Neural Networks from Models to Applications, L. PERSONNAZ, G. DREYFUS, Eds. (IDSET, Paris, 1989).

TOWARDS A NEURAL NETWORK CHIP : A PERFORMANCE ASSESSMENT AND A SIMPLE EXAMPLE

L. PERSONNAZ, A. JOHANNET, G. DREYFUS
Ecole Supérieure de Physique et de Chimie Industrielles de la Ville de Paris
Laboratoire d'Electronique
10, rue Vauquelin
75005 PARIS
FRANCE

M. WEINFELD
Ecole Polytechnique
Laboratoire de Physique Nucléaire des Hautes Energies
91120 PALAISEAU
FRANCE

ABSTRACT

The design of a digital VLSI feedback neural network with on-chip learning is discussed ; the accuracy of the synaptic coefficients, which is a crucial issue in neural network design, is studied in detail. The architecture of such a neural network, which is currently in the design phase, is described ; it is shown that the wiring complexity of the network can be overcome by a multiplexing scheme. It is shown that, with such a design, it is possible to implement both the learning and the retrieval processes. In addition, it is suggested that such networks can easily be made cascadable.

1. INTRODUCTION

It is well known that formal neural networks exhibit properties mimicking elementary functions of the brain, such as error correction, associative memory, pattern recognition, etc. They are made of numerous elementary processors (the neurons) which are individually able of performing a low level task, namely a weighted sum of several inputs, and make a decision according to result of this sum. Among the various architectures appearing in the literature, the fully connected network with feed-back proposed by Hopfield [1] has a number of very interesting features : first, the properties of these networks were shown to be amenable to detailed analytical treatments, so that the behaviour of such a network can be predicted accurately ; secondly, their design is made relatively straightforward by the regularity of their structure and the simplicity of the elementary process taking place in each neuron ; finally, the existence of simple, local, efficient learning rules makes the design of a network with on-chip learning feasible. Robustness with respect to connection and cell degradations or even failure, which is an important property from the electronic design point of view, is due to the redundancy of the representation of the information in the network.

It is clear that massive parallelism and fault tolerance, which are the main assets of artificial neural networks, will be taken advantage of only if real circuits are manufactured : simulations on standard or parallel computers are extremely useful for research purposes, but are too slow, in general, for real-time operation. If one wants to investigate higher hierarchical structures (networks of networks, having probably more operative properties than today's networks), the availability of integrated circuits exhibiting some fundamental characteristics such as associative memory or classification will be extremely helpful. These considerations are the motivations of the work which is presented here, as well as for other similar projects, already described or to be published [2-6]. The present project describes a fully digital implementation with the unique capability of having the learning algorithm integrated on the chip itself.

In the present paper, we shall first recall briefly the properties of the model network and the learning process ; in a second part, we shall discuss two questions which are of central importance in the actual design of an integrated network :

i) If the computation of the synaptic coefficients (the learning) is performed off-line, with, say, double-precision floating point real number representation, what round-off error can be tolerated when the synaptic weights are transferred to the device and stored there?

ii) If the computation of the synaptic coefficients is performed by the same device as the neural network, round-off errors occur at each stage of the learning phase, thus accumulating steadily ; what is the minimum precision required in the computation and in the storage of the weights ?
Finally, the chosen architecture will be described.

2. THE NETWORK MODEL

We consider a fully connected network, with feedback, of n formal binary neurons. The decision-making rule of neuron i is as follows :

$$\sigma_i(t+\tau) = \text{sign}\,(\textstyle\sum_j C_{ij}\,\sigma_j(t)) = \text{sign}\,(v_i(t)) \qquad (1)$$

$\sigma_i(t)$ is the state $\{-1;+1\}$ of the neuron i at time t,
$v_i(t)$ is the potential of the neuron i at time t,
C_{ij} is the synaptic coefficient of neuron i receiving information from neuron j.

