A Self-Scaling Procedure in Unsupervised Correlational Neural Networks

Sylvain Chartier & Robert Proulx
chartier@leina.uqam.ca  proulx.robert@uqam.ca
Université du Québec à Montréal

Abstract

In neural networks, categorization is generally achieved by learning directly from the prototypes. However, in a natural setting the categories should emerge from learning from a set of exemplars instead of prototypes. Still even if the problem of learning from correlated items has been solved [1], the selection of the right size for a category remain an open question. In this study, we test the hypothesis that the introduction of a vigilance parameter that specifies the degree to which patterns must be similar in order to be considered exemplars of the same prototype can be implemented in a general correlational neural network. The results show that this is the case and the number of the resulting categories varies as a function of the value of the vigilance parameter. It is thus concluded that such a vigilance parameter may constitute the key to self-scaling adaptation in unsupervised correlational neural network.

Introduction

One of the advantages of using unsupervised learning algorithms in artificial neural networks lie in their ability to naturally implement adaptive categorization without the need of postulating an access to a pre-existing information from outside the system. In general, there are two kinds of models used for categorization: The competitive models and the correlational autoassociative models.

Competitive models are characterized by the fact that each memorized category is associated with one specific unit of the network. Consequently, losing a unit results in the losing of the associated category. For that reason, such models are very sensitive to the instability of their components and thus lack the robustness that characterizes most of the biological systems. On the other hand, in correlational models the categories are distributed among the connections. This feature makes those models less sensitive to specific unit loss.

Still both kinds of models have problems when categorizing in noisy environments. To overcome this problem Crick & Mitchison [2] proposed an unlearning procedure to remove noise in correlational network. This algorithm has been implemented and tested by Hopfield, Feinstein & Palmer [3] within correlational networks. However, such a procedure can only be applied to random noise, and is not applicable to selectively decide the variability of a given set of exemplars to be members of the same category.

One way to deal with the choice of variability is used by the ART models developed by Grossberg [4]. Such models include a vigilance parameter that enables the network to adjust the size of a category for a specific unit. Moreover, it has the advantage to develop categories even when the exemplars do not fall in the same quadrant of the hypercube, but has the disadvantage to relies on non-local information.

The purpose of this study is to test the possibility to implement a vigilance parameter in a specific recurrent correlational network, that is, the Eidos model developed by Bégin & Proulx [1]. However, before pursuing further we need to present the operating principles of this model.

The Eidos model

Following Anderson, Silverstein, Ritz and Jones [5] and their subsequent work on the BSB model, the Eidos model is a general autoassociative neural network. As with any other neural networks, it is completely specified by its architecture, its transmission rule and its learning rule.

Architecture
Its architecture is that of a one layer recurrent network of the Hopfield’s type that uses a positive feedback loop to iteratively force any initial input state to converge toward one the invariant final states that define the categories memorized by the system. This is illustrated in figure 1 below.

![Figure 1. The Eidos model](image)

**Transmission rule**

The transmission rule is non-linear and may be express by the equation

\[ x_{t+1} = L[Ax_t + \psi x_0] = L[A + \psi I]x_0, \quad t = 1...T \]  

where \( x_0 \) is the vector of the activity of the units at time \( t \), \( A \), the interconnection matrix between the units, \( \psi \), a decay parameter \( (0 \leq \psi \leq 1) \), and \( L \) a saturation function that maintains the activity of the units inside an hypercube of which sides are defined by two arbitrary limits equal to \( \pm \gamma \).

**Learning rule**

The learning rule is what distinguishes the Eidos model from its predecessors. Nearly all of the interesting characteristics of the model results from the use of such a rule. In short, it is postulated that at each learning trial, the weights of the connection matrix are modified by a dual Hebbian and Anti-Hebbian process of the form

\[
\Delta A = \alpha x_0 x_0^T + \beta x_0 x_0^T
\text{ and}
\]

\[ A_{k+1} = \zeta A_k + \Delta A_k \]  

where \( \alpha \) and \( \beta \) correspond respectively to the general learning parameters of the Hebbian and anti-Hebbian adaptation \( (\alpha \geq 0 \text{ and } \beta \leq 0) \), and \( p \) and \( n \), to the iteration at which each occurs. The quantity \( \zeta \) is a general forgetting parameter \( (0 \leq \zeta \leq 1) \), and the value of \( k \) represents the number of the learning trial.

It has been demonstrated [1] that provided that the absolute value of the magnitude of \( \alpha \) is greater than that of \( \beta \), and that \( p \) is less than \( n \), then the eigenvalues of the connection matrix will always converge to define the proper stimulus space, whatever the degree of correlation between the individual items and without any constraint on the representation of the stimulus set.

Moreover, even after over-learning of an old set of items, the presentation of a new linearly independent stimulus will always result in the emergence of a new positive eigenvalue in the spectrum and thus a new dimension in the feedback space.

However, if such modifications solved the problem of learning correlated stimuli, it cannot decide by itself when two exemplars are sufficiently close to generate only one prototype. The system always ends in developing the same number of prototypes then that of the exemplars. It is therefore interesting to see if the introduction of a vigilance parameter will enable the network to deal with this problem.

**Introduction of a vigilance parameter**

To introduce a vigilance parameter, we must have a way to measure the distance between an input and the resulting category. This comparison of the input and the output may be express by

\[ d = \sum \left| x_0 - x_t \right|_p \]  

where \( x_0 \) is the input vector, and, \( x_t \), is the output vector at the convergence time \( t \).

