In *IEEE Workshop on Neural Network for Signal Processing (NNSP)*
Falmouth, MA, 2001

ENHANCEMENT OF HOPFIELD NEURAL NETWORKS USING STOCHATIC NOISE PROCESSES

Vladimir Pavlović
Melrose, MA 02176
vladimir@pavlovic.net

Dan Schonfeld and Gary Friedman
University of Illinois at Chicago
Chicago, IL 60607
{*ds,gary*}*@eecs.uic.edu*

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 work we present a stochastic process enhanced bipolar HNN. Presence of the stochastic process in the network enables us to describe its evolution using the Markov chains theory. When faced with a fixed network topology, the desired final distribution of states can be reached by modulating the network's stochastic process. Guided by the desired final distribution, we propose a general L^2 norm error density function optimization criterion for enhancement of the Hopfield neural network performance. This criterion can also be viewed in terms of *stability intervals* associated with the desired and non-desired stable states of the network. Because of the complexity of the general criterion we relax the optimization to the set of non-desired states. We further formulate a stochastic process design based on the stability intervals, which satisfies the optimization criterion and results in enhanced performance of the network. Our experimental simulations confirm the predicted improvement in performance.

INTRODUCTION

Neual networks are a class of non-linear function approximators whose origins date back to work by McCulloch and Pitt [11], Hebb [3], and Rosenblatt [12, 13]. Hopfield defined a single-layer network consisting of interconnected individual perceptrons and modified perceptrons (with sigmoidal nonlinearities) [4, 5, 6]. The basis for network operation as a *content addressable memory* is the Hebbian learning algorithm. The idea is to choose network connections in a way that the energy function associated with the network is minimized for a set of desired network states. Unfortunately, because of its nonlinear character, the network has also exhibited non-desirable, local minima. This has shown to affect the network performance, both in its capacity and its ability to address its content [1, 2, 9, 16]. Several approaches based on *simulated annealing* and other techniques have been proposed that deal with the problem of local minima [7, 8, 10, 15, 17]. In these approaches, an inherent assumption of the final network state (Gibbs) distribution is presumed. The motivation for these assumptions is that the Gibbs distribution provides a mechanism for the characterization of the global minima. In many applications, such as neural networks, 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. A modification of this approach can nonetheless be used to enhance the performance of neural networks.

In his work on the network robustness Schonfeld observed an interesting phenomenon [14]. By applying stochastic perturbation on network thresholds, performance of bipolar Hopfield network was improved. In the present work we extend the study of the effects of stochastic noise on bipolar HNN. Presence of a stochastic process enables us to describe the evolution of the network by using the Markov chains theory. Distribution of the final system states depends on the chain topology, which in turn depends on network topology and the distribution of noise. Given the desired final distribution, it is necessary to determine either the network topology or the noise distribution, or both, so that modified network achieves this final distribution. If the network topology is fixed, the desired final distribution can be achieved by an appropriate stochastic process design.

In this work we introduce a way to stochastically enhance performance of the bipolar Hopfield neural networks (HNNs). We propose a *weighted least-square error density function* optimization criterion for enhancement of the Hopfield neural network performance. The criterion can be related to *stability intervals* associated with the desired and non-desired 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 quasi-optimal HNN based on the minimization of undesirable state final probabilities.

The paper is organized as follows: In Section 2 we define a general stochastic Hopfield neural network, introduce the notion of state transition intervals and its relation to the Markov chain associated with the network. We propose a general optimization *weighted least-square error density function* criterion. In Section 3 we relate the bipolar HNN's state stability to the transition intervals and the network's Markov chain. A criterion that relaxes some constraints of the general optimization criterion is then proposed. Finally, we suggest a way to satisfy this criterion through an appropriate design of the underlying network stochastic process. The approach is verified through a set of experiments detailed in Section 4. This is followed by concluding remarks in Section 5.

STOCHASTIC HOPFIELD NEURAL NETWORKS

Stochastic Hopfield Neural Network (SHNN) as considered in this work is a generalization of the original Hopfield model [4, 5]. It is assumed that the network is a discrete-time, discrete-valued system.