In the case of parallel and synchronous updating of all neurons, the decision-making rule of the network can be formulated in matrix form :

$$\underline{\sigma}(t+\tau) = \text{sign}\,(C\,\underline{\sigma}(t)) = \text{sign}\,(\underline{v}(t)) \qquad (2)$$

$\underline{\sigma}(t)$ is the n-vector state of the network at time t,
C is the (n,n) synaptic matrix.

The associative memory properties of such a network, based on the existence of attractor fixed points in state space, have been extensively studied [7,8]. The problem of designing an auto-associative memory may be summarized as follows :
Consider a given set of p prototype vectors $\{\underline{\sigma}^k\}$ and a network defined by a synaptic matrix C and thresholds equal to zero ; if C is symmetric positive semi-definite, we can construct a Lyapunov function of this non linear dynamical

system as :

$$E(\underline{\sigma}) = -(1/2)\,\underline{\sigma}^T C\,\underline{\sigma}.$$

The problem of designing an associative memory consists in finding a matrix C so that the $\{\underline{\sigma}^k\}$ are attractor fixed points. The algorithms designed for the computation of the C matrix by multiple presentations of the prototype vectors are called learning rules. It is well known [8] that if C is the orthogonal projection matrix into the subspace spanned by the prototype vectors, the network satisfies the previous condition even with correlated patterns, in the limit of p<n. More precisely, with the projection matrix, each prototype $\underline{\sigma}^k$ is an attractor state and corresponds to the absolute minimum of the Lyapunov function : $E(\underline{\sigma}^k) = -n/2$. Some other attractors with energy values higher than -n/2 appear while prototypes are stored, but their basins of attraction are related to ambiguous patterns [9].

The projection matrix may be computed iteratively by a gradient-type learning rule[10] :

- the C_{ij} coefficients are initialized to 0;
- when the prototype $\underline{\sigma}^k$ is presented, matrix C(k-1) is available, so that one can compute :

$$\Delta C_{ij}(k) = (1/n)\,(\sigma_i^k - v_i^k)\,\sigma_j^k \qquad \text{for i and j = 1 to n ,} \qquad (3)$$

with $$v_i^k = \sum_j C_{ij}(k-1)\,\sigma_j^k \qquad \text{for i = 1 to n .}$$

- all prototypes are presented in sequence several times until the algorithm converges.

It should be noticed that the classical Widrow-Hoff algorithm, which takes the same form, works in the hetero-associative case with the condition p>>n ; it converges to the unique least-mean-squares solution which minimizes $\|\sigma_i^k - v_i^k\|^2$ for k = 1 to p and i = 1 to n. In our case, the system of equations :

$$v_i^k = \sigma_i^k \qquad \text{for k = 1 to p and i = 1 to n,}$$

has an infinity of solutions in general ; the solution of interest is obtained with the condition $C_{ij}(0) = 0$ for all i and j.

3. CIRCUIT ARCHITECTURE

3.1 Choice of technology : analog vs. digital

Most published designs describe analog implementation of fully connected networks, following ideas of Hopfield [11]. These designs use operational amplifiers as neural cells, and resistors as connecting links. In this case, the corresponding conductances are the synaptic coefficients. Such a design makes the system maximally parallel : parallelism is present within each neuron, and all neurons operate in parallel ; moreover, this design alleviates the connectivity problem to a large extent. However, these circuits exhibit the problems inherent to analog circuitry, especially in feedback systems, such as instability and noise sensitivity.

Still more important, the above designs do not allow easy adjustments of the synaptic coefficients : these circuits do not have any learning abilities. This very important factor, together with the fact that it might seem desirable to implement neural networks in standard, proven technologies, led us to the design of the fully digital custom circuit with on-chip learning which will be described below.

3.2 General considerations on VLSI digital architectures

Various digital architectures were previously discussed in the literature [4,12,13]. The main differences between these designs are related to the topological distribution of the computational tasks on silicon.
If binary neurons are to be implemented (σ_i=+1 or -1), the function that must be performed by the synapse betwen neurons j and i reduces to transmitting the value of C_{ij} (if σ_j=+1) or its opposite (if σ_j=-1) to neuron i. This is indeed a very simple function, which leads to various possible designs :

- if speed is of fundamental importance, the computation of the potential v_i can be performed in parallel at the level of each neuron i, thereby requiring n-1 adders per neuron ;
- if silicon area is a limiting factor, one can sacrifice speed to some extent by using one single adder per neuron and computing the potential in n sequential additions.

