Extended Abstracts of the 1994 International Conference on Solid State Devices and Materials, Yokohama, 1994, pp. 358-360

Invited

Digital Implementation of Neural Networks: From Generic Chips to Specific Blocks

Marc DURANTON

Laboratoires d'Electronique Philips,
22, avenue Descartes
94453 Limeil-Brévannes Cedex (France)

Neural networks have recently known a major development due to advances in mathematical techniques, better knowledge in neurobiology and to the increase of computing power in the recent years. This paper gives some hints for digital implementations of neural networks according to the field of application, the time constraints and the cost. It shows that neural networks can be performed by a wide range of systems, from software only solutions running on classical processors to very dedicated blocks.

1. INTRODUCTION

After a first bright period that followed the formal neuron model invention, then dark ages, Neural Networks (NN) have recently known a new growth due to several reasons : new algorithms, better understanding of neurological phenomenons, real industrial applications and new hardwares,that allow to perform the neural network approach far more easily.

Early in this stage, hardware implementations were studied because NN and their new paradigm of information processing, their intrinsic parallelism open new ways for chip architecture and realizations. A lot of technologies such as optics[1], superconducting devices, quantum well and so on can be of great help for making neurochips.

In the area of silicon neurochips[2], there is also a race between the analog and the digital approach. The two have advantages and drawbacks : analog systems are more compact because they generally use physical phenomenons to make computations[3], they make the most of technological particularities, they may mimic biological process[4], or they can directly embed sensors to build intelligent retina[5] for example. Digital systems are more accurate, implementations of algorithms used in the majority of practical applications are easier, and programmability and reconfiguration are also important factors of the digital chips. This paper focusses only on silicon based chips, and more precisely on digital realizations.

2.WHICH KIND OF SYSTEM ?

The versatility of the digital approach allows to fit the hardware to the application, and the use of parallelism leads to a great number of systems[6]. Neural networks algorithms can be performed by :

1. Software only solutions running on conventional computers or on a network of computers.
2. Softwares running on parallel or vectorial computers.
3. Softwares running on machines built with components "off-the-self" but with an architecture specialized for NN operations.
4. Softwares running on dedicated machines built with programmable neurochips.
5. Systems using dedicated neurochips.
6. Programmable neurochips added to a "classical" system.
7. NN oriented blocks that are embedded in a foreign architecture,
8. and so on...

It is often a compromise between performances, availability of the hardware, price that help in choosing the approach. Highly tuned designs reach very high performances, but also

lack flexibility.

People often justify dedicated hardware by the speed improvement compared to workstations.Comparing accurately with other systems is difficult, and meaningless on such general criteria like OPS (Operation Per Second). The only kind of honest benchmark is to program the same algorithm on the different machines. Furthermore, raw speed is not the only criterion : low cost, low power requirement, high integration... are also very important for the performance figures, and it is often a very complex issue. It also seems that new areas for NN-technology are in embedded, control or portable applications, which call for different architectures (small, embedded, evolutive, low power...).

The system price is also a key point : even a chip with a huge silicon area, but with all the system functions integrated can lead to a final system (printed circuit board and other chips) less expensive.

All this kind of remarks must be taken into account when designing a chip which is not only for research purpose.

3. FOR WHICH APPLICATION ?

Some users of NN use pure software implementations, running on conventional hardware. This is the case of most of current applications that run today without the help of dedicated hardware. For development and research purpose, a dedicated NN machine should combine the flexibility of workstations with the power of supercomputers.

Other users require hardware accelerators, but at a relatively low cost. The speed improvement, however, must be higher than what can be reached with commercially available systems of the same range of price. The hardware must be easy to use and should be integrated with the system that the users already have. This leads to realize accelerator boards that are plugged into existing systems such as PCs, workstations. For these systems, the target is not only the raw speed, but also the user-friendliness : a developer may prefer to use his workstation with its compiler rather than a specialized box, that is faster but requires to learn a dedicated language or to port floating point algorithms into integers (chips with floating point are more easy to use, but they lead to a waste of silicon area for NN applications).

From an industrial point of view, one can afford the development of chips if the market is big enough or if the final application cannot be realized without these chips.

