연속시간계 순환 뉴럴 네트워크를 이용한 공정 확인

이용준, 성수환, 박선원
한국과학기술원, 생명화학공학과, 초미세 화학공정 연구센터

System Identification using Continuous-time Recurrent Neural Network

Yong Joon Lee, Su Whan Sung, and Sunwon Park
Department of Chemical & Biomolecular Engineering, KAIST
The Center for Ultramicrochemical Process Systems.

1. Introduction

Artificial Neural Networks(ANNs) have been widely applied to many fields of science and engineering due to their ability to describe any nonlinear dynamics between process inputs and process outputs. It is notable that most previous neural network dynamic models are for the discrete-time processes. There are some reasons why discrete-time approaches are more popular than continuous-time approaches. First, the mathematics for discrete-time approaches is easier than that of continuous-time approaches. The stochastic theories of discrete-time approaches are much simpler than those of continuous-time approaches. Also, time derivatives of the process output and process input should be calculated in continuous-time approaches, which makes continuous-time approaches complicated. Second, if our main purposes are only to design controllers, it is not important to estimate physical parameters of continuous-time physical models. For these reasons, discrete-time approaches have been recognized more efficient than continuous-time approaches[2-4]. However, it has been proven that discrete-time approaches have serious disadvantages for several situations. First, when we estimate physical parameters, continuous-time approaches are definitely preferred. Second, when we deal with a process input of a physical signal such as temperature, pressure and level, continuous-time approaches should be used since it is a continuous-time signal rather than a discrete-time signal. Third, discrete-time approaches cannot escape numerical ill-conditions for a small sampling time and the modeling errors increase inversely proportional to a sampling time for a certain type of noise[2-4]. In this context, we propose a continuous-time recurrent neural network(CRNN) to overcome the disadvantages of the previous discrete-time neural networks. It is organized as follows. Section 2 presents the framework of CRNN and we derive the training rule of CRNN. Section 3 compares the performance of the proposed CRNN with that of previous neural networks from real data of micro-PCR reactor. Section 4 concludes this research.

2. Continuous-time Recurrent Neural Network

In this section, we propose a framework for continuous-time recurrent neural network model and derive the supervisory training rule of the proposed CRNN.

2.1 Framework of continuous-time recurrent neural network

The framework of SISO CRNN is shown in Figure 1. The input layer is composed of the state and the input and there is one hidden layer. The output layer consists of the first order time derivative of the state and the model output. The model state can be calculated from the first order derivatives of the state through the Euler integration method, which goes back to the input nodes for the next time calculation. Because of the above features, we call this model continuous-time recurrent neural network. The number of input nodes is equal to that of output nodes. If the dimension of the state is n, the number of the input nodes becomes $n+1$.

2.2 Training rule of continuous-time recurrent neural network

In this research, the backpropagation with the generalized delta rule is used to train the weights. The goal of the supervisory learning algorithm is to minimize the average squared error between the values of the measured process output and the model output as follows.

The objective function to be minimized for training CRNN is as follows.

$$E = 0.5\,(y(t) - y_m(t))^2 \tag{1}$$

subject to

$$z_p(t) = \sum_{q=1}^{n} V_{p,q} X_q(t) + V_{p,n+1}\, u(t) \tag{2}$$

$$h_p(t) = f(z_p(t)) \quad \text{where, } f(x) = \frac{1}{1+e^{-x}} \quad \text{(logistic sigmoid function)} \tag{3}$$

$$y_m(t) = \sum_{p=1}^{m} W_{n+1,p} h_p(t) \tag{4}$$

$$\frac{dX_q(t)}{dt} = \sum_{p=1}^{m} W_{q,p} h_p(t) \tag{5}$$

$$X_q(t+\Delta t) = X_q(t) + \sum_{p=1}^{m} W_{qp} h_p(t)\,\Delta t \tag{5-1}$$

where, $y(t)$, $y_m(t)$ and $u(t)$ denote the process output, the model output and the process input, respectively. $X \in R^n$ denotes the n-dimensional state. W and V represent the output weight matrix and the input weight matrix. n and m are the number of states and the number of hidden nodes. $h_p(t)$ is the output of p-th hidden node.

