Noise Injection into Inputs in Sparsely Connected Hopfield and Winner-Take-All Neural Networks

Lipo Wang

***Abstract*—In this paper, we show that noise injection into inputs in *unsupervised learning* neural networks *does not improve* their performance as it does in *supervised learning* neural networks. Specifically, we show that training noise degrades the classification ability of a sparsely connected version of the Hopfield neural network, whereas the performance of a sparsely connected winner-take-all neural network does not depend on the injected training noise.**

I. Introduction

One of the most useful properties of artificial neural networks (NN's) is their ability to generalize; for instance, to classify patterns that have not been presented during training. It has been established recently that the performance of *supervised learning* NN's can be improved by introducing noise into the training patterns, which is called *noise injection*. In the case of the back-propagation NN, experimental evidences [1]–[6] appeared first in the literature, which was followed by extensive theoretical analysis [7], [8]. Similar conclusions have been reached for Hopfield-like recurrent NN's that use perceptron learning algorithms [9]–[12].

The work mentioned above relates to *supervised learning* NN's. Will noise injection into *unsupervised learning* NN's also have a constructive impact on their performance? It is the purpose of the present Correspondence to provide an answer to this question. Specifically, we will study the effects of training noise on two popular types of unsupervised learning NN's: a sparsely connected version [13] of the Hopfield NN [14] (Section II) and a sparsely connected winner-take-all NN that we shall propose (Section III). We will show that noise injection in these unsupervised learning NN's does not improve their performance as it does in supervised learning NN's. Thus noise injection should not be adopted when unsupervised learning NN's are used in practical applications.

II. Noise Injection into Inputs in a Sparsely-Connected Hopfield Neural Network

The Hopfield NN [14] consists of N binary neurons that are either active or quiescent, i.e., $S_i(t) = \pm 1$, where $S_i(t)$ is the state of neuron i at time t and $i = 1, \ldots, N$. Each neuron receives signals from other neurons in the network, and the signals are affected by synaptic weights T_{ij}. The neuron then either fires if the total input h_i exceeds a threshold, or remains quiescent otherwise. The Hopfield NN, though crude compared to biological neural systems, has been widely studied and applied (e.g., [15]–[18]).

Specifically, the neurons update their states according to

$$S_i(t + \Delta t) = \mathrm{sign}[h_i(t)] \tag{1}$$

where $\mathrm{sign}(x) = -1$ for negative x and $\mathrm{sign}(x) = +1$ for positive x; and the total input signal h_i for neuron i at time t is given by

$$h_i(t) = \sum_{j=1}^{N} T_{ij} S_j(t) + \eta_i \tag{2}$$

where η_i represents the noise in neuronal signals due to probabilistic releases of synaptic vesicles that account for the spontaneous firing of a neuron [19]. We assume η_i is Gaussian-distributed with an average zero and a standard deviation σ_o.

The original Hopfield model is fully connected and the synapses are formed in the spirit of Hebbian learning [20], an *unsupervised learning* rule

$$T_{ij}^{H} = \sum_{\mu=1}^{p} S_i^{\mu} S_j^{\mu} \tag{3}$$

where $\vec{S}^{\mu}$ is the μth stored pattern, and p is the number of patterns stored. For our present investigation, we also assume that patterns $\vec{S}^{\mu}$ are randomly generated.

In a sparsely connected version of the Hopfield NN [13], the synapses are randomly cut off to model the asymmetric and sparse connectivity in biological NN's. We choose to study the sparsely connected Hopfield NN here because it is possible to derive an exact dynamical equation for the network's evolution from its initial condition, i.e., an analytical description on the dynamical response of the network to an input pattern.

We generalize the Hebbian rule given by (3) to incorporate noise in the training patterns (noise injection)

$$T_{ij}' = \frac{C_{ij}}{q_1 q_2} \sum_{\nu_1=1}^{q_1} \sum_{\nu_2=1}^{q_2} \sum_{\mu=1}^{p} S_i^{\mu\nu 1} S_j^{\mu\nu 2} \tag{4}$$

where $\vec{S}^{\mu\nu_1}$ and $\vec{S}^{\mu\nu_2}$ are the noisy training patterns. Equation (4) reduces to the original Hopfield prescription equation (3) if $\vec{S}^{\mu\nu_1} = \vec{S}^{\mu\nu_2} = \vec{S}^{\mu}$ for all μ, ν_1, and ν_2. The random numbers C_{ij} assume 1 with probability C/N and 0 with probability $(1 - C/N)$, where $c < N$. Thus only C of N synapses remain for each neuron. The noisy training patterns are