4. FROM DEDICATED BLOCKS...

One of new development for NN is the possibility of "on-line" learning, which could lead to many applications which are limited today by the fact that all the database has to be presented again in case of adaptation of the application, with time, and size problems. Most of these new fields are for embedded applications, where a powerful workstation cannot be used, for example in automotive. New markets may be the PDAs (Personal Digital Assistants) like the Apple's Newton. They use handwriting recognition, which is currently very poor, but the performances can be increased by using new NN algorithms and a better adaptation to the user. The main problem, apart from the speed, is the power requirement. The CPUs of PDAs use less than 0.3 W to operate, compared to the 5 W of a Intel 486. The final system must remain small and the NN part is not the preponderant one in the system : in this case, small dedicated NN blocks[7], directly linked and embedded in the system seems to be a good response.

5. ...TO DIGITAL NEUROCHIPS

Nowadays, most of usable neural chips are based on an SIMD structure. Other architectures, while remaining interesting research topics, do not seem to be as usable, due to programming problems and to the fact that the most used neural algorithms (like the Error Back Propagation) are based on matrix and vector operations, and not on local operations. MIMD machines are generally limited to some tens of processors for efficient programming and communication, while SIMD machines can still use efficiently more than hundreds.

The neural chip computing elements must also be able to perform other treatments such as image processing,

data processing... which represent a significant part of a whole application.

The ideal chip should have the following characteristics: it should have better performances (through parallelism) than scalar microprocessors, at least for neural oriented applications. It should lead to cheaper systems, by decreasing the system complexity, and by maximizing the silicon area effectively used for the targeted algorithms. A good compromise should be chosen between high efficiency for a particular application (that lead to the development of a dedicated chip), and a "general purpose" ability (i.e. a chip dedicated to NN algorithms and not to one particular application), which allows to decrease the development costs (which are shared by all the applications in which the chip may be used), but may also decrease the efficiency.

The optimization of an architecture must not only be done at the level of the NN function, but at the application level: several NN chips have very high performances in NN processing, but are very inefficient[6] for communication or are badly integrated in the system, which lead to decrease the overall performances. When designing a parallel NN chip, the "Amdahl law"[8] must always be considered.

6. CONCLUSION

The choice of a good architecture for digital neurochips is not obvious and heavily depends on the aim of the system : developing a research tool, accelerators for applications, building blocks in classical systems, building blocks for high performance or real-time systems, for dedicated applications...

Within Philips, several approaches are explored: the use of analog and optical techniques[9], as well as digital realizations. In this category, software implementations of NN on parallel machines (interpretation of infrared spectra on the POOMA machine), realization of NN oriented machines (Philips General Purpose Neuro-Computer) based on Digital Signal Processors (TMS 320C40) and dedicated ASICs (L-Neuro 1.0 and 2.3) were done.

As far as chips are concerned, several approaches are also taken : dedicated chips in coprocessor boards for PCs[10] or in parallel machines[11], development of a high performance general purpose vectorial signal processor, well suited for NN algorithms and fuzzy logic operations[12], or the design of specific blocks, that can be embedded with a classical CPU core. Few mm^2 of NN cells can also be added to non neural chips for performing special functions like reducing signal crosstalk, improving colors images or approximating functions[13].

7. REFERENCES

1) K. Kyuma and J. Ohta, Optical and Electrical Neurochips, in this symposium.
2) U. Ramacher, Development of neural network microelectronics in Europe, in this symposium.
3) L. Tarassenko and A. Blake, Proc. of the IEEE Int. Conf. on Robotics and Automation, Sacramento, (1991) 540.
4) T. Yagi, Image processing in analog systems : What do we learn from the retina ? in this symposium.
5) C. Mead, Analog VLSI and Neural Systems, (Addison-Wesley, 1989).
6) P. Ienne, Architectures for Neuro-Computers, Technical Report n° 93/21, Ecole Polytechnique Fédérale de Lausanne, January 1993.
7) M. Hervieu and J.Y. Brunel, Deeply Embedded Handwriting Recognition, in this symposium.
8) G. Amdahl, AFIPS Conf. Proc., **30** (1967) 483.
9) J. Schleipen, Injection laser Neural Networks, Philips Workshop on Neural Networks and fuzzy Logic, De Ruwenberg, The Netherlands, January 1994.
10) N. Mauduit, M. Duranton, J. Gobert, J.A. Sirat, IEEE Trans. on Neural Networks, **3** (1992) 414.
11) M. Duranton, F. Aglan, N. Mauduit, Computing with T.Node Parallel Architecture, eds. D. Heidrich and J.C. Grossetie (Kluwer Academic Press, 1991) 235.
12) C. Dejean and F. Caillaud, Parallel implementations of Neural Networks using the L-Neuro 2.0 architecture, in this symposium.
13) Y. Deville, Microelectron. J. **24** (1993) 259.