the current artificial neural networks. Since then, a huge proliferation in the ANN methodologies has come about.

Many physical systems in the real world, such as rainfall systems (Hense, 1987), chemical reactions (Argoul et al., 1987), biological systems (Glass et al., 1983) and traffic flow systems (Dendrinos, 1994), have complicated dynamical behaviors and their mathematical models are usually difficult to derive. If the system has only one measurement available, to obtain knowledge about its dynamical behaviors, one usual method is to reconstruct the state space by using delay coordinates: $\{x(t), x(t-\tau), x(t-2\tau), \dots, x(t-(q-1)\tau)\}$, where $x(t)$ is time series of the measurement, q is the dimension of the reconstructed state space and τ is the time delay (Moon, 1992; Alligood et al., 1997). By gradually increasing the dimension of the state space, the fractal dimension of the chaotic attractor (Moon, 1992; Grassberger and Proccacia, 1983) will approach an asymptote. This asymptote is often considered to be the chaotic attractor dimension of the dynamical system. According to Takens (1981), the embedding dimension of the chaotic attractor, or the dimension required to reconstruct the state space, must be at least twice larger than the chaotic attractor dimension so that the generic delay plots could be obtained.

That chaotic behaviors exist in the traffic flow system has been known for decades. Gazis et al. (1961) developed a generalized car-following model, known as the GHR (Gazis-Herman-Rothery) model, whose discontinuous behavior and nonlinearity suggested chaotic solutions for a certain range of input parameters. Traffic systems without signals, bottlenecks, intersections, etc. or with a coordinate signal network, modeled by the traditional GHR traffic-flow equation, were tested for presence of chaotic behaviors (Disbro and Frame, 1989; Fu et al., 2005). Chaos was also observed in a platoon of vehicles described by the traditional GHR model with a nonlinear inter-vehicle separation dependent term added (Addison and Low, 1996).

Up to the present, there have been a variety of methodologies devoted to short-term chaotic time series prediction, including local linear models (Farmer and Sidorowich, 1987), polynomial models (Acquirre and Billings, 1994) and neural network-based black-box models (Principe et al., 1992; Albano et al., 1992; Deco and Schurmann, 1994; Bakker et al., 1996), just to name a few. However, this chapter will only focus on two easy-to-implement methods: (i) the feedforward backpropagation neural network and (ii) the multiple linear regression.

2. Embedding dimension

To get the right dimension of the reconstructed state space to embed the attractor (chaotic or periodic), the dimension of the attractor must first be found. There are a number of ways to measure the attractor dimension (Moon, 1992; Grassberger and Proccacia, 1983). Among them, this chapter only demonstrates two measures easily processed with the aid of the computer: (i) the pointwise dimension; (ii) the correlation dimension.

2.1 Pointwise dimension

A long-time trajectory in the state space is considered, as shown in Fig. 1, where a sphere of radius r is placed at some point on the orbit. The probability of finding a point inside the sphere is (Moon, 1992)

$$P(r) = \frac{N(r)}{N_0} \quad (1)$$

where $N(r)$ is the number of points within the sphere and N_0 the total number of points in the orbit. When the radius r becomes smaller, and $N_0 \rightarrow \infty$, there exists a relationship between $\ln P(r)$ and $\ln r$, which can be expressed as (Moon, 1992):

$$\tilde{d}_p = \lim_{r \rightarrow 0} \frac{\ln P(r)}{\ln r} \quad (2)$$

In order to have better results, the averaged pointwise dimension is usually used; that is

$$d_p = \lim_{r \rightarrow 0} \frac{\ln[(1/M) \sum P(r)]}{\ln(r)} \quad (3)$$

where M is the number of randomly selected points on the orbit. In practice, M is approximately one tenth of the total number of points on the orbit.

2.2 Correlation dimension

As in the definition of pointwise dimension, the orbit is discretized to a set of points in the state space. A sphere of radius r is first placed at each and every point of the orbit and then the Euclidean distances between pairs of points are calculated. A correlation function is defined as (Moon, 1992; Grassberger and Proccacia, 1983)

$$C(r) = \lim_{n \rightarrow \infty} \frac{1}{N(N-1)} \sum_i \sum_j u(r - |\mathbf{x}_i - \mathbf{x}_j|) \quad (4)$$

$(i \neq j)$

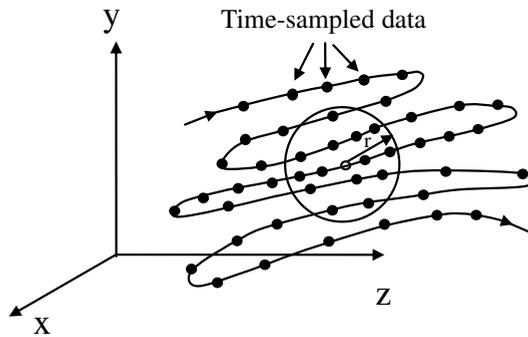


Fig. 1. A trajectory in the state space.

where $|x_i - x_j|$ is the Euclidean distance between points x_i and x_j of the orbit and u is the unit step function. For a lot of attractors, this function $C(r)$ exhibits a power law dependence on r , as $r \rightarrow 0$; that is

$$\lim_{r \rightarrow 0} C(r) = ar^d \tag{5}$$

Based on the above relationship, a correlation dimension is defined by the expression

$$d_c = \lim_{r \rightarrow 0} \frac{\ln C(r)}{\ln r} \tag{6}$$

The dimension of the attractor found by Eq. (3) or (6) will approach an asymptote d with the conjectured dimension of the reconstructed state space gradually increasing. To represent the attractor one to one without causing self-intersection, the embedding dimension of the attractor must be at least $2d+1$ (Takens, 1981). For a chaotic attractor, the dimension d is always fractal, not an integer. Therefore, the appropriate dimension for the reconstructed state space will be the smallest integer greater than or equal to $2d+1$.

3. Forecasting models

3.1 Neural networks

The first forecasting model introduced in this chapter is a two-layer feedforward neural network with the backpropagation training algorithm, as shown in Fig. 2. The transfer function used in the single hidden layer is the tan-sigmoid function for mapping the input to the interval $[-1, 1]$ of the following form

$$a_i = f(n_i) = \frac{1 - e^{-n_i}}{1 + e^{-n_i}} \quad , \quad i = 1, 2, 3, \dots, s \tag{7}$$

where $n_i = w_{i1}x_1 + w_{i2}x_2 + \dots + w_{iR}x_R + b_i$, x_1, x_2, \dots, x_R are the inputs, s is the number of neurons, $w_{i1}, w_{i2}, \dots, w_{iR}$ are the weights connecting the input vector and the i th neuron of the hidden layer, and b_i is its bias. The output layer with a single neuron uses the linear transfer function

$$a = f(n) = n \quad (8)$$

where $n = W_{11}a_1 + W_{12}a_2 + \dots + W_{is}a_s + b$, $W_{11}, W_{12}, \dots, W_{is}$ are the weights connecting the neurons of the hidden layer and the neuron of the output layer, and b is the bias of the output neuron.

