Proc. IEEE World Congress on Computational Intelligence (IJCNN'98), May 4-9, Anchorage, Alaska, pp. 2378-2383.

The Role of Excitatory and Inhibitory Learning in EXIN Networks

Guilherme de A. Barreto Aluízio F. R. Araújo
University of São Paulo - Electrical Engineering Department
C. P. 359 - São Carlos - SP - Brazil
{gbarreto, aluizioa}@sel.eesc.sc.usp.br

Abstract

In this paper we propose modifications for the learning rules of Marshall's EXIN (excitatory + inhibitory) neural network model in order to decrease its computational complexity and understand the role of the weight updating learning rules in correctly encoding familiar, superimposed and ambiguous input patterns. The MEXIN (Modified EXIN) models introduce mixtures of competitive and Hebbian updating rules. In this case, only the weights of the unit with highest activation are updated. Hence, the MEXIN networks require less computation than the original EXIN model. A number of simulations are carried out with the aim of showing how the models respond to overlapping, superimposed and ambiguous patterns.

Keywords: EXIN networks, anti-hebbian learning, competitive learning, uncertainty, distributed coding.

1. Introduction

Different authors [1], [2], [3], [4] have proposed self-organizing artificial neural network (ANN) models with trainable excitatory and inhibitory weight connections. Most of them deal with the problem of decorrelating output units applied to principal components analysis (PCA models) [5]. The basic idea is to use a Hebbian learning rule to update excitatory weights and anti-Hebbian learning rule to update inhibitory weights. According to the Hebbian rule, an excitatory synapse connecting two neurons are strengthened if the activities of the two neurons are correlated and weakened if they are anti-correlated. The anti-Hebbian rule, states that the change in a synaptic strength is proportional to the correlation of the activities of the two neurons, but the direction of the change is opposite to that in the Hebbian rule.

Information received by ANN from the environment might carry redundancy and uncertainty. In neural networks, uncertainty occurs when an incomplete, noisy, or ambiguous incoming signal has more than one likely classification. The ability to deal with uncertainty is a desirable network property. An ANN must also be able to handle context [6] and multiple patterns occurring simultaneously in the data. Marshall [7] proposed a self-organizing neural network model that deals with overlapping, superimposed and ambiguous patterns. The EXIN (excitatory + inhibitory) network consists basically of a set of coupled differential equations that governs the dynamics of the neural model. These equations are characterized by a Hebbian learning rule to update the feedforward excitatory weights, an anti-Hebbian learning rule to update the inhibitory weights, and a shunting equation [8].

One of the major drawbacks of implementing EXIN model is the high computational cost of finding numerical solutions to the differential equations governing its activation [9]. This limits its practicality and the network efficacy can only be demonstrated on small problems. Other limitations of EXIN models are the absence of a clear objective function and the difficult to determine, except empirically, the system effectiveness and stability [10].

In this paper, we propose a model to reduce the computation time for EXIN networks and to help us understanding the role of excitatory and inhibitory learning rules. The basic idea is to substitute the Hebbian-like weight updating rule by a simpler competitive learning rule [11] and to update only the weights of the output unit with highest activation value.

The paper is organized as follows. First we summarize the original EXIN model in Section 2. Then, in Section 3 we present the modified versions. In Section 4, we discretize the MEXIN (Modified EXIN) models and organize them in a procedure for implementation purpose. In Section 5, we show the simulation results. Then, we discuss the results and conclude the paper in Section 6.

2. The original EXIN model

The EXIN neural network model has been proposed by J. Marshall [7], [12], [13], and comprises two layers of processing units, a non-linear system of coupled differential equations for activating the output neurons, modifiable excitatory feedforward and inhibitory feedback connections. Each group of connections is trained through a different Hebbian-like learning rule. EXIN can deal with binary and analog inputs. The network has m inputs: u_j, and n output units: a_i. The activation of the output units follows the shunting equation:

$$\frac{da_i}{dt} = -Aa_i + \beta(B - a_i)E_i - \gamma(C + a_i)I_i \qquad (1)$$

where A is a decay term, B and C are the maximum and minimum possible activities, E_i and I_i represent the total excitatory and inhibitory input to the unit i, β and γ are parameters that describe the overall influence of the excitatory and inhibitory terms in (1). E_i is defined as:

$$E_i = \frac{\sum_{j=1}^{m}\left(\left[u_j\right]w_{ji}\right)}{1 + \sum_{j=1}^{m} w_{ji}} \qquad (2)$$

where the u_j represents the activation level of the input unit j, $[u_j] \equiv \max(u_j, 0)$, and w_{ji} is the excitatory connection between input j and output unit i. I_i is defined as:

$$I_i = \sum_{j=1}^{n}\left(\left[a_j\right]p_{ji}\right) \qquad (3)$$

where a_j represents the activity level of the output unit j, $[a_j] \equiv \max(a_j, 0)$, and p_{ji} is the inhibitory connection between output units j and i.

It is worth noting that the activation of the output units cannot be calculated in a single step because of the influence of the feedback links and the non-linearity of the other units. Thus, a transient must be simulated by numerically solving the differential equation (1).

The feedforward excitatory weights from the input to the output layer are updated according to the following Hebbian rule:

$$\frac{dw_{ji}}{dt} = \alpha f(a_i)\left[g\left(u_j\right) - w_{ji}\right] \qquad (4)$$

where α is a small positive the learning rate constant, $f(a_i)=[\max(a_i, 0)]^2$ and $g(u_j)=\max(Gu_j, 0)$, G is a constant.

The inhibitory weights connecting the output units are updated according to an anti-Hebbian rule:

$$\frac{dp_{ji}}{dt} = \eta q\left(a_j\right)\left[h\left(a_i\right) - p_{ji}\right] \qquad (5)$$

where $0 < \eta << \alpha$ is the learning rate, $q(a_j)=\max(a_j, 0)$ and $h(a_i)=\max(Ha_i, 0)$, H is a constant.

In the next sections we propose some combinations of the weight updating rules together with a mechanism of finding the highest activation output unit.

3. The modified EXIN models

In this section we introduce the modified versions of the EXIN model. The goal of these changes is to reduce the computation effort for the EXIN model, and to understand the role played by the excitatory and inhibitory learning rules in classifying the superimposed and ambiguous input patterns.

3.1. The first modified EXIN model

The first modification in the original EXIN is the introduction of a mechanism to find the output unit with the highest activation value. Then, we substitute the feedforward excitatory Hebbian updating rule (4) by a purely competitive rule, namely

$$\frac{dw_{jv}}{dt} = \alpha\left[g\left(u_j\right) - w_{jv}\right] \qquad (6)$$

where v is the index of the output unit with the highest activation. The inhibitory learning is performed by equation (5).

Equation (6) can be considered a particular realization of equation (4) when $f(a_i) = 1$, for all i.

3.2. The Second Modified EXIN Model

The second modification in the original EXIN also employs a mechanism to find the output unit with the highest activation value. Then, the excitatory learning is performed by equation (4) updating the weights of the unit with highest activity, however the inhibitory weights are updated according to an "anti-competitive" rule, namely:

$$\frac{dp_{ji}}{dt} = \eta\left[h(a_i) - p_{ji}\right] \qquad (7)$$

In this case, according to the inhibitory equation (7), the output units will compete for the right of not responding to a given input pattern. Equation (7) can be considered a particular realization of equation (5) when $q(a_j) = 1$, for all j.

3.3. The third modified EXIN model

The third modification, as in the other two MEXIN models, employs a mechanism to find the output unit with the highest activation value. In this case, we use the competitive equation (6) to update the excitatory weights, and the anti-competitive equation (7) to perform inhibitory learning.

4. Discretizing the models

In order to simulate the models presented above on a general purpose computer, we must discretize the differential equations of the EXIN and MEXIN models. So, using a forward-difference method [14] the equation (1) get the following form

$$\frac{a_i^{k+1} - a_i^k}{T} = -Aa_i^k + (B - a_i^k)E_i^k - (C + a_i^k)I_i^k \qquad (8)$$

where the superscript k refers to the discrete time and T is the integration step size constant. After some algebraic manipulations the final form of the equation (8) is

$$\Delta a_i^k = -A^* a_i^k + \beta^*(B - a_i^k)E_i^k - \gamma^*(C + a_i^k)I_i^k \qquad (9a)$$

$$a_i^{k+1} = a_i^k + \Delta a_i^k \qquad (9b)$$

where $A^* = A.T$, $\beta^* = \beta.T$, and $\gamma^* = \gamma.T$.

The same procedure is valid to equations (4) and (5). In this case the final forms are

$$\Delta w_{ji}^k = \alpha^* f(a_i^k)\left[g(u_j^k) - w_{ji}^k\right] \qquad (10)$$

$$\Delta p_{ji}^k = \eta^* q(a_j^k)\left[h(a_i^k) - p_{ji}^k\right] \qquad (11)$$

and the discrete versions of equations (6) and (7) are

$$\Delta w_{jv}^k = \alpha^*\left[g(u_j^k) - w_{jv}^k\right] \qquad (12)$$

$$\Delta p_{ji}^k = \eta^*\left[h(a_i^k) - p_{ji}^k\right] \qquad (13)$$

where $\alpha^* = \alpha.T$, and $\eta^* = \eta.T$ in both sets of weight updating rules.

Marshall [15] has also worked with adaptive step-size numerical integration methods (Gear method, for instance) however the speed difference was not worth the extra effort.

4.1. The MEXIN algorithm

The activation and the weight updating rules are organized and summarized below in order to correctly simulate the dynamics of MEXIN neural networks.

1. Set the following initial values for $k = 0$:

$$w_{ji}^0 = 1.0 + 0.01(2R_{ji} - 1), \ \textit{for all } i, j$$

$$p_{ji}^0 = \begin{cases} 0.25 + 0.01(2R_{ji} - 1) & \textit{if } \ i \neq j \\ 0 & \textit{if } \ i = j \end{cases}$$

where R_{ji} is a random value between 0 and 1.

2. Initialize the output activations: a_i=0, i=1, ..., n

3. Present the randomly chosen input pattern $\mathbf{u} = (u_1, \ldots, u_m)^T$, in which T is the transpose.

4. With the pattern clamped, numerically solve the differential equation (1), cycling the discretized shunting equation (9) until the variation of the activation reaches a given near-zero value:

Calculate E_i^k according to equation (2)
DO:
  Calculate I_i^k according to equation (3)
  Calculate Δa_i^k according to equation (9a)
  Calculate a_i^{k+1} according to equation (9b)
WHILE: $|\Delta a_i^k| > \varepsilon$, for all i.

where ε is a very small value.

5. Find the output unit with the highest activation:

$$v = \underset{i}{\arg\max}\{\, a_i \,\}$$

6. Update only the excitatory weights of the output unit determined in step 5 according to the first, second, or third MEXIN model.

7. Update inhibitory weights according to the first, second, or third MEXIN model.

8. Increment k. If $k > k_{max}$ (in which k_{max} is the maximum number of cycles), the algorithm stops; otherwise go to step ***2***.

6. Simulation results

The simulations carried out here are based on those performed by Marshall [7] and Harpur [9]. All the experiments refer to a network with $m = 6$ inputs (labeled *A*-*F*) and $n = 6$ outputs. In all simulations, the patterns are randomly chosen and presented to the network repeatedly for a fixed number of times specified by k_{max}. For all the simulations the constants used are: k_{max} = 9000 (≅ 1500 for each pattern), A = 2.25, B = 1.0, C = 0.1, $\beta = 1.25$, $\gamma = 750$, $\alpha = 112.5$, $\eta = 16.125$, $\varepsilon = 10^{-5}$, $G = 1$, and $H = 1$. The MEXIN models were simulated in ANSI C in a SUN workstation ULTRA-1, 166MHz and 128 Mbytes of RAM.

Simulation 1: Coding Overlapping Patterns

In this experiment the MEXIN model uses equation (12) to update the excitatory weights and equation (11) to update the lateral inhibitory connections. The training set is composed of 6 overlapping binary patterns *A*, *AB*, *ABC*, *CD*, *DE*, and *DEF*. The results are shown in Figure 1. Once these weights are reached they become completely stable. It is worth noting that inhibition is strongest between neurons coding overlapping patterns and weakest between neurons coding non-overlapping patterns. The inhibitory weight matrix is approximately symmetric. For this simulation we used a step size T = 0.0014. Figure 1a illustrates the final excitatory weight configuration, and Figure 1b shows the final inhibitory weight matrix.

