In Proceedings of World Congress on Neural Networks 1993, Portland, volume 4, pages 332-336, 1993.

Using Recurrent Network in Time Series Prediction

Evan Fung-Yu YOUNG and Lai-Wan CHAN
Computer Science Department
Chinese University of Hong Kong, Shatin, HONG KONG
Email : lwchan@cs.cuhk.hk

Abstract

A recurrent network is useful in memorizing past information. We have studied the effectiveness of the recurrent network for modeling and forecasting time series. In this paper, we used a ring-structured recurrent backpropagation network (RRN). We present the results of a non-chaotic time series with white noise, a deterministic chaotic time series and the sunspots activity data. The network generates accurate predictions and outperforms a traditional statistical approach and a feedforward connectionist approach.

1 Introduction

Existing approaches for the time series prediction problem include the linear modelling method, *e.g.* autoregressive moving average [BJ70] and the non-linear model building, such as the bilinear models and the threshold autoregressive models [TL80]. Another new and promising concept is that of non-linear signal processing using neural network. Lapades [Lap87] had shown in an empirical study that neural networks can be used for modeling and forecasting non-linear time series. More researches are now on using feedforward connectionist net in modeling and forecasting time series.

In this paper, we study the feasibility of using recurrent neural network in time series prediction. Recurrent networks have an advantage over feedforward nets in time series prediction; there is no need to determine *in priori* the number of past values needed for the prediction. In recurrent networks, only the value at time t-1 is used to predict the value at time t. Since the recurrent links in the network can bring the appropriate previous values back to itself, the number of input unit is thus reduced from d to 1. In the following, we study the predictive power of one type of recurrent networks called RRN, in three different kinds of time series. We also compare the performance of RRN with other methods in these tasks.

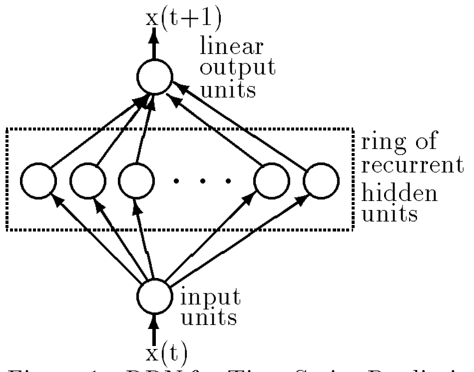

Figure 1 : RRN for Time Series Prediction

2 Network Topology

The network used is a ring-structured recurrent network (RRN) [CY93] with one input unit and one output unit (Fig. 1). The current value x_t is fed into the input unit and the prediction for the next value x_{t+1} is made at the output unit. There are recurrent feedback among the hidden units and they form a ring-structure. The output unit is linear but the hidden units are sigmoidal. Non-linearities are due to the sigmoid hidden units and this makes a connectionist approach different from other statistical linear models. Other work has shown that RRN has the advantage of having low computation and storage complexity of O(n) when compared with other recurrent networks [CY93].

3 Training Samples

In this paper, we study three different examples.

- **A Non-Chaotic Series with White Noise (Series 1)**

The series is generated from discrete points on a sine curve, having an amplitude of unity, and adding to it at each step a uniformly distributed random variable in the interval [-0.5, 0.5] ($i.e. y(t) = sin(10t) + random[-0.5, +0.5]$, $t = 0, 1, 2, ...$). This series is non-chaotic but the periodicities are masked by noise and it displays character similar to a chaotic system.

- **A Deterministic Chaotic Series (Series 2)**

A deterministic chaotic series is produced by a completely deterministic, noise-free system, but the series looks random. Very accurate predictions of the next value are possible by fitting a function through the

Series	δ_t^2	r_t	δ_p^2	r_p
1	0.017067	0.983961	0.032094	0.973552
2	2.5271×10^{-5}	0.999974	2.9020×10^{-5}	0.999972
3	130	0.942927	313	0.944053

Table 1: Single-Step Prediction - Residual Variance and Correlation Coefficient in both the Training and Forecasting Period.

Model	δ_t^2	δ_{35}^2	δ_{55}^2
TAR	149	152	1302
CNAR	131	141	2016
RRN	130	140	1176

Table 2: δ_t^2, δ_{35}^2 and δ_{55}^2 for the Three Models TAR, CNAR and RRN

points y_{t-1} and y_t. Deterministic chaos is characterized by an exponential divergence of nearby trajectories. If one bit of information is lost in each iteration, the uncertainty in prediction will increases exponentially with time and any long-term predictability is precluded. The deterministic chaos we used in this test is generated by $y_t = 1 - 2y_{t-1}^2, \quad t = 0, 1, 2, ...$ All values generated by this iterative quadratic formula lie within the unit interval if the initial value is between -1 and +1.