Let $\underline{\mathbf{x}}^{(k)}$ denote a state vector at time k of some system. Let Υ be an elementwise non-decreasing operator such that $\Upsilon\ :\ \mathcal{R} \rightarrow \mathcal{C}(\mathcal{R})$, where $\mathcal{C}(\mathcal{R})$ is a finite subset of $\mathcal{R}$. Let $\underline{\mathbf{u}}^{(k)} \in \mathcal{R}^N$ be a random vector, each component of which has a probability distribution function (pdf) $p_{U_i}(u_i, k)$. Let $\mathbf{C}$ be a matrix of connection weights c_{ij} such that $\mathbf{C} = [c_{ij}]_{N\times N} = [\underline{\mathbf{c}}_1 \underline{\mathbf{c}}_2 \cdots \underline{\mathbf{c}}_N]'$. The Stochastic Synchronous Hopfield Neural Network (SHNN) $\mathcal{H}_s$ model is given by

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

The set of all possible network states is then denoted by $\mathcal{S} = \mathcal{C}(\mathcal{R})^N$. The network is said to be stochastic because of the existence of random vector $\underline{\mathbf{u}}^{(k)}$, the *threshold vector*. The network is also synchronous, which refers to the case where every state of the system is updated at the same time. It is also possible to define a similar model where at any one time instance only one set of states can be updated.

Threshold vector $\underline{\mathbf{u}}^{(k)}$ acts as an input to the system. Control over this vector enables one to (possibly) control the network. Given two consecutive state-vectors of our general SHNN, a set of all threshold vectors that will force the system to make the desired transition is called the *transition set*.

$$\mathcal{T}(\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)}) = \{\underline{\mathbf{u}} \ : \ \underline{\mathbf{x}}^{(k+1)} = \Upsilon(\mathbf{C} \cdot \underline{\mathbf{x}}^{(k)} + \underline{\mathbf{u}})\}. \tag{2}$$

Stable state of the network corresponds to the fixed point of the mapping defined by the network $\underline{\mathbf{x}}_s = \Upsilon(\mathbf{C} \cdot \underline{\mathbf{x}}_s + \underline{\theta})$, for some deterministic threshold $\underline{\theta} \in \mathcal{R}^N$. Clearly, the stable states of the network are determined by the network matrix $\mathbf{C}$ and the network operator Υ. A SHNN can have numerous stable states. All the stable states form the *stable set* of the network, denoted by $\mathcal{S}_s$. In general, it is extremely difficult to determine the stable state set associated with a given network.

It is easy to show that the state $\underline{\mathbf{x}}_s$ in SHNN $\mathcal{H}_s$ is stable iff

$$\underline{\mathbf{0}} \in \mathcal{T}(\underline{\mathbf{x}}_s \rightarrow \underline{\mathbf{x}}_s). \tag{3}$$

Alternatively, we say that the state $\underline{\mathbf{x}}_c$ in SHNN $\mathcal{H}_s$ is *conditionally* stable iff

$$\exists u \in \mathcal{R} \ : \ u \cdot \underline{\mathbf{1}} \in \mathcal{T}(\underline{\mathbf{x}}_c \rightarrow \underline{\mathbf{x}}_c). \tag{4}$$

The term *conditional* is used here to emphasize a possibility of such state being forced to be stable by applying a constant external input (threshold).

Distribution of network states

The evolution of $\mathcal{H}_s$ can be described using a first-order Markov model defined over the space of permissible state of $\mathcal{H}_s$. From 2 it follows that the state transition probability matrix of this Markov chain now becomes

$$p_{ij}(k) = Pr\{\underline{\mathbf{u}}^{(k)} \in \mathcal{T}(\underline{\mathbf{X}}_i \rightarrow \underline{\mathbf{X}}_j)\}. \tag{5}$$

This model represents an inhomogeneous Markov chain. A homogeneous chain arises when $p_{ij}(k) = p_{ij}, \forall k$. This case can be readily achieved when dealing with a *zero-threshold* SHNN (ZSHNN), a network with $\underline{\mathbf{u}}^{(k)} = 0, \ w.p.1$. The following theorem is related to the class of ZSHNN.

Let $\mathcal{H}_s$ be a zero-threshold Synchronous Hopfield Neural Network (ZSHNN), and let $\mathcal{S}_s$ be its stable state set given as $\mathcal{S}_s = \{\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, ..., \underline{\mathbf{X}}_K\}$.