$$S_i^{\mu\nu a} = S_i^{\mu} + \delta_i^{\mu\nu a}, \quad a = 1, 2 \tag{5}$$

where we assume that the difference between the noisy pattern and the "clean" pattern, i.e., $\delta_i^{\mu\nu a}$, which may take values $0, \pm 2$, is a random number with a zero average and a standard deviation δ_a, i.e.,

$$\left\langle \delta_i^{\mu\nu a} \right\rangle = 0 \tag{6}$$

and

$$\left\langle \left(\delta_i^{\mu\nu a} - \left\langle \delta_i^{\mu\nu a} \right\rangle \right)^2 \right\rangle = \left\langle \left(\delta_i^{\mu\nu a} \right)^2 \right\rangle = \delta_a^2 \tag{7}$$

where $a = 1, 2$. For instance, $\delta_1^2 = 1$ and $\delta_2 = 0$ indicate that $\vec{S}^{\mu\nu_2}$ is the same as the corresponding clean pattern $\vec{S}^{\mu}$ and $\delta_1^2/4 = 25\%$ of the bits in $\vec{S}^{\mu\nu_1}$ are randomly chosen and flipped, since each bit flipped gives a $(\delta_i^{\mu\nu a})^2 = 4$. Thereafter we assume that $\delta_1^2 < 0.5$ and $\delta_2^2 < 0.5$ so that despite the noise in the training patterns, $\vec{S}^{\mu\nu'}$ is more similar to $\vec{S}^{\mu\nu}$ than any $\vec{S}^{\mu'\nu''}$ is to $\vec{S}^{\mu\nu}$ for any set of (ν, ν', ν'') if $\mu \neq \mu'$. Thus the noisy training pattern form broadened bands around the clean patterns $\vec{S}^{\mu}$, with band gaps greater than $0.25N$ bits.

Manuscript received June 30, 1995; revised August 29, 1996.

The author is with the School of Computing and Mathematics, Deakin University, Clayton, Victoria 3168, Australia.

Publisher Item Identifier S 1083-4419(97)05207-2.

We now derive and analyze the dynamical equation that describes how the sparsely connected Hopfield NN evolves from its initial condition, in other words, how the network responds to an input pattern, after the network is trained with noisy patterns and is then presented with an input for recognition. Let us evaluate the similarity between the state of the network and a stored pattern $\vec{S}^{\mu}$, that is, the overlap or the normalized dot product

$$m^{\mu}(t) = \frac{1}{N}\vec{S}^{\mu} \cdot \vec{S}(t) = \frac{1}{N}\sum_{j=1}^{N} S_j^{\mu} S_j(t). \tag{8}$$

Suppose the input pattern, which is represented by the initial state of the network, is most similar to one of the stored pattern $\vec{S}^1$, i.e., $m^1(0) = \max\{m^{\mu}(0) \mid \mu = 1, 2, \ldots, p\}$. Substituting (4) and (5) into (2) and isolating the terms related to $\vec{S}^1$, we obtain

$$\begin{aligned} h_i(t) = {} & C m^1(t) S_i^1 + \sum_j \sum_{\mu=2}^{p} S_i^{\mu} S_j^{\mu} S_j(t) \\ & + \frac{1}{q_2}\sum_j \sum_{\mu=1}^{p} \sum_{\nu_2=1}^{q_2} S_i^{\mu}\delta_j^{\mu\nu_2} S_j(t) \\ & + \frac{1}{q_1}\sum_j \sum_{\mu=1}^{p} \sum_{\nu_1=1}^{q_1} \delta_i^{\mu\nu_1} S_j^{\mu} S_j(t) \\ & + \frac{1}{q_1 q_2}\sum_j \sum_{\mu=1}^{p} \sum_{\nu_1=1}^{q_1} \sum_{\nu_2=1}^{q_2} \delta_i^{\mu\nu_1}\delta_j^{\mu\nu_2} S_j(t) + \eta_i \end{aligned} \tag{9}$$

where $\sum_j$ covers the remaining synapses after the random synaptic disruption. The first term [13] in the right hand side of (9) represents the *signal*, which drives the system toward the memory state $\vec{S}^1$, and the rest represents noise that interferes with this converging process. Although the noise terms are random, the correlations among them prevents us from getting an exact solution when the network is fully connected. The disruption process eliminates correlations among neurons and noise terms become *independent* random variables [13]. Thus the sum of these terms are Gaussian-distributed when N is large, according to the central limit theorem.