- **A Non-Deterministic Chaotic Series (Series 3)**

The sunspots series has been serving as a benchmark of non-deterministic chaotic series. It is believed that the statistical mechanism by which the sunspots numbers are generated is non-linear and non-stationary and the series has been used as a yardstick to compare new modeling and forecasting methods.

For Series 1 and 2, we generate 100 data points to train the neural network and the next 20 points are used for prediction. For Series 3, we use the data from 1700 to 1979 which has a cyclical behavior with a mean cycle of 11.2 years. We use the data from 1700 through 1920 for training and the remaining data for evaluation of the prediction, in order to have a fair comparison with the other methods.

4 Performance on Time Series Prediction

4.1 Method of Analysis

We adjusted the weights and thresholds to achieve the smallest deviation between the target values and the predicted values. The training data is fed into the network repeatedly until the root mean square error of the predictions is less than a threshold. The testing data is then used to evaluate the predictive power. We use two different measures for evaluations, the residual variance and the correlation coefficient.

- **Residual Variance, δ^2.** The residual e_i is the difference between the observed value and the predicted value. The variance of these residuals gives some measure on the goodness of fitting the model to the observed data ($i.e. \delta^2 = \frac{1}{n-1}\sum_{i=1}^{n} e_i^2$).

- **Correlation Coefficient, r.** The correlation coefficient between actual and predicted values is widely used as a measure of predictive skill. If we have a sample of n pairs of values, (X_i, Y_i), i=1, 2, ..., n, the correlation coefficient, r, between X and Y is define by $r = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2}\sqrt{\sum_i (X_i - \bar{X})^2}}$ where $\bar{X} = \frac{1}{n}\sum_i X_i$ and $\bar{Y} = \frac{1}{n}\sum_i Y_i$ are the sample means of X and Y respectively. r is also satisfying the inequalities $-1 \leq r \leq 1$ and that $\mid r \mid = 1$ if and only if all of the n sample points lie on a straight line. The further from unity is the absolute value of r, the further do the values depart from this linear relationship.

4.2 Result

We compare the residual variance and the correlation coefficient in both the training period and forecasting period. There are two ways to do prediction, single-step prediction and multi-step prediction.

- **Single-Step Prediction.** The term single-step prediction is used when all input units are given values of the observed time series during the prediction. Fig. 2, 3 and 4 plot the the original time series and the predicted values of the series 1, 2 and 3 respectively. The residual variance and correlation coefficient of both the training (δ_t^2 and r_t) and forecasting (δ_p^2 and r_p) period are recorded and summerized in Table 1.

- **Multi-Step Prediction.** To achieve predictions several steps into the future, the predicted output is fed back as input for the next prediction and all other inputs are shifted one unit back. Hence the input is consisted of predicted values as opposed to actual observations of the original time series. The prediction error is now dependent on the time lagged after prediction is started. We are interested in the maximum number of steps into which predictions can be made accurately and this can be obtained by computing the correlation coefficient or the residual variance as a function of prediction time.

4.3 Prediction Result

• **Series 1 - Sine Function with White Noise**
From Figure 2, although the predicted values do not fit the original series very well, the predicted values followed quite smoothly the shape of a pure sine curve in both the single-step and multi-step prediction cases. It is due to the large amplitude of the white noise (from -0.5 to +0.5) in comparison with the pure sine function, so the network cannot optimize its weights and thresholds to fit the series at all points perfectly. However the performance is still satisfactory and the network seems to neglect the noise. The correlation coefficient is above 0.97 in both the training and forecasting period (Table 1).

Figure 5 plots the correlation coefficients against the prediction time in multi-step prediction. Except the drop at the very beginning, the correlation coefficient is above 0.9 throughout the whole period. This suggests an independence of predictive skill with time. The error remains almost the same regardless of how far into the future one tries to predict.

• **Series 2 - Iterative Quadratic Series**
In Fig. 3, we can see a very good fitting between the observations and the predictions in single-step prediction. The network can extract the deterministic behavior of the series although it appears to be chaotic. The correlation coefficients in both periods of Series 2 are very close to one (Table 1). The network can predict so perfectly because the series is deterministic and noise free. For multi-step prediction, we find that the network can predict six to seven steps into the future quite accurately. This is shown more clearly in Figure 6 which plots the correlation coefficient against the prediction time. We can see that the correlation coefficient is very close to one in the first six to seven steps, then it drops to a small value and remains almost unchanged there. We cannot expect accurate predictions too far into the future because the original series is chaotic and a small prediction error will increase exponentially with time. However it is interesting to compare Figure 5 with Figure 6. We find that the correlation coefficient drops with time in a chaotic series while that in a non-chaotic series remain almost unchanged regardless of the prediction length. This suggests a method to distinguish chaos from noise in time series records. It is also possible to quantitatively compare the rates of degradation in predictive skill as an indication of the amount of chaos in a system.