The first solution requires a large silicon area to implement n(n-1) adders, possibly sixteen-bit wide each ; most of the silicon area is taken up by the adders and the communication busses between them ; therefore, this solution is hardly feasible in standard VLSI technology for large networks (n >> 100). The second solution requires only n adders, and each neuron is relatively small ; in this case, most of the silicon area of an elementary neuron processor is taken up by the memory of the synaptic coefficients. The main advantage of this solution consists in the fact that the information circulating from a neuron processor to an another is a one-bit information. Moreover, the iterative learning rule we have proposed is local relatively to this architecture : its implementation on the chip does not requires significantly more communication of information beetwen the neuron processors than the implementation of the retrieval function alone.
If the state of the neuron is multi-valued [4], the first solution with processor-synapses is clearly imposed because the necessity of a real multiplication.

3.3 Accuracy of the synaptic coefficients

In all computer simulations of neural networks, the computation of the neuron potentials and of the synaptic coefficients (during the learning phase) and the computation of the potential only (during the retrieval phase) are usually carried out with the standard precision available on any computer. When attempting to implement a network of binary neurons, one cannot usually indulge in using the floating-point number representation, which would result in bulky circuits. Instead, one has to determine what is the minimum number of bits necessary to encode the synaptic coefficients for the network to operate satisfactorily. This problem must be approached in two slightly different ways, depending on whether the learning will be achieved on the chip or whether it will be performed off-line on a computer. These two aspects will be discussed below.

In order to be able to assess the efficiency of the network whose synaptic coefficients are coded on a limited number of bits, we compared the behaviour of such a network in the neighbourhood of the prototype states to the behaviour of a network whose synaptic coefficients were computed on a standard computer. We considered randomly chosen prototypes : proba(σ_i^k=+1)=x; proba(σ_i^k=-1)=1-x (the corresponding average overlap between prototypes is $\langle q \rangle = (1-2x)^2$).

We present in figure 1 the average behavior of a reference "perfect" network in the vicinity of the prototype states. The simulations are performed on a network of 64 neurons; 16 prototypes (α=p/n=.25) were randomly chosen with x=.5 (the prototypes are quasi-orthogonal). Both learning and retrieval were performed with floating point computation. The figure shows two histograms summarizing 10,000 retrievals with initialization of the network state at a normalized Hamming distance $h_i=H_i/n$ from a given prototype state. $h_f=H_f/n$ is the normalized Hamming distance after convergence. The results are averaged over prototypes.

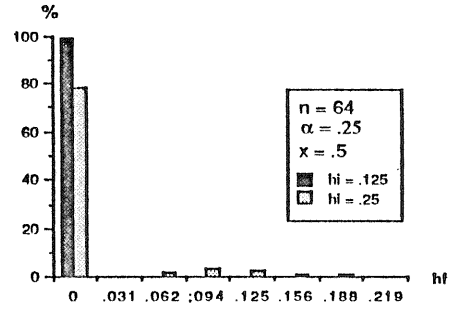

Figure 1.

The following results should be considered with reference to Figure 1. Figure 2 shows results obtained with a network whose synaptic coefficients were computed with floating-point computation during the learning phase, and subsequently truncated to m=4 bits including the sign bit (Figure 2a) or m=6 bits (Figure 2b). It is clear that in the second case the behaviour of the network is quite similar to that of the reference network. In the first case, excessive inaccuracy in the synaptic weights decreases the retrieval capabilities.

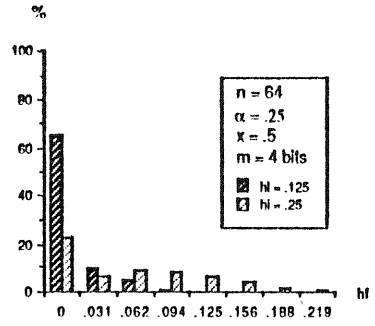

Figure 2a.

