

Bayesian Neural Network with Recurrent Architecture for Time Series Prediction

Chan-Young Hong*, Jung-Hun Park*, Tae-Sung Yoon** and Jin-Bae Park*

* Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
(Tel : +82-2-2123-2773; E-mail: cyhong@control.yonsei.ac.kr)

**Department of Electrical Engineering, Changwon National University, Changwon, Korea
(Tel : +82-55-279-7513; E-mail: tsyoon@sarim.changwon.ac.kr)

Abstract: In this paper, the Bayesian recurrent neural network (BRNN) is proposed to predict time series data. Among the various traditional prediction methodologies, a neural network method is considered to be more effective in case of non-linear and non-stationary time series data. A neural network predictor requests proper learning strategy to adjust the network weights, and one need to prepare for non-linear and non-stationary evolution of network weights. The Bayesian neural network in this paper estimates not the single set of weights but the probability distributions of weights. In other words, we sets the weight vector as a state vector of state space method, and estimates its probability distributions in accordance with the Bayesian inference. This approach makes it possible to obtain more exact estimation of the weights. Moreover, in the aspect of network architecture, it is known that the recurrent feedback structure is superior to the feedforward structure for the problem of time series prediction. Therefore, the recurrent network with Bayesian inference, what we call BRNN, is expected to show higher performance than the normal neural network. To verify the performance of the proposed method, the time series data are numerically generated and a neural network predictor is applied on it. As a result, BRNN is proved to show better prediction result than common feedforward Bayesian neural network.

Keywords: time series prediction, Bayesian inference, recurrent neural network, Bayesian neural network

1. INTRODUCTION

The prediction techniques for time series enable to forecast the future behavior of a system, which is the reason why the time series analysis gathers wide interest in various field including mathematics and economics. The classical method for time series analysis has used the statistical approach to figure out the optimal coefficients of the system model. The recent researches, however, have proved that the artificial neural network ensures higher accuracy for the prediction of non-linear system. Therefore, various types of neural network have been attempted to the time series analysis in modified manners[1~3].

The Bayesian neural network (BNN) is suggested by Mackay with Gaussian approximation[4], and expanded by Neal with Markov chain Monte Carlo method[5]. In the Bayesian learning approach for neural network, one can acquire the probability distribution over weight values. This characteristic of BNN bears some advantages over common neural network[6]; First, it is needless to separate the data into training set and validation set, which is helpful for more accurate model. Second, network output has the predictive distribution and confidence interval. Therefore, there has been diverse attempts for the solution of Monte Carlo method to implement BNN[7].

Most researches about BNN have adopted the fundamental network architecture of feedforward multi-layer perceptron[8~9]. However, the recurrent, or feedback network structure is considered to be more effective to manage the time series data. In this paper, we apply the Bayesian framework to the recurrent neural network (RNN) for the purpose of time series prediction.

This paper is organized as follows. In Section 2, the principles of Bayesian inference on neural network and recursive Monte Carlo method are discussed. In Section 3, the structure and propriety of the proposed neural network system is explained. In Section 4, the method is validated through the simulation with non-linear and non-stationary time series data, and Section 5 concludes the paper.

2. BAYESIAN INFERENCE FOR NEURAL NETWORK

The neural network approaches to the optimal system model by updating the network weights whenever new data is obtained. The general update method is the backpropagation algorithm based on the delta rule. But this adjusting process can be interpreted as a process of the state space method[10]. Namely, the network weights vector w_k is regarded as state vector of state space model of Eqs. (1) and (2).

$$\text{process equation:} \quad w_{k+1} = w_k + d_k \quad (1)$$

$$\text{measurement equation:} \quad y_k = g(w_k, x_k) + v_k, \quad (2)$$

Equation (1) means the evolution of the weights and Eq. (2) means the nonlinear relation of input x_k and output y_k of network. The nonlinear mapping $g(\cdot)$ represents the neural network. d_k denotes the uncertainty of evolution and v_k denotes the corruption by noise. They are considered to be zero mean Gaussian distribution with covariance Q and R , respectively.