The overlap between the state of the network and the attracting pattern $\vec{S}^1$ at the next time step $t+1$ can now be easily obtained by substituting the updating (1) into the definition for overlap given by (8). For parallel updating, i.e., when the states of all neurons are updated simultaneously, the dynamics of the network is governed by

$$m(t+1) = \mathrm{erf}\left[\frac{m(t)}{\sqrt{2}\sigma}\right] \tag{10}$$

where $m(t)$ is a statistical average of the overlap $m^1(t)$. The total standard deviation of the noise terms is

$$\sigma^2 \equiv \frac{(p-1)}{C} + \frac{p}{C}\left(\frac{\delta_1^2}{q_1} + \frac{\delta_2^2}{q_2} + \frac{\delta_1^2\delta_2^2}{q_1 q_2}\right) + \left(\frac{\sigma_o}{C}\right)^2 \tag{11}$$

which includes the effects of random synaptic disruption, interference between the stored patterns, noise in the training patterns, and signal transmission noise. In (10), $\mathrm{erf}(y) = (2/\sqrt{\pi})\int_0^y e^{-x^2}dx$ is the standard error function.

Starting from any initial condition $(t = 0)$ the overlap thereafter can be calculated from (10) iteratively. After a few iterations, the network reaches equilibrium characterized by the "fixed point" of the dynamical equation, m, or the "final overlap." Fixed points of the system can be obtained from (10) by letting $m(t+1) = m(t) = m$, as a function of the number of stored patterns p and the results are given in Fig. 1 for various choices of training noise. For simplicity, in Fig. 1 we neglect the background noise or the spontaneous neuronal activities, i.e., $\sigma_o = 0$. Fig. 1 shows that the final overlap m, which measures the networks ability of converging to the correct memory pattern, decreases as noise in training patterns δ_H increases, where

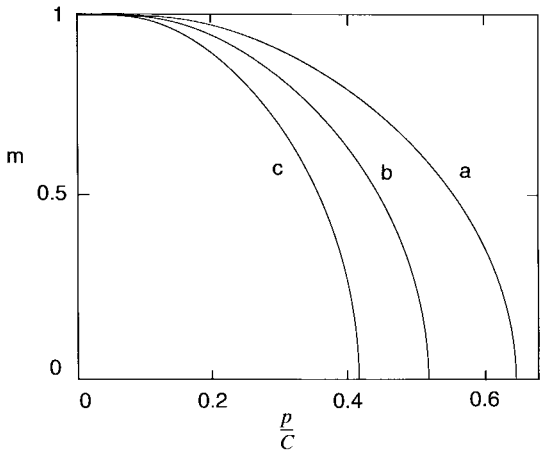

Fig. 1. The average final overlap m (fixed points) as a function of the ratio between the number of stored patterns p and the average number of synapses per neuron C, according to dynamical equations (10)–(12) for the sparsely connected Hopfield neural network. (a) $\delta_H = 0$; (b) $\delta_H^2 = \delta_1^2 = 0.25$, $\delta_2 = 0$; and (c) $\delta_H^2 = 0.563$.

$$\delta_H^2 \equiv \frac{\delta_1^2}{q_1} + \frac{\delta_2^2}{q_2} + \frac{\delta_1^2\delta_2^2}{q_1 q_2} \tag{12}$$

measures the combined effects of the training noise. The performance is best when δ_H vanishes, or both δ_1 and δ_2 vanish. The total noise deviation is now simply written as $\sigma = \sqrt{p/C}\sqrt{1+\delta_H^2}$, since usually $p \gg 1$.

None-zero fixed points exist if $\sigma < \sigma_c \equiv \sqrt{2/\pi}$. In particular, $m = 1$ for $\sigma = 0$, and m decreases monotonously as σ increases. For $\sigma \geq \sigma_c$, $m = 0$, so the network loses its ability to perform as a classifier for $\sigma \geq \sigma_c$. The maximum number of patterns that can be stored in and retrieved from the system (the memory capacity of the network), P, is obtained by letting $\sigma = \sqrt{2/\pi}$

$$P = \frac{2C}{\pi\left(1+\delta_H^2\right)}, \quad \text{or} \quad \frac{P}{C} = \frac{0.637}{1+\delta_H^2}. \tag{13}$$

It is clear from (12) and (13) that in the absence of noise in training patterns, i.e., $\delta_1^2 = \delta_2^2 = \delta_H^2 = 0$, the memory capacity P reaches maximum, which is $0.637C$, and decreases monotonically as δ_H^2 increases. For instance, if $\delta_1^2 = \delta_2^2 = 0.25$ and $q_1 = q_2 = 1$, we have $\delta_H^2 = 0.563$, the memory capacity is $0.407C$, according to (13)—a 36% reduction. The basis of attraction of any non-negative fixed point is always $0 < m(t=0) \leq 1$, which is independent of training noise δ_H.