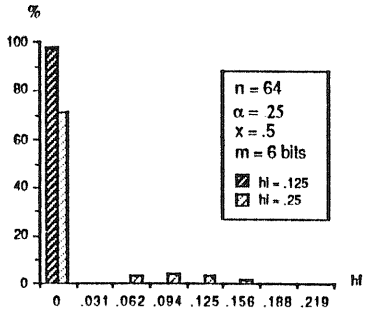

Figure 2b.

If both the computation of the synaptic coefficients and the retrieval are performed with limited precision, the number of bits required is higher than in the previous case. This is due to the fact that the learning algorithm is iterative ; therefore one has to compute $\Delta C_{ij}(k)$ at each iteration ; if these computations are performed with limited precision, roundoff errors accumulate. It is shown on Figure 3 that, all other things being equal, an accuracy of 9 bits (including sign) is required in order to obtain the same results as with floating-point computation. The results are basically the same with $0.3 < x < 0.7$, which corresponds to an average overlap between 0 and 0.16.

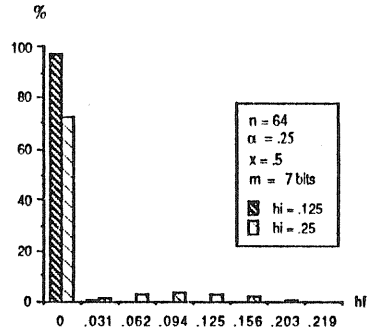

Figure 3a.

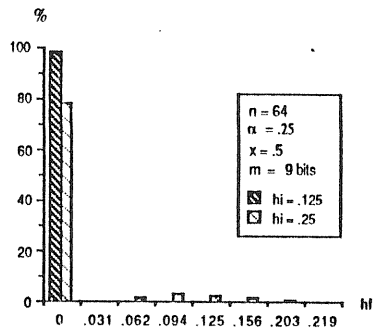

Figure 3b.

3.4 Proposed implementation

3.4.1 General considerations

Most current designs implement sign programming (-1,0,+1), by use of switches, for the synaptic coefficients. As shown above, this is far from satisfactory for associative memory purposes. There are current projects which try to implement the coefficients by storing charges in capacitors, the corresponding voltage driving transistors whose transconductance represents the synaptic coefficients [14,15]. It has also been suggested that resistors could be replaced by switched capacitors [16]. These solutions are interesting but technology dependent, and does not allow for a large precision on the coefficients.

Fully digital implementations, on the other hand, are more flexible as far as the accuracy of the coefficients is concerned, and can be performed with standard technologies. Moreover, on-chip learning can be implemented since the learning rule (3) involves basically the same arithmetic as that required by the retrieval, with little additional hardware. All multiplications are simply identities or complementations, the whole arithmetic being integer. Moreover, if we choose a network size which is an integer power of 2, the divisions by n are only shifts. The remaining problem is to implement the required connectivity.

3.4.2 Network architecture

As mentioned above, the technological constraints lead to a tradeoff between speed and silicon area, whereby the network will include n identical neuron-processors, each one with full arithmetic capability for learning and updating, and also including a local memory containing the n relevant synaptic coefficients. The only information that neurons have to exchange is one component of the network state, coded on one bit only, thus simplifying greatly signal routing and reducing chip complexity. The basic circuit diagram is shown on Figure 4.

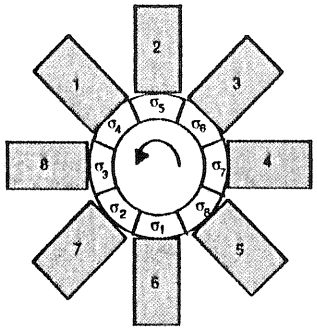

Figure 4

The network state is stored in a circular n x 1 bit shift register (each neuron processor contains one cell of the state register). A simultaneous partial potential update in every neuron processor is performed at each elementary shift of the register ; after a complete revolution of the state of the network, each neuron makes its decision and reloads its shift register cell with its new state (the updating of the state of the network is performed simultaneaously by the n neurons).