Simulation 2: Network Response to Trained Patterns

Using the weights obtained in the first simulation, the response of the network to familiar patterns was evaluated. When each input pattern was presented to the developed network only the corresponding neuron coding the whole pattern became active (see Figure 2a) For instance, when ***AB*** was presented, the output unit ***AB*** became fully active and inhibited the activations of the others output units.

Simulation 3: Parsing Superimposed Patterns

Additional tests were performed considering as inputs unfamiliar patterns. Unfamiliar patterns mean an exact combination of two training patterns. In this case, the two corresponding neurons coding the superimposed patterns became active (see Figure 2b). For example, when ***ABDE*** was presented, both units ***AB*** and ***DE*** became fully active. The network parses unfamiliar patterns in terms of the familiar patterns [7]. Hence, the MEXIN network, as well as the original EXIN, allows multiple patterns to be represented in a distributed fashion. The use of a inhibitory learning rule (Hebbian, competitive, etc.) allows multiple neurons to win a competition, instead of forcing a single winner.

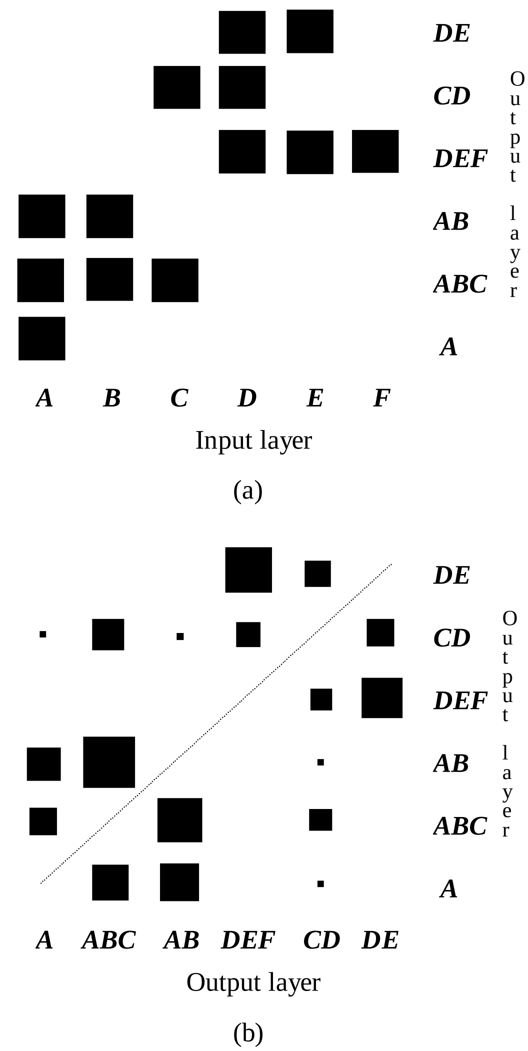

Figure 1. Final weight in simulation 1 for (a) excitatory weights; (b) inhibitory weights. The weight values are to the length of the sides of each square.

Simulation 4: Representation of Uncertainty

The fourth test evaluated the response of the network to ambiguous patterns (Figure 2c). When pattern ***D*** was presented to the network, both units ***CD*** and ***DE*** became a little active. These are the nearest known patterns to input ***D***. The activation of other neurons (***DEF,*** for instance) was suppressed. The network represents its uncertainty about the classification of an input pattern simultaneously activating multiple output units. This is a very useful property because in some perceptual environments, a network can successfully self-organize only when multiple hypotheses about a uncertain classification can be simultaneously represented [7].

