Paper presented to INNS first annual meeting, Boston, September 6-10 88_C2

AN ALGORITHM FOR PATTERN RECOGNITION WITH VLSI NEURAL NETWORKS.

B. Sirletti (*), M. Verleysen (*), A. Vandemeulebroecke (**), P. Jespers. Université Catholique de Louvain, laboratoire de microélectronique, place du Levant 3, 1348 Louvain-la-Neuve, Belgium.

A very promising way to realize content-addressable memories (CAM) is the use of highly interconnected neural networks. Neural networks CAM can be very useful in the field of pattern recognition because of their inherent ability to retrieve pre-recorded information from damaged or partial input patterns.

In 1982, Hopfield ([1]) proposed an electrical model of biological neural networks and an information storage algorithm derived from the Hebb's rule ([2]). This algorithm is able to memorize about 0.15N patterns in a N-neuron fully interconnected neural network. However, as patterns too close to each other tend to merge, the Hamming distance (number of different bits) between two patterns to memorize must be high enough to retrieve them correctly. Furthermore, input patterns which never converge are frequent with this algorithm if we suppress Hopfield's hypothesis that two changes in the neuron values cannot occur simultaneously, which is impossible to foretell in an analog asynchronous VLSI neural network (fig.1 shows the percentage of well retrieved, non retrieved and unstable states reported to the Hamming distance from input patterns in a 12-bit network with 3 recorded patterns).

We propose here another algorithm which suppresses these drawbacks. The coefficients of the connection matrix are computed column by column. Each neuron has an input threshold (usually 0); above (under) it, its output value is 1 (-1). The feature of the method is to use a linear algebra algorithm (simplex) to determine the coefficients in order to maximize simultaneously the input of each neuron, i.e. the weighed sum of the other neuron values, with regard to its threshold, keeping the sign of this difference so as to have the desired patterns at the output of the network. To avoid unbounded solutions, we fix a lower and an upper bound (-1 and +1) to each connection weight. Fig.2 shows the results of this method, for the same recorded and input patterns as in fig.1.

A significant advantage of this method appears when we use an integer simplex method instead of a continuous-values one. If we limit indeed the possible values of the connections to -1,0 and +1, we observe no significant differences in the results of the method. This is due to the fact that even with the continuous-values method, at least (n-k) variables will take either the lower or the upper bound value, where n is the number of neurons and k the number of recorded patterns. This limitation of the possible values to -1, 0 and +1 is very interesting when we intend to implement the CAM with a VLSI neural network. The restriction to only three different values limits indeed the area of each synapse (connection) cell: only two one-bit registers are necessary to store the three different states, while the Hebb's rule needs $\log_2(2k+1)$ registers to memorize the 2k+1 possible values for the connection. Since k may be considerable in large networks, $\log_2(2k+1) >> 3$ and the area gain is very significant.

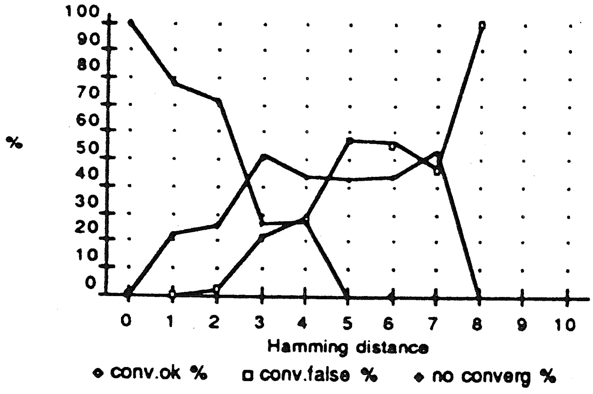

fig.1

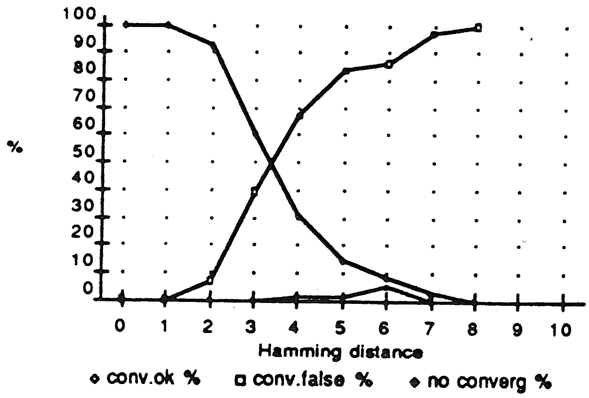

fig.2

[1] Hopfield J.J. (1982) Neural networks and physical systems with emergent collective computational abilities (Proc. Natl. Acad. Sci. USA)

[2] Hebb D. (1949) The Organization of Behavior (Wiley, New York)

(*) sponsored by IRSIA. (**) sponsored by FNRS.