3.4.3 The neuron processor

The ALU includes a parallel adder, complementers, and a shifter (Figure 5). The "potential" v_i is stored in an accumulator, the most significant bit of which encodes the new neuron state, when the updating is completed. This state is loaded in the local one-bit cell of the state register σ(t), before a new updating revolution can takes place (operation symbolized by switch S1). It is also stored in the σ(t-1) one-bit register, in order to be able to detect the convergence of the system. This can be implemented with an exclusive-or operation simultaneously in each neuron, all the individual convergence signals x being or'ed. The synaptic memory is represented by C(i), and contains 64 9-bit words. Since this memory is always addressed sequentially, both in the learning and in the retrieval phase, it is implemented with shift registers. This implementation can be made almost as economical (in terms of power consumption) as a classical static random access memory, despite the fact that all n^2 coefficients have to be moved at each elementary time step, instead of only the n strictly needed : the auxiliary circuitry overhead needed with an addressable memory design is rather costly when the total memory capacity is low, as it is in our case. On the diagram, S2 symbolizes the switching between the two modes of operation, learning and retrieving.
Note that in the learning phase, three revolutions of the state register are necessary in order to compute one coefficient increment : one for initializing the state register with the prototype to be learnt, one for evaluating the potential, and one for updating the synaptic coefficients. The diagram does not represent the circuitry needed for testing the convergence of the iterative learning process; the computed increment is simply compared to zero, and all the local tests are combinatorially or'ed.

Figure 6 shows a tentative sixteen-neuron network design. The I/O tasks are performed by a serial-parallel register R, which can be used for loading or unloading the network, or can be disconnected from the network loop during the learning or retrieving phases by using switch S. Wire ξ carries the convergence signal. For simplicity, other control signals were omitted, as well as the sequencer, clock generator, and other ancillary circuits.

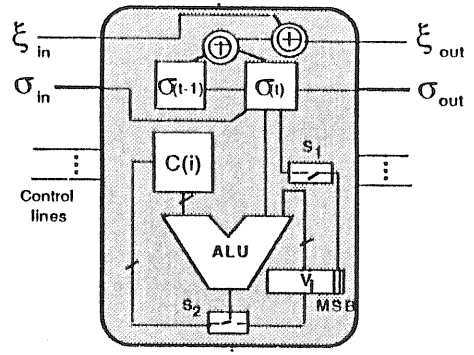

Figure 5

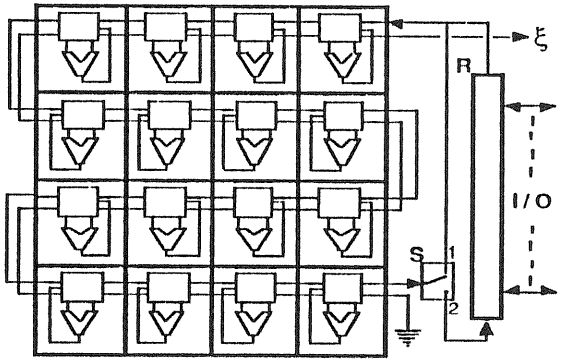

Figure 6

3.4.4 Technology and design strategy

An existing commercial technology, namely two metal CMOS, with 2 or 1.5 µm characteristic dimension, will be used. The overall clock rate, in this case, will be around 20MHz, which will give a typical convergence time of 10 to 20 microseconds.

One neuron is to be designed and processed via the French multi-chip project (CMP-GCIS). It is expected to require 4000 to 5000 transistors, which is a rather low number for testing and verification purposes. The testability will be an interesting issue, since such a circuit can be functional even in case of processing failures, provided these failures are located for instance in the memory.