• **Series 3 - Sunspots Data**
The results of single-step prediction in sunspots data is given in Figure 4. Although the testing set includes the highest cycle ever observed which is a situation not contained in the training set, the prediction fits the original series very accurately for the single step case.

For multi-step prediction, Figure 4 shows that the prediction is no longer fitting the original series accurately but the predicted values can still follow the observed series cycle by cycle. Figure 7 plots the correlation coefficient against the prediction time. We can see that the coefficient drops very slowly with the prediction length and this suggests a small amount of chaos in the sunspots activity.

4.4 Comparison with other methods

4.4.1 Comparison of RRN, TAR and CNAR on Sunspot Data

The sunspots series opens up the area of autoregressive modeling and many attempts have been undertaken to model the data by a linear ARMA process. Recent study shows that a better modeling can be done by a non-linear process. We compare the performance of RRN with the threshold autoregressive models (TAR) [Ton83] which is a non-linear statistical model, and a feedforward connectionist model called feedforward autoregressive connectionist net CNAR [dGW91].

The comparison is made based on the residual variance in the training period δ_t^2 and the residual variance of forecasting 35 points δ_{35}^2 in single-step prediction (Table 2). We can see that the predictive skill of RRN and CNAR are very similar and both of them are better than TAR in both periods. In multi-step prediction, Table 2 gives the residue variances of forecasting 55 steps ahead into the future δ_{55}^2 for the three models. The best 55 steps ahead forecast is achieved by RRN, followed closely by TAR and then CNAR.

4.4.2 Comparison of RRN and RTRL in Time Series Prediction

We have used the three series as described above to compare the performance of RRN and the Real Time Recurrent Learning Algorithm (RTRL) [WZ89] in time series prediction. The study emphasizes on both the training speed and the predictive power. In order to have a fair comparison, RTRL is slightly modified in such a way that only the hidden layer is fully recurrent. The connections running from the input layer to the hidden layer or from the hidden layer to the output layer are just ordinary feedforward links.

The performance of RRN and RTRL in the training period is described in Table 3. Training RTRL to learn the sunspots number needs about four days but RRN needs about 9 hours. This speed up is significant and is even larger in comparison with that of sequence recognition [CY93]. It is because the large training set (at least 100 data points, 220 in the sunspots data) increases the computational complexity of RTRL significantly. RRN is shown to be practically more useful then RTRL for larger problems.

	RRN					RTRL				
series	no. of hidden units	no. of recurrent links	no. of cycles	time taken (sec)	final rms error	no. of hidden units	no. of recurrent links	no. of cycles	time taken (sec)	final rms error
1	10	30	90000	25675	0.008448	10	100	30000	430451	0.008320
2	10	30	37305	24194	0.000013	10	100	30000	148575	0.000012
3	10	30	82800	33103	0.3232	10	100	30000	336922	0.2616

Table 3: Performance of RRN and RTRL in the Training Period of Time Series Prediction

Series	δ_t^2 in RRN	δ_p^2 in RRN	δ_t^2 in RTRL	δ_p^2 in RTRL
1	0.017067	0.032094	0.016808	0.025880
2	2.5271×10^{-5}	2.9020×10^{-5}	2.4870×10^{-5}	2.5040×10^{-5}
3	130	313	105	303

Table 4: Performance of Simple RRN and RTRL in Single-Step Prediction

For the predictive power of RRN and RTRL in single-step prediction, Table 4 shows that the performance of RTRL is better than that of RRN in both the training and forecasting period. This is expectable because RTRL has more recurrent links and it is actually doing more work than RRN. However there are still many factors affecting the results. One obvious example is that the number of hidden unit used (i.e. 10 in the above cases), may not be optimal for RRN or RTRL.

In multi-step prediction, we have plotted the correlation coefficient against prediction time for both methods on Figures 5, 6, 7 and it is found that the correlation coefficient of RTRL drops slower than that of RRN.

5 Conclusion

We have used the recurrent network RRN to model and forecast time series. We have applied it to three typical examples, a non-chaotic time series with a white noise component, a deterministic chaotic time series and the sunspots activity data. In the first case, the network can extract the noise from the series and the predictive skill remains unchanged regardless of how many steps ahead into future the prediction is made. In the second case, the network shows excellent forecasting ability in single-step prediction but the predictive skill drops after forecasting six to seven steps ahead into the future. This marked difference between a chaotic and non-chaotic series can serve as a way to distinguish chaos from noise. In the third case, the network can model the complex features inherent in the sunspots data quite satisfactorily and it outperforms the threshold autoregressive model TAR and the feedforward autoregressive connectionist net CNAR in both single-step and multi-step prediction.

