SELECTING THE BEST SIGNIFICANT FRAGMENT TO THE INCREMENTAL HETEROASSOCIATIVE NEURAL NETWORK (RHI)

García Chamizo, J.M.; Satorre Cuerda, R.; Ibarra Picó, F.; Cuenca Asensi, S.
Departamento de Tecnología Informática y Computación.
Universidad de Alicante
Apdo. 99. Alicante. Spain
e-mail: juanma@dtic.ua.es

Abstract. The generality of the artificial neural networks models infers the requests based in the totality of the characteristics of the patterns. The RHI model infers just with a limited set of this characteristics, the significant fragment. This reason make RHI really appropriated by resolution of control and active vision problem. Although RHI model present high sensibility to distortion. In this paper it is developed the formalism to obtain the significant fragment in such a way it improve the noise tolerance.

Introduction

In general, an associative memory infers by using every component of input patterns used in recalling process. This is the case when hebbian correlations between the components of the learning patterns are used build the weight matrix [6,12,15]. The classification in this model is difficult because the linear dependencies between patterns bring up noise. Due to it, the recalling process consists sometimes of the orthogonalisation [9,10] of the learning patterns.

The RHI considers the components of each pattern which allow us to distinguish one pattern from the other and through these components, the model can infer, at least partially, the associated pattern to a given one. New components of this given pattern are then inferred in new iterations.

Total recall is not always possible, however, the model can indicate which components can not be obtained in an incremental manner.

Other artificial neural network models [1,4,7,8,9,11,13,14,16] use all the components of the patterns and, as a consequence, they can not used to extract the changing components in a set of patterns.

There could exist several significant fragments (a set of relevant components that are necessaries to infer a response). The components of the significant fragment particularised to the learning patterns represent a projection of them in a hyper-space.

In this work, an algorithm is proposed that finds a hyper-space to take out the projection of the learning patterns in such a manner that the relevant components corresponding every learning pattern form the maximum angles between them. That is the way to obtain the best distortion tolerance.

The RHI model

The RHI model is resumed: There could be a way to obtain a weight matrix \( m_{kj} \) by solving equation systems so that, in the recalling process, the response were satisfactory.

To that end, let \( a_{ik} \) be the matrix whose row vectors \( A_i \) are the input to a table with \( q \) associated pairs, the dimension of which is \( n \); let \( b_j \) be the matrix whose rows are the \( p \)-dimensioned \( B_j \) vectors associated to \( A_i \) and let \( m_{ik} \) be the matrix resulting from the correlations between \( A_i \) and \( B_j \) pairs of vectors in the following way:

\[
b_{ij} = \sum_{k=1}^{p} a_{ik} m_{kj} \quad \forall i = 1, \ldots, q; \quad j = 1, \ldots, p
\]

Since we want to obtain \( m_{ik} \) to validate the expression, we have a system with \( q \cdot p \) linear equations and \( n \cdot p \) variables.

Alternatively, the equation system can be rewritten as a set of \( p \) simpler subsystems, one for each column of the second member of the matrix expression:

\[
b_{ij} = \sum_{k=1}^{n} a_{ik} m'_{kj} \quad \forall i = 1, \ldots, q
\]

A system with \( q \) equations is obtained for each \( j \) value. Note that \( j \) can change between \( 1 \) and \( p \). That is, an equation system can be obtained for each column of the matrix \( m_{ik} \) of the variables; in other words, for each component of the \( B_j \) vectors.

It can be the case that some of the proposed equation subsystems could be solved whereas the rest could be incomposables.

Let \( p'_s \) be the number of solvable columns and \( p'_u \) the number of unsolvable columns. Then: \( p'_s + p'_u = p \).

It follows that the \( m'_{ik} \) matrix with \( p'_s \) columns and \( n \) rows could be obtained. As a consequence, the corresponding \( p' \) components of the vector \( B'_j \) associated to any \( A_i \) could be calculated by making the product between \( A_i \) and \( m'_{ik} \). They will be called subvectors \( B'_j \), the dimension of which will be
p_i. All the other p_i components of vectors B_i can not be calculated. They will be called subvectors B'_i, the dimension of which will be p_i. It is as, if the system can "remember" a part of the associated pattern from the variable pattern.

At this point, it is the time to go back to the that part the system has not been able to recall and treat to do it from the updated information; that is: the input pattern variable in addition to the associated pattern the system has been able to actually recall. What is then requested is that a new set of p_i subsystem be built having q linear equations and n + p_i variables.

However, the coefficients for the added up p_i variables can not be the corresponding components for the column vectors already recalled. These have already been calculated by linear combination of the input vectors and as the result of the initial equations systems. The new p_i equations systems will remain incompatible, the way they previously were in the above argumentation.

To eliminate this limitation of algebraic kind and be able to obtain additional information from the already recalled associated pattern, it will be necessary to build de new p_i equation systems with the n initial variables in addition to p_i variables so that the coefficients of the later ones be non-linear transformations of the recalled column vector. Discarded then the linear dependence, the possibility of actually recalling the missing information of the associated pattern can be expected.

By applying this technique repeatedly, the system could either definitively solve the problem or fail to do it because no one new component of the B_i vectors can be calculated in some iteration. In the latter case, the system can be told either to mark them with special symbols or to solve the problem by means of some heuristics.

**Selecting the best significant fragment**

In the precedent paragraphs not one hypothesis is made about the selection of the significant fragment, even of its possible repercussion about the quality of inference during recall. In this section we discuss about the existence of at least one optimal significant fragment, in order to get minimal distortion during the recalling of a noisy pattern.

Given a set of n learning patterns the significant fragment determine n vectors. These constitute the basis of one hyperspace the dimension of which is the cardinal of the significant fragment. If the angles among these vectors are too smalls, then any distortion could transform one of them in other vector of the base. The RHI answer will be wrong. We impose that the vectors of the significant fragment get a hyper-poliedra with maximum volume to obtain the best significant fragment; that is, the angles between two of these vectors will be as big as possible.

**NOTATION**

\[
A_i \in \mathbb{R}^n; \quad i = 1..q; \quad A = \begin{bmatrix} A_1 \\ \vdots \\ A_q \end{bmatrix}
\]

ALGORITHM

\[n = \text{relevants\_components\_numbers (RHI, A)}\]

INPUT: \(A'\)

OUTPUT: \(FR \in \mathbb{R}^{m\times}\) submatrix of optimal relevants components set.

Submatrix of optimal relevants

METHOD:

\[
\text{comb} = C_{p,n} \quad (\text{number of combinations})
\]

\[d_{max} = 0\]

FOR \(i=1..\text{comb}\)

\[
\text{generate } C_i = \text{matrix of columns of i-esima combination} \quad \text{NC}_i = \text{Normalised (C)}_i \quad d = \text{determinant (NC)}_i
\]

IF (\(d > d_{max}\))

\[\text{THEN } d_{max} = d \quad FR = C_i\]

END_IF

END_FOR

END_METHOD

**Example**

Let the following pairs of patterns be used to teach a RHI:

\[
A = \begin{bmatrix} A_1 = [010111010] \\ A_2 = [000111000] \\ A_3 = [010101010] \end{bmatrix} \rightarrow A' = \begin{bmatrix} 0111 \\ 0011 \\ 1001 \end{bmatrix}
\]

In this case, only three not null different determinants could be built with the directions

\[
\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} % & % & % \\ 0 & % & % \end{bmatrix} \begin{bmatrix} % & % & % \\ 0 & % & % \end{bmatrix} = %
\]

PREVIOUS: Eliminate null columns and repeated columns from A, because they generate null determinants.