From (1)-(5), the following equations with respect to W can be derived.

$$\frac{\partial E}{\partial W_{kj}} = -\,(y(t) - y_m(t))\frac{\partial y_m(t)}{\partial W_{kj}} \tag{6}$$

$$\frac{\partial y_m(t)}{\partial W_{kj}} = h_j(t) \qquad \text{for} \quad k = n+1 \tag{7-1}$$

$$= \sum_{p=1}^{m} W_{n+1,p} \frac{\partial h_p(t)}{\partial W_{kj}} \qquad \text{for} \quad k = 1, 2, \cdots, n \tag{7-2}$$

The following three equations are for $k = 1, 2, \cdots, n$

$$\frac{\partial h_p(t)}{\partial W_{kj}} = h_p(t)\,(1 - h_p(t))\frac{\partial z_p(t)}{\partial W_{kj}} \tag{8}$$

$$\frac{\partial z_p(t)}{\partial W_{kj}} = \sum_{q=1}^{n} V_{pq} \frac{\partial X_q(t)}{\partial W_{kj}} \tag{9}$$

$$\frac{\partial X_q(t+\Delta t)}{\partial W_{kj}} = \frac{\partial X_q(t)}{\partial W_{kj}} + \left(h_j(t) + \sum_{p=1}^{m} W_{qp} \frac{\partial h_p(t)}{\partial W_{kj}} \right)\Delta t \qquad \text{for} \quad k = q \tag{10-1}$$

$$= \frac{\partial X_q(t)}{\partial W_{kj}} + \sum_{p=1}^{m} W_{qp} \frac{\partial h_p(t)}{\partial W_{kj}}\,\Delta t \qquad \text{for} \quad k \neq q \tag{10-2}$$

From the above equations, the weight W can be updated as follows.

$$W_{kj}^{\,t+1} = W_{kj}^{\,t} - \eta \frac{\partial E}{\partial W_{kj}}\bigg|_{W_{kj}^{\,t}} \tag{11}$$

Also, we can derive the following equations with respect to V.

$$\frac{\partial E}{\partial V_{ji}} = -\,(y(t) - y_m(t))\frac{\partial y_m(t)}{\partial V_{ji}} \tag{12}$$

$$\frac{\partial y_m(t)}{\partial V_{ji}} = \sum_{p=1}^{m} W_{n+1,p} \frac{\partial h_p(t)}{\partial V_{ji}} \tag{13}$$

$$\frac{\partial h_p(t)}{\partial V_{ji}} = h_p(t)(1-h_p(t)) \frac{\partial z_p(t)}{\partial V_{ji}} \tag{14}$$

The above three equations are for $k = 1, 2, \cdots, n+1$.

$$\frac{\partial z_p(t)}{\partial V_{ji}} = \sum_{q=1}^{n} V_{pq} \frac{\partial X_q(t)}{\partial V_{ji}} + X_i(t) \quad \text{for} \quad i = 1, 2, \cdots, n \quad \text{and} \quad p = j \tag{15-1}$$

$$= \sum_{q=1}^{n} V_{pq} \frac{\partial X_q(t)}{\partial V_{ji}} \quad \text{for} \quad i = 1, 2, \cdots, n \quad \text{and} \quad p \neq j \tag{15-2}$$

$$= \sum_{q=1}^{n} V_{pq} \frac{\partial X_q(t)}{\partial V_{ji}} + u(t) \quad \text{for} \quad i = n+1 \quad \text{and} \quad p = j \tag{15-3}$$

$$= \sum_{q=1}^{n} V_{pq} \frac{\partial X_q(t)}{\partial V_{ji}} + X_i(t) \quad \text{for} \quad i = n+1 \quad \text{and} \quad p \neq j \tag{15-4}$$

$$\frac{\partial X_q(t+\Delta t)}{\partial V_{ji}} = \frac{\partial X_q(t)}{\partial V_{ji}} + \sum_{p=1}^{m} W_{qp} \frac{\partial h_p(t)}{\partial V_{ji}} \Delta t \quad \text{for} \quad i = 1, 2, \cdots, n+1 \tag{16}$$

From the above equations, the weight V can be updated as follows.

$$V_{ji}^{t+1} = V_{ji}^{t} - \eta \frac{\partial E}{\partial V_{ji}}\Big|_{V_{ji}^{t}} \tag{17}$$

3. Experimental Study

In this paper, the micro-PCR (Polymerase Chain Reaction) reactor is considered. We identified the PCR process in order to compare the model performances of a previous discrete-time ANN and those of CRNN. The process input is the square root of voltage, and the process output is the temperature of the chamber in the micro-PCR reactor. Figure 2 shows the process input and the process output, where the sampling time is 0.055sec. Each model structure is shown in Figure 3. Figure 4 and Table 1 show the model performances for each approach. The solid line and the dot line in Figure 4 denote the process output and the model output respectively. The performance of the discrete-time NN is very poor, while that of CRNN is very excellent.

Table 1. Conditions and result of simulation

	Input node	Hidden node	Output node	η	Average squared error
Discrete-time NN	4	10	1	0.0001	0.00330
CRNN	3	5	3	0.0001	0.00009

4. Conclusion

We have developed the continuous-time recurrent neural network (CRNN) which is able to identify efficiently continuous-time processes. The training rule for CRNN is derived, which uses the backprogagation method with the generalized delta rule to update the weights. In order to demonstrate the performance of CRNN and compare CRNN with discrete-time NN, we carried out experiments with a micro-PCR reactor. The performance of CRNN is much better than that of the discrete-time NN when the sampling time is small.

