Lecture Series on

Intelligent Control

Lecture 15

Diagonally Recurrent Neural Networks

Kwang Y. Lee
Professor of Electrical & Computer Engineering
Baylor University
Waco, TX 76706, USA
Kwang_Y_Lee@baylor.edu

1

Diagonally Recurrent Neural Networks

I. Introduction

II. DRNN-Based Control System

III. Convergence and Stability of the Closed-loop System

IV. Simulation Results

V. Conclusion

Ku, C. C. and K. Y. Lee, "Diagonal Recurrent Neural Network for Dynamic Systems Control," IEEE Transactions on Neural Networks, Vol. 6, pp. 144-156, January 1995.

2

2

Artificial Neural Network Paradigms

1. Feedforward Neural Network (FNN)
   . Static mapping
   . Can not represent a dynamic response w/o tapped delays

2. Fully Connected Recurrent Neural Network (FRNN)
   . Can naturally represent dynamic systems
   . Difficult to train and to converge in a short time

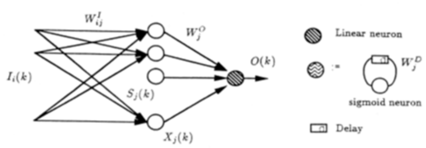
3. Diagonal Recurrent Neural Network (DRNN)
   . Fewer weights and shorter training time
   . Can be implemented easily for real-time control

3

3

Feedforward Neural Network (FFNN)

1. Feedforward Neural Network (FFNN)

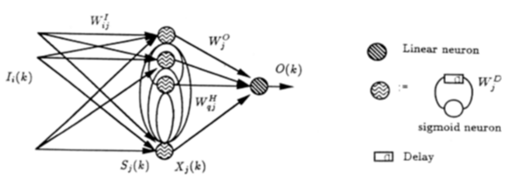


* Static mapping
* Combine with tapped delays to perform dynamic mapping

4

4

Fully-Connected Recurrent Neural Network (FRNN)

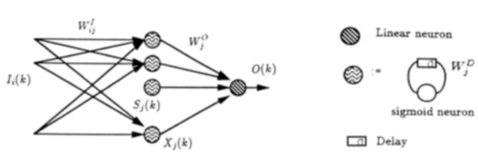


* Dynamic mapping
* Convergence problem
* Stability problem

5

5

Diagonal Recurrent Neural Network (DRNN)

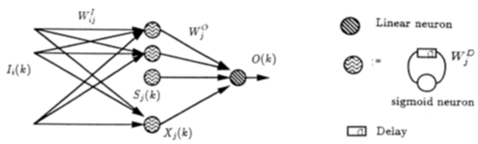


* Dynamic mapping
* Requires fewer weights
* Convergence is enhanced

6

6

Dynamic Representation of DRNN



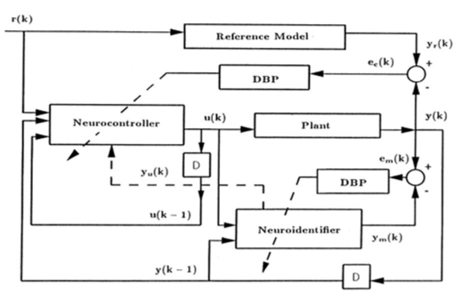
$$O(k) = \sum_j W_j^O X_j(k), \quad X_j(k) = f(S_j(k)), \qquad (1)$$

$$S_j(k) = W_j^D X_j(k-1) + \sum_i W_{ij}^I I_i(k), \qquad (2)$$

7

7

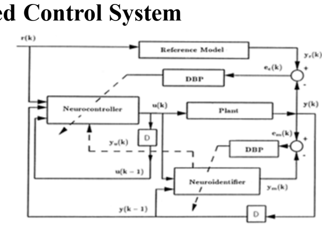
DRNN-based Control System



8

8

DRNN-based Control System



Let $y_r(k)$ and $y(k)$ be the desired and actual responses of the plant, respectively, then an error function for DRNC can be defined as

$$E_c = \frac{1}{2}(y_r(k) - y(k))^2. \qquad (3)$$

The error function (3) is also modified for the DRNI by replacing $y_r(k)$ and $y(k)$ with $y(k)$ and $y_m(k)$, respectively, where $y_m(k)$ is the output of the DRNI, i.e.,

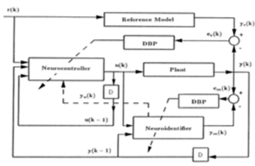
$$E_m = \frac{1}{2}(y(k) - y_m(k))^2, \qquad (4)$$

9

9

DRNN-based Control System

$$E_c = \frac{1}{2}(y_r(k) - y(k))^2. \quad (3)$$



The gradient of error in (3) with respect to an arbitrary weight vector $W \in \Re^n$ is represented by

$$\frac{\partial E_c}{\partial W} = -e_c(k)\frac{\partial y(k)}{\partial W} = -e_c(k)y_u(k)\frac{\partial O(k)}{\partial W}, \quad (5)$$

where $e_c(k) = y_r(k) - y(k)$ is the error between the desired and output responses of the plant, and the factor $y_u(k) \equiv \frac{\partial y(k)}{\partial u(k)}$ represents the *sensitivity* of the plant with respect to its input. Since the plant is normally unknown, the sensitivity needs to be estimated for the DRNC. However, in the case of the DRNI, the gradient of error in (4) simply becomes

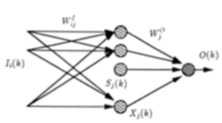
$$\frac{\partial E_m}{\partial W} = -e_m(k)\frac{\partial y_m(k)}{\partial W} = -e_m(k)\frac{\partial O(k)}{\partial W}, \quad (6)$$

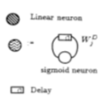
$$E_m = \frac{1}{2}(y(k) - y_m(k))^2, \quad (4)$$

10

10

DRNN-based Control System





$$O(k) = \sum_j W_j^O X_j(k), \quad X_j(k) = f(S_j(k)),$$
$$S_j(k) = W_j^D X_j(k-1) + \sum_i W_{ij}^I I_i(k),$$

$$\frac{\partial E_c}{\partial W} = -e_c(k)\frac{\partial y(k)}{\partial W} = -e_c(k)y_u(k)\frac{\partial O(k)}{\partial W}, \quad (5)$$

$$\frac{\partial E_m}{\partial W} = -e_m(k)\frac{\partial y_m(k)}{\partial W} = -e_m(k)\frac{\partial O(k)}{\partial W}, \quad (6)$$

The output gradient $\frac{\partial O(k)}{\partial W}$ is common in (5) and (6), and needs to be computed for both DRNC and DRNI. The gradient with respect to output, recurrent, and input weights, respectively, are computed using the following equations :

$$\frac{\partial O(k)}{\partial W_j^O} = X_j(k) \quad (7a)$$
$$\frac{\partial O(k)}{\partial W_j^D} = W_j^O P_j(k) \quad (7b)$$
$$\frac{\partial O(k)}{\partial W_{ij}^I} = W_j^O Q_{ij}(k), \quad (7c)$$

where $P_j(k) \equiv \frac{\partial X_j(k)}{\partial W_j^D}$ and $Q_{ij} \equiv \frac{\partial X_j(k)}{\partial W_{ij}^I}$, and satisfy

$$P_j(k) = f'(S_j)\Big(X_j(k-1) + W_j^D P_j(k-1)\Big), \quad (8a)$$

$$Q_{ij}(k) = f'(S_j)\Big(I_i(k) + W_j^D Q_{ij}(k-1)\Big), \quad (8b)$$

11

DRNN-based Control System

B. Dynamic backpropagation for DRNI

From (6), the negative gradient of the error with respect to a weight vector in $\Re^n$ is

$$-\frac{\partial E_m}{\partial W} = e_m(k)\frac{\partial O(k)}{\partial W}, \quad (9)$$

where the output gradient is given by (7) and (8), and W represents W^O, W^D, or W^I in $\Re^{n_o}$, $\Re^{n_d}$, or $\Re^{n_i}$, respectively.