Moreover if \( d \) is greater of the value of the vigilance parameter \( \rho \), then we will say that the input pattern, \( x_0 \), is an exemplar of a new category and let the network performs an usual learning trial. On the other hand, if \( d \) is less of the value of \( \rho \), we assume that \( x_0 \) is an exemplar of the same category. In this case the new exemplar should only serve to update its associate prototype. Consequently, we must incorporate a function that will compute an adjusted mean including the exemplar to existing prototype. This function can be express by the equation

\[ \overline{x}_{adj} = (x_t + \phi x_0)/(1 + \phi) \]  

Where \( \overline{x}_{adj} \) is the adjusted mean, and, \( \phi \), a weighting parameter in the interval \([0.0 \ldots 1.0]\). This parameter is used so that after every learning trial the activation pattern will be modified as a function of the correct exemplar. In other word, the frequency of an exemplar will influence the resulting prototype.

The learning procedure just described above is schematically illustrated in the figure 2 below.
Standards and Technology (NIST). Those exemplars are bitmap digits taken from the National Institute of Stimuli categories “0” and “1” (see figure 3 below).

At a value of \( \rho \)

More precisely, by varying the value of the vigilance parameter, the network selected a random exemplar and then performed a recall trial followed by the comparison between the input and the output patterns. If \( d \) was less than \( \rho \) then the network applied the equation 5 followed by a learning trial; if not, the network performed an usual learning trial. The procedure was repeated for 1000 learning trials.

The value used for \( \rho \) were 0.1, 2.0 and 5.0. The reason for a choice of 5.0 as maximum value of \( \rho \) was because it is the limit of the distance between the desired categories. In the present case the upper value for two categories was determined by calculating the distance between the means of the category “0” and the category “1” which was 5.5.

Prior to the simulation the system started by learning a random exemplar because the network must have something in memory before making any comparison.

Dependent variables

The dependent variables retained for the analyses were the number and magnitude of the eigenvalues of the connection matrix, which represent the effective number of dimensions of the feedback space, as well as the categorization performance of the system measured by the percentage of correct recognition in recall trials.

Results

For the simulation with \( \rho = 0.1 \), following our expectation, the network developed 24 positives eigenvalues, out of a possibility of 24. Thus, there is as much categories as exemplars. This is illustrated in the figure 4 below.

Method

Stimuli

To test our hypothesis, we use 24 exemplars of 8x8 bitmap digits taken from the National Institute of Standards and Technology (NIST). Those exemplars are taken from a random sample of elements of the categories “0” and “1” (see figure 3 below).

Procedure

The general procedure used for the learning and recall task was the same as the one previously used for the simulations of the Eidos model [1]. In other words, the system was tested using the following values for the free parameters of the model (\( \alpha=0.005, \beta=0.0025, \rho=5, \ zeta=0.9995, \) and \( \phi=0.01 \)). In the case of the vigilance parameter, the network selected a random exemplar and then performed a recall trial followed by the comparison between the input and the output patterns. If \( d \) was less than \( \rho \) then the network applied the equation 5 followed by a learning trial; if not, the network performed an usual learning trial. The procedure was repeated for 1000 learning trials.

The value used for \( \rho \) were 0.1, 2.0 and 5.0. The reason for a choice of 5.0 as maximum value of \( \rho \) was because it is the limit of the distance between the desired categories. In the present case the upper value for two categories was determined by calculating the distance between the means of the category “0” and the category “1” which was 5.5.

Prior to the simulation the system started by learning a random exemplar because the network must have something in memory before making any comparison.

Dependent variables

The dependent variables retained for the analyses were the number and magnitude of the eigenvalues of the connection matrix, which represent the effective number of dimensions of the feedback space, as well as the categorization performance of the system measured by the percentage of correct recognition in recall trials.
By looking at the figure 5, it is easy to see that when \( \rho \) is set to a value 2.0, then the network has more prototypes than the number of desired categories, but less than the number of exemplars.

Finally, when we test the system at a value of \( \rho = 5.0 \), the spectrum reduces to only two important values, as can be seen in figure 6. It thus means that the network has only two categories. The minor remaining positive eigenvalues are due to the small value of the general forgetting parameter. Moreover, the network always recalled the appropriate prototype when given an exemplar.

It is therefore possible to introduce a vigilance parameter for obtaining the good categories.

Discussion

The preceding experiments clearly demonstrated that it is possible to implement a vigilance parameter in a correlational autoassociative model which accomplish the same task as the one in the ART model. However, when incorporated, the network keeps its robustness property.

In addition, another feature of this modification is the introduction of a computation of an adjusted mean that permit to modify the prototype as a function of the previously learned exemplars. This is an advantage over the ART models because these last models are not able to show this qualitative modification of the prototypes since the activity of a unit corresponds to the category. Thus, even if the category qualitatively changes the network will never reveal it.

However, these two modifications, introduce at the same time two new parameters that must be tuned by the experimenter. But this do not jeopardize the unsupervised character of the procedure. Nevertheless, for the parameter \( \phi \) in equation 5, its value is not bounded to a rigid value, and it is not critical to the proper functioning of the network.

As for the vigilance parameter computation, it must rely on non local information, as well as the one in ART. However, the procedure proposed by Oja [6] on PCA neural networks has shown that it is possible to normalize a weight matrix only from local modification. Further research should then look at this possibility to see if in the case of a correlational model Oja's method could be applied to locally normalize a vector. If this is the case, then we could calculate the \( d \) at a local level. As for the value of the vigilance parameter, because it is used for making a decision on the degree of similarity we can
suggest that it represents an attention level that we can let free in the simulations.

It is thus concludes that as it is the case of competitive neural networks it seems possible to incorporate a vigilance parameter in a general correlational autoassociative neural network.

References