Now the decision of weights is the problem of estimating a state vector given the measurements $Y_k = \{y_1, y_2, \dots, y_k\}$. BNN is the procedure of solving the nonlinear state space equations through Bayesian learning, or Bayesian inference in other words. The main interest of Bayesian inference is the probability density function (pdf) of $p(w_k | Y_k)$, which enables us to know various statistical properties of weights, including centroids, modes, medians and confidence intervals. As the estimation of the network weights gets more precise, so does the estimation of network output.

Bayesian approach to train neural network is similar to the filtering scheme. In order to obtain the predictive distribution of weight state, it recursively iterates the two stages: prediction and update. The prediction stage propagates the past probability density of $p(w_{k-1} | Y_{k-1})$ to prior pdf of

$p(w_k | Y_{k-1})$ at time step k through Eq. (3).

$$\begin{aligned} p(w_k | Y_{k-1}) &= \int \frac{p(w_k, w_{k-1}, Y_{k-1})}{p(Y_{k-1})} dw_{k-1} \\ &= \int \frac{p(w_k, w_{k-1}, Y_{k-1})}{p(w_{k-1}, Y_{k-1})} \cdot \frac{p(w_{k-1}, Y_{k-1})}{p(Y_{k-1})} dw_{k-1} \\ &= \int p(w_k | w_{k-1}) p(w_{k-1} | Y_{k-1}) dw_{k-1} \end{aligned} \quad (3)$$

Here, the transition density $p(w_k | w_{k-1})$ is defined as following derivation.

$$\begin{aligned} p(w_k | w_{k-1}) &= \int \frac{p(w_k, d_{k-1}, w_{k-1})}{p(w_{k-1})} dd_{k-1} \\ &= \int \frac{p(w_k, d_{k-1}, w_{k-1})}{p(d_{k-1}, w_{k-1})} \cdot \frac{p(d_{k-1}, w_{k-1})}{p(w_{k-1})} dd_{k-1} \\ &= \int p(w_k | d_{k-1}, w_{k-1}) p(d_{k-1} | w_{k-1}) dd_{k-1} \\ &= \int \delta(w_k - d_{k-1} - w_{k-1}) p(d_{k-1}) dd_{k-1} \end{aligned} \quad (4)$$

In Eq. (4), $\delta(\cdot)$ is Dirac delta function, and $p(d_{k-1} | w_{k-1})$ is same with $p(d_{k-1})$ because of the assumption that noise terms d_k and v_k are independent of past and present values of the states.

The update stage starts after the observation of the new data. In this stage, the prior pdf is updated to posterior pdf in accordance with the Bayes' rule:

$$p(w_k | Y_k) = \frac{p(y_k | w_k) p(w_k | Y_{k-1})}{p(y_k | Y_{k-1})} \quad (5)$$

Here, the likelihood density $p(y_k | w_k)$ and the evidence density $p(y_k | Y_{k-1})$ are defined as following derivations of Eq. (6) and Eq. (7), respectively.

$$\begin{aligned} p(y_k | w_k) &= \int \frac{p(y_k, v_k, w_k)}{p(w_k)} dv_k \\ &= \int \frac{p(y_k, v_k, w_k)}{p(v_k, w_k)} \cdot \frac{p(v_k, w_k)}{p(w_k)} dv_k \\ &= \int p(y_k | v_k, w_k) p(v_k | w_k) dv_k \\ &= \int \delta(y_k - g(w_k, x_k) - v_k) p(v_k) dv_k \end{aligned} \quad (6)$$

$$\begin{aligned} p(y_k | Y_{k-1}) &= \int \frac{p(y_k, w_k, Y_{k-1})}{p(Y_{k-1})} dw_k \\ &= \int \frac{p(y_k, w_k, Y_{k-1})}{p(w_k, Y_{k-1})} \cdot \frac{p(w_k, Y_{k-1})}{p(Y_{k-1})} dw_k \\ &= \int p(y_k | w_k) p(w_k | Y_{k-1}) dw_k \end{aligned} \quad (7)$$