Acknowledgment

This work was supported by the BK 21 Project and Center for Ultramicrochemical Process Systems sponsored by KOSEF.

References

(1) Bart Kosko, "*Neural networks and fuzzy systems*," Prentice-Hall, Englewood Cliffs, NJ, 1992.
(2) Sung, S. W., Lee, S. T., Kwak, H. J. and Lee, I. "*Continuous-time subspace system identification method*," Ind. Eng. Chem. Res. 40, 2886, 2001
(3) Middlen, R. H. and Goodwin, G. C., "*Digital control and estimation*," Prentice-Hall, Englewood Cliffs, NJ, 1990.
(4) Sung, S. W. and Lee, I., "*Prediction error identification method for continuous-time processes with time delay*," in press, Ind Eng. Chem. Res. 2001
(5) Kang, M. G., "*Development of hybrid model and optimization of operating conditions for a styrene monomer reactor*," M.S. Thesis, KAIST, Teajon, Korea

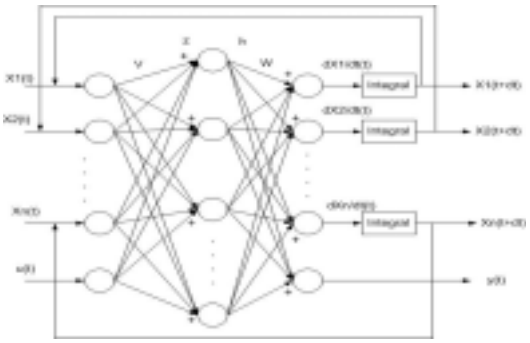

Figure 1. Framework of CRNN

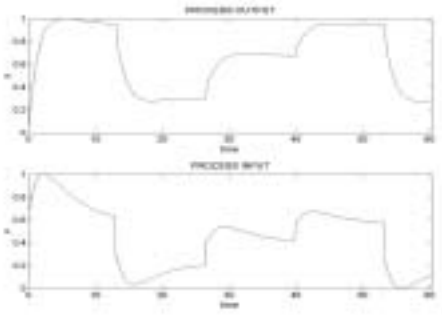

Figure 2. Output and input data for training

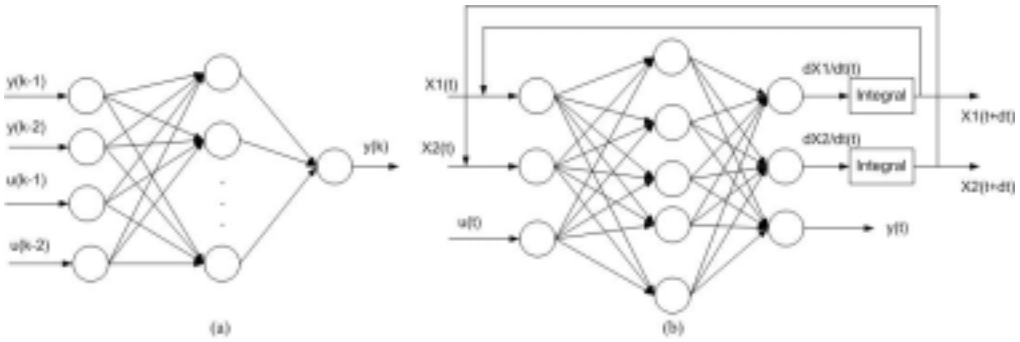

Figure 3. Frameworks: (a) discrete-time NN, (b) CRNN

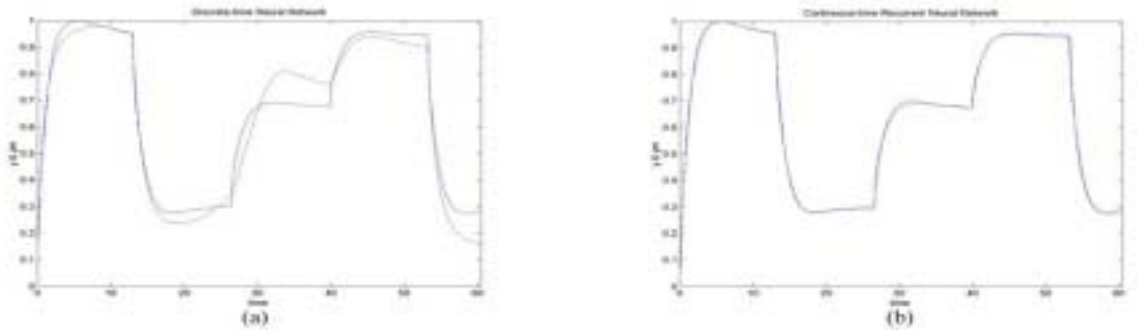

Figure 4. Model performances: (a) discrete-time NN, (b) CRNN