Theorem 1 *Each stable states of a ZSHNN forms a single class of essential indices of the Markov chain associated with ZSHNN. Unstable states of ZSHNN correspond to the chain's inessential indices. Limiting state distribution vector of a ZSHNN exists and has the form*

$$\underline{\mathbf{p}}^{(\infty)} = \begin{bmatrix} \underline{\mathbf{p}}_s^{(\infty)} \\ \underline{\mathbf{0}} \end{bmatrix}, \tag{6}$$

where $\underline{\mathbf{p}}_s^{(\infty)}$ is a non-zero limiting state distribution vector corresponding to the stable states of $\mathcal{S}$.

Proof: The proof follows readily from the definition of essential and inessential states of a Markov chain. Following that, it is also known that the limiting distribution of essential (stable) states exists and is non-zero. Similarly, the limiting distribution vector of the unstable (inessential) states is zero. □ The above results establish a relation between the stable state set and the convergent states. They also classify all stable states as essential and all other states as inessential. The importance of this result will become more clear as we develop the notion of network optimization.

"Bad" and "Good" States

The stable state set is determined by the choice of the network matrix, if the network network function is fixed. Given a set of desired stable states $\mathcal{S}_t = \{\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, ..., \underline{\mathbf{X}}_M\}$, a procedure of finding the connection weights matrix $\mathbf{C}$ such that $\underline{\mathbf{X}} \in \mathcal{S}_t \Rightarrow \underline{\mathbf{X}} \in \mathcal{S}_s$ is called the *learning procedure.*

It is well known that all the learning procedures associated with HNN exhibit one property: designed stable states form, in the best case, a subset of the network's stable state set. It is also known that as the number of desired, "to-be-stable" states increases, learning algorithms approach saturation [1, 2, 4, 5, 6, 9]. We consider only those cases where the learning algorithm saturation point is not reached. Such SHNN is then said to be *proper*. In this case all the desired stable states are said to be *"good"* and all the undesired ones are said to be *"bad"*. We will use this notion in defining the optimality criterion for the network.

Optimal Network

Because of the existence of so called "bad" (stable) states, and the fact that the network converges to one of the states from the stable state set ("good" or "bad"), it is obvious that the network is not performing its task flawlessly. Starting at some initial state, a ZSHNN will sometimes converge to an undesired "bad" state. However, it is possible to control the network by applying some external input (threshold). If the input is designed such that the limiting distribution of the "bad" states becomes zero, then the "bad" states can be altogether avoided.

In general we say that a SHNN $\mathcal{H}_s^*(\Upsilon, \mathbf{C}, p_U^*)$ is weighted (W) least-square error density function (WLSED($\mathbf{W}$)) optimal iff

$$p_U^* = \arg\min_{p_U \in \mathcal{P}} \left(\tilde{\underline{\mathbf{p}}}^{(\infty)\prime} \mathbf{W} \tilde{\underline{\mathbf{p}}}^{(\infty)} \right), \tag{7}$$

where

$$\tilde{\underline{\mathbf{p}}}^{(\infty)} = \left[\begin{array}{c} \underline{\mathbf{p}}_{t,d}^{(\infty)} \\ \underline{\mathbf{0}} \end{array} \right] - \underline{\mathbf{p}}^{(\infty)}, \tag{8}$$

$\underline{\mathbf{p}}^{(\infty)}$ is the limiting state distribution of $\mathcal{H}_s(\Upsilon, \mathbf{C}, p_U)$, and $\mathbf{W}$ is a $K \times K$ matrix of weights. More importantly, to only focus on the "bad" states we can use

$$\mathbf{W} = \mathbf{W}_0 = \mathrm{diag}(0, ..., 0_M, 1_{M+1}, 1_{M+2}, ...). \tag{9}$$

It is important to notice that final distribution of the "good" states is not addressed in WLSED($\mathbf{W}_0$) case, as opposed to the general WLSED($\mathbf{W}$) case.

Design of a network that satisfies this criterion is not trivial. Nontheless, it is easy to see that a proper SHNN can be made WLSED($\mathbf{W}_0$) optimal if pdf p_U^* is chosen such that the states of the "bad"-state set are made inessential, and all the other states belong to their ZSHNN classes of states. These conditions might seem to be fairly obvious, however, it is actually very difficult to satisfy them because of the cardinality of the sets involved, especially the inessential state set. To deal with this problem more efficiently, we will restrict our discussion to a subclass of SHNN, known as *bipolar HNNs*.