4. CONCLUSION

This paper presents a first performance assessment of a fully digital VLSI neural network with on-chip learning, and an actual design which is currently being

completed. We studied the required precision for a satisfactory operation of the network as an associative memory. We describe the design of a network of 64 neuron processors with a ring architecture. The basic idea consists in a tradeoff between speed and wiring complexity, which, in addition, leads to an architecture which greatly simplifies the implementation of the learning algorithm on the circuit.
Design work is still in progress along two lines. First, it should be noticed that such circuits can be made easily cascadable by implementing only a part of the shift register on the chip and arranging several such chips in a ring ; the number of neurons will be limited only by the amount of memory available on the chip. Second, a feedforward classifier with integrated learning can easily be implemented with a small number of alterations to the chip.

Acknowledgements

This work is sponsored by G.CIS, by contract 87.C.185 from Ministère de la Recherche et de l'Enseignement Supérieur, and contract 87/34/187 from Ministère de la Défense (DRET).

REFERENCES

[1] Hopfield, J. J., "Neural networks and physical systems with emergent collective computational abilities", Proc. Natl. Acad. Sci. USA, vol. 79, pp. 2554-2558, 1982.

[2] Alspector, J., Allen, R. B., Hu, V. and Satyanarayana, S., "Stochastic learning networks and their electronic implementation", IEEE Conf. on Neural Information Processing Systems, Natural and Synthetic, Denver, 1987.

[3] Moopenn, A., Langenbacher, H., Thakoor, A. P. and Khanna, S. K., "A programmable binary synaptic matrix chip for electronic neural networks", IEEE Conf. on Neural Information Processing Systems, Natural and Synthetic, Denver, 1987.

[4] Murray, A. F., Smith, V. W. and Butler, Z. F., "Bit-serial neural networks", IEEE Conf. on Neural Information Processing Systems, Natural and Synthetic, Denver, 1987.

[5] Schwartz, D. B., Howard, R. E., Denker, J. S., Epworth, R. W., Graf, H. P., Hubbard, W., Jackel, L. D., Straughn, B. and Tennant, D. M., "Dynamics of microfabricated electronic neural networks", Appl. Phys. Lett., vol. 50, pp. 1110-1112, 1987.

[6] Sivilotti, M., Emerling, M. R., and Mead,C., "A novel associative memory implemented using collective computation", in Proc. Chapel Hill Conf. on VLSI, pp. 329-342, 1985.

[7] Amit, D. J., Gutfreund, H. and Sompolinsky, H., "Storing infinite numbers of patterns in a spin-glass model of neural networks", Phys. Rev. Lett., vol. 55, no. 14, pp. 1530-1533, 1985.

[8] Personnaz, L., Guyon, I. and Dreyfus, G., "Collective computational properties of neural networks : new learning mechanisms", Phys. Rev. A, vol. 34, pp. 4217-4228, 1986.

[9] Personnaz, L., Guyon, I. and Dreyfus, G., "Neural Networks for Associative Memory Design", in "Computational Systems - Natural and Artificial", ed. H. Haken (Springer, 1987).

[10] Diederich, S. and Opper, M., "Learning of correlated patterns in spin-glass networks by local learning rules", Phys. Rev. Lett., vol. 58, no. 9, pp. 949-952, 1987.

[11] Hopfield, J. J., "Neurons with graded response have collective computational properties like those of two-state neurons", Proc. Natl. Acad. Sci. USA, vol. 81, pp. 3088-3092, 1984.

[12] Blayot, F. and Hurat, Ph., "A VLSI Systolic Array Dedicated to Hopfield Neural Network", in "international Workshop on VLSI for Artificial Intelligence", Oxford 1988.

[13] Gobert, J., private communication.

[14] Schwartz, D. B. and Howard, R. E., "Analog VLSI for adaptive learning", Neural Networks for Computing, Snowbird, 1988.

[15] Tam, S., Holler, M. and Canepa, G., "Neural networks synaptic connections using floating gate non-volatile elements", Neural Networks for Computing, Snowbird, 1988.

[16] Tsividis, Y.P., Anastassiou, D., "Switched-capacitor Neural Network", Electronics Letters vol. 23, pp. 958-959, 1987.