34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Nonlinear Model for Dynamic Synapse Neural Network

Hyung O. Park, Alireza A. Dibazar, and Theodore W. Berger, *Fellow, IEEE*

***Abstract*— This paper presents a simplified nonlinear model for Dynamic Synapse Neural Network (DSNN) which is based on nonlinear dynamics of neurons in the hippocampus, using a recurrent neural network. The proposed model will be utilized in place of DSNN for various applications which require simpler implementation and faster training, maintaining the same performance as a nonlinear system model, classifier, or pattern recognizer. This model was tested in two different structure and training methods, by learning the input-output relationship of a few DSNNs with sets of experimentally-determined coefficients. The results showed that this model can capture DSNN's complicated nonlinear dynamics in a temporal domain with less computational cost and faster training.**

I. Introduction

Biologically inspired technologies prevail based on their ability to mimic the biological functions of nature. Specifically, biologically-inspired computational models, in particular those that mimic the human brain, have been widely used in signal processing to take advantage of the capability of real neurons. The capability comes from a nonlinear transformation between input and output sequence of temporal patterns, forming neurons' memory through nonlinear molecular mechanisms connecting them.

The Dynamic Synapse Neural Network (DSNN) by Berger and Liaw is based on experimentally-determined nonlinear dynamics of neurons in the hippocampus, the brain region responsible for forming pattern recognition memories [1]. In this study, processing elements are assumed to transmit information by variation in a series of temporal patterns, and connections between processing elements are modeled as a set of dynamic processes with different time courses of decay derived from experimental studies. These multiple time courses determine the composite dynamics of each synaptic connection, and as a result, synaptic output becomes a function of the time since past input events. Thus, each network connection transforms a sequence of input events into a different sequence of output events.

Although several studies applied the DSNN to the recognition of speech and specific type of vibration, which have presented encouraging results, these methods require lossy signal-to-spike transformations and complicated processes to provide preliminary information for the classification process of the DSNN [2-6]. In order to deal with practical applications which require low computational cost and simple signal transformation, a Discrete Synapse Recurrent Neural Network (DSRNN) was implemented and tested as a classifier in the task of footstep and vehicle recognition [7, 8]. A DSRNN represents the relationship between input and output as a nonlinear function of past input and output history in a nonparametric method without the complicated nonlinear modeling of synaptic transmission. Therefore, this model can be utilized at more general applications which require a nonlinear modeling of input-output relationship.

In this paper, we describe how a DSRNN is trained as a nonlinear system model to learn the nonlinear input-output relationship of a DSNN, to not only show that it can sufficiently replicate a DSNN's function as a simplified model for DSNN, but also show its capability as a general nonlinear model.

II. Design of DSRNN

In the DSNN, four important synaptic mechanisms namely calcium response, facilitation I, facilitation II, and inhibition have been modeled with differential equations. For signal processing applications, the time and weight scales of the differential equations are adapted for particular tasks. This can be considered as a temporal processing of the input. The process of adaptation (or learning) is involved with finding the relationship between the current and the past history of input signal.

For a discrete algorithm, 4 difference equations for the DSNN's presynapse are derived instead of differential equations using the impulse-invariant transformation [2] and described in (1), (2), (3), and (4). Each equation represents calcium response, facilitation I, facilitation II per input action potential (A_P), and modulation per inhibitory action potential (A_{Pinh}) i.e. feedback respectively. Equation (5) describes the output of presynapse as the overall model of the presynapse components.

$$R(n+1) = e^{-h/\tau_R} R(n) + K_R(1-e^{-h/\tau_R})A_p(n) \tag{1}$$

$$F_1(n+1) = e^{-h/\tau_{F_1}} F_1(n) + K_{F_1}(1-e^{-h/\tau_{F_1}})A_p(n) \tag{2}$$

$$F_2(n+1) = e^{-h/\tau_{F_2}} F_2(n) + K_{F_2}(1-e^{-h/\tau_{F_2}})A_p(n) \tag{3}$$

H. O. Park is with the Laboratory for Neural Dynamics, University of Southern California, Los Angeles, CA 90089 USA (corresponding author to provide phone: 213-740-9359; e-mail: hyungpar@usc.edu).

A. A. Dibazar is with the Laboratory for Neural Dynamics, University of Southern California, Los Angeles, CA 90089 USA. (e-mail: dibazar@usc.edu).

T. W. Berger is with the Center for Neural Engineering, University of Southern California, Los Angeles, CA 90089 USA, (e-mail: berger@bmsr.usc.edu).

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

$$Mod(n+1) = e^{-h/\tau_{Mod}} Mod(n) + K_{Mod}(1 - e^{-h/\tau_{Mod}}) A_{P_{inh}}(n) \tag{4}$$

$$P_R(n) = R(n) + F_1(n) + F_2(n) + Mod(n) \tag{5}$$

From the difference equations, the input-output relationship of the presynapse can be re-organized in terms of the past values of input (A_P, A_{Pinh}), output (P_R), and error (E_m) as following.

$$P_R(n+1) = \sum_{k=0}^{m} \alpha_{k+1} A_p(n-k) + \sum_{k=0}^{m} \beta_{k+1} A_{P_{inh}}(n-k) + E_m \tag{6}$$

where new variables are defined as below.

$$\alpha_k = e^{-(k-1)h/\tau_R} K_R^E + e^{-(k-1)h/\tau_{F_1}} K_{F_1}^E + e^{-(k-1)h/\tau_{F_2}} K_{F_2}^E \tag{7}$$

$$\beta_k = e^{-(k-1)h/\tau_{Mod}} K_{Mod}^E \tag{8}$$

$$\begin{aligned} E_m = {} & e^{-(m+1)h/\tau_R} R(n-m) + e^{-(m+1)h/\tau_{F_1}} F_1(n-m) + \\ & e^{-(m+1)h/\tau_{F_2}} F_2(n-m) + e^{-(m+1)h/\tau_{Mod}} Mod(n-m) \end{aligned} \tag{9}$$

$$K_X^E = K_X(1 - e^{-h/\tau_X}) \tag{10}$$

As Equation (6) shows, the present value of output can be represented by the weighted sum of past values of input and feedback from output. This is another interpretation of the temporal signal processing of DSNN's presynapse and is considered as the discrete synapses, the input layer of DSRNN with recurrent connections. The other part of DSNN modeling is regarding the nonlinearity from synaptic transmission mechanism including thresholding for release, and quantal release, refractory period by depletion, and thresholding for generation of excitatory post-synaptic potentials (EPSPs) in the post-synapse [1]. Adding these nonlinear activities to the new model, as a result, its final structure ends up with a lumped nonlinear function of the past values of input and the feedback from the output plus an estimation error as in Equation (11). The superscript *t*-1 means the set of all past values of itself. This nonlinear function *f* is expected to be capable enough to incorporate spike transformation and a feature extractor as a lumped model.

$$y(t) = f(x^{t-1}, y^{t-1}) + e(t) \tag{11}$$

The nonlinear function *f* is replaced with general neural networks here and trained accordingly. Figure 1 and Equation (12-14) shows one of DSRNNs with the structure of a recurrent neural network.

$$y_k(t) = a\left(\sum_{i=1}^{n_R} w_{ki}^{(o)} R_i(t) + \sum_{j=1}^{n_x} w_{kj}^{(o)} x_j^{\,t-1} \right) \tag{12}$$

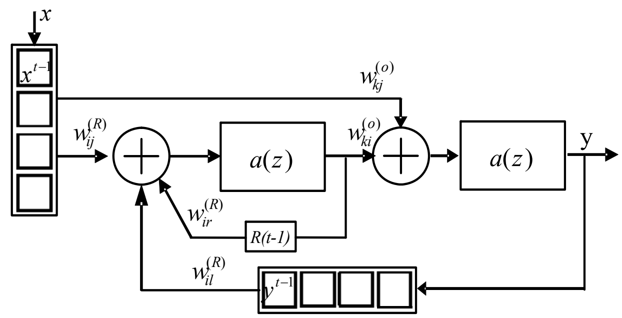

Figure 1. A DSRNN with recurrent network; it consists of a recurrent-structured neural network with recurrent components and a hidden layer, and feedback loop. The final output and hidden nodes' output will be fed through feedback loop to the input of the network. The superscript *t-1* on the input and feedback means the set of all past values of them.

where $R(t)$ is the post synaptic potential at a hidden layer which has recurrent connections to itself with a delay of one time step, and $a(z)$, logistic sigmoid function is used as below.

$$R_i(t) = a\left(\sum_{j=1}^{n_x} w_{ij}^{(R)} x_j(t) + \sum_{l=1}^{n_y} w_{il}^{(R)} y_l^{\,t-1} + \sum_{r=1}^{n_R} w_{ir}^{(R)} R_r(t-1) \right) \tag{13}$$

$$a(z) = \frac{1}{1+\exp(-z)} \tag{14}$$

In the DSRNN with feedforward network, all recurrent connections around the hidden layer, $w_{ir}^{(R)}$ are zero. However, it still has recurrent connections from the output as a feeback term by the definition of DSRNN.

III. EXPERIMENT: MODELING DSNNS

For a validation of the DSRNN which was defined from a mathematical derivation of the DSNN, multiple DSRNNs were trained to model various DSNNs to see if the DSRNNs can capture DSNN's complicated nonlinear dynamics in temporal signal processing.

First, the DSRNN with feedforward neural network is trained using LM algorithm to mimic single DSNN's output as a nonlinear model. Although the DSRNN is a simplified version of DSNN, it is totally different from DSNN since input and output of DSRNN are analog whereas the DSNN's are spikes. However, it should be able to learn somehow DSNN's characteristic embedded with the capability for the nonlinear modeling of temporal patterns. For fair comparison, DSNN is set up to generate an analog membrane potential rather than the spike on the output. Also, a uniformly distributed random signal is chosen as an input signal on the presynapse instead of spikes. From the typical set of parameters of DSNN, the input and output from DSNN were prepared to be trained. The sampling frequency was 2 kHz and the length of data was 10000 samples, which was 5 seconds. The length of the delay of input and feedback were 50 and 5 respectively. The number of nodes in the hidden layer was 8. After training, the Normalized Mean Square Error (NMSE) between the original output from DSNN and

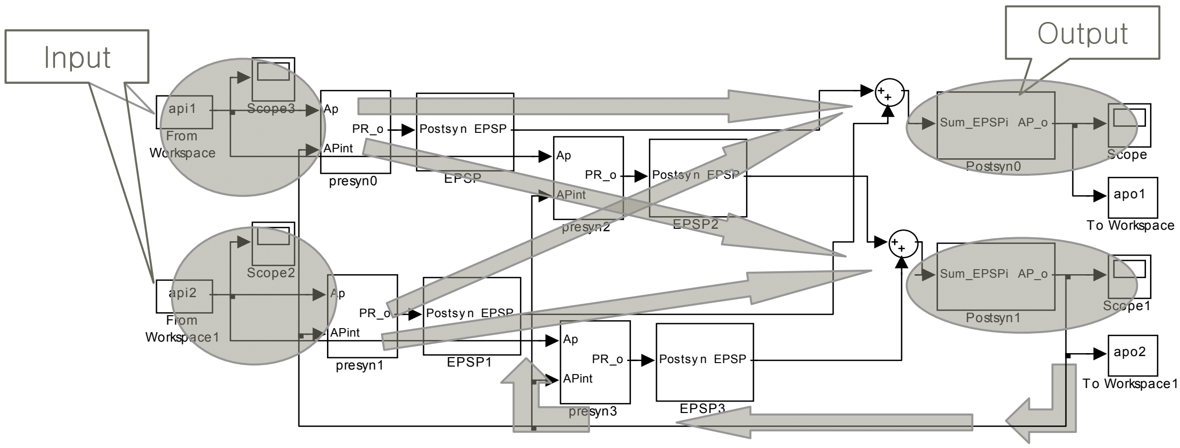

Figure 2. Block diagram of 2x2 DSNN; gray areas represent two inputs, one output, four pathways for each synaptic connection, and inhibitory stimulus from another output. The equation and typical values are from Berger and Liaw [1].

the output from the trained DSRNN was measured for a quantitative measure of its modeling error. Also, NMSE was measured again on the same DSRNN but for the different input than the one during training. The same training-test process was repeated for the different single DSNNs which have different sets of parameters from each other and the results were presented in Table I and Figure 3.

Another DSRNN with recurrent network was trained using Extended Kalman Filter algorithm in a similar way. Since it has additional recurrent connections, it is expected to have more capability for nonlinearity and dynamicity in system modeling. A 2x2 DSNN shown in Figure 2 was chosen to be modeled. For this, the two inputs of DSNN were set identical. One output was chosen for modeling and the other output was used for feedback, that is, stimulating inhibitory pathway. From the typical set of parameters of DSNN, the input and the chosen output from DSNN were prepared to be trained. The sampling frequency was 2 kHz and the length of data was 50000 samples, which was 25 seconds. The length of the delay of input and feedback were 50 and 10 respectively. The number of nodes in the hidden layer was 10. After training and testing, NMSEs were measured and presented in Table II and their test samples were shown in Figure 4.

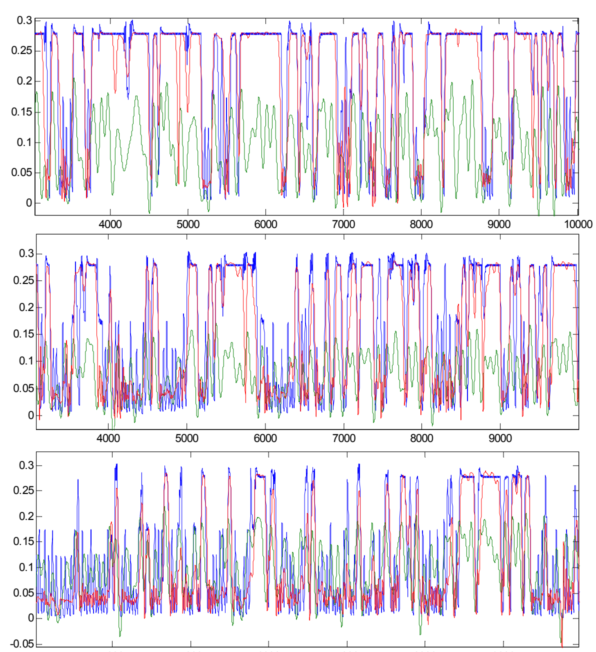

Figure 3. Test samples from the task for learning 3 different single DSNNs (A, B, C in order of top to bottom); Green line: input signal, blue line: output signal of DSNN, red line: reproduced output signal from trained DSRNN, X-axis represents time in the unit of sample (0.5ms per sample).

TABLE I. Normalized mean square error for the output signal of three different trained DSRNNs with feedforward network

DSRNN	A	B	C
Weights of DSNN's presynapse	[30, 0.16, 50, 20]	[10, 0.16, 80, 20]	[10, 0.16, 50, 20]
NMSE(%) with trained input	2.1	12.5	15.1
NMSE(%) with test input	3.2	11.7	16.3

TABLE II. Normalized mean square error for the output signal of three different trained DSRNNs with recurrent network

DSRNN	A	B	C
Weights of DSNN's presynapse	[10 0.16 50 20; 15 0.3 30 10; 5 0.05 70 15; 20 0.3 10 30]	[13 0.2 40 15; 19 0.3 30 11; 21 0.02 53 10; 15 0.4 9 22]	[18 0.1 30 15; 22 0.2 15 13; 15 0.09 43 20; 5 0.2 3 32]
NMSE(%) with trained input	2.47	1.67	1.28
NMSE(%) with test input	2.60	1.77	1.41

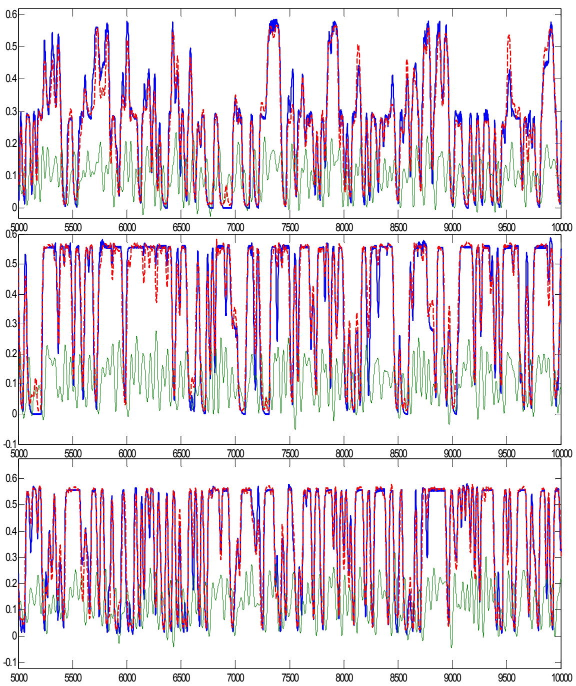

Figure 4. Test samples from the task for learning 3 different 2x2 DSNNs (A, B, C in order of top to bottom); Green line: input signal, blue line: output signal of DSNN, red line: reproduced output signal from trained DSRNN, x-axis represents time in the unit of sample (0.5ms per sample).

IV. Results and Conclusions

From the first experiment, all of test showed that the DSRNN successfully learned each single DSNN, being capable of capturing temporal dynamics of each DSNN. Most of error was generated from the subthreshold region since the DSRNN which has analog signals was not able to follow the fluctuation of subthreshold membrane potential. By the same reason, NMSE gets bigger exposing more subthreshold region as the DSNN becomes less excitatory, that is, the weights of DSNN get lower. Considering the subthreshold region is less important in terms of transfer of information, this result is not necessarily an issue. From the view of lumped modeling, this experiment showed the capability for modeling DSNNs to a certain degree. Even when it comes to the network of multiple DSNNs, actual nonlinear function still remains same per a single output only with additional input nodes. Also, including spike transformation to the DSRNN is actually unnecessary or it was already included to the DSRNN built since the original signal was analog signal. Unless the original signal is spike, there is no reason to use spike transformation since the signal-to-spike conversion is a lossy transformation in terms of information.

From the second test with the DSRNN with recurrent network, each DSRNN model shows the better capability to capture the nonlinear temporal dynamics of DSNN with less error than from the DSRNN with feedforward network, even though the 2x2 DSNN has much more nonlinearity to learn compared to a single DSNN. A 2x2 DSNN’s output is the nonlinear sum of two different synaptic transmission mechanism, whereas a single DSNN’s output can be considered as the nonlinear synaptic transmission function of input. Also, a 2x2 DSNN has four times the number of weights and 4 different inhibitory pathways stimulated by another output which was not trained directly, whereas a single DSNN has a single inhibitory pathway simulated by a trained output. Considering these facts, it is obvious that the DSRNN with recurrent network outperforms the DSRNN with feedforward network, incorporating richer nonlinear modeling ability.

From the view of a general nonlinear model, the choice of DSRNN's nonlinear function has a significant impact on the performance as expected. Efficient recurrent connections inside the network should supply enough temporal dynamics and memory depth to store the past data, without increasing computational complexity significantly.

Overall, a DSRNN proved its capability as a nonlinear system model and its inheritance of DSNN's functionality. In addition, there is a significant benefit on its implementation, considering a DSNN classifier's training time using Genetic Algorithm from other application is more than ten times of DSRNN's using Extended Kalman Filter algorithm.

Acknowledgment

This research was supported by the awards from ONR (award no: N00014-10-1-0685) and the Navy through a CPP grant (award no: N00014-09-C-0209).

References

[1] J. S. Liaw and T. W. Berger, "Dynamic synapse: A new concept of neural representation and computation," *Hippocampus*, vol. 6, pp. 591-600, 1996.

[2] H. H. Namarvar, L. Jim-Shih, and T. W. Berger, "A new dynamic synapse neural network for speech recognition," presented at Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on, 2001.

[3] H. Heidari Namarvar, A. A. Dibazar, and T. W. Berger, "The Gauss-Newton learning method for a generalized dynamic synapse neural network," presented at Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, 2002.

[4] A. A. Dibazar, H. H. Narnarvar, and T. W. Berger, "A new approach for isolated word recognition using dynamic synapse neural networks," presented at Neural Networks, 2003. Proceedings of the International Joint Conference on, 2003.

[5] A. A. Dibazar, H. O. Park, and T. W. Berger, "The Application of Dynamic Synapse Neural Networks on Footstep and Vehicle Recognition," presented at Neural Networks, 2007. IJCNN 2007. International Joint Conference on, 2007.

[6] S. George, A. Dibazar, J. S. Liaw, and T. W. Berger, "Speaker recognition using dynamic synapse based neural networks with wavelet preprocessing," presented at Neural Networks, 2001. Proceedings of the 2001 International Joint Conference on, 2001.

[7] H. O. Park, A. A. Dibazar, and T. W. Berger, "Protecting Military Perimeters from Approaching Human and Vehicle Using Biologically Realistic Dynamic Synapse Neural Network," presented at Technologies for Homeland Security, 2008 IEEE Conference on, 2008.

[8] H. O. Park, A. A. Dibazar, and T. W. Berger, "Discrete Synapse Recurrent Neural Network for Nonlinear System Modeling and Its Application on Seismic Signal Classification," presented at Neural Networks, 2010. Proceedings of the 2010 International Joint Conference on, 2010.