Summarizing the above two stages, one can get the prior pdf at time step k from past pdf through Eq. (3) (prediction), and obtain the posterior pdf from prior pdf and likelihood through Eq. (5) (update). The estimation of weight state vector

\hat{w}_k is determined by taking a mean value of posterior pdf. This procedure is depicted in Fig.1.

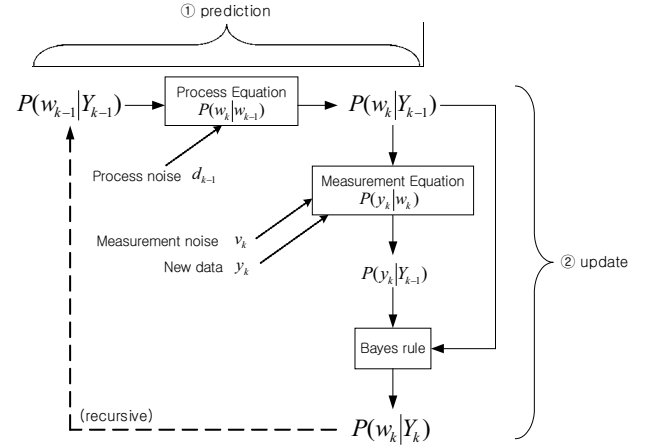


Fig. 1 Recursive Bayesian inference: prediction & update

The most serious problem in realization of the Bayesian inference is that there exist multi-dimensional integrations in prediction and update formulas. To overcome this difficulty, Monte Carlo integration method has been used. The Monte Carlo method transforms the integration into numerical summation by drawing samples from the density function. The accuracy of this approximation depends on the number of drawn samples. As the number of samples increases to infinite, the normalized histogram of the samples converge into the pdf. Therefore, one needs an efficient sampling strategy from pdf. Among the various kinds of Monte Carlo techniques, we use the recursive Monte Carlo method which adopts the sampling algorithm of recursive SIR.

The recursive SIR algorithm also follows Bayesian framework like next steps. First, for time step 0, M samples $\{w_0^i\}_{i=1}^M$ are generated from $p(w_0)$. Second, the likelihood weights $u_i = p(y_k | w_k^i)$ are computed for $i=1,2,\dots,M$, and normalized as $\tilde{u}_i = u_i / \sum_{j=1}^M u_j$. Third, a new set $\{w_k^{i*}\}_{i=1}^N$ is generated by resampling with replacement N times from $\{w_k^j\}_{j=1}^M$ where $\Pr(w_k^{i*} = w_k^j) = \tilde{u}_j$. Fourth, each resampled states are predicted independently r times for the generation of $\{w_{k+1}^j\}_{j=1}^M$, where $w_{k+1}^{(i-1)r+h} \sim p(w_{k+1} | w_k^{i*})$ for $h=1,2,\dots,r$ and $i=1,2,\dots,N$. Fifth, increase time k and iterate to the second step.

3. BAYESIAN RECURRENT NEURAL NETWORK

The performance of neural network relies not only the learning method but also the network architecture. Especially for the time series analysis, the recurrent network model is known to be better than general feedforward network. The basic forms of RNN are appeared in Fig.2.