BIPOLAR STOCHASTIC HOPFIELD NEURAL NETWORKS

Bipolar Hopfield neural networks along with binary Hopfield neural networks are two of the most widely used models of a single-layer neural network. Binary network model appeared in the original Rosenblatt and Hopfield papers [4, 5, 13, 12]. The model can be viewed as a special case of SHNN where υ assumes the form of the binary hard threshold function and threshold $\underline{\mathbf{u}}_k$ is deterministic. Bipolar Hopfield neural network is a modification of the original binary network with υ being the bipolar threshold function:

$$\upsilon(\bullet) = h(\bullet) = \begin{cases} 1 & , \quad \bullet > 0; \\ -1 & , \quad \bullet < 0. \end{cases} . \tag{10}$$

Here υ is a scalar version of the elementwise operator Υ and h is the bipolar hard limiter[1]. W.L.O.G., we can also assume that all the vectors in the state space of this network belong to the N-dimensional bipolar vector space $\mathcal{S} \subseteq \{-1, 1\}^N$. Since h is a subset of υ, all the assertions from the previous section also hold for the bipolar SHNN.

The original bipolar HNN was considered in its asynchronous version. However, we will constrain our discussion to the synchronous case. The latter is easily described by using matrix notation in accordance with the theoretical developments of the previous section.

Within the scope of BSHNN several results can be related to the transition sets[2] The most important one provides us with an alternative definition of the transition sets:

Proposition 1 *Element $\underline{\mathbf{u}}$ belongs to the transition set $\mathcal{T}(\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)})$ of bipolar SHNN iff*

$$\mathrm{diag}(x_1^{k+1}, x_2^{k+1}, ..., x_N^{k+1}) \cdot \underline{\mathbf{u}} \; > \; -\mathrm{diag}(x_1^k, x_2^k, ..., x_N^k) \cdot \mathbf{C}\underline{\mathbf{x}}^{(k)}. \tag{11}$$

It is interesting to know if and when an intersection of the transition set with the main diagonal of the space of threshold vectors occurs. The main diagonal simply corresponds to a subset of threshold vectors with all identical components. The following theorem defines those conditions:

Let

$$\mathcal{I}^+ = \{u \; : \; u = -\underline{\mathbf{c}}_i' \underline{\mathbf{x}}^{(k)} \; , \; x_i^{(k+1)} > 0\}, \quad \text{and} \tag{12}$$

$$\mathcal{I}^- = \{u \; : \; u = -\underline{\mathbf{c}}_j' \underline{\mathbf{x}}^{(k)} \; , \; x_j^{(k+1)} < 0\} \tag{13}$$

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

$$i_{min} = \sup \mathcal{I}^+, \quad \text{and} \tag{14}$$

$$i_{max} = \inf \mathcal{I}^-. \tag{15}$$

[1] Case of $h(0)$ is left undefined here. However, it is assumed that in the case of $\underline{\mathbf{c}}_i{}'\underline{\mathbf{x}}^{(k)} + u_i^{(k)} = 0$ we have $x_i^{(k+1)} = x_i^{(k)}$.

[2] We have to provide most of those results without the acompanying proofs, for the sake of bravity.

Theorem 2

$$u \cdot \underline{\mathbf{1}} \in \mathcal{T}(\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)}) \tag{16}$$

iff

$$i_{min} < u < i_{max}. \tag{17}$$

It is now also easy to prove the following two corollaries that establish conditions for (conditional) stability.

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

$$i_{min} = \sup\{u \ : \ u = -\underline{\mathbf{c}}_i'\underline{\mathbf{x}}_c \ , \ \underline{\mathbf{x}}_{c,i} > 0\} \tag{18}$$

$$i_{max} = \inf\{u \ : \ u = -\underline{\mathbf{c}}_j'\underline{\mathbf{x}}_c \ , \ \underline{\mathbf{x}}_{c,j} < 0\}. \tag{19}$$

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

Corollary 2 *State $\underline{\mathbf{x}}_s$ in bipolar SHNN $\mathcal{H}_s$ is stable iff it is conditionally stable and*

$$0 \in (i_{min}, i_{max}). \tag{20}$$

The conditional stability interval associated with stable state $\underline{\mathbf{x}}_s$ is called the stability interval associated with $\underline{\mathbf{x}}_s$.

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

Optimal Network

Because all the properties of a general SHNN also hold for BSHNN, the optimization procedure remains defined the same way as it is in the case of SHNN.

