ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

Perspectives on dedicated hardware implementations.

D.Anguita, M.Valle
Department of Biophysical and Electronic Engineering

Via Opera Pia 11A, I-16145 Genova, Italy
{anguita,valle}@dibe.unige.it

Abstract. Algorithms, applications and hardware implementations of neural networks are not investigated in close connection. Researchers working in the development of dedicated hardware implementations develop simplified versions of otherwise complex neural algorithms or develop dedicated algorithms: usually these algorithms have not been thoroughly tested on real-world applications. At the same time, many theoretically sound algorithms are not feasible in dedicated hardware, therefore limiting their success only to applications where a software solution on a general-purpose system is feasible. The paper focuses on the issues related to the hardware implementation of neural algorithms and architectures and their successful application to real world-problems.

1 Introduction

Depending on the circuit design style, dedicated VLSI Neural Networks (DNNs) could be subdivided in two main categories: digital and analog. Digital VLSI NNs (DVNNs) use/operate on digital signals (i.e. signals which can assume only two states, on and off) and over discrete time intervals. Analog VLSI NNs (AVNNs) operate on signals that are graded in their states and continuous in time. Between these two extremes, many VLSI implementations of NNs exist; they can utilize both mechanisms for the coding of information and utilize circuits that operate on digital electrical variables (e.g. for the storage of synaptic values) and others that operate on analog electrical variables (e.g. for the feedforward computations).

DVNNs have been mainly used as hardware accelerators. In principle, using DVNNs it is possible to configure neural architectures of any size with no precision constraints; in practice, silicon area and speed constraints can limit their feasibility.

AVNNs intend to create biologically inspired structured neural systems that perform (specific) computations with high efficiency. Digital and analog VLSI implementations are compared in Table 1.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

Table 1: Digital vs. Analog VLSI Implementations.

	Digital technology	Analog technology
signal representation	numbers (symbols)	physical signals
time	sampling	continuous / sampling
signal amplitude	quantized	continuous / quantized
signal regeneration	along path	degradation
precision	cheap and easy	area and power expensive
area per processing element	large	small
transistor mode of operation	switch mode	all modes
design and test	easy	difficult / expensive

We will try to briefly outline which are main features (i.e. drawbacks and advantages) of both implementation approaches and to identify future trends and promising applications. In Sections 2 and 3 we target, respectively, analog and digital implementations.

2 Analog VLSI implementations of neural networks

Analog VLSI technology looks attractive for the efficient implementation of artificial neural systems for the following reasons.

- Massively parallel neural systems are efficiently implemented in analog VLSI technology: they can achieve high processing speed. The neural processing elements are smaller than their digital equivalent, so it is possible to integrate on the same chip Neural Networks (NNs) composed of a large number (i.e., thousands) of interconnections (i.e., synapses).
- Fault tolerance: to ensure fault tolerance to the hardware level it is necessary to introduce redundant hardware and, in analog VLSI technology, the cost of additional nodes is relatively low.
- Low power: weak inversion operated MOS transistors reduce the power consumption and achieve low power operation.
- Real-world interface: analog neural networks eliminate the need for A/D and D/A converters and can be directly interfaced to sensors and actuators.

The low values of Signal/Noise (S/N) ratio featured by analog implementations of neurons and synapses are not critical since in a neural system the overall precision in the computation is determined not by the single computational nodes, but by the collective computation of the whole network [31].

With reference to the learning phase, we can distinguish four types of analog implementations of NNs [14].

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

- Non-learning neural networks: the synapses have fixed weight values implemented by the size of the transistors of the synaptic modules. It is not possible to change the weight values once the circuit has been realised.
- Off-chip learning networks: the analog circuit implementation performs only the feed-forward phase and has externally adjustable weight values. An host computer with a neural simulator program performs the off-chip learning phase. The matching between the neural simulator program and the analog circuit implementation is improved with respect to the non-learning NNs.
- On-chip learning networks: the NN performs both the feed-forward and the learning phase. The advantages are the high learning speed due to the analog parallel operation and the absence of the interface with a host computer for the weight update. On-chip learning networks are suited to implementing adaptive neural systems, i.e., systems that are continuously taught while been used.
- Biological inspired or neuromorphic NNs [18]: analog implementation of the biological process inherent to the visual and audio perception.

We can identify three hierarchical levels at which analog circuits can play an effective role in learning systems:

- low level: adaptive sensors transduce signals from the environment and extract invariant representation of the external world e.g. silicon retinas and cochleae [25].
- intermediate level: adaptive analog processing implements self-organisation and unsupervised learning. Main tasks at this level are: pattern processing for co-ordinate transformation, decorrelators, principal component analysers, etc.
- high level: non linear mappings between two data spaces for classification and decision making. In the following we will concentrate on this level.