RNN has the feedback connections among units. This dynamic property is appropriate for dealing with time that is important in many analysis tasks such as speech, control, and time series data. Elman network activates hidden units at time

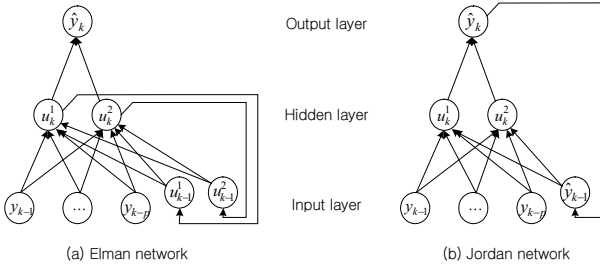


Fig. 2 Fundamental RNN architectures

$t-1$ and uses them as additional inputs to the network at time t . So the activated hidden units, context units, imply a time delay. On the other hand, Jordan network takes context units from the output layer for additional inputs. Each network can be represented as Eqs. (8) and (9).

$$\hat{y}_k = g(y_{k-1}, \dots, y_{k-p}; u_1(k-1), \dots, u_h(k-1); w) \quad (8)$$

$$\hat{y}_k = g(y_{k-1}, \dots, y_{k-p}; \hat{y}_{k-1}; w) \quad (9)$$

We apply the recurrent architecture of Elman network to BNN for the purpose of time series prediction as shown in Fig.3. This BRNN has one hidden layer, and hidden units go back to input layer as context units. The inputs are past observations $y_k, y_{k-1}, \dots, y_{k-p}$, and generate one estimation output for next time step. The bias input is added in input and hidden layer.

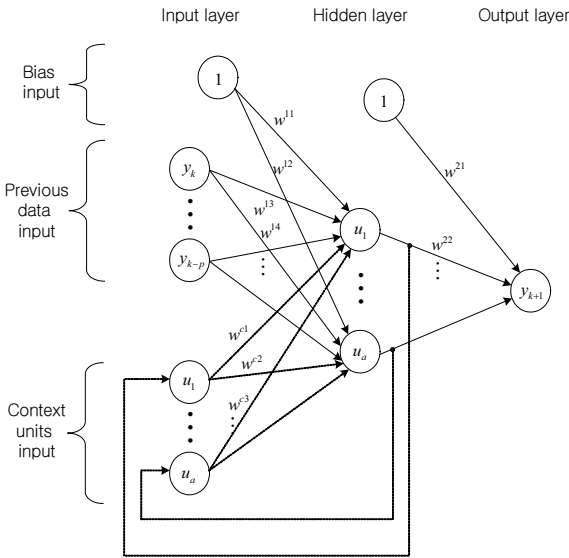


Fig. 3 Bayesian recurrent neural network for time series prediction

In the standpoint of learning method, Bayesian inference can be more effective than standard backpropagation because the statistical properties of weights can be traced. Furthermore, in the standpoint of network architecture, recurrent structure can be more effective than standard feedforward structure because the time delay factor is contained in it. Therefore, the Bayesian network with recurrent structure is expected to show better performance than traditional neural networks.

4. COMPUTER SIMULATION

We simulated the BNN and BRNN predictors with non-linear and non-stationary time series data. The target data y_k is artificially generated through Eq. (10), where $S(\cdot)$ is the one cycle of square wave function to add a non-stationary factor. The initial value is $y_0 = 1$, and sampling rate is $\Delta k = 1$. To compare the performance of BRNN to that of BNN, initial conditions are set to be same.

$$x_k = 0.5x_{k-1} + \frac{25x_{k-1}}{(1+x_{k-1}^2)} + 8 \cos(1.2(k-1)) \quad (10)$$

$$y_k = (x_k + 6S(0.05(k-1)) + 3) / 10$$

The estimation is one-step-ahead prediction, and the networks takes on-line learning. The estimation results with BNN and BRNN are as shown in Fig. 4 and Fig. 5.