III. Noise Injection into a Sparsely Connected Winner-Take-All Neural Network

In the section, we study another unsupervised NN-a NN that uses competitive learning. Many authors have discussed competitive learning [21]–[27]. For the present analysis we shall propose a new sparsely connected winner-take-all NN instead of using an existing competitive learning NN, since our sparsely connected winner-take-all NN is easier to study analytically than others and yet it captures the essence of competitive learning.

Our sparsely connected winner-take-all NN consists of p binary neurons and N input nodes. Each neuron is connected to a randomly chosen group of C input nodes ($C < N$). Thus the total input to neuron i is

$$h_i(t) = \sum_j T_{ij} I_j(t) \tag{14}$$

where I_j denotes the signal at input node j. Only the neuron with the largest total input wins the competition and responds to the input pattern. Equation (14) implies that the neuron whose synapses are most similar to the input pattern wins the competition.

The synaptic updating algorithm for the present system is a modified version of the algorithm first proposed by von der Malsburg [22] and used in the Rumelhart–Zipser model [26]. At the τ_ith modification of the synapses of neuron i, we let the synapses give up some portion, i.e., $1/\tau_i$, of its weights and these weights are then distributed among the synapses in proportion to the training input pattern. Following Grossberg [28], we do not normalize the sum of the synapses to 1. Hence the updated synapses are

$$\begin{aligned} T_{ij}(\tau_i) &= \frac{1}{\tau_i} \sum_{\tau'=1}^{\tau_i} I_{jk}(\tau') \\ &= \left(1 - \frac{1}{\tau_i}\right) T_{ij} + \left(\frac{1}{\tau_i}\right) I_{jk}(\tau) \end{aligned} \tag{15}$$

which implies that the synapse vector is an *overall* average of the contributing training patterns [29] and all contributing training patterns—independent of the temporal order in which the training patterns are presented—contribute *equally* to learning.

After the network is trained with the noisy patterns given by (5), the synapses of neuron μ are, according (15)

$$T_{\mu j} = S_j^{\mu} + (q_1 + q_2)^{-1} \sum_{\nu=1}^{q_1+q_2} \left(\delta_j^{\mu\nu 1} + \delta_j^{\mu\nu 2}\right). \tag{16}$$

Thus the synapses of the participating neurons "cluster" around the clean patterns $\{\vec{S}^{\mu}\}$, the differences being Gaussian-distributed and having standard deviations

$$\delta_D^2 = \delta^2/(q_1 + q_2). \tag{17}$$

Since $(q_1 + q_2) \geq 1$ and we have assumed that $\delta^2 < 0.5$, we have $\delta_D^2 < 0.5$. Thus the cluster exemplars stored in the synapses differ from each other by at least 25% of the N bits and the present sparsely connected winner-take-all NN of p neurons can successfully classify p classes of randomly generated patterns after being trained with noisy patterns. It is also clear from the above analysis that the performance of the present system does not depend on noise injection into the training inputs.

IV. Summary and Conclusion

In summary, effects of training noise on the performance of a sparsely connected Hopfield NN and a sparsely connected winner-take-all NN are discussed. By deriving and solving an exact dynamical equation, we show that the training noise degrades the classification abilities of the sparsely connected Hopfield NN. We have also proposed a simple sparsely connected winner-take-all NN. By analytically calculating the synaptic efficacies after training with noisy patterns, we show that the performance of the system does not depend on the training noise. We thus conclude that noise injection into the training inputs does not improve the performances of *unsupervised learning* NN's such as the typical ones presented here, in contrast to *supervised learning* NN's.

References

[1] D. C. Plaut, S. J. Nowlan, and G. E. Hinton, "Experiments on learning by back-propagation," Tech. Rep. CMU-CS-86-126, 1986.