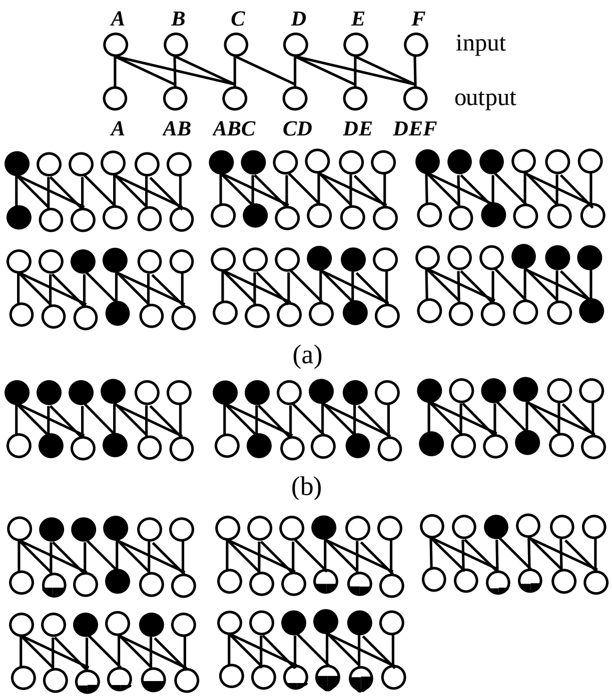

Figure 2. **The responses of the network in simulation 2-4 for (a) familiar patterns; (b) parsing of familiar patterns; and (c) ambiguous patterns (representation of uncertainty). The inhibitory connections are not shown for clarity sake.**

Simulation 5:

In this experiment the MEXIN model uses equation (10) to update the excitatory weights, and equation (13) to update the lateral inhibitory connections. The training set is composed of the same 6 overlapping binary patterns used in simulation 1. The excitatory weights coded the overlapping patterns identically to the MEXIN model in simulation 1. However, the inhibitory weight matrix is very different (see Figure 3a) The inhibition between neurons coding overlapping patterns and between neurons coding non-overlapping patterns were quite strong. This leads to a decreasing in the network ability to classify superimposed patterns (see simulation 6).

Simulation 6:

This test verifies the response of the model trained in simulation 5 to familiar and superimposed input patterns. The response to familiar overlapping patterns is identical to that of Figure 2a in simulation 2. However, the response to superimposed patterns does not activate only the output units which coded the superimposed pattern. For example, the presentation of pattern ***ABDE*** activates the output units that coded the patterns ***AB*** and ***DE***, as well as the units that coded patterns *A* and *ABC*. This occurs due to strong inhibition between units encoding non-overlapping patterns.

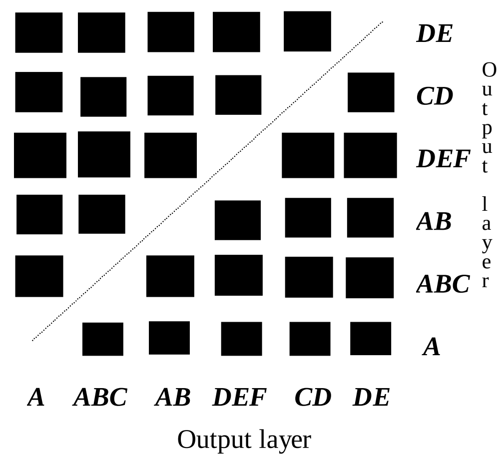

Figure 3. Inhibitory weights in simulation 5.

Simulation 7:

This experiment used a combination of equation (12) to update excitatory weights and equation (13) as a inhibitory weight updating rule. The integration step size was set to T = 0.014. We repeated the sequence of simulations 1-4 and assessed the results as in the previous simulations. The network was able to code the six binary patterns as the others MEXIN models did, however the networks was unable to classify correctly the familiar patterns (see Figure 4). The inhibitory weight matrix had the same structure of that of Figure 3, that is, inhibition between neurons coding non-overlapping patterns were strong, and had the same order of magnitude of the inhibition between neurons coding overlapping patterns.

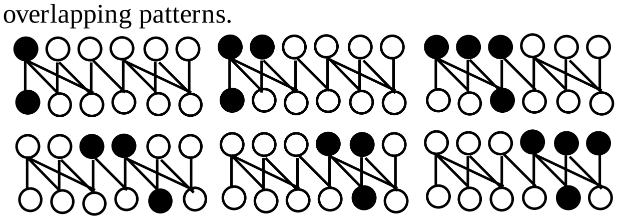

Figure 4. Response of the network to familiar patterns in simulation 7.