The weights can now be adjusted following any gradient method such as the steepest descent method, i.e., the update rule of the weights becomes

$$W(n+1) = W(n) + \eta(-\frac{\partial E_m}{\partial W}) + \alpha\Delta W(n), \quad (10)$$

where η is a learning rate, α is a momentum factor, and $\Delta W(n)$ represents the change in weight in the n^{th} iteration.

12

12

DRNN-based Control System

C. Dynamic backpropagation for DRNC

In the case of DRNC, from (5), the negative gradient of the error with respect to a weight vector in $\Re^n$ is

$$-\frac{\partial E_c}{\partial W} = e_c(k) y_u(k) \frac{\partial O(k)}{\partial W}. \quad (11)$$

Since the plant is normally unknown, the sensitivity term $y_u(k)$ is unknown. This unknown value can be identified by using the DRNI. When the DRNI is trained, the dynamic behavior of the DRNI is close to the unknown plant, i.e., $y(k) \approx y_m(k)$, where $y_m(k)$ is the output of the DRNI.

Therefore, the sensitivity was approximated in [2] and shown to be

$$y_u(k) \equiv \frac{\partial y(k)}{\partial u(k)} \approx \frac{\partial y_m(k)}{\partial u(k)} = \sum_j W_j^O f'(S_j(k)) W_{1j}^I, \quad (12)$$

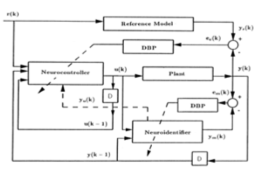
where the variables and weights are those found in DRNI.

13

13

Example 1 – BIBO Nonlinear Plant

Example 1: A BIBO nonlinear plant



Reference Model:

$$y_r(k+1) = 0.6 y_r(k) + r(k)$$

$$r(k) = sin\left(\frac{2\pi k}{25}\right) + sin\left(\frac{2\pi k}{10}\right)$$

Plant Model:

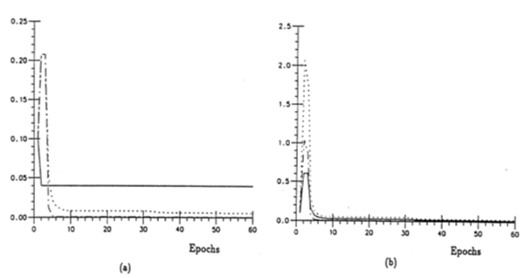
$$y(k+1) = \frac{y(k)}{1.0 + y^2(k)} + u^3(k)$$

In this example, $Z_c = \{r(k), u(k-1), y(k-1)\}$ and $Z_I = \{u(k), y(k-1)\}$, thus $n_c = 3$ and $n_i = 2$. Also, $N_T = 14$ and $W_T = 67$. $\eta_c = 0.1$ and $\eta_I = 0.1$.

14

14

Example 1 – BIBO Nonlinear Plant



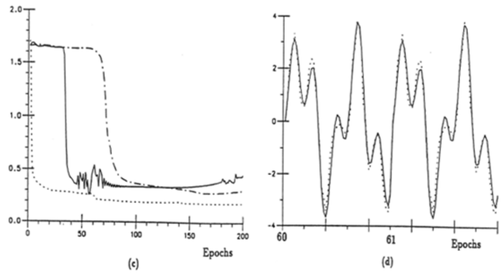
DRNI

DRNC

15

15

Example 1 – BIBO Nonlinear Plant

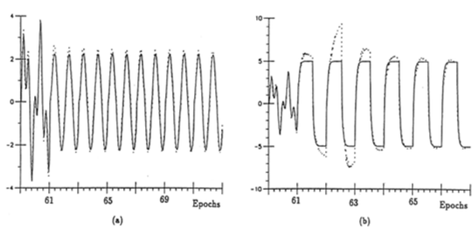


16

16

Example 1 – BIBO Nonlinear Plant

The on-line adapting ability of DRNN based control



17

17

Example 2 – Model Reference Control

Reference model:

$$y_r(k+1) = 0.6y_r(k) + r(k)$$

Desired reference:

$$r(k) = 0.5sin(\frac{2\pi k}{50.0}) + 0.5sin(\frac{2\pi k}{100.0})$$

Plant :

$$y(k+1) = 0.2y^2(k) + 0.2y(k-1) + 0.4sin[0.5(y(k)+ y(k-1))]\cdot cos[0.5(y(k)+y(k-1))] + 1.2u(k)$$

18

18

Example 2 – Model Reference Control

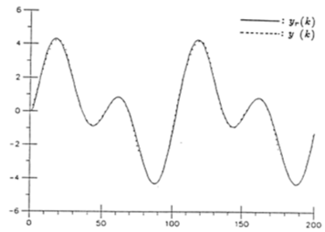


Fig. 4 (a) Outputs of reference model and plant

19

19

Example 2 – Model Reference Control

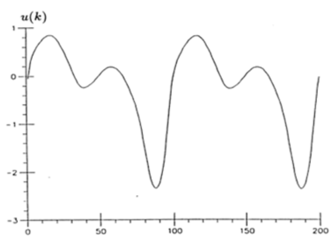


Fig. 4 (b) Control signal generated from DRNC

20

20

Example 3 – Flight Control

Example 3: Flight Control

Reference model:

$$H(s) = \frac{4.0}{s^2 + 2.82s + 4.0}$$

Plant:

$$H(s) = \frac{1.0}{s^2 + 2.0s + 1.0}.$$

training sets $(x_1(0), x_2(0)) =$ (0.0, 0.0), (0.1, 0.3), and (0.5, 0.75)

testing set $(x_1(0), x_2(0)) = (0.8, 1.0)$

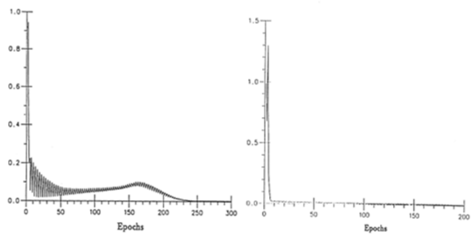
The step input is applied to the reference model, and $r(t) = 1$

$N_T = 16$ and $W_T = 84$. η_c and η_I , 20 , and both biases, b_c and b_I, 1.0.

21

21

Example 3 – Flight Control



Fixed learning rates   Adaptive learning rates

22

22

Example 3 – Flight Control

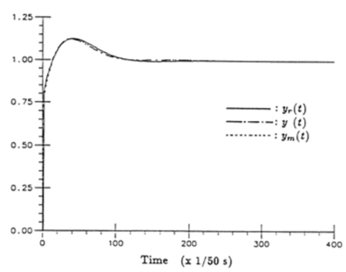


Fig. 3 (b) Outputs of reference model, plant, and DRNI

23

23

Example 4 – Controlled Van der Pol Dynamics

Example 2: A Controlled Van der Pol equation

The objective of this example is to investigate the ability of DRNN based control system in controlling a nonlinear plant with a Van der Pol dynamics.

Plant model:

$$\ddot{x}(t) - \mu(1 - x^2(t))\dot{x}(t) + x(t) = u(t)$$

or

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -x_1(t) + \mu(1 - x_1^2(t))x_2(t) + u(t)$$

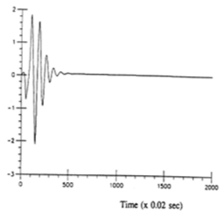
and

$$y(t) = x_1(t) + x_2(t).$$

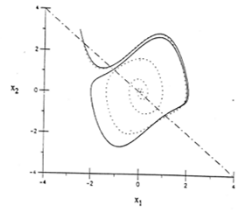
24

24

Example 4 – Controlled Van der Pol Dynamics



The control signal generated from DRNC



The plots of (x_1, x_2); u(t)=0 (solid line), controlled tracking output (dotted line), and the line $x_1 + x_2 = 0$ (dotted-dashed line)

25

25

Conclusion

1. A new neural network paradigm of DRNN is developed as a minimal realization of the fully recurrent neural network.
2. The proposed paradigm has the desired features of simplicity and recurrence. This makes the convergence and stability possible.
3. Adaptive learning algorithm is developed for on-line approach.
4. Convergence theorems are developed which not only guarantees the error to converge to an arbitrary small value, but also guarantees the closed-loop stability of the BIBO stable system.

26

26