Proposition 2 *Let $\mathcal{H}_s$ be the proper bipolar zero-threshold SHNN, with the "good"-state set $\mathcal{S}_t$ of M elements. Let $\mathcal{H}_s^q$ be a bipolar SHNN, such that the threshold vector is the random variable*

$$\underline{\mathbf{u}}^{(k)} = u\underline{\mathbf{1}} \ : \ u \sim p_U(u^{(k)}). \tag{21}$$

The network $\mathcal{H}_s^q$ is WLSED($\mathbf{W}_0$) optimal if $p_U(u)$ is such that

$$Pr\{u \in (i_{min}, i_{max})\} = 1 \quad , \quad \underline{\mathbf{X}} \in \mathcal{S}_t \tag{22}$$

$$Pr\{u \in (i_{min}, i_{max})\} = 0 \quad , \quad \underline{\mathbf{X}} \in \mathcal{S}_b \tag{23}$$

$$Pr\{\underline{\mathbf{u}}^{(k)} \in \mathcal{T}(\underline{\mathbf{X}} \rightarrow \underline{\mathbf{X}})\} = 0 \quad , \quad \text{otherwise}. \tag{24}$$

Proposition 2 relates robustness of the "good" and "bad" stable states to a possibility of forcing the "bad" states to become inessential, while keeping the "good" ones essential. However, satisfying all three conditions at the same time is not a trivial task. This is particularly true for the third condition which guarantees instability of the originally non-stable states.

We now suggest a heuristic approach that could lead to WLSED($\mathbf{W}_0$) optimal network.

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

$$Pr\{u \in (i_{min}, i_{max})\} = 1 \quad , \quad \underline{\mathbf{X}} \in \mathcal{S}_t \tag{25}$$
$$Pr\{u \in (i_{min}, i_{max})\} < 1 \quad , \quad \underline{\mathbf{X}} \in \mathcal{S}_b. \tag{26}$$

Then, the network is quasi-WLSED($\mathbf{W}_0$) optimal.

Condition in Equation 25 is the same as the condition in Equation 22. Equation 26, however, poses a weaker condition on $p_U(u)$ than the one in Equation 23 resulting in possibly non-inessential "bad" states. Finally, what distinguishes Heuristic 1 from Proposition 2 is the non-existence of the condition given by Equation 24. This actually means that there is no guarantee that some originally non-stable states would not become essential. A possibility, thus, exists for some conditionally stable states to become stable. Still, experimental results show that in large majority of cases the first two conditions tend to be sufficient.

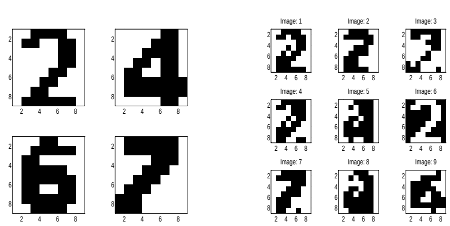
As an example of a pdf that satisfies the conditions of Heuristic 1 we suggest a uniformly distributed stochastic process, given as follows

Example 1 *Let $p_U(u)$ be such that $u \sim \mathcal{U}(0, \sigma/\sqrt{3})$, where $\max_{\underline{\mathbf{X}} \in \mathcal{S}_b}\{|i_{min}|, |i_{max}|\} \leq \sigma$ and $\min_{\underline{\mathbf{X}} \in \mathcal{S}_t}\{|i_{min}|, |i_{max}|\} \geq \sigma$.*

From the fact that u has a uniform distribution, with pdf equal to zero for $u > \pm\sigma$, and Corollary 2, it follows that the conditions for Heuristic 1 hold. Therefore, the network is quasi-WLSED($\mathbf{W}_0$) optimal. □

EXPERIMENTS

A Hebbian Binary Synchronous Hopfield Neural Network was designed to perform bipolar image classification under the following setup: Network dimension is chosen to be $N = 64$; network is designed to classify $M = 4$ desired images, depicted in Figure 1(a); Initial states are obtained from the desired images by reversing the state of individual elements with probability $p_{init} = 35\%$. A set of 200 different initial states was generated; Pdf associated with thresholds is chosen to be zero-mean uniform with variance in the range determined by the stability intervals of stable states (see Example 1); Number of network iterations is set to 40.