7. Conclusions and further work

We have presented a set of modifications to the original EXIN model in order to decrease its computational cost and assess the role played by the excitatory and inhibitory weight updating rules in encoding overlapping, superimposed and ambiguous input patterns. The nature of the inhibitory learning rule plays a very important role in learning such patterns.

Table 1 summarizes the results of the several models presented. Such results suggest that correct classification and encoding of familiar, superimposed and ambiguous binary patterns demands the inhibitory Hebbian learning rule. The excitatory learning rule can be either Hebbian or competitive type. The absence of a Hebbian learning in both rules is the worst case. It is worth emphasizing that the combinations of different kinds of learning rules, together with shunting activation equation, is possible only if we determine the output unit with highest activation. If we try to use the combinations proposed in Section 3 in the original model it will fail.

The MEXIN1 model requires less computational effort than the original EXIN for a same task. But, one could argue that updating only the weights of the unit with highest activation value limit the representational capacity of the network because it is implementing a kind of winner-take-all behavior. The simulations suggest that even doing so the MEXIN1 model is still able to deal with overlapping, superimposed and ambiguous binary patterns. Further work must be developed in order to simulate the MEXIN models with real-valued inputs and with applications domains in which we have a network with a significantly higher number of output units.

Table 1. Summary of features and results

	EXIN	MEXIN 1	MEXIN 2	MEXIN 3
Excitatory Learning	Hebb	Compet.	Hebb	Compet.
Inhibitory Learning	Anti Hebb	Anti Hebb	Anti Compet.	Anti Compet.
Encode overlapping Patterns	YES	YES	YES	YES
Classify overlapping Patterns	YES	YES	YES	NO
Parsing of Multiple Patterns	YES	YES	NO	NO
Classify Ambiguous Patterns	YES	YES	NO	NO

Acknowledgments: I would like to thank FAPESP and CAPES, for providing financial support for this research.

8. References

[1] A. Carlson. Anti-Hebbian learning in a non-linear neural network, *Biological Cybernetics*, 64:171-176, 1990.

[2] J. Rubner, and P. Tavan. Development of feature detectors by self-organization: A network model, *Biological Cybernetics*, 62:193-199, 1990.

[3] P. Földiák. Forming sparse representations by local anti-Hebbian learning, *Biological Cybernetics*, 64:165-170, 1990.

[4] F. Palmieri, J. Zhu, and C.H. Levy. Anti-Hebbian learning in topologically constrained linear networks: A tutorial, *IEEE Transactions on Neural Networks*, 4(5):748-761, 1997.

[5] J. Rubner, and P. Tavan. A self-organizing network for principal-component, *Europhysics Letters*, 10(7):693-698, December 1989.

[6] I.J. Myung, C. Kim, and W.B. Levy. Context-dependent recognition in a self-organizing recurrent network, *Proceedings of the Annual Conference of Cognitive Science Society*, 7-10, San Francisco, CA, August 1997.

[7] J.A. Marshall. Adaptive perceptual pattern recognition by self-organizing neural networks: context, uncertainty, multiplicity, and scale, *Neural Networks*, 8:335-362, April 1995.

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

[9] G.F. Harpur, and R.W. Prager. A fast method for activating competitive self-organizing neural networks, *Proceedings of the International Symposium on Artificial Neural Networks*, 412-418, Tainan 1994.

[10] G.F. Harpur. *Low Entropy Coding with Unsupervised Neural Networks*, PhD. Thesis, University of Cambridge, February 1997.

[11] D.E. Rumelhart, and D. Zipser, "Feature discovery by competitive learning", *Cognitive Science*, 9:75-112, 1985.

[12] J. A. Marshall, "A self-organizing scale sensitive neural network", *Proceedings of the International Joint Conference on Neural Networks*, 3:649-654, San Diego, June 1990.

[13] J. A. Marshall, "Representation of uncertainty in self-organizing", *Proceedings of the International Neural Network Conference*, 809-812, Paris, France, July 1990.

[14] W.E. Boyce and R.C. Diprima, *Elementary Differential Equations and Boundary Value Problem*, John Wiley & Sons, 6th Edition, August 1996.

[15] J.A. Marshall, *Personnal communication*, September, 1997.