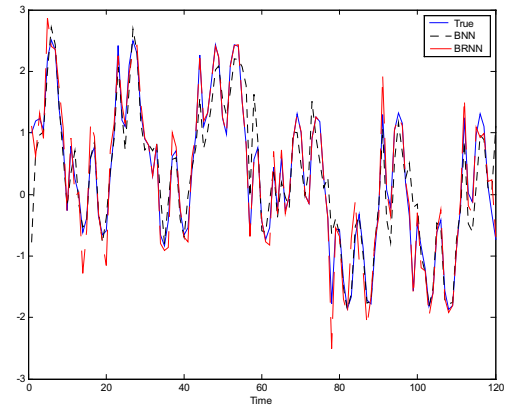


Fig. 4 Prediction performance of BNN and BRNN

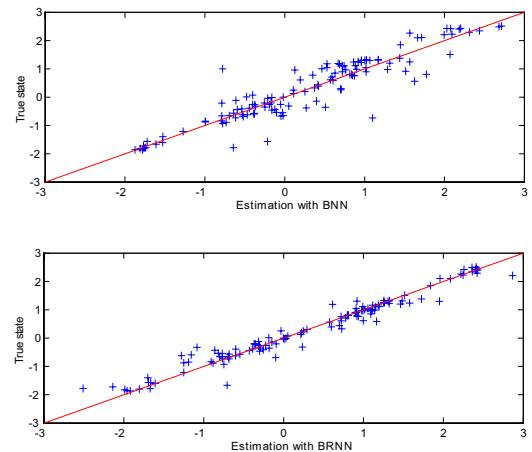


Fig. 5 True value versus estimates from BNN and BRNN

The estimate result indicates that BRNN has lower prediction error as depicted in Fig. 5. The mean square errors of BNN and BRNN are calculated to be 0.1907 and 0.0828, respectively. Therefore, the prediction estimate of BRNN

shows better performance than that of BNN. It means that the recurrent network architecture is also effective in case of Bayesian network model for time series prediction.

5. CONCLUSION

The general learning algorithm of backpropagation in neural network decides the weights as a single set of fixed value. But Bayesian learning algorithm offers the weights as a form of probability distribution, and it enables one to analyze the estimation in a statistical manner. In some cases, therefore, the Bayesian neural network can show the higher performance than other neural network. Especially, this paper expanded the range of BNN to recurrent network structure besides the basic feedforward structure in order to make a better prediction for time series data. Through the simulation with non-linear time series data, the BRNN proved its usefulness. We can conclude that the BRNN is one of the suitable neural network systems for time series analysis.

REFERENCES

- [1] T. D. Cimpanu, "Using Neural Networks For Time Series Prediction", *Proceedings of the 6th International Conference on Optimization of Electrical and Electronic Equipments*, Vol. 2, pp. 533-536, 1998.
- [2] P. Cristea, R. Tuduce, and A. Cristea, "Time series prediction with wavelet neural networks", *Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering*, pp. 5-10, 2000.
- [3] J. Neves and P. Cortez, "An artificial neural network -genetic based approach for time series forecasting", *IVth Brazilian Symposium on Neural Networks*, pp. 9-13, 1997.
- [4] D. J. C. Mackay, "A practical Bayesian framework for backpropagation networks", *Neural Computation*, Vol. 4, No. 3, pp. 448-472, 1992.
- [5] R. M. Neal, *Bayesian learning for neural network*, Lecture Notes in Statistics No. 118, Springer-Verlag, New York, 1996.
- [6] C. M. Bishop, *Neural network for pattern recognition*, Oxford University Press, 1995.
- [7] N. Bergman, *Recursive Bayesian estimation : navigation and tracking applications*, Linköping Univ., 1999.
- [8] T. Zhang and A. Fukushige, "Forecasting time series by Bayesian neural networks", *International Joint Conference on Neural Networks, Proceedings of the 2002*, Vol. 1, pp. 382-387, 2002.
- [9] E. H. Tito, G. Zaverucha, M. Vellasco and M. Pacheco, "Bayesian neural networks for electric load forecasting", *6th International Conference on Neural Information*, Vol. 1, pp. 407-411, 1999.
- [10] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*, Oxford University Press, 2001.