We have also compared RRN and RTRL in time series prediction. We find that RTRL has a better predictive skill than RRN. This may be due to the larger number of recurrent links used in RTRL and other factors, like initial conditions and number of hidden units, may also affect the results.

References

[BJ70] G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, San Francisco, CA, 1970.

[CY93] Lai-Wan Chan and Evan F.Y. Young. Ring structured recurrent backpropagation network. In *paper in submission*, 1993.

[dGW91] C. de Groot and D. Wurtz. Analysis of univariate time series with connectionist nets: A case study of two classical examples. *Neurocomputing*, pages 177–192, 1991.

[Lap87] A. Lapades. Nonlinear signal processing using neural networks: Prediction and system modelling. Technical Report LA-UR87-226, Los Alamos, 1987.

[TL80] H. Tong and K.S. Lim. Threshold autogressive, limit cycles and cyclical data. *J.R. Stat. Soc.*, 245, 1980.

[Ton83] H. Tong. *Threshold Models in Non-linear Time Series Analysis, Lecture Notes in Statistics Vol.21.* Springer, Berlin, 1983.

[WZ89] R.J. William and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1989.

Figure 2: Observed and predicted values of Series 1 using RRN

Figure 3: Observed and predicted values of Series 2 using RRN

Figure 4: Observed and predicted values of Series 3 using RRN

Figure 5: Absolute correlation coefficient in multi-step prediction of Series 1

Figure 6: Absolute correlation coefficient in multi-step prediction of Series 2

Figure 7: Absolute correlation coefficient in multi-step prediction of Series 3

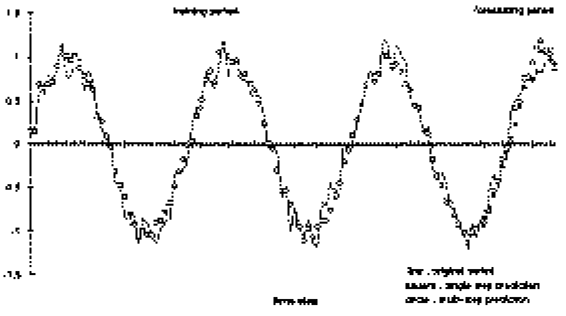

Figure 2: Observed and predicted values of Series 1 using RRN

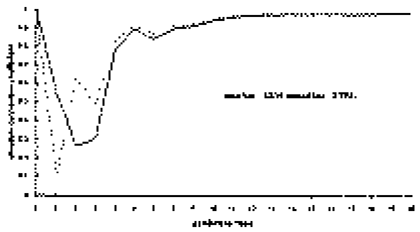

Figure 5: Absolute correlation coefficient in multi-step prediction of Series 1

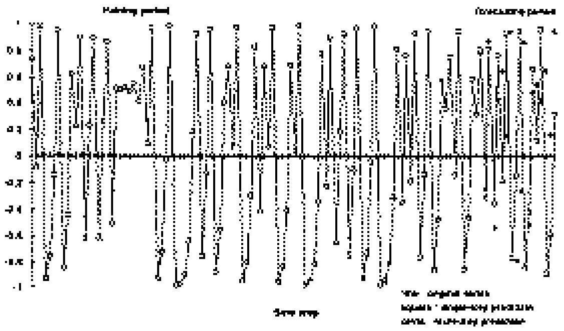

Figure 3: Observed and predicted values of Series 2 using RRN

Figure 6: Absolute correlation coefficient in multi-step prediction of Series 2

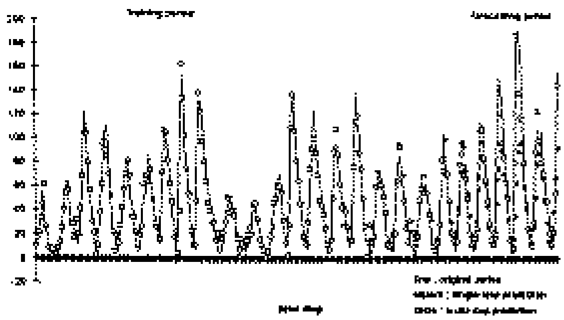

Figure 4: Observed and predicted values of Series 3 using RRN

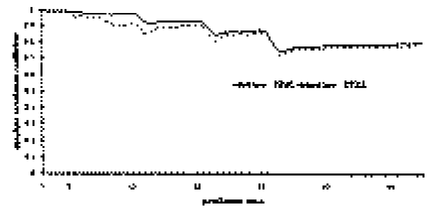

Figure 7: Absolute correlation coefficient in multi-step prediction of Series 3