Stochastic Noise Process Enhancement of Hopfield Neural Networks

Vladimir Pavlović, *Member, IEEE*, Dan Schonfeld, *Member, IEEE*, and Gary Friedman, *Member, IEEE*

***Abstract*—Hopfield neural networks (HNN) are a class of densely connected single-layer nonlinear networks of perceptrons. The network's energy function is defined through a learning procedure so that its minima coincide with states from a predefined set. However, because of the network's nonlinearity, a number of undesirable local energy minima emerge from the learning procedure. This has shown to significantly effect the network's performance. In this brief, we present a stochastic process-enhanced binary HNN. Given a fixed network topology, the desired final distribution of states can be reached by modulating the network's stochastic process. We design this process, in a computationally efficient manner, by associating it with *stability intervals* of the nondesired stable states of the network. Our experimental simulations confirm the predicted improvement in performance.**

***Index Terms*—Hopfield neural networks (HNNs), stochastic HNNs (SHNNs).**

I. Introduction

HOPFIELD neural networks (HNNs) are a class of nonlinear function approximators represented by a single-layer network consisting of interconnected individual perceptrons and modified perceptrons (with sigmoidal nonlinearities) [1]. The basis for its operation is the Hebbian learning algorithm which selects network weights to minimize the network energy function for a set of desired states. Unfortunately, because of its nonlinear character, the network also exhibits nondesirable, local minima. This has shown to affect the network performance, both in its capacity and its ability to address its content [2]. Several approaches based on stochastic modifications of the network [3], [4] have been proposed that deal with the problem of local minima. Alternatively, the stochastic HNNs can be viewed as a Boltzmann machine [5] and Gibbs distribution of the final network states in global minima. Network learning and convergence can then be studied in probabilistic terms. In many applications, however, the desired final network state distribution corresponds to particular local minima, and not necessarily to the global minima. The use of Gibbs distribution is thus undesirable in many applications.

In his work on the network robustness, Schonfeld [6] observed a phenomenon where a stochastic perturbation on network thresholds improved performance of a binary HNN. In this brief, we introduce a new way to stochastically enhance performance of binary HNNs, first reported in [7] and [8]. Presence of a stochastic process allows us to describe the evolution of the network as a Markov chain. We propose an optimization criterion for enhancement of the HNN performance based on a desired limiting distribution of the Markov chain. The criterion can be related to *stability intervals* associated with the desired and nondesired stable states of the network, given a fixed network topology. Given the stability intervals, we suggest a noise process design which will result in a suboptimal HNN based on the minimization of undesirable state final probabilities.

Our approach is related to the notion of stability and performance in Hopfield networks, which have been studied in literature [9]–[12]. Aiyer *et al.* have studied the dynamics of Hopfield networks in terms of the eigenvalues of the connection matrix. Discussion of stability of feasible solutions was presented in [10] as a function of network's energy function and, hence, connection coefficients. In a seminal study [11] of stability of feasible and infeasible solutions of combinatorial optimization problems represented as Hopfield networks Matsuda proposed general conditions on network coefficients for altering stability of desired feasible solutions. He also proposed similar criteria for hysteresis Hopfield networks [12] by adding hysteresis size as a variable. Unlike these approaches, our method assumes fixed connection weights determined by some other method and concentrates of the design of the input noise process. By considering stochastic inputs we hope to circumvent a potentially complex simultaneous optimization of all network parameters.

II. Stochastic HNN

Stochastic HNN (SHNN) is a generalization of the original Hopfield model [1]. The network is defined as

$$\mathcal{H}_s(\Upsilon, \mathbf{C}, p_{U_i}) : \underline{\mathbf{x}}^{(k+1)} = \Upsilon\left(\mathbf{C} \cdot \underline{\mathbf{x}}^{(k)} + \underline{\mathbf{u}}^{(k)}\right)$$

where $\underline{\mathbf{x}}^{(k)} \in \mathcal{R}^N$ is its state at time k and Υ a monotonically nondecreasing operator[1]. $\underline{\mathbf{u}}^{(k)} \in \mathcal{R}^N$ is the random *threshold vector* with pdf $p_U(\underline{\mathbf{u}}^{(k)})$ and $\mathbf{C}$ is a fixed matrix of connection weights $c_{ij}, \mathbf{C} = [c_{ij}]_{N\times N} = [\underline{\mathbf{c}}_1 \ \underline{\mathbf{c}}_2 \ \cdots \ \underline{\mathbf{c}}_N]'$. The set of all possible network states is denoted by $\mathcal{S}$.

Control over the threshold vector enables one to (possibly) control the network. Given two consecutive state vectors of our

[1]While we assume synchronous update of the network state, our analysis also holds for the asynchronous update model.

Manuscript received June 23, 2003; revised September 13, 2004. This paper was recommended by Associate Editor G. Cauwenberghs.

V. Pavlović is with the Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA (e-mail: vladimir@cs.rutgers.edu).

D. Schonfeld is with the Department of Electrical and Computer Engineering, University of Illinois Chicago, IL 60607 USA.

G. Friedman is with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104 USA.

Digital Object Identifier 10.1109/TCSII.2004.842027

1057-7130/$20.00 © 2005 IEEE

general SHNN, a set of all threshold vectors that will force the system to make the desired transition is called the *transition set*

$$\mathcal{T}\left(\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)}\right) = \left\{\underline{\mathbf{u}} : \underline{\mathbf{x}}^{(k+1)} = \Upsilon\left(\mathbf{C}\cdot\underline{\mathbf{x}}^{(k)} + \underline{\mathbf{u}}\right)\right\}.$$

Stable state of the network corresponds to the fixed point of the mapping defined by the network $\underline{\mathbf{0}} \in \mathcal{T}(\underline{\mathbf{x}}_s \rightarrow \underline{\mathbf{x}}_s)$. Conversely, self-transition sets of nonstable states do not contain $\underline{\mathbf{0}}$. Stable states of SHNN are determined by the network matrix $\mathbf{C}$, for fixed Υ, and form the *stable set* of the network, $\mathcal{S}_s$. Given a set of the desired or "*good*" stable states, a learning procedure can be used to select an optimal value of $\mathbf{C}$. However, in the best case, a learning procedure yields a subset of the network's stable state set, $\mathcal{S}_t \subset \mathcal{S}_s$[2] and a set of undesired, "*bad*" stable states $\mathcal{S}_b$.

A. Distribution of Network States and Optimal Network

Dynamics of state transitions in $\mathcal{H}_s$ can be described as a Markov chain with the state mixing matrix $P(k) = [p_{i,j}(k)], p_{ij}(k) = \Pr\{\underline{\mathbf{u}}^{(k)} \in \mathcal{T}(\underline{\mathbf{X}}_i \rightarrow \underline{\mathbf{X}}_j)\}$. The limiting distribution of all network states can now be related to the structure of the state transition probability matrix $[p_{ij}(k)]$. Namely, $p_{ii} = 1$ for all stable states i in a deterministic HNN.

If $p_U(\underline{\mathbf{u}}^{(k)})$ is designed such that the limiting distribution of the "bad" states becomes zero, then the "bad" states can be altogether avoided. We call such a network optimal. The optimality may be achieved if.

Proposition 1: The optimality of a SHNN is imposed by $\Pr\{\underline{\mathbf{u}}^{(k)} \in \mathcal{T}(\underline{\mathbf{x}} \rightarrow \underline{\mathbf{x}}_t)\} > 0, \Pr\{\underline{\mathbf{u}}^{(k)} \in \mathcal{T}(\underline{\mathbf{x}}_t \rightarrow \underline{\mathbf{x}}_t)\} = 1$ for all $\underline{\mathbf{x}} \in \mathcal{X} - \mathcal{S}_t$ and $\underline{\mathbf{x}}_t \in \mathcal{S}_t$.

While the conditions seem simple, enforcing them by optimizing the noise process $p_U(u)$ is not trivial because of the complex dependence of $[p_{ij}]$ on p_U. However, in a BSHNN, we can consider an alternative approach outlined below.

III. Binary SHNN

Binary SHNN, with Υ a binary threshold function and $\underline{\mathbf{x}} \in \{-1, +1\}^N$, allows a simpler definition of the transition sets:

Proposition 2: Element $\underline{\mathbf{u}}$ belongs to the transition set $\mathcal{T}(\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)})$ of a BSHNN iff for each i

$$x_i^{k+1}\cdot u_i > -x_i^{k+1}\cdot\left[\mathbf{C}\underline{\mathbf{x}}^{(k)}\right]_i = -x_i^{k+1}\cdot\underline{\mathbf{c}}_i'\underline{\mathbf{x}}^{(k)}$$

where $[\underline{\mathbf{x}}]_i$ denotes the i-th element of vector $\underline{\mathbf{x}}$.

This proposition formulates a simple condition which we now use to define the notion of *stability intervals* for each state. Notions similar to stability intervals such as domains or basins of convergence have been introduced in literature [9]–[12]. Our notion of stability intervals provide an alternative, simple measure of a state's robustness that involves only one dimensional concepts.

Let

$$\mathcal{I}^+\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right) = \left\{u : u = -\underline{\mathbf{c}}_i'\underline{\mathbf{x}}^{(k)}, x_i^{(k+1)} > 0\right\} \quad (1)$$

$$\mathcal{I}^-\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right) = \left\{u : u = -\underline{\mathbf{c}}_j'\underline{\mathbf{x}}^{(k)}, x_j^{(k+1)} < 0\right\} \quad (2)$$

be two sets associated with the consecutive states $\underline{\mathbf{x}}^{(k)}$ and $\underline{\mathbf{x}}^{(k+1)}$ in binary SHNN $\mathcal{H}_s$, and let

$$i_{\min}\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right) = \sup \mathcal{I}^+\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right) \quad (3)$$

$$i_{\max}\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right) = \inf \mathcal{I}^-\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right). \quad (4)$$

Then, the condition for the threshold u being in the transition set is simply

$$i_{\min}\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right) < u < i_{\max}\left(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}\right). \quad (5)$$

The proof can be found in [7]. Consequently, $\Pr_U(u\cdot\underline{\mathbf{1}} \in \mathcal{T}(\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)})) = \Pr_U(i_{\min}(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}) < u < i_{\max}(\underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}))$.

The following two corollaries establish stability conditions for a BSHNN.

Corollary 1: State $\underline{\mathbf{x}}_c$ in binary SHNN $\mathcal{H}_s$ is conditionally stable iff there exists interval $(i_{\min}, i_{\max})$ associated with $\underline{\mathbf{x}}_c$ such that

$$i_{\min}(\underline{\mathbf{x}}_c) = \sup\{u : u = -\underline{\mathbf{c}}_i'\underline{\mathbf{x}}_c, \underline{\mathbf{x}}_{c,i} > 0\} \quad (6)$$

$$i_{\max}(\underline{\mathbf{x}}_c) = \inf\{u : u = -\underline{\mathbf{c}}_j'\underline{\mathbf{x}}_c, \underline{\mathbf{x}}_{c,j} < 0\}. \quad (7)$$

Interval $(i_{\min}, i_{\max})_{\underline{\mathbf{x}}_c} = (i_{\min}(\underline{\mathbf{x}}_c), i_{\max}(\underline{\mathbf{x}}_c))$ is called the conditional stability interval associated with $\underline{\mathbf{x}}_c$.

Corollary 2: State $\underline{\mathbf{x}}_s$ in binary SHNN $\mathcal{H}_s$ is stable iff it is conditionally stable and $0 \in (i_{\min}, i_{\max})_{\underline{\mathbf{x}}_c}$.

Corollary 2 gives us a very simple way to assert stability and robustness of certain states. The wider the interval, the more robust the state. The notion of robustness will become particularly interesting when dealing with network optimization.

A. Optimal Binary Stochastic HNN

Because all the properties of a general SHNN also hold for binary stochastic network, the optimization procedure remains defined the same way as it is in the case of SHNN. Alternatively, we now relate the stability intervals to HNN's optimality in the following proposition.

Proposition 3: Let $\mathcal{H}_b$ be the proper BSHNN, with the "good"-state set $\mathcal{S}_t$ and the "bad" state set $\mathcal{S}_b$, and the threshold vector distributed according to the distribution p_U. The network is optimal if $p_U(u)$ is such that

$$\Pr\left(u \in \bigcap_{\underline{\mathbf{X}}\in\mathcal{S}_t} (i_{\min}, i_{\max})_{\underline{\mathbf{X}}}\right) = 1 \quad (8)$$

$$\Pr\left(u \in \bigcup_{\underline{\mathbf{X}}\in\mathcal{S}_b} (i_{\min}, i_{\max})_{\underline{\mathbf{X}}}\right) = 0 \quad (9)$$

$$\Pr\left(\underline{\mathbf{u}}^{(k)} \in \bigcup_{\underline{\mathbf{X}}\in\mathcal{X}-\mathcal{S}_t-\mathcal{S}_b} \mathcal{T}(\underline{\mathbf{X}} \rightarrow \underline{\mathbf{X}})\right) = 0. \quad (10)$$

Proposition 3 follows from Proposition 1, and is one way to satisfy some of its conditions. However, concurrently satisfying all three conditions is not trivial. This is particularly true for the third condition which guarantees instability of the originally nonstable states. We now suggest a heuristic approach that could lead to the optimal network.

[2]When the network is not saturated. Furthermore, it is extremely difficult to determine the stable state set for a given network (cf., [9], [11], [12].)

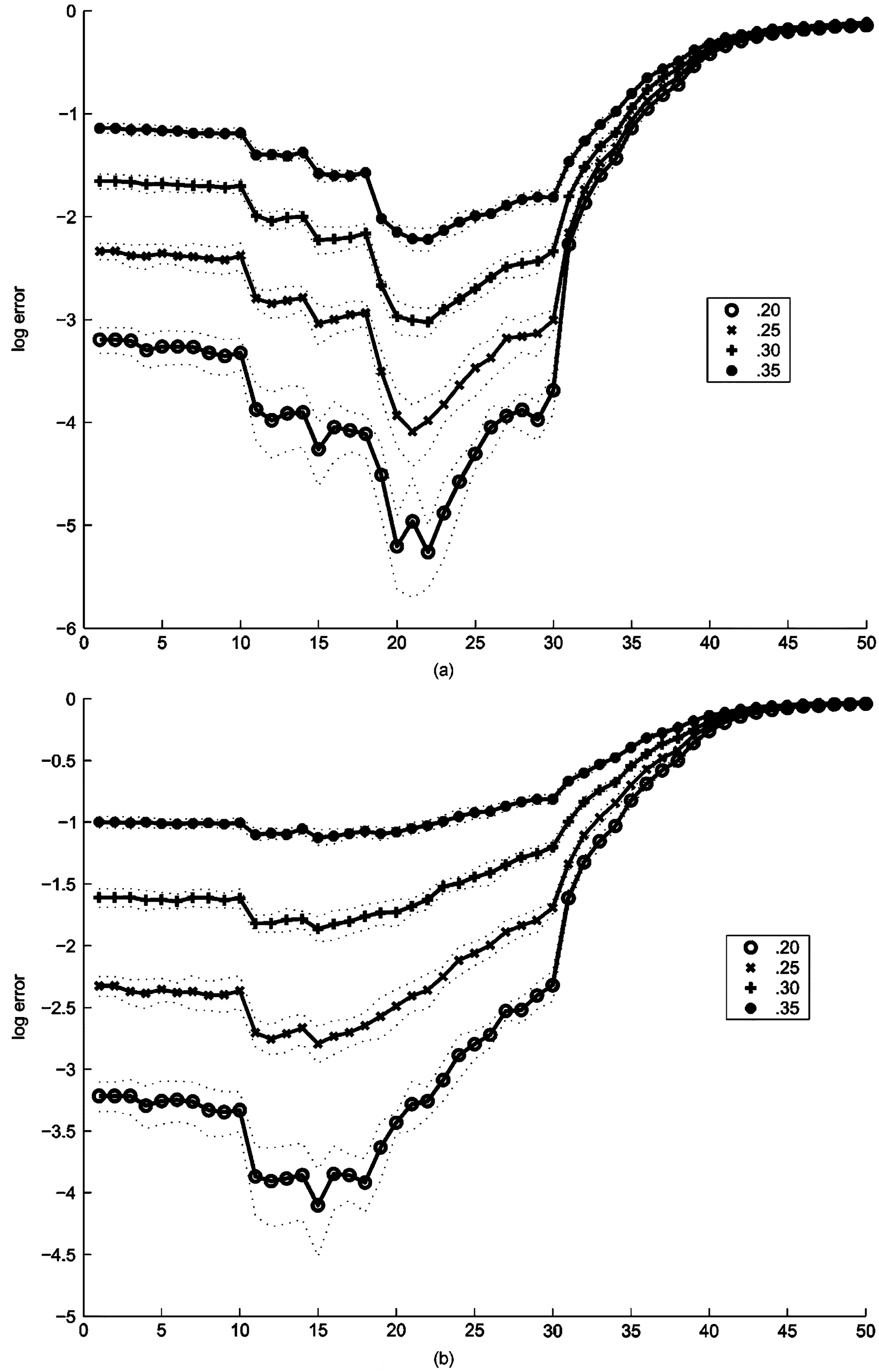

Fig. 1. Classification error for the "Bad-Good" (a) and "Exact" (b) error measures as a function of the range of support, $[-R, R]$, of the uniform noise for $R = 0, 1, \ldots, 50$. Results shown are for four different noise levels ($p_{\text{init}} = \{20\%, 25\%, 30\%, 35\%\}$) used to create initial states. Also shown are one-standard-deviation bounds.

Heuristic 1: Let $p_U(u)$ be such that

$$\Pr\left(u \in \bigcap_{\underline{\mathbf{X}} \in \mathcal{S}_t} (i_{\min}, i_{\max})_{\underline{\mathbf{X}}}\right) = 1, \qquad \underline{\mathbf{X}} \in \mathcal{S}_t \tag{11}$$

$$\Pr\left(u \in \bigcup_{\underline{\mathbf{X}} \in \mathcal{S}_b} (i_{\min}, i_{\max})_{\underline{\mathbf{X}}}\right) = p_e, \qquad \underline{\mathbf{X}} \in \mathcal{S}_b. \tag{12}$$

where $0 < p_e < 1$ is an arbitrary parameter. Then, the network is quasi-optimal.

Condition (12) poses a weaker condition on $p_U(u)$ than the one in (9) resulting in possibly stable "bad" states with some probability p_e. Finally, what distinguishes Heuristic 1 from Proposition 3 is the nonexistence of the condition given by (10). Hence, there is no guarantee that some originally nonstable states would not become stable. Still, experimental results show that in large majority of cases the first two conditions tend to be sufficient.

B. Optimality and Hebbian Learning

One of the most widely applied learning procedures related to HNN is due to Hebb [13]. The strength of network connections is determined based on correlation of desired states. For the set $\mathcal{S}_t$ of M desired stable states $\mathbf{C} = \sum_{i=1}^{M} \underline{\mathbf{X}}_i \underline{\mathbf{X}}_i' - M\mathbf{I}$ where $\mathbf{I}$ is an identity matrix.

It is easy to show that such choice of the connection matrix results in network convergence for any initial state. Unfortunately, one is not in principle guaranteed that the network is proper nor whether it has converged to undesired, "bad" states. In order to eliminate such states using the optimization procedure from the previous section, we need to estimate the "bad" states from the known set of the "good" ones. We now propose two tests that address these issues.

The capacity of Hopfield networks with Hebbian learning has been studied extensively. It was shown that the capacity of such networks is $O(N \log N)$ [14]. We provide an alternative, more specific test, for an easy-to-compute upperbound on the number of desired states that can be learned using the Hebbian procedure as follows.

Proposition 4: A binary Hebbian SHNN is proper if $M < N - \max_j \{\sum_{i=1, i\neq j}^{M} |\underline{\mathbf{X}}_i' \underline{\mathbf{X}}_j|\}, \underline{\mathbf{X}}_i, \underline{\mathbf{X}}_j \in \mathcal{S}_t$.

The proof of this proposition can be found in [7]. While the bound appears to be conservative, $O(N)$ vs. $O(N \log N)$, it nevertheless is useful for an initial estimate of proper functioning.

The second test attempts to infer the elements of the "bad" stable set, $\mathcal{S}_b$ from the "good" states in $\mathcal{S}_t$.

Proposition 5: Given a proper Hebbian binary SHNN, where $\underline{\mathbf{X}}_i' \underline{\mathbf{X}}_j > 0, \forall \underline{\mathbf{X}}_i, \underline{\mathbf{X}}_j \in \mathcal{S}_t$ the elements of the stable set $\mathcal{S}_s$ satisfy

$$\underline{\mathbf{X}}_k \in \mathcal{S}_s \Rightarrow \underline{\mathbf{X}}_k = \text{median}\,\{\mathcal{S}_k\}, \qquad \mathcal{S}_k \subseteq \mathcal{S}_s.$$

This test is heuristic. However, the procedure itself can be used as a part of quasi-optimization suggests in Proposition 3, which requires the knowledge of "bad" states. We successfully used this procedure in our experimental studies.

IV. Experiments

A Hebbian binary synchronous HNN was designed to perform an image classification task: noisy versions of numbers 4, 5, 6, and 7 were to be classified into one of the four classes. The noisy images were obtained from the four desired images by reversing the state of individual elements with probabilities $p_{\text{init}} = \{20\%, 25\%, 30\%, 35\%\}$. Ten sets of 1000 different images were generated in this manner. Two different error measures were used for the evaluation of the performance of the this network: (1) "Exact" error measure. This measure requires exact matching of the final state with the instance used to obtain the initial state of the network. (2) "Bad-Good" error measure. According to this measure, an error occurs whenever the final state of the network differs from *any* one of the four images.

To determine the set "bad" states we subjected a zero-threshold binary network to 200 random initial states and recorded nondesired stable network states. For all stable states we computed the corresponding stability intervals (see [7] for details). Using the conditions of the Heuristic 1 this resulted in the intersection of the "good" stability intervals

$$\cap_{\underline{\mathbf{X}} \in \mathcal{S}_t} (i_{\min}, i_{\max})_{\underline{\mathbf{X}}} = (-30, 30). \tag{13}$$

Similarly, for "bad" states we have

$$\cup_{\underline{\mathbf{X}} \in \mathcal{S}_b} (i_{\min}, i_{\max})_{\underline{\mathbf{X}}} = (-14, 14). \tag{14}$$

Next, a stochastic version of the original network was applied to the ten sets of initial states. Range of support of a uniform pdf noise process $[-R, R]$ was $R = 0, 1, \ldots, 50$. For each range the network performed a fixed number (40) of iterations, without any annealing on the noise process. Results of classification based on the two different error measures proposed are shown in Fig. 1(a) and (b).

In the case of "Bad-Good" error measure, a drastic improvement in performance occurred for noise levels between 14 and 25. The lower bound corresponds to the highest stability interval limit of the states in the "bad" state set in (14). The upperbound is slightly lower than the lowest limit of states in the training set, as shown in (13). This was expected, following the analysis of the quasi-optimal network. The "exact" error measure shows an improvement in a narrower interval starting at about 10 and ending at 18. This can be explained by the fact that the quasi-optimization procedure does not have full control over the final distribution (see additional results in [8]). However, the overall convergence to the elements of training set still shows good results. Additional experiments described in [7], [8] also confirm the importance of condition (12) from our heuristic.

V. Conclusion

In this brief, we studied the effects of stochastic noise on the performance of binary HNN. We propose an optimization criterion for the enhancement of HNN performance based on *stability intervals* associated with the desired and nondesired stable states of the network. Because of the complexity of the general criterion we restrict the optimization to the set of nondesired states. Given the stability intervals, we formulate a stochastic process design which satisfies the restricted optimization criterion and results in enhanced performance of the network. Conducted experimental simulations confirm the predicted improvement in performance despite simplification of the final criterion.

Nevertheless, the general optimization procedure, as proposed in Section II, still remains an interesting open problem. Solution to this problem could enable realization of HNNs with arbitrary final state distribution. Extension of this approach to multi-layer networks can lead to the enhancement of a class of nonlinear iterative operators and systems in numerous applications such as nonlinear feedback control and optimization algorithms.

REFERENCES

[1] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, pp. 2554–2558, 1982.

[2] S. Amari and K. Maginu, "Statistical neurodynamics of associative memory," *Neural Netw.*, vol. 1, pp. 63–73, 1988.

[3] T. Nakamura, A. Tsuneda, and T. Inoue, "A new stochastic binary neural network based on Hopfield model and its application," in *Proc. ITC CSCC*, Phuket, Thailand, 2002, pp. 28–34.

[4] S. Hu, X. Liao, and X. Mao, "Stochastic Hopfield neural networks," *J. Phys. A: Math. Gen.*, vol. 36, pp. 2235–2249, 2003.

[5] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in boltzmann machines," in *Parallel Distributed Processing*, D. E. Rumelhart and J. E. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 282–317.

[6] D. Schonfeld, "On the hysteresis and robustness of Hopfield neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 2, no. 11, pp. 745–748, Nov. 1993.

[7] V. Pavloviæ, "Stochastic noise process enhancement of Hopfield neural networks," Master's thesis, University of Illinois at Chicago, 1993.

[8] V. Pavlovic, D. Schonfeld, and G. Friedman, "Enhancement of Hopfield neural networks using stochatic noise processes," in *Proc. Neural Networks for Signal Processing Soc. Workshop*, Sep. 2001, pp. 173–182.

[9] S. V. B. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the Hopfield model," *IEEE Trans. Neural Netw.*, vol. 1, no. 2, pp. 204–215, Feb. 1990.

[10] S. Abe, "Global convergence and suppression of spurious states of the Hopfield neural networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 40, no. 4, pp. 246–257, Apr. 1993.

[11] S. Matsuda, ""Optimal" Hopfield network for combinatorial optimization with linear cost function," *IEEE Trans. Neural Networks*, vol. 9, no. 6, pp. 1319–1330, Jun. 1998.

[12] ——, "Theoretical considerations on the dynamics of hysteresis binary Hopfield networks for combinatorial optimization," in *Proc. Int. Joint Conf. Neural Networks*, Jul. 2000, pp. 480–485.

[13] D. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[14] S. S. Venkatesh and D. Psaltis, "Linear and logarithmic capacities in associative neural networks," *IEEE Trans. Inf. Theory*, vol. 35, no. 3, pp. 558–568, Mar. 1989.