(a) Good (b) Bad

Figure 1: "Good" (a) and "bad" (b) stable states.

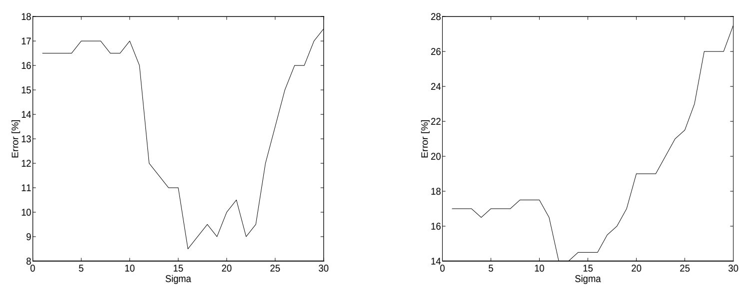
Two different error measures were used for the evaluation of the performance of the HBSHNN: (1) "Exact" error measure. This measure requires exact matching of the final state with the exampler used to obtain the initial state of the network. (2) “Bad-Good” error measure. According to this measure, an error occurs whenever the final state of the network differs from one of the four images. If the final state is equal to *any* of the desired four images, the network is said to be successful in classifying the initial state.

First, a zero-threshold HBSHNN was subjected to 200 initial states. The network converged in at most 4 iterations. A set $\mathcal{S}_s$ of stable states of the network was obtained. Based on the set of four images, $\mathcal{S}_t$, and the set of stable states, a set of “bad” states was determined. Images corresponding to the set of “bad” states are depicted in Figure 1(b). For each of the stable states, a corresponding stability interval was calculated. Results are presented in Table 1.

Table 1: Stable states and corresponding stability intervals (images are listed in the left column).

Image	Interval	Image	Interval	Image	Interval	Image	Interval	Image	Interval
Exm. 1	(-30,38)	1	(-10,14)	5	(-10,10)	9	(-14,14)	13	(-14,6)
Exm. 2	(-38,38)	2	(-14,14)	6	(-30,38)	10	(-30,30)	14	(-38,38)
Exm. 3	(-38,38)	3	(-6,6)	7	(-10,10)	11	(-14,14)	15	(-6,6)
Exm. 4	(-34,34)	4	(-10,10)	8	(-14,14)	12	(-10,10)	16	(-14,14)

Next, a stochastic version of the original network was applied to the same initial state set. Variance of the pdf was changed in steps of one in zero to 30 range. For each variance the network performed a fixed number of iterations. Results of classification based on the two different error measures proposed are shown in Figures 2(a) and 2(b). In addition, the relative frequencies of occurrence of images are shown in Figure 3.



(a) Bad-Good

(b) Exact

Figure 2: Classification error for the “Bad-Good” (a) and “Exact” (b) error measures.

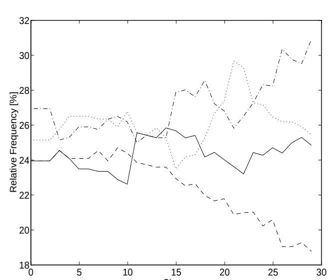


Figure 3: Relative frequencies of occurrence for “Good” states. Solid, dotted, dash-dotted, and dashed lines correspond to images ’2’,’4’, ’6’, and ’7’ respectively.

In the case of “Bad-Good” error measure, a drastic improvement in performance occurred for noise levels between 14 and 25 (see Figure 2(a)). The lower bound corresponds to the highest stability interval limit of the states in the “bad” state set. However, we should note that this does not include reverse images of the original images. The upper bound is slightly lower than the lowest limit of states in the exampler set. This was expected, following the analysis of quasi-optimal BSHNN. With noise levels between 14 and 25, all “bad” states were forced to become inessential, except for the reverse images. All the original images and their reverse images remained essential. The “exact” error measure shows an improvement in a narrower interval starting at about 10 and ending at 18 (see Figure 2(b).) This can be explained by the fact that the quasi-optimization procedure does not have full control over the final distribution. As the noise level increases over 18, it is clear from Figure 3 that some of the desired images become dominant. However, the overall convergence to the elements of exampler set still shows good results. Simulations also showed that for the chosen range of noise variance, the number of iterations was sufficiently high to guarantee convergence of the associated Markov chain. The same set of simulations was conducted with 60 network iterations, and only an insignificant decrease in recognition error was noticed for higher noise variances.

CONCLUSION

Hopfield neural networks (HNN) are a class of densely connected single layer nonlinear networks of perceptrons often representing a basis for content addressable memory systems. The basic principle behind HNN lies in defining the energy function associated with the network that has its minima over some predefined set of states. Unfortunately, because of its nonlinearity, a number of undesirable local minima also occur. This has shown to adversely affect the network’s performance [1, 2, 9, 16].

In this work we studied the effects of stochastic noise on the performance of bipolar HNN. Presence of a stochastic process enables us to describe the evolution of the network using the theory of Markov chains. Distribution of the final network states depends on the Markov chain topology, which is in turn determined by the topology of the network and the underlying stochastic process. If the network topology is fixed, the desired final distribution can be reached by modulating the network’s stochastic process. Guided by the desired final distribution we propose a general L^2 norm *weighted least-square error density function* optimization criterion for the enhancement of Hopfield neural network performance. The criterion can also be viewed in terms of *stability intervals* associated with the desired and non-desired stable states of the network. Because of the complexity of the general criterion we restrict the optimization to the set of non-desired 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.

Nevertheless, the general optimization procedure, as proposed in Section 2, still remains an interesting open problem. Future work on the solution to this problem would be of utmost importance, since it could enable realization of HNN with arbitrary final state distribution. Extension of this approach to a class of superior neural networks (such as multi-layer, back-propagation neural networks) is another important further endeavor. Considering the growing successful applications of such neural networks, yet keeping in mind that they exhibit the same problems addressed in this work, the need for optimization becomes increasingly attractive. Further abstraction of neural networks and stochastic noise enhancement 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] S. Amari, "Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements," **IEEE Transactions on Computers**, vol. C-21, no. 11, pp. 1197–1206, November 1972.

[2] S. Amari and K. Maginu, "Statistical Neurodynamics of Associative Memory," **Neural Networks**, vol. 1, pp. 63–73, 1988.

[3] D. Hebb, **The Organization of Behavior**, New York, New York: John Wiley and Sons, 1949.

[4] J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," **Proceedings of the National Academy of Science USA**, vol. 79, pp. 2554–2558, April 1982.

[5] J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," **Proceedings of the National Academy of Science USA**, vol. 81, pp. 3088–3092, May 1984.

[6] J. Hopfield and D. Tank, "Computing with neural circuits: A model," **Science**, vol. 233, pp. 625–633, 1986.

[7] I. Arizono, A. Yamamoto, and H. Ohta, "Scheduling for minimizing total actual flow time by neural networks," **International Journal of Production Research**, vol. 30, no. 3, pp. 503–511, March 1992.

[8] B. Lee and B. Sheu, "Modified Hopfield Neural Networks for Retrieving the Optimal Solution," **IEEE Transactions on Neural Networks**, vol. 2, no. 1, pp. 137–142, January 1991.

[9] R. Lippmann, "An Introduction to Computing with Neural Nets," **IEEE Acoustics, Speech and Signal Processing Magazine**, pp. 4–22, April 1987.

[10] M. Lu, Y. Zhan, and G. Mu, "Bipolar Optical Neural Network with Adaptive Threshold," **Optik**, vol. 91, no. 4, pp. 178–182, 1992.

[11] W. McCulloch and W. Pitts, "A logical calculus of the ideas imminent in nervous activity," **Bulletin of Mathematical Biophysics**, vol. 5, pp. 115–133, 1943.

[12] R. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," **Psychological Review**, vol. 65, pp. 386–408, 1958.

[13] R. Rosenblatt, **Principles of Neurodynamics**, New York, New York: Spartan Books, 1959.

[14] D. Schonfeld, "On the Hysteresis and Robustness of Hopfiled Neural Networks," **IEEE Transactions on Circuits and Systems – II : Analog and Digital Signal Processing**, vol. 2, pp. 745–748, November 1993.

[15] S.L. Lin, J.M. Wu, and C.Y. Liou, "Annealing by Perturbing Synapses," **IEICE Transactions on Information and Systems**, vol. E75-D, no. 2, pp. 210–218, March 1992.

[16] S. Venkatesh and D. Psatis, "On Reliable Computations with Formal Neurons," **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. 14, no. 1, pp. 87–91, January 1992.

[17] E. Wong, "Stochastic Neural Networks," **Algorithmica**, pp. 466–478, June 1991.