[2] S. M. Peeling, R. K. Moore, and M. J. Tomlinson, "The multi-layer perceptron as a tool for speech pattern processing research," in *Proc. IOA Autumn Conf. Speech and Hearing*, 1986.

[3] J. Sietsma and R. J. F. Dow, "Neural network pruning–-Why and how," in *Proc. IEEE Int. Conf. Neural Networks*, 1988, vol. I, pp. 325–333.

[4] A. Lindon and J. Kindermann, "Inversion of multilayer nets," in *Proc. Int. Joint Conf. Neural Networks*, 1989, pp. 425–430.

[5] M. M. Moya and L. D. Hostetler, "One-class generalization in second-order backpropagation networks for image classification," in *Proc. Int. Joint Conf. Neural Network*, 1990, pp. 221–224.

[6] J. Sietsma and R. J. F. Dow, "Creating artificial neural networks that generalize," *Neural Networks*, vol. 4, pp. 67–79, 1991.

[7] K. Matsuoka, "Noise injection into inputs in back-propagation learning," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 436–440, May/June 1992.

[8] L. Holmström and P. Koistinen, "Using additive noise in back-propagation training," *IEEE Trans. Neural Networks*, vol. 3, no. 1, pp. 24–38, 1992.

[9] E. Gardner, N. Stroud, and D. J. Wallace, "Training with noise: Application to word and text storage," in *Neural Computation*, R. Eckmiller and C. V. D. Malsburg, Eds. New York: Springer-Verlag, 1987, pp. 251–260.

[10] ——, "Training with noise and the storage of correlated patterns in a neural network model," *J. Phys. A, Math. Gen.*, vol. 22, pp. 2019–2030, 1989.

[11] K. Y. M. Wong and D. Sherrington, "Training noise adaptation in attractor neural networks," *J. Phys. A, Math. Gen.*, vol. 23, pp. L175–L182, 1990.

[12] R. Hecht-Nielsen, *Neurocomputing*. New York: Addison-Wesley, 1990, pp. 84–85.

[13] B. Derrida, E. Gardner, and A. Zippelius, "An exactly solvable asymmetric neural network model," *Europhys. Lett.*, vol. 4, pp. 167–173, July 1987.

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

[15] E. Mjolsness, C. D. Garrett, and W. L. Miranker, "Multiscale optimization in neural nets," *IEEE Trans. Neural Networks*, vol. 2, pp. 263–274, Mar. 1991.

[16] S. Abe, "Global convergence and suppression of spurious states of the Hopfield neural networks," *IEEE Trans. Circuits Syst.*, vol. 40, pp. 246–257, Apr. 1993.

[17] L. Wang and J. Ross, "Synchronous neural networks of nonlinear threshold elements with hysteresis," in *Proc. Nat. Acad. Sci.*, 1990, vol. 87, pp. 988–992; and "Interactions of neural networks: Models for distraction and concentraction," in *Proc. Nat. Acad. Sci.*, 1990, vol. 87, pp. 7110–7114.

[18] *Artificial Neural Networks: Oscillations, Chaos, and Sequence Processing*, L. Wang and D. L. Alkon, Eds. Los Alamitos, CA: IEEE Computer Soc. Press, 1993.

[19] M. Abelles, *Local Cortical Circuits*. New York: Springer-Verlag, 1982, p. 21.

[20] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949, p. 44.

[21] S. Grossberg, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*. Boston, MA: Reidel, 1982.

[22] C. von der Malsburg, "Self-organization of orientation sensitive cells in the striate cortex," *Kybernetik*, vol. 14, pp. 85–100, 1973.

[23] T. Kohonen, *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag, 1984.

[24] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, pp. 193–202, 1980.

[25] G. A. Carpenter and S. Grossberg, "Neural dynamics of category learning and recognition: Attention, memory consolidation, and amnesia," in *AAAS Symp. Brain Structure, Learning, and Memory*, J. Davis, R. Newburgh, and E. Wegman, Eds., 1986, pp. 233–290.

[26] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cogn. Sci.*, vol. 9, pp. 75–112, 1985.

[27] R. Hecht-Nielsen, *Neurocomputing*. Reading, MA: Addison-Wesley, 1990, p. 63.

[28] S. Grossberg, "Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors," *Biol. Cybern.*, vol. 23, pp. 121–134, 1976.

[29] L. Wu and F. Fallside, "On the design of connectionist vector quantizers," *Comput. Speech Language*, vol. 5, pp. 207–229, 1991.