2.1 Analog supervised on–chip learning implementation

Usually, analog circuit implementation of learning algorithms suffers on the drawback of the limited precision of computation. Anyway, experimental results have shown that learning can be achieved inspite of hardware related imperfections (see [9, 8]). The weight adjustment can be achieved by using the Back Propagation (BP, the gradient of the error function is "computed") or the Weight Perturbation (WP) or stocastic error descent [16, 7, 1] (the gradient of the error function is estimated) rules: they adjust the weight values according to the gradient of an error function.

The WP rule looks attractive for the analog on-chip implementation because the learning circuitry is simple [6] and the estimation of the error function

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

gradient is not dependent on the linearity of the transfer function of neurons and synapses. The WP algorithm was developed with the aim of making easier the analog on-chip learning implementation. As a consequence, even if a lot of research activity has been done on the circuit implementation, few papers report on the performance in real applications. On the contrary, BP has been extensively and successfully used to solve real world tasks but it does not fully deal with non-ideality of analog VLSI circuits.

Despite the many advantages, which stem from the analog on-chip learning implementation of NNs, this approach suffers on the fundamental limitations of analog integrated circuits. In fact analog hardware does not suffer the drawback of digital circuits in which the minimum (non-null) signal value is linked to the digital precision in bits; the resolution is limited by noise and mismatch between components. Moreover, analog integrated circuits suffer on temperature variations, processing parameters values spread, device non-linearity, parasitic capacitance, etc. The inherent feedback structure given by learning can in principle compensate for most of the non-ideal effects and errors; though non-ideal behaviour of learning circuits cannot be compensated by the learning feedback.

One of the most critical issues is the effect of zero offsets. VLSI circuits present two kinds of offsets: random and systematic offsets. Random offsets are due to random errors resulting from the limited resolution of the photolitographic process and/or from physical parameter variations randomly distributed over the whole die. Systematic offsets are due to an improper circuit design and/or systematic errors in the photolitographic processing and etching of the wafer and to process gradients (i.e., layers thickness gradients, doping concentration, etc.). It is possible to decrease the effects of random process variations at the expenses of a larger silicon area. On the other hand, systematic offsets can be dealt with by proper circuit design and suitable layout techniques, determining also in this case a trade-off between silicon area and precision.

In a context quite close to the one we are dealing with (LMS algorithms for adaptive filters), it is well known that dc offsets degrade the performance of analog adaptive filters [29]. Then the designer must carefully select the LMS algorithm and its hardware implementation.

In the analog on-chip supervised learning architecture it is necessary to distinguish between offsets in the forward or backward paths. Offsets in the forward path can be easily compensated for by the learning algorithm through the bias synapses weight values [27]. Offsets in the backward path are much more critical: the non-ideal behaviour of learning circuits themselves cannot be fully compensated for by the learning feedback. More precisely, offsets in the weight update circuit can change the value of the sign of the weight update term (when it is small) thus changing the direction of the gradient descent trajectory; this effect likely can prevent from reaching a satisfactory minimum of the error function.

Few results have been presented in the literature concerning the effect of offsets in on-chip learning ANNs. In [10] the effect of offsets in BP learning is

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

investigated: the given results indicate that BP is unable to reach convergence with offsets larger than 10the maximum signal values.

3 Digital implementations of neural networks

Digital hardware cannot compete with analog (or biological) implementations when efficiency of computation is considered, even taking in account the latest advances in Micro- or Digital Signal Processors (μPs or DSPs). Analog hardware is a clear winner in this sense. Let us consider, for example, the projected the power dissipation of DSPs: some experts [13] forecast that in 2004 this technology is supposed to reach an outstanding ratio of 10^{-2} mW/MIPS (Millions of Instructions Per Second). Despite these achievements in digital VLSI, biological neural networks (which are the inspiration of neural analog computation) are far more efficient: a human brain performs on the order of 3.6×10^{15} OPS (Operations Per Second) with a consumption of only 12 W [26] that corresponds to approximately 3×10^{-6} mW/OPS.

Even though digital hardware suffers from power and silicon area consumption problems, it shows its superiority when the physics of analog hardware cannot be readily exploited for implementing the desired neural functions or algorithms. In this case, the greater flexibility and precision of digital hardware overcome easily any advantage of analog implementations.

3.1 State of the art

The research on digital neural implementations seems to have reached its peak at the beginning of '90s when several devices came out from the research laboratories and entered the market. Some of the best–known examples were: Adaptive Solution CNAPS, IBM ZISC, Philips L-Neuro, Siemens SYNAPSE[1]. Unfortunately, after a certain amount of success, most of the commercially available systems disappeared from the market. There are at least two reasons for this brief and not–so–bright success: the first is a direct consequence of the competition between general–purpose μPