Letters

Random Neural Networks With State-Dependent Firing Neurons

Sungho Jo, Jijun Yin, and Zhi-Hong Mao

***Abstract*—This letter studies the properties of the random neural networks (RNNs) with state-dependent firing neurons. It is assumed that the times between successive signal emissions of a neuron are dependent on the neuron potential. Under certain conditions, the networks keep the simple product form of stationary solutions and exhibit enhanced capacity of adjusting the probability distribution of the neuron states. It is demonstrated that desired associative memory states can be stored in the networks.**

***Index Terms*—Associative memory, random neural networks (RNNs), spiking neurons, state-dependent firing rate.**

I. Introduction

Networks of spiking neurons have been proposed as alternatives to the classical perceptron networks [14]. A spiking neuron computes by transforming dynamical input into a train of spikes. This represents closely the manner that signals are encoded and transmitted in real neuronal networks: neural signals travel as action potentials (voltage spikes) rather than fixed analog levels. The spiking neural networks are therefore more biologically plausible than the perceptron-type models [8]. Though carrying less information than real-valued signals, binary-valued spiking signals benefit from the higher rate at which information may be reliably sent. It has been indicated that the spiking signals are more efficient for information transfer than the high-resolution, real-valued analog signals [15]. Furthermore, several types of spiking neural networks have proved to be more computationally powerful than the perceptron networks [12], [13].

This letter considers a specific model of spiking neural networks—the random neural network (RNN) model, which was proposed by Gelenbe more than a decade ago [2]. Compared with the models that aim at describing accurately cellular dynamics of neuron firing and may, thus, be computationally expensive (see [9] for an excellent review), the RNN model reasonably balances the biological plausibility and computational efficiency. The RNN model is based on a direct point process representation of signals and a discrete state–space representation of neurons. It neglects some details of pre-/post-synaptic interactions, and focuses on behaviors of networks rather than single cell dynamics. The power of the RNN model has been demonstrated in terms of efficiency in computation [2], capability in universal function approximation [5], competence in learning [3], and tractability in hardware implementation [11]. In addition, the RNN model has been developed for effective applications in a number of domains such as network communication [4] and image processing [7].

Manuscript received August 31, 2004; revised December 28, 2004. This work was supported by the Zakhartchenko Fellowship at Massachusetts Institute of Technology.

S. Jo is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: shjo@mit.edu).

J. Yin is with HRL Laboratories, Malibu, CA 90265 USA (e-mail: jijun@hrl.com).

Z.-H. Mao is with the Harvard-MIT Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: maozh@mit.edu).

Digital Object Identifier 10.1109/TNN.2005.849829

TABLE I
Main Notation Used in the Random Neural Network Model

Notation	Definition
$k_i(t)$	Potential of neuron i at time t
Λ_i	Rate of positive signals arriving at neuron i from outside of the network
λ_i	Rate of negative signals arriving at neuron i from outside of the network
λ_i^+	Average arrival rate of positve signals to neuron i
λ_i^-	Average arrival rate of negative signals to neuron i
r_i	Firing rate of neuron i
p_{ij}^+	Probability that a signal leaving neuron i heads for neuron j as a positive signal
p_{ij}^-	Probability that a signal leaving neuron i heads for neuron j as a negative signal
d_i	Probability that a signal leaving neuron i departs from the network
q_i	Probability that the potential of neuron i is positive

In this letter, we generalize the original RNN model by taking into account state-dependent firing neurons. In the extended model, the times between successive signal emissions of a neuron are dependent on the neuron potential. We prove that under certain conditions the extended RNN still has simple product form stationary distribution. Compared with the original model, the extended model possesses enhanced capability of adjusting the probability structure of the network stationary distribution. This brings new features to the computational power of the RNN.

II. Random Neural Network Model

Consider a network with n neurons in which positive and negative signals circulate. Positive signals $(+1)$ represent excitation, and negative signals (-1) represent inhibition. Each neuron i $(i = 1, \ldots, n)$ is represented at time t by its potential $k_i(t)$, which is a nonnegative integer. An arriving positive signal adds 1 to the neuron potential; an arriving negative signal reduces by 1 the neuron potential or has no effect on the neuron potential if the potential is already zero. If the neuron potential is positive, the neuron may "fire," sending signals out toward other neurons or to the outside of the network. As signals are sent, the neuron potential decreases by the same number of the sent signals. The times between successive signal emissions are exponentially distributed with the firing rate $r_i > 0$. Excitatory and inhibitory signals also arrive at neuron i from the outside of the network at rate Λ_i and λ_i, respectively.

A signal leaving neuron i heads for neuron j as a positive signal with probability p_{ij}^+ or as a negative signal with probability p_{ij}^-, or departs from the network with probability d_i. Assume that the signals leaving a neuron will not return directly to the same neuron, i.e., $p_{ii}^+ = p_{ii}^- = 0$ for all i, and assume that d_i is greater than 0 for all i—this is to reflect the loss of information or energy during the transmission of signals. It is obvious that $\sum_j [p_{ij}^+ + p_{ij}^-] + d_i$ equals 1, $i = 1, \ldots, n$. In this model, the exogenous arrival of signals and firing of neurons are independent of each other. The main notation used in the RNN model has been summarized in Table I.

Let $k(t)$ be the vector of neuron potentials at time t, i.e., $k(t) = (k_1(t), \ldots, k_n(t))'$, $k = (k_1, \ldots, k_n)'$ be a particular value of the vector, and K be the set of all states (i.e., the state–space). Denote

$p(k) = \lim_{t\to\infty} P(k(t) = k)$, and $p(k_i) = \lim_{t\to\infty} P(k_i(t) = k_i)$, where $P(X)$ is the probability of event X.

Gelenbe proved that the stationary distribution of the RNN can be written as the product of the marginal probabilities of the state of each neuron. This leads to simple expression for the network state distribution. Let λ_i^+ and λ_i^- be the average arrival rate of positive and negative signals to each neuron, respectively, and q_i the probability that the potential of neuron i is positive. Gelenbe showed [2] that λ_i^+, λ_i^-, and q_i satisfy the following system of nonlinear simultaneous equations:

$$\begin{aligned} q_i &= \frac{\lambda_i^+}{r_i + \lambda_i^-} \\ \lambda_i^+ &= \sum_j q_j r_j p_{ji}^+ + \Lambda_i \\ \lambda_i^- &= \sum_j q_j r_j p_{ji}^- + \lambda_i \end{aligned} \tag{1}$$

for $i = 1, \ldots, n$, and if a unique nonnegative solution $\{\lambda_i^+, \lambda_i^-, i = 1, \ldots, n\}$ exists to (1) such that $q_i < 1$, then

$$p(k) = \prod_{i=1}^{n} (1 - q_i) q_i^{k_i}. \tag{2}$$

III. Extended Model of RNN With State-Dependent Firing Neurons

In the RNN model presented previously, the times between successive signal emissions of neuron i are exponentially distributed with rate r_i, and r_i is fixed; in other words, r_i is independent of the neuron potential as long as the potential is greater than 0. According to neurophysiology, however, it may be more biologically plausible to assume that the firing rate of a neuron depends on the neuron potential level. Normally, the higher a neuron's excitation level is, the stronger the neuron fires signals. Therefore, we make a natural generalization of the RNN model by including a new feature in the model such that the firing rate of a neuron is now a function of the neuron potential.

We assume that the times between successive signals emissions are exponentially distributed with rate $r_i f_i(k_i(t))$, where $k_i(t)$ is the potential of neuron i and $f_i(\cdot)$ is a function of $k_i(t)$ satisfying $f_i(k_i) > 0$ for $k_i > 0$. A factor $f_i(\cdot)$ is multiplied to r_i in order to represent that the signal emissions are dependent on the neuron potential. Note that the role of function $f_i(\cdot)$ in the extended RNN model is similar to that of the activation function [14] in the conventional perceptron networks: both functions indicate the dependency of neuron firings on the neuron potentials. Like the activation function, function $f_i(\cdot)$ may take the form of threshold functions, e.g., $f_i(k_i) = 1$ for $k_i \geq b$ and $f_i(k_i) = 0.1$ for $k_i < b$ (b is some threshold), the form of sigmoidal functions, e.g., $f_i(k_i) = 1/(1 + e^{-k_i})$, or other forms of functions that characterize neuron firing properties.

It can be tested [1] that, in an RNN with potential-dependent signal emissions, the product form solution may not hold under the original definition of negative signals. Therefore, an updated interpretation has to be given to the effect of negative signals in order to retain the simple product form of solutions, which is computationally efficient. In this letter, we assume that the cancellation effect of negative signals also depends on the potential of the targeting neuron. In fact, this assumption is biologically plausible since a neuron with high excitation level tends to be influenced by inhibitory signals to a greater extent than a neuron with low excitation level. In particular, we assume that (i) $f_i(k_i)$ is bounded from above by B_i, a positive constant, for any $k_i > 0$, and (ii) the negative signals have the following effect: when a negative signal arrives at neuron i, it reduces by 1 the potential of the neuron with probability $f_i(k_i)/B_i$ if $k_i > 0$, otherwise no effect.

Theorem 1: Under the proposed assumptions on the state-dependent firing of neurons and negative signals, if the network stationary distribution exists, it is given by

$$p(k) = c \prod_{i=1}^{n} \prod_{m=1}^{k_i} \frac{q_i}{f_i(m)} \tag{3}$$

with

$$q_i = \frac{\lambda_i^+}{r_i + \frac{\lambda_i^-}{B_i}} \tag{4}$$

where q_i, λ_i^+, and λ_i^- satisfy the system of nonlinear simultaneous equations

$$\begin{aligned} \lambda_i^+ &= \sum_j q_j r_j p_{ji}^+ + \Lambda_i \\ \lambda_i^- &= \sum_j q_j r_j p_{ji}^- + \lambda_i \end{aligned} \tag{5}$$

and c is a normalizing constant defined by

$$c = \left(\sum_{k\in K} \prod_{i=1}^{n} \prod_{m=1}^{k_i} \frac{q_i}{f_i(m)} \right)^{-1}. \tag{6}$$

The state stationary distribution $p(k)$ exists if and only if $c > 0$.

Proof: Since $\{k(t), t \geq 0\}$ is a continuous time Markov chain, it satisfies Fokker–Planck equations. Thus, in steady state it can be seen that $p(k)$ satisfies the following global balance equation:

$$\begin{aligned} p(k) \sum_i & \left\{ \Lambda_i + \left[\lambda_i \frac{f_i(k_i)}{B_i} + r_i f_i(k_i) \right] 1(k_i > 0) \right\} \\ = \sum_i & \Bigg\{ p\left(k_i^+\right) r_i f_i(k_i + 1) \\ & \times \Bigg[1 - \sum_j p_{ij}^+ \\ & \quad - \sum_j p_{ij}^- \frac{f_j(k_j) 1(k_j > 0) + B_j 1(k_j = 0)}{B_j} \Bigg] \\ & + p\left(k_i^-\right) \Lambda_i 1(k_i > 0) + p\left(k_i^+\right) \lambda_i \frac{f_i(k_i + 1)}{B_i} \\ & + \sum_j \Big[p\left(k_{ij}^{+-}\right) r_i f_i(k_i + 1) p_{ij}^+ 1(k_j > 0) \\ & \quad + p\left(k_{ij}^{++}\right) r_i f_i(k_i + 1) p_{ij}^- \frac{f_j(k_j + 1)}{B_j} \\ & \quad + p\left(k_i^+\right) r_i f_i(k_i + 1) p_{ij}^- 1(k_j = 0) \Big] \Bigg\} \end{aligned} \tag{7}$$

where the vectors used are defined by

$$\begin{aligned} k_i^+ &= (k_1, \cdots, k_i + 1, \cdots, k_n)' \\ k_i^- &= (k_1, \cdots, k_i - 1, \cdots, k_n)' \\ k_{ij}^{+-} &= (k_1, \cdots, k_i + 1, \cdots, k_j - 1, \cdots, k_n)' \\ k_{ij}^{++} &= (k_1, \cdots, k_i + 1, \cdots, k_j + 1, \cdots, k_n)' \end{aligned}$$

and $1(X)$ is the characteristic function that takes the value 1 if X is true and 0 otherwise. Since $\{k(t), t \geq 0\}$ is an irreducible Markov chain, if a nonnegative stationary solution exists, it is unique. Now we only need to verify that (3) satisfies (7). We omit the verification, which is similar to that in [2]. The proof of the latter part of Theorem 1, i.e., existence of $p(k)$, is similar to the proof of Jackson Theorem [6]. ■

Note that, unlike the quantity q_i in the original RNN model, q_i in Theorem 1 is no longer equal to the probability that the potential of neuron i is positive. However, it can be tested that λ_i^+ and λ_i^- in Theorem 1 still represent the average rate of positive and negative signal arrival to neuron i, respectively.

The following two theorems present some sufficient conditions for the existence of the RNN stationary distribution.

Theorem 2: There exists a nonnegative solution $\{q_i, \lambda_i^+, \lambda_i^-, i = 1, \ldots, n\}$ to the system of (4) and (5). If $\varliminf_{m\to\infty} f_i(m) > q_i$ for $i = 1, \ldots, n$, the network stationary distribution exists and is given by (3) and (6).

Proof: The existence of a nonnegative solution to (4) and (5) follows Brouwer's fixed-point theorem. If $\varliminf_{m\to\infty} f_i(m) > q_i$ for $i = 1, \ldots, n$, then $\sum_k \prod_{i=1}^n \prod_{m=1}^{k_i} (q_i/f_i(m))$ converges and, thus, $c > 0$. By Theorem 1, the network stationary distribution exists and is given by (3) and (6). ∎

According to Theorem 2, we may ensure the existence of the network stationary distribution by constructing appropriate $f_i(\cdot)$ that satisfies $\varliminf_{m\to\infty} f_i(m) > q_i$ for $i = 1, \ldots, n$. However, the value of q_i is not independent of the choice of $f_i(\cdot)$, since B_i in (4) is an upper bound of $f_i(\cdot)$, which depends on $f_i(\cdot)$. It is still not clear so far if we can find $f_i(\cdot)$ such that $\varliminf_{m\to\infty} f_i(m)$ is greater than q_i. To answer this question, we propose the following theorem.

Theorem 3: Consider an RNN with $\{q_i, \lambda_i^+, \lambda_i^-, i = 1, \ldots, n\}$ being a nonnegative solution to (4) and (5). Then each q_i is no greater than q_{0i} : $\{q_{0i}, i = 1, \ldots, n\}$ is a solution to the following system of equations:

$$q_{0i} r_i = \sum_j q_{0j} r_j p_{ji}^+ + \Lambda_i, \quad i = 1, \ldots, n. \tag{8}$$

Further, if $\varliminf_{m\to\infty} f_i(m) > q_{0i}$ for $i = 1, \ldots, n$, the network stationary distribution exists and is given by (3) and (6).

Proof: First prove the existence of a solution to (8). Denote $n \times n$-matrix $P^+ = (p_{ij}^+)'$ (the transpose of matrix (p_{ij}^+)), $\Lambda = (\Lambda_1, \ldots, \Lambda_n)'$, $\lambda_0^+ = (q_{01} r_1, \ldots, q_{0n} r_n)'$. Then we have $\lambda_0^+ = P^+ \lambda_0^+ + \Lambda$, i.e., $(I - P^+)\lambda_0^+ = \Lambda$. Since $d_i > 0$, it can be tested that $\lim_{m\to\infty} (P^+)^m = 0$, so $I - P^+$ has an inverse and $(I - P^+)^{-1} = \sum_{m=0}^{\infty} (P^+)^m$ [10]. Thus, we can write $\lambda_0^+ = \sum_{m=0}^{\infty} (P^+)^m \Lambda$, which implies the existence of the nonnegative $q_{0i}, i = 1, \ldots, n$.

If we have $q_i \le q_{0i}$, $i = 1, \ldots, n$, then the conclusion that $\sum_k \prod_{i=1}^n \prod_{m=1}^{k_i} (q_{0i}/f_i(m))$ converges leads to the conclusion that $\sum_k \prod_{i=1}^n \prod_{m=1}^{k_i} (q_i/f_i(m))$ converges and, thus, the second part of Theorem 3 holds.

So the only thing left is to prove $q_i \le q_{0i}, i = 1, \ldots, n$. Let $G = \mathrm{diag}\{g_1, g_2, \ldots, g_n\}$, where $g_i = (r_i/(r_i + \lambda_i^-)/(B_i)) \le 1$, and denote $\lambda^+ = (\lambda_1^+, \ldots, \lambda_n^+)'$. Then we have $\lambda^+ = P^+ G \lambda^+ + \Lambda$ and, thus, $\lambda^+ = \sum_{m=0}^{\infty} (P^+ G)^m \Lambda$, following the same argument as shown previously. Since $\lambda_0^+ = \sum_{m=0}^{\infty} (P^+)^m \Lambda$ as shown previously and $g_i \le 1$, we have $\lambda_i^+ \le q_{0i} r_i, i = 1, \ldots, n$, thus, $q_i = (\lambda_i^+/(r_i + \lambda_i^-)/(B_i)) \le (\lambda_i^+/r_i) \le (q_{0i} r_i / r_i) = q_{0i}, i = 1, \ldots, n$. ∎

Theorem 3 provides an upper bound of q_i, i.e., $q_{0i}, i = 1, \ldots, n$, for the network, and shows that the existence of the stationary distribution of an RNN can be guaranteed by choosing appropriate $f_i(\cdot)$ such that $\varliminf_{m\to\infty} f_i(m) > q_{0i}$. One nice thing about introducing q_{0i} is that q_{0i} is independent of the choice of $f_i(\cdot), i = 1, \ldots, n$. This property makes it easy to ensure existence of the network stationary distribution.

IV. Discussion

In the original model of RNN, the network stationary distribution is completely determined if $q_i, i = 1, \ldots, n$, are determined. In the extended model of RNN with state-dependent firing neurons, however, the network stationary distribution depends not only on q_i, but also on $f_i(\cdot), i = 1, \ldots, n$. As shown in the following, the introduction of the state-dependent factor $f_i(\cdot)$ brings new features to the computational power of the RNN. We will find soon that the extended model of RNN has enhanced capacity to adjust the probability structure of the network stationary distribution. As an example, this section demonstrates how desired associate memory states can be stored in the network dynamics.

What are associative memory states? In this letter, an associative memory state is defined as a state with the maximum probability in the steady-state distribution. The associative memory state of an RNN with state-dependent firing neurons depends on both the inputs and parameters of the network. Denote $U = (\Lambda, \lambda)'$ the input vector, where $\Lambda = (\Lambda_1, \ldots, \Lambda_n)'$ and $\lambda = (\lambda_1, \ldots, \lambda_n)'$, and denote W the vector containing all adjustable parameters of the network. For instance, in the original RNN, W is comprised of p_{ij}^+, p_{ij}^-, and $r_i, i, j = 1, \ldots, n$; while in the RNN with state-dependent firing neurons, since $f_i(\cdot)$ may have adjustable parameters, these parameters should also be included in W. Then denote k_M an associative memory state of the RNN, given the input vector U and parameter vector W.

Consider the capacities of the original RNN and the extended RNN in storing the associate memory states. In the original RNN, for any input vector U and parameter vector W, if the stationary distribution exists, the associative memory state k_M equals $(0, \ldots, 0)'$. This follows the fact that the stationary distribution $p(k)$ determined by (2) is monotone decreasing for any $k_i, i = 1, \ldots, n$, and $p(k)$ takes the maximum value when $k = (0, \ldots, 0)'$. In comparison, it can be tested that in the extended RNN with state-dependent firing neurons, given any $k = (k_1, \ldots, k_n)'$, if $f_i(\cdot)$ and q_i satisfy $f_i(m) < q_i$ for any $m \le k_i$ and $f_i(m) > q_i$ for any $m > k_i$, $i = 1, \ldots, n$, then the associate memory state k_M equals k.

The previous argument implies that the original RNN does not have the ability to adjust associative memory state (the only associative memory state that can be stored in the network is $(0, \ldots, 0)$), while the extended RNN has the ability to store any associative memory state, with appropriate setting of network parameters. For example, let $f_i(\cdot)$ be monotone increasing and amplitude modulatable (e.g., $f_i(m) = d_i/(1+e^{-m})$, where d_i is an adjustable parameter), then any state k can be made an associative memory state of the RNN if the amplitude of $f_i(\cdot)$ can be modulated to satisfy $f_i(k_i) < q_i < f_i(k_i + 1)$, $i = 1, \ldots, n$.

Moreover, we may prove that in the extended RNN with state-dependent firing neurons, if $f_i(\cdot)$ is allowed to take arbitrary positive values at points $k_i = 1, 2, \ldots$, then the stationary distribution of neuron i can be modulated to approximate any probability distribution of nonnegative integers. The stationary distribution of neuron i is $p(k_i) = c_i \prod_{m=1}^{k_i} (q_i/f_i(m))$, where c_i is a normalizing constant. For any given probability distribution of nonnegative integers $\hat{p}(k_i) > 0$, let $f_i(k_i) = (q_i \hat{p}(k_i - 1))/\hat{p}(k_i)$, $k_i > 0$, then we have $p(k_i) = \hat{p}(k_i)$.

Compared with the original RNN, which has a stationary distribution of geometric decreasing structure, the extended RNN has enhanced capability to present a variety of probability structures in the stationary distribution of neuron states. However, an increase of the number of network parameters tends to increase the complexity of the network structure and consequently raise difficulty in training of parameters. This may lead to limitations in the application and generalization of the RNN. Therefore, appropriate choice of $f_i(\cdot)$ and its adjustable parameters is a critical component of the RNN modeling.

Acknowledgment

The authors would like to thank the editor and reviewers for their valued comments.

References

[1] X. L. Chao, "Networks of queues with customers, signals and arbitrary service time distributions," *Oper. Res.*, vol. 43, no. 3, pp. 537–544, 1995.

[2] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Computat.*, vol. 1, no. 4, pp. 502–511, 1989.

[3] E. Gelenbe and K. Hussain, "Learning in the multiple class random neural network," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1257–1267, Nov. 2002.

[4] E. Gelenbe, R. Lent, and Z. Xu, "Measurement and performance of a cognitive packet network," *Comput. Netw.*, vol. 37, no. 6, pp. 691–701, 2001.

[5] E. Gelenbe, Z.-H. Mao, and Y.-D. Li, "Function approximation with spiked random networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 1, Jan. 1999.

[6] E. Gelenbe and G. Pujolle, *Introduction to Queueing Networks*. New York: Wiley, 1987.

[7] E. Gelenbe, M. Sungur, C. Cramer, and P. Gelenbe, "Traffic and video quality in adaptive neural video compression," *Multimedia Syst.*, vol. 4, no. 6, pp. 357–369, 1996.

[8] W. Gerstner and W. Kistler, *Spiking Neuron Models—Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[9] E. M. Izhikevich, "Which model to use for cortical spiking neurons?," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.

[10] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. New York: Van Nostrand, 1960.

[11] T. Kocak, J. Seeber, and H. Terzioglu, "Design and implementation of a random neural network routing engine," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1128–1143, Sep. 2003.

[12] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.

[13] W. Maass and C. M. Bishop, Eds., *Pulsed Neural Networks*. Cambridge, MA: MIT Press, 1998.

[14] J. Sima and P. Orponen, "General-purpose computation with neural networks: a survey of complexity theoretic results," *Neural Computat.*, vol. 15, pp. 2727–2778, 2003.

[15] W. R. Softky, "Fine analog coding minimizes information transmission," *Neural Netw.*, vol. 9, no. 1, pp. 15–24, 1996.

An Efficient Parameterization of Dynamic Neural Networks for Nonlinear System Identification

Victor M. Becerra, Freddy R. Garces, Slawomir J. Nasuto, and William Holderbaum

***Abstract*—Dynamic neural networks (DNNs), which are also known as recurrent neural networks, are often used for nonlinear system identification. The main contribution of this letter is the introduction of an efficient parameterization of a class of DNNs. Having to adjust less parameters simplifies the training problem and leads to more parsimonious models. The parameterization is based on approximation theory dealing with the ability of a class of DNNs to approximate finite trajectories of nonautonomous systems. The use of the proposed parameterization is illustrated through a numerical example, using data from a nonlinear model of a magnetic levitation system.**

***Index Terms*—Approximation theory, architectures and algorithms, dynamic systems, neural networks.**

Manuscript received May 23, 2003; revised June 11, 2004.

V. M. Becerra, S. J. Nasuto, and W. Holderbaum are with the University of Reading, Reading RG6 6AY, U.K. (e-mail: v.m.becerra@reading.ac.uk).

F. R. Garces is with Spiral Software Ltd., Cambridge CB4 1DL, U.K.

Digital Object Identifier 10.1109/TNN.2005.849844

I. Introduction

Due to the nonlinear nature of many systems, physical and otherwise, there has been extensive research covering the field of nonlinear system identification [1]–[3]. It is here that the use of neural networks emerges as a feasible solution. The universal approximation properties of static neural networks [4] make them a useful tool for modeling nonlinear systems. The problem of nonlinear modeling using static neural networks has been extensively researched [5] and many approaches have used multilayer perceptrons and radial basis functions [2], [6], [7]. The inputs to these static networks are usually delayed values of the inputs and outputs of the plant. This approach, however, has some disadvantages. First, the input structure is not easy to choose. Moreover, discrete time nonlinear models require retraining when the sampling time is changed. Furthermore, if the models are to be employed as part of a nonlinear control scheme, methods for discrete time nonlinear control are not as well developed as continuous time nonlinear control methods [8].

Continuous time Hopfield-type dynamic neural networks (DNNs) [9] and their variations [10]–[13], do not present the previous disadvantages. Several techniques have been proposed that have characterized the nonlinear modeling properties of DNNs [5], [14], [15]. A number of control applications of this family of neural networks have also been proposed [16], [17]. Research work has demonstrated that DNNs can approximate finite trajectories of n-dimensional autonomous dynamic systems of the form $\dot{x}(t) = f(x(t))$ [18], [19]. This letter presents implications of approximation theory presented in [16] on the ability of a class of DNNs to approximate finite trajectories of nonautonomous systems of the form $\dot{x}(t) = f(x(t), u(t))$. The letter introduces an efficient parameterization of a class of DNNs and illustrates its use through a numerical example.

The letter is organized as follows. Section II introduces the class of DNNs of interest in this letter. Section III discusses theoretical results on the approximation ability of DNNs. Section IV discusses offline training methods for DNNs. Section V introduces an efficient parameterization of a class of DNNs. Section VI presents an numerical example. Finally, Section VII gives concluding remarks.

II. DNNs

DNNs are made of interconnected dynamic neurons. The class of neuron of interest in this letter is described by the following differential equation:

$$\dot{x}_i = -\beta_i x_i + \sum_{j=1}^{N} \omega_{ij}\sigma(x_j) + \sum_{j=1}^{m} \gamma_{ij} u_j \tag{1}$$

where β_i, ω_{ij}, and γ_{ij} are adjustable weights, with $1/\beta_i$ a positive time constant and x_i the activation state of the ith unit, $\sigma : \mathbb{R} \to \mathbb{R}$ a sigmoid function and $u_1, \ldots, u_m$ the input signals. Fig. 1 shows the block diagram of a dynamic neuron.

A DNN is formed by a single layer of N units. The states of the first n units are taken as the outputs of the network, leaving $N - n$ units as hidden neurons. The network is defined by the following vectorized expression:

$$\begin{aligned}\dot{x} &= -\beta x + \omega\sigma(x) + \gamma u \\ y_n &= Cx\end{aligned} \tag{2}$$

where x are coordinates on $\mathbb{R}^N$, $\beta \in \mathbb{R}^{N\times N}$ is a diagonal matrix with diagonal elements $\{\beta_1, \ldots, \beta_N\}$, $\omega \in \mathbb{R}^{N\times N}$, $\gamma \in \mathbb{R}^{N\times m}$ are weight matrices, $\sigma(x) = [\sigma(x_1), \ldots, \sigma(x_N)]^T$ is a vector sigmoid function, $u \in \mathbb{R}^m$ is the input vector, $y_n \in \mathbb{R}^n$ is the output vector, $C = [I_{n\times n}, 0_{n\times(N-n)}]$.