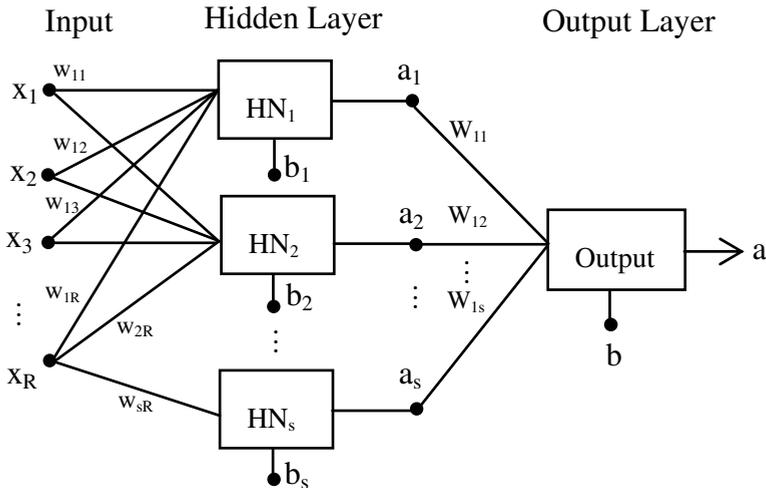


Fig. 2. The feedforward neural network with two layers

There are many variations of the backpropagation algorithm, which is aimed at minimizing the network performance function, i.e., the mean square error between the network outputs and the target outputs, which is

$$MSE = \frac{1}{m} \sum_{j=1}^m (t_j - a_j)^2 \quad (9)$$

where t_j and a_j are the j th target output and network output, respectively. The Levenberg-Marquardt algorithm (Hagan and Menhaj, 1994; Levenberg, 1944; Marquardt, 1963) is selected as the training function to minimize the network performance function. This algorithm interpolates between the Newton's algorithm and the gradient descent method. If a tentative step increases the performance function, this algorithm will act like the gradient descent method, while it shifts toward Newton's method if the reduction of the performance function is successful. In this way, the performance function will always be reduced at each iteration of the algorithm. To make the neural networks more efficient, it is quite beneficial to scale inputs and targets so that they will fall within a specific range. For example, the following formula

$$k' = 2 \left(\frac{k - \min}{\max - \min} \right) - 1 \quad (10)$$

is often used to scale both inputs and targets, where k is the original value, k' is the scaled value, and \max and \min are the maximum and minimum of the inputs or targets,

respectively. Eq. (10) produces inputs and targets in the range [-1, 1]; the scaled outputs of the trained networks will be usually converted back to the original units.

There are two methods to improve the network generalization: Bayesian regularization (MacKay, 1992) and early stopping. The Bayesian regularization provides a measure of how many network parameters (weights and biases) are being effectively used by the network. From this effective number of parameters, the number of neurons required in the hidden layer of the two-layer neural network can be derived by the following equation

$$(Rs+s)+(s+1)= P \tag{11}$$

where R is the number of elements in the input vector, s is the number of neurons in the hidden layer, and P is the effective number of parameters found by the Bayesian regularization. With the strategy of early stopping incorporated into the neural network, the error on the validation set is monitored during the training process. When the network begins to overfit the training data, the error on the validation set typically also begins to rise. Once the validation error increases for a specified number of iterations, the training stops and the weights and biases at the minimum of validation error are returned. To evaluate the performance of the trained network, the regression analysis between the network outputs and the corresponding targets is frequently adopted, and the result is usually displayed by the scatter plot or correlation coefficient (Mendenhall et al., 1986).

3.2 The multiple linear regression

Because nearby states in the state space have analogous behaviors (Alligood et al. 1997; Farmer and Sidorowich, 1987), as shown in Fig. 3, a multiple linear regression model (Mendenhall et al., 1986), fitted to the delay coordinates of nearby states in the reconstructed state space is another good choice to forecast the short-term behavior of the chaotic dynamical system. Assuming that q is the appropriate embedding dimension of the strange attractor, the multiple linear regression to predict the future behaviors of a trajectory has the following form

$$\hat{y}_i = \beta_0 + \beta_1x_{i1} + \beta_2x_{i2} + \dots + \beta_qx_{iq}, \dots, i = 1, \dots, p \tag{12}$$

where \hat{y}_i is the predicted value of the observed response y_i , which corresponds to the first delay coordinate of the i th nearest state some time units later, the independent variables $x_{i1}, x_{i2}, \dots, x_{iq}$ correspond to the delay coordinates of the i th nearest state: $\{x(t), x(t-\tau), x(t-2\tau), \dots, x(t-(q-1)\tau)\}$, respectively, p is the total number of data, and $\beta_0, \beta_1, \beta_2, \dots, \beta_q$ are unknown parameters to be decided by the method of least squares (Mendenhall et al., 1986).

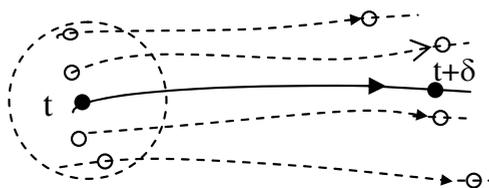


Fig. 3. Analogues of nearby trajectories in a short period of time.

4. Numerical results

Examples chosen for demonstration are the westbound passing traffic volume at the intersection of Xingai Road and Guanfu S. Road, Taipei City, Taiwan. The data were collected by the vehicle traffic counter from August 22, 2005 to September 2, 2005, totaling 10 weekdays excluding the weekend. There are three time scales involved: 5-min, 10-min, and 15-min.

4.1 Reconstruction of the traffic flow system

Time series of the three different time intervals show no repeat of themselves and have the tendency to be aperiodic. For example, Fig. 4 shows time series of the 5-min traffic volume for the training set (first 7 days totaling 2016 observations). The state-space dimension n for the delay coordinate reconstruction of the traffic flow system is increased gradually from 3 to 22. For each reconstruction, both the pointwise dimension and correlation dimension of the strange attractor are found for comparison. Fig. 5(a) shows the limiting behavior of the function $\ln[(\sum P(r))/M]$ as $r \rightarrow 0$ for different state-space dimension n with time delay τ fixed at 20, and Fig. 5(b) shows the pointwise dimension approaching an asymptote of 6.449. Fig. 6(a) shows the limiting behavior of the function $\ln C(r)$ as $r \rightarrow 0$ for different state-space dimension n with time delay τ fixed at 20, and Fig. 6(b) shows the correlation dimension approaching an asymptote of 6.427. The results shown in Figs. 5 and 6

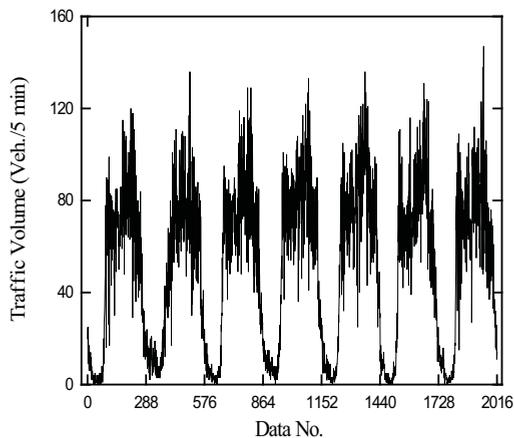


Fig. 4. Time series of the 5-min traffic volume.

are valid only for the 5-min traffic volume and time delay $\tau = 20$. The asymptotes of the pointwise dimension and correlation dimension for a variety of different time intervals and time delays are presented in Tables 1 and 2, respectively, where the limiting dimension of the chaotic attractor ranges from 6.307 to 6.462. Different time intervals and time delays lead to almost the same fractal dimension. According to Takens (1981), the embedding dimension of chaotic attractors of the traffic flow system is therefore found to be at least 14. Because different time delays result in almost identical limiting fractal dimension of the chaotic attractor, the choice of time delay in fact is actually not decisive, except to avoid the natural period of the system.

Time Interval \ Time Delay	Time Delay						
	$\tau=10$	$\tau=20$	$\tau=30$	$\tau=40$	$\tau=50$	$\tau=60$	$\tau=70$
5-min	6.445	6.449	6.449	6.429	6.434	6.431	6.449
10-min	6.435	6.432	6.331	6.307			
15-min	6.445	6.427	6.454				

Table 1. Asymptotes of pointwise dimension for different time intervals and delays.

Time Interval \ Time Delay	Time Delay						
	$\tau=10$	$\tau=20$	$\tau=30$	$\tau=40$	$\tau=50$	$\tau=60$	$\tau=70$
5-min	6.408	6.427	6.447	6.433	6.432	6.462	6.440
10-min	6.423	6.415	6.416	6.419			
15-min	6.444	6.443	6.430				

Table 2. Asymptotes of correlation dimension for different time intervals and delays.

4.2 Neural networks

The neural network toolbox of MATLAB (Demuth et al., 2010) software is used to build up neural networks and perform the training. The elements of the input vector are composed of 14-dimensional delay coordinates: $x(i)$, $x(i-\tau)$, $x(i-2\tau)$, ..., $x(i-13\tau)$, where $x(i)$ is the i th observation of the time series of traffic volume, and τ is the time delay, which is chosen to be 20, 10 and 5 for 5-min, 10-min, and 15-min traffic volumes, respectively. The network target corresponding to this input is $x(i+1)$. All forecasts are only one time interval ahead of occurrence, i.e., 5-min, 10-min or 15-min ahead of time. When using the strategy "early stopping" to monitor the training process, the allowed number of iterations for the validation error to increase is set to be 5. The data collected is divided into three sets: the training set (the first 7 days), the validation set (the 8th and 9th days), and the prediction set (the 10th day).

4.2.1 5-min traffic volume

First of all, a feedforward backpropagation neural network with the Bayesian regularization is created to get the effective number of network parameters. The network inputs and targets are imported from the 14-dimensional delay coordinates: $x(i)$, $x(i-20)$, $x(i-40)$, ..., $x(i-260)$, and $x(i+1)$, respectively. The results are shown in Fig. 7, which indicates only approximately 216 effective parameters are required in this network; therefore, the appropriate number of neurons in the hidden layer is found by Eq. (11) to be 14 (equal to the number of elements in the input vector). Then, replace the number of neurons in the hidden layer with 14 and train the network again by the Levenberg-Marquardt algorithm coupled with the strategy "early stopping." The training process stops at 10 epochs because the validation error already has increased for 5 iterations. Fig. 8 shows the scatter plot for the training set with correlation coefficient $\rho=0.90249$. Lastly, simulate the trained network with the prediction set. Fig. 9 shows the scatter plot for the prediction set with the correlation coefficient $\rho=0.83086$. Time series of the observed value (network targets) and the predicted

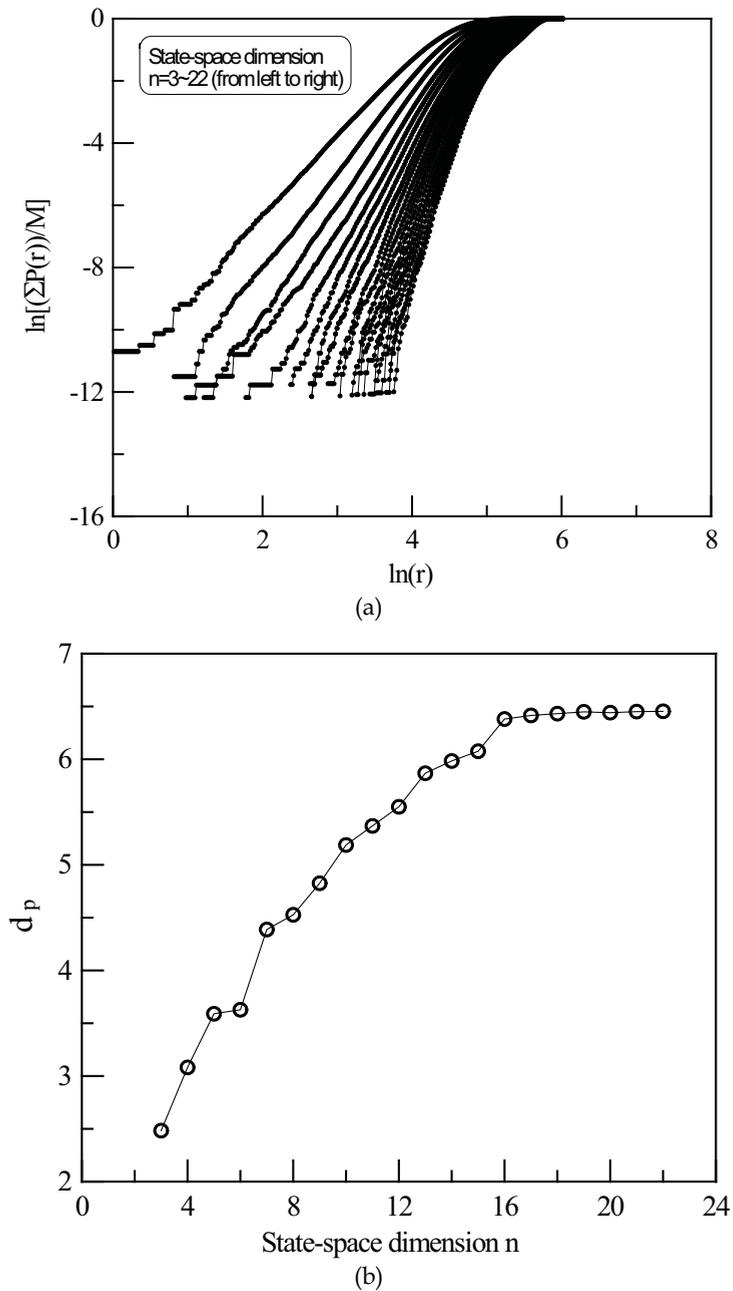


Fig. 5. (a) Limiting behavior of the function $\ln[(\Sigma P(r))/M]$ as $r \rightarrow 0$ for time delay $\tau = 20$ and (b) the asymptote of pointwise dimension, with the state-space dimension n increasing from 3 to 22 for the 5-min traffic volume.

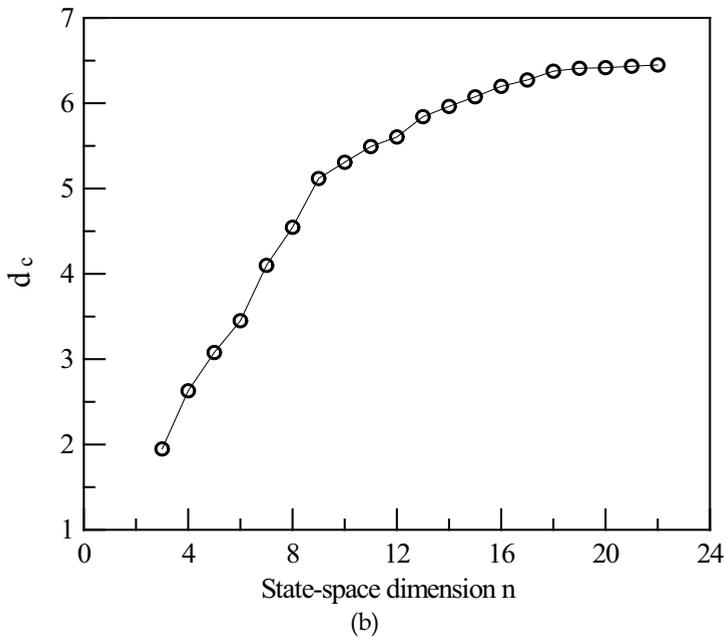
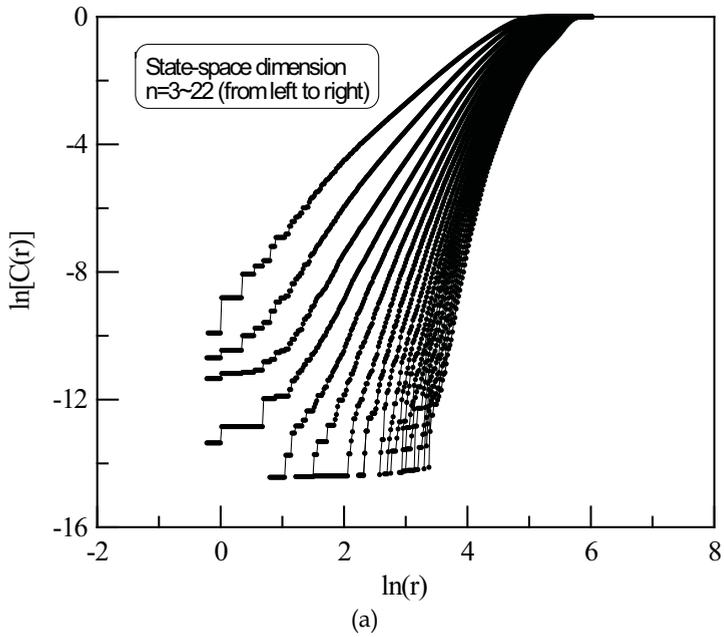


Fig. 6. (a) Limiting behavior of the function $\ln C(r)$ as $r \rightarrow 0$ for time delay $\tau = 20$ and (b) the asymptote of correlation dimension, with the state-space dimension n increasing from 3 to 22 for the 5-min traffic volume.

value (network outputs) are shown in Fig. 10. If the strategy “early stopping” is disregarded and 100 epochs is chosen for the training process, the trained network performance indeed improves for the training set, but gets worse for the validation and prediction sets. If the number of neurons in the hidden layer is increased to 28 and 42, the performance of the network for the training set tends to improve, but does not have the tendency to significantly improve for the validation and prediction sets, as listed in Table 3.

No. of Neurons	14	28	42
Data			
Training Set	0.90249	0.90593	0.94371
Validation Set	0.86535	0.86614	0.86757
Prediction Set	0.83086	0.85049	0.82901

Table 3. Correlation coefficients for training, validation and prediction data sets with the number of neurons in the hidden layer increasing (5-min traffic volume).

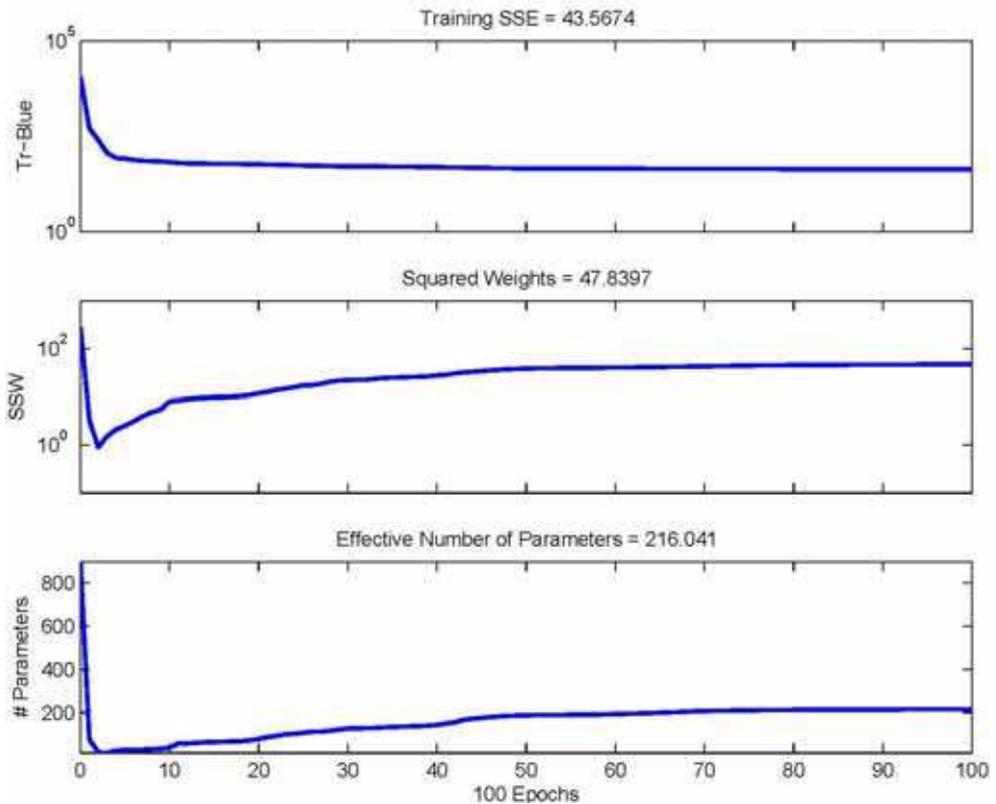


Fig. 7. The convergence process to find effective number of parameters used by the network for the 5-min traffic volume

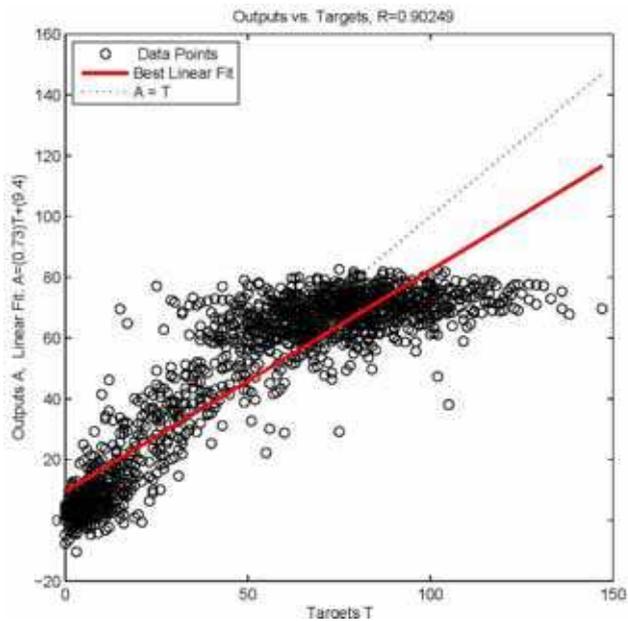


Fig. 8. The scatter plot of the network outputs and targets for the training set of the 5-min traffic volume.

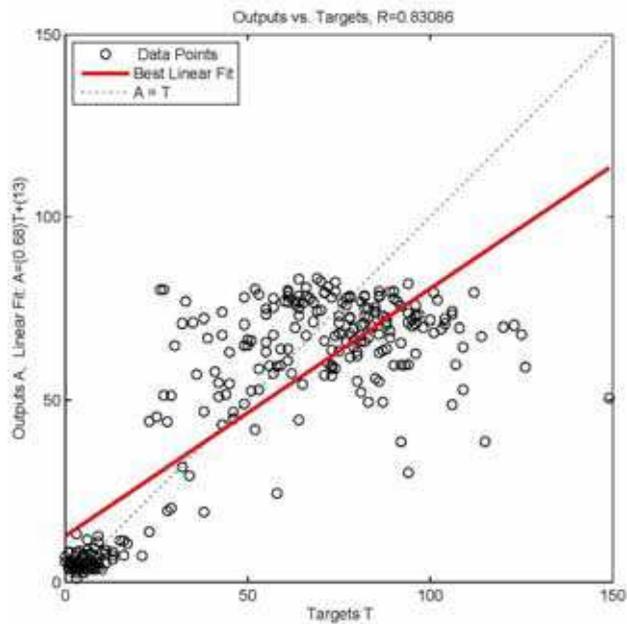


Fig. 9. The scatter plot of the network outputs and targets for the prediction set of the 5-min traffic volume.

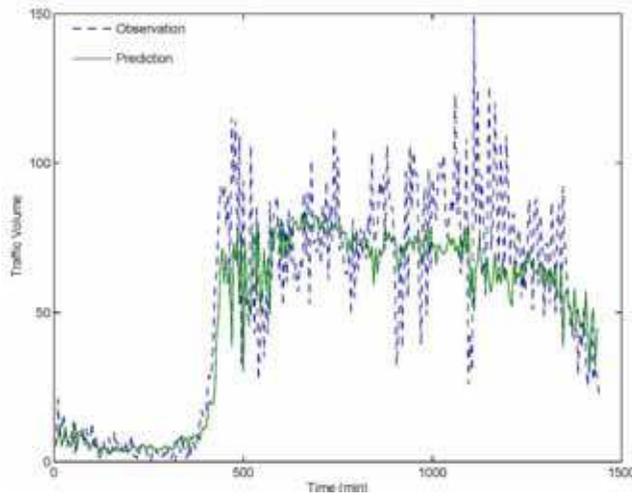


Fig. 10. Time series of the observed value (network targets) and the predicted value (network outputs) for the 5-min traffic volume.

4.2.2 10-min traffic volume

The network inputs and targets are the 14-dimensional delay coordinates: $x(i)$, $x(i-10)$, $x(i-20), \dots, x(i-130)$, and $x(i+1)$, respectively. Similarly, by using Bayesian regularization, the effective number of parameters is first found to be 108, as shown in Fig. 11; therefore, the appropriate number of neurons in the hidden layer is 7 (one half of the number of elements in the input vector). Replace the number of neurons in the hidden layer with 7 and train the network again. The training process stops at 11 epochs because the validation error has increased for 5 iterations. Fig. 12 shows the scatter plot for the training set with correlation coefficient $\rho=0.93874$. Simulate the trained network with the prediction set. Fig. 13 shows the scatter plot for the prediction set with the correlation coefficient $\rho=0.91976$. Time series of the observed value (network targets) and the predicted value (network outputs) are shown in Fig. 14. If the strategy “early stopping” is disregarded and 100 epochs is chosen for the training process, the performance of the network improves for the training set, but gets worse for the validation and prediction sets. If the number of neurons in the hidden layer is increased to 14 and 28, the performance of the network for the training set tends to improve, but does not have the tendency to improve for the validation and prediction sets, as listed in Table 4.

No. of Neurons		7	14	28
Data				
Training Set	0.93874	0.95814	0.96486	
Validation Set	0.92477	0.87930	0.88337	
Prediction Set	0.91976	0.90587	0.91352	

Table 4. Correlation coefficients for training, validation and prediction data sets with the number of neurons in the hidden layer increasing (10-min traffic volume).

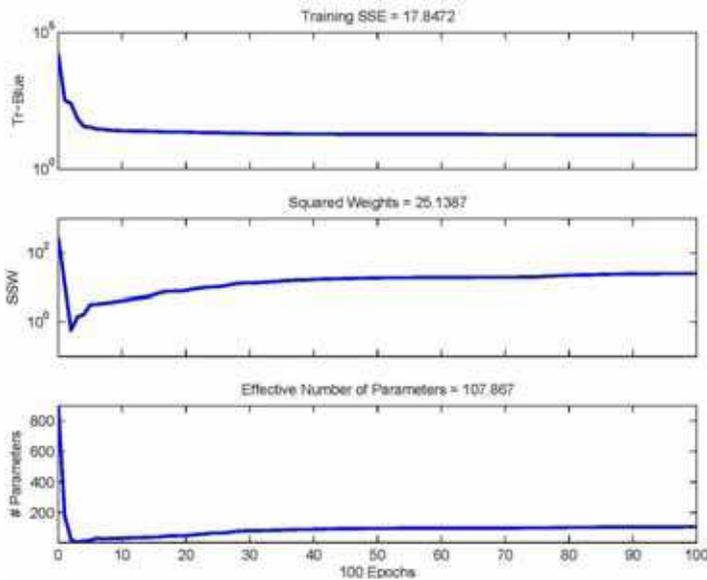


Fig. 11. The convergence process to find effective number of parameters used by the network for the 10-min traffic volume.

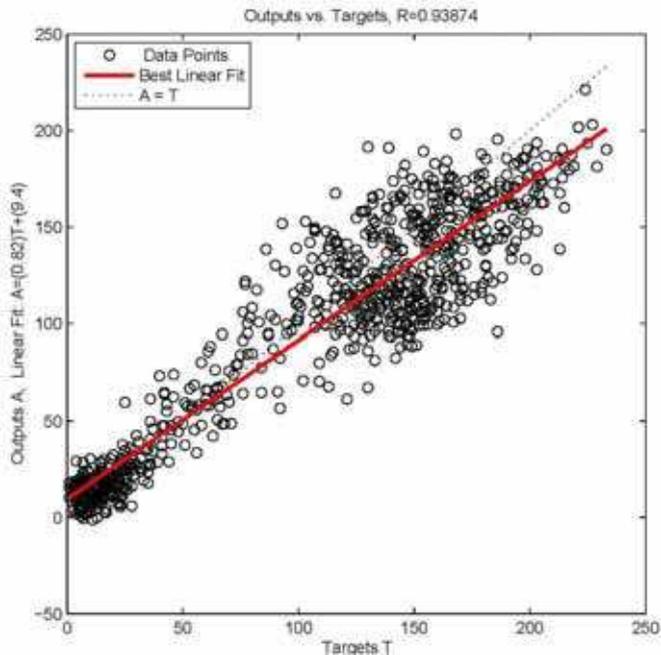


Fig. 12. The scatter plot of the network outputs and targets for the training set of the 10-min traffic volume.

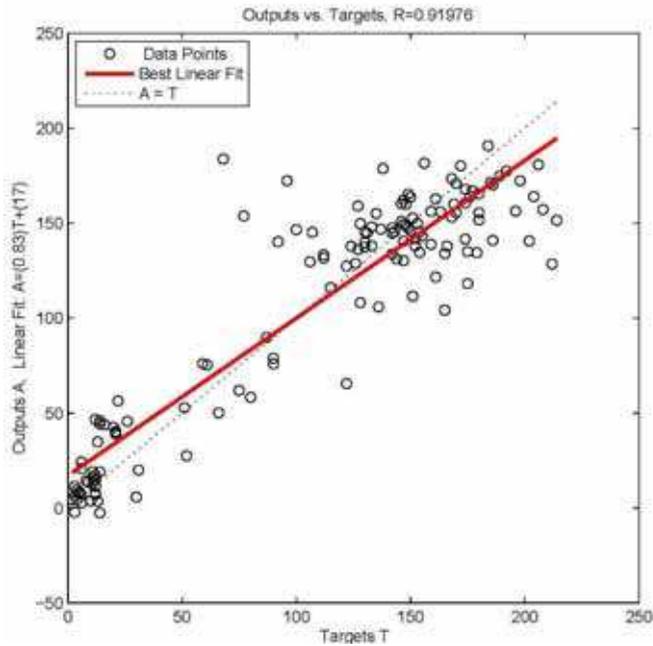


Fig. 13. The scatter plot of the network outputs and targets for the prediction set of the 10-min traffic volume.

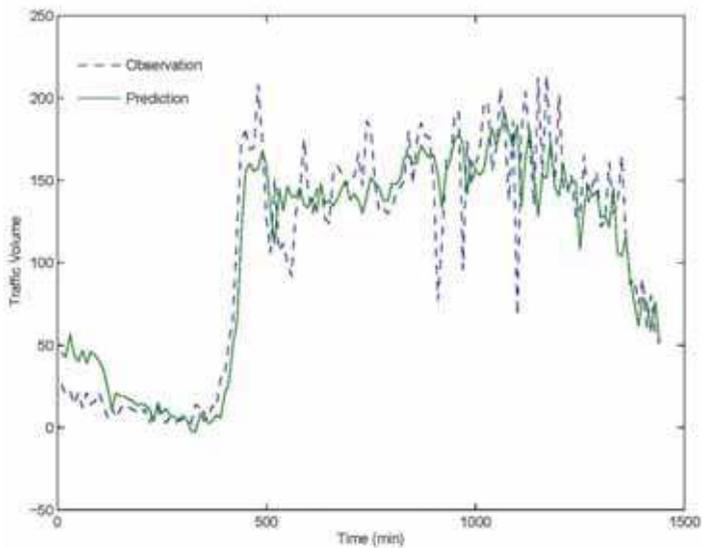


Fig. 14. Time series of the observed value (network targets) and the predicted value (network outputs) for the 10-min traffic volume.

4.2.3 15-min traffic volume

The network inputs and targets are the 14-dimensional delay coordinates: $x(i)$, $x(i-5)$, $x(i-10), \dots, x(i-65)$, and $x(i+1)$, respectively. In a similar way, the effective number of parameters is found to be 88 from the results of Bayesian regularization, as shown in Fig. 15. Instead of using 6 neurons obtained by Eq. (11), 7 neurons (one half of the number of elements in the input vector), are used in the hidden layer for consistence. Replace the number of neurons in the hidden layer with 7 and train the network again. The training process stops at 11 epochs because the validation error has increased for 5 iterations. Fig. 16 shows the scatter plot for the training set with correlation coefficient $\rho=0.95113$. Simulate the trained network with the prediction set. Fig. 17 shows the scatter plot for the prediction set with the correlation coefficient $\rho=0.93333$. Time series of the observed value (network targets) and the predicted value (network outputs) are shown in Fig. 18. If the strategy “early stopping” is disregarded and 100 epochs is chosen for the training process, the performance of the network gets better for the training set, but gets worse for the validation and prediction sets. If the number of neurons in the hidden layer is increased to 14 and 28, the performance of the network for the training set tends to improve, but does not have the tendency to significantly improve for the validation and prediction sets, as listed in Table 5.

No. of Neurons				
		7	14	28
Data	Training Set	0.95113	0.96970	0.97013
	Validation Set	0.88594	0.93893	0.92177
	Prediction Set	0.93333	0.94151	0.94915

Table 5. Correlation coefficients for training, validation and prediction data sets with the number of neurons in the hidden layer increasing (15-min traffic volume).

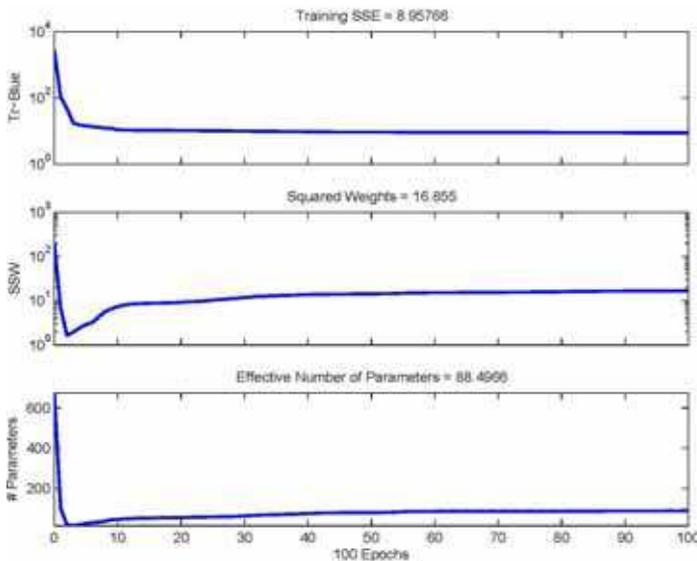


Fig. 15. The convergence process to find effective number of parameters used by the network for the 15-min traffic volume.

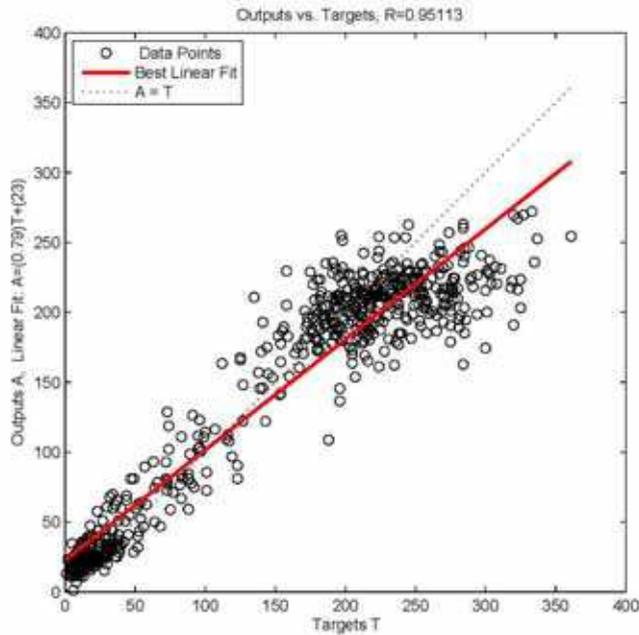


Fig. 16. The scatter plot of the network outputs and targets for the training set of the 15-min traffic volume.

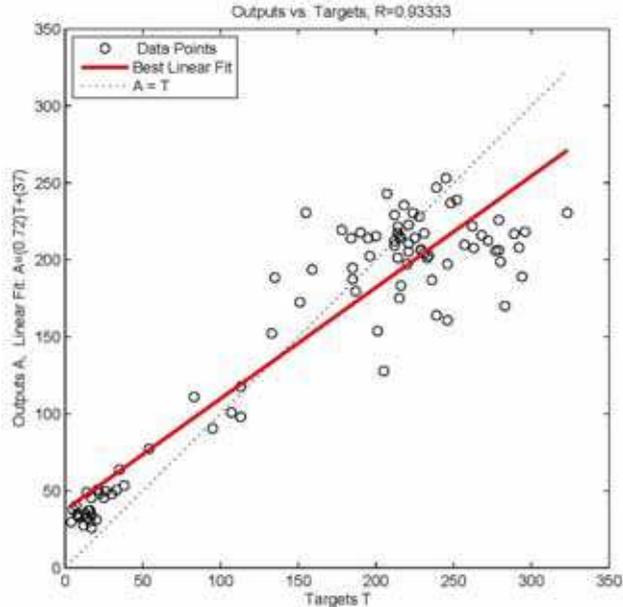


Fig. 17. The scatter plot of the network outputs and targets for the prediction set of the 15-min traffic volume.

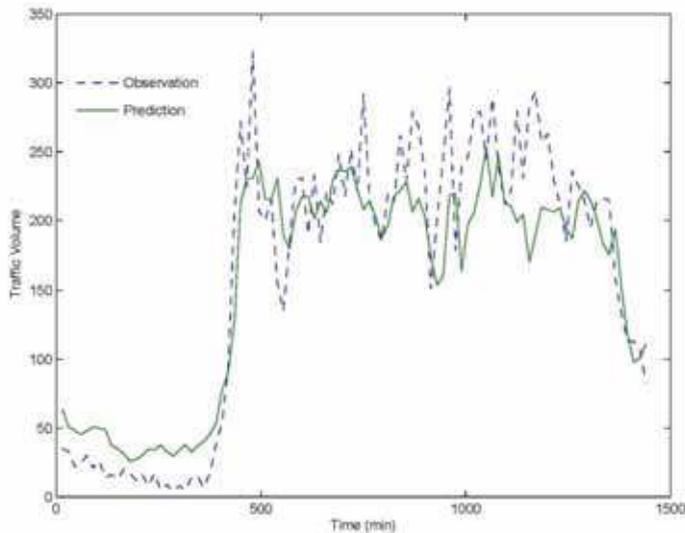


Fig. 18. Time series of the observed value (network targets) and the predicted value (network outputs) for the 15-min traffic volume.

4.3 The multiple linear regression

Data collected for the first nine days are used to build the prediction model, and data collected for the tenth day to test the prediction model. To forecast the near future behavior of a trajectory in the reconstructed 14-dimensional state space with time delay $\tau=20$, the number of 200 nearest states of the trajectory, after a few trials, is found appropriate for building the multiple linear regression model. Figs. 19-21 show time series of the predicted and observed volume for 5-min, 10-min, and 15-min intervals whose correlation coefficients ρ 's are 0.850, 0.932 and 0.951, respectively. All forecasts are all one time interval ahead of occurrence, i.e., 5-min, 10-min and 15-min ahead of time. These three figures indicate that the larger the time interval, the better the performance of the prediction mode. To study the effects of the number K of the nearest states on the performance of the prediction model, a number of K 's are tested for different time intervals. Figs. 22-24 show the limiting behavior of the correlation coefficient ρ for the three time intervals. These three figures reveal that the larger the number K , the better the performance of the prediction mode, but after a certain number, the correlation coefficient ρ does not increase significantly.

5. Conclusions

Numerical experiments have shown the effectiveness of the techniques introduced in this chapter to predict short-term chaotic time series. The dimension of the chaotic attractor in the delay plot increases with the dimension of the reconstructed state space and finally reaches an asymptote, which is fractal. A number of time delays have been tried to find the limiting dimension of the chaotic attractor, and the results are almost identical, which indicates the choice of time delay is not decisive, when the state space of the chaotic time series is being reconstructed. The effective number of neurons in the hidden layer of neural networks can be derived with the aid of the Bayesian regularization instead of using the trial and error.

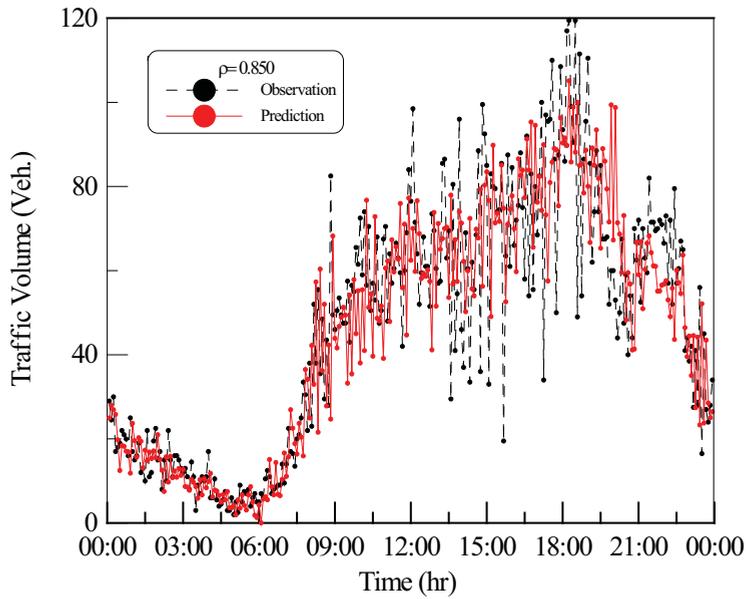


Fig. 19. Time series of the predicted and observed 5-min traffic volumes.

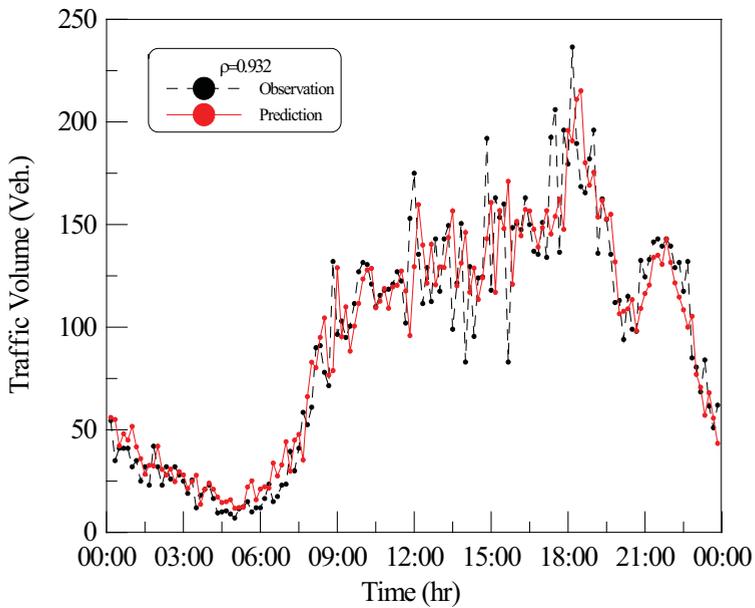


Fig. 20. Time series of the predicted and observed 10-min traffic volumes.

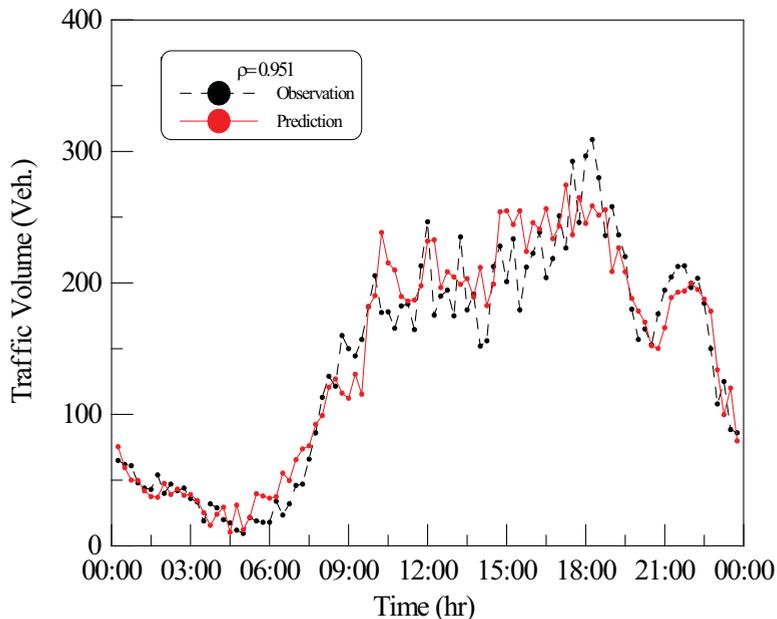


Fig. 21. Time series of the predicted and observed 15-min traffic volumes.

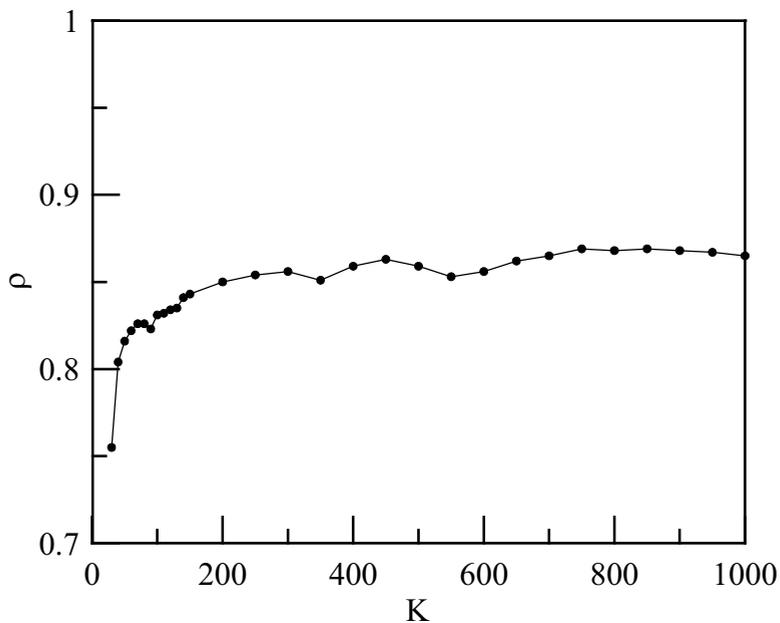


Fig. 22. The limiting behavior of the correlation coefficient ρ with K increasing for the 5-min traffic volume.

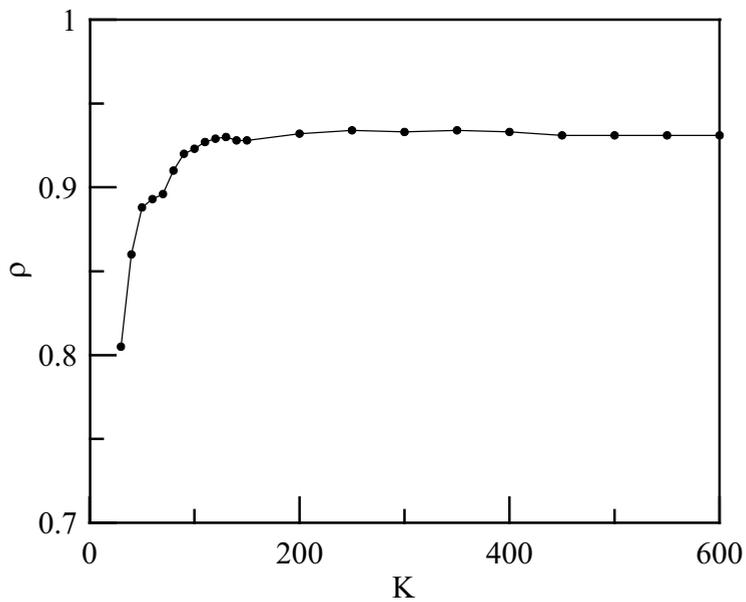


Fig. 23. The limiting behavior of the correlation coefficient ρ with K increasing for the 10-min traffic volume.

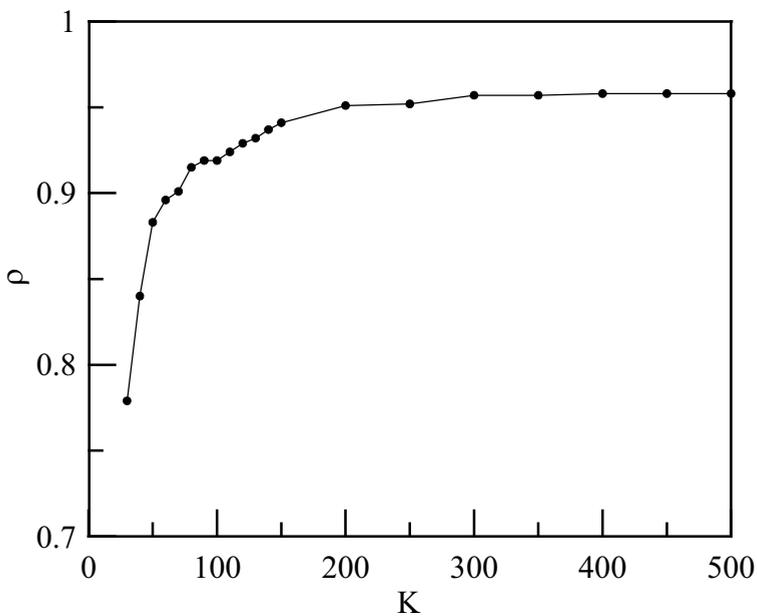


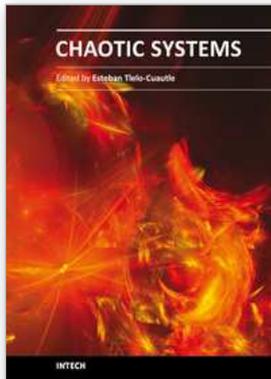
Fig. 24. The limiting behavior of correlation coefficient ρ with K increasing for the 15-min traffic volume.

Using neurons in the hidden layer more than the number decided by the Bayesian regularization can indeed improve the performance of neural networks for the training set, but does not necessarily better the performance for the validation and prediction sets. Although disregarding the strategy “early stopping” can improve the network performance for the training set, it causes worse performance for the validation and prediction sets. Increasing the number of nearest states to fit the multiple linear regression forecast model can indeed enhance the performance of the prediction, but after the nearest states reach a certain number, the performance does not improve significantly. Numerical results from these two forecast models also show that the multiple linear regression is superior to neural networks, as far as the prediction accuracy is concerned. In addition, the longer the traffic volume scales are, the better the prediction of the traffic flow becomes.

6. References

- Addison, P. S. and Low, D. J. (1996). Order and Chaos in the Dynamics of Vehicle Platoons, *Traffic Engineering and Control*, July/August, pp. 456-459, ISSN 0041-0683.
- Albano, A. M., Passamante, A., Hediger, T. and Farrell, M. E. (1992). Using Neural Nets to Look for Chaos, *Physica D*, Vol. 58, pp. 1-9, ISSN 0167-2789.
- Alligood, K. T., Sauer, T. D., and Yorke, J. A. (1997). *Chaos: An Introduction to Dynamical Systems*, Springer-Verlag, ISBN 3-540-78036-x, New York.
- Aquirre, L. A. and Billings, S. A. (1994). Validating Identified Nonlinear Models with Chaotic Dynamics, *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, Vol.4, No. 1, pp. 109-125, ISSN 0218-1274.
- Argoul, F., Arnedo, A., and Richetti, P. (1987). Experimental Evidence for Homoclinic Chaos in Belousov-Ehabotinski Reaction, *Physics Letters, Section A*, Vol. 120, No. 6, pp.269-275, ISSN 0375-9601.
- Bakker, R., Schouten, J. C., Takens, F. and van den Bleek, C. M. (1996). Neural Network Model to Control an Experimental Chaotic Pendulum, *Physical Review E*, 54A, pp. 3545-3552, ISSN 1539-3755.
- Deco, G. and Schurmann, B. (1994). Neural Learning of Chaotic System Behavior, *IEICE Transactions Fundamentals*, Vol. E77-A, No. 11, pp.1840-1845, ISSN 0916-8508.
- Demuth, H., Beale, M., and Hagan, M. (2010). *Neural Network Toolbox User's Guide*, The MathWorks, Inc., ISBN 0-9717321-0-8, Natick, Massachusetts.
- Dendrinis, D. S. (1994). Traffic-Flow Dynamics: A Search for Chaos, *Chaos, Solitons, & Fractals*, Vol. 4, No. 4, pp. 605-617, ISSN 0960-0779.
- Disbro, J. E. and Frame, M. (1989). Traffic Flow Theory and Chaotic Behavior, *Transportation Research Record* 1225, pp. 109-115. ISSN: 0361-1981
- Farmer, J. D. and Sidorowich, J. J. (1987). Predicting Chaotic Time Series, *Physical Review Letters*, Vol. 59, pp. 845-848, ISSN 0031-9007.
- Fu, H., Xu, J. and Xu, L. (2005). Traffic Chaos and Its Prediction Based on a Nonlinear Car-Following Model, *Journal of Control Theory and Applications*, Vol. 3, No. 3, pp. 302-307, ISSN 1672-6340.
- Gazis, D. C., Herman, R., and Rothery, R. W. (1961). Nonlinear Follow-The-Leader Models of Traffic Flow, *Operations Research*, Vol. 9, No. 4, pp. 545-567, ISSN 0030-364X.
- Glass, L., Guevau, X., and Shrier, A. (1983). Bifurcation and Chaos in Periodically Stimulated Cardiac Oscillator, *Physica 7D*, pp. 89-101, ISSN 0167-2789.

- Grassberger, P. and Proccacia, I. (1983). Characterization of Strange Attractors, *Physical Review Letters*, No. 50, pp. 346-349, ISSN 0031-9007.
- Hagan, M. T. and Menhaj, M. (1994). Training Feedforward Networks with the Marquardt Algorithm, *IEEE Transactions on Neural Networks*, Vol.5, No.6, pp. 989-903, ISSN 1045-9227.
- Hebb, D. O. (1949). *The Organization of Behavior*, John Wiley & Sons, ISBN 0-8058-4300-0, New York.
- Hense, A. (1987). On the Possible Existence of a Strange Attractor for the Southern Oscillation, *Beitr Physical Atmosphere*, Vol. 60, No. 1, pp. 34-47, ISSN 0005-8173.
- Hopfield, J. J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proceedings of the National Academy of Sciences of the USA*, Vol. 79, No. 8, pp. 2554-2558, ISSN 0027-8424.
- Hopfield, J. J., Feinstein D. I. and Palmers, R. G. (1983). Unlearning Has a Stabilizing Effect in Collective Memories, *Nature*, Vol. 304, pp. 158-159, ISSN 0028-0836.
- Levenberg, K. (1944). A Method for the Solution of Certain Problems in Least Squares, *Quarterly of Applied Mathematics*, No.2, pp.164-168, ISSN 0033-569X.
- MacKay, D. J. C. (1992). Bayesian Interpolation, *Neural Computation*, Vol. 4, No. 3, pp. 415-447, ISSN 0899-7667.
- Marquardt, D. (1963). An Algorithm for Least Squares Estimation of Nonlinear Parameters, *SIAM Journal on Applied Mathematics*, Vol.11, pp.431-441, ISSN 0036-1399.
- McCulloch, W. S. and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133, ISSN 0007-4985.
- Mendenhall, W., Scheaffer, R. L., and Wackerly, D. D. (1986). *Mathematical Statistics with Application*, Third Edition, Duxbury Press, ISBN 0-87150-939-3, Boston, Massachusetts.
- Moon, F. C. (1992). *Chaotic and Fractal Dynamics: An Introduction for Applied Scientists and Engineer*, John-Wiley and Sons, ISBN 0-471-54571-6, New York.
- Principe, J. C., Rathie, A. and Kuo, J. M. (1992). Prediction of Chaotic Time Series with Neural Networks and the Issue of Dynamic Modeling, *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, Vol.2, pp. 989-996, ISSN 0218-1274.
- Rosenblatt, F. (1958). The Perception: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, Vol. 65, No. 6, pp. 386-408, ISSN 0033-295X.
- Rumelhart, D. E. and McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1 (Foundations)*, The MIT Press, ISBN 0-262-68053-x, Cambridge, Massachusetts.
- Takens, F. (1981). *Detecting Strange Attractors in Turbulence*, *Lecture Notes in Mathematics*, No. 898, pp. 366-381.



Chaotic Systems

Edited by Prof. Esteban Tlelo-Cuautle

ISBN 978-953-307-564-8

Hard cover, 310 pages

Publisher InTech

Published online 14, February, 2011

Published in print edition February, 2011

This book presents a collection of major developments in chaos systems covering aspects on chaotic behavioral modeling and simulation, control and synchronization of chaos systems, and applications like secure communications. It is a good source to acquire recent knowledge and ideas for future research on chaos systems and to develop experiments applied to real life problems. That way, this book is very interesting for students, academia and industry since the collected chapters provide a rich cocktail while balancing theory and applications.

How to reference

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

Jiin-Po Yeh (2011). Short-Term Chaotic Time Series Forecast, Chaotic Systems, Prof. Esteban Tlelo-Cuautle (Ed.), ISBN: 978-953-307-564-8, InTech, Available from: <http://www.intechopen.com/books/chaotic-systems/short-term-chaotic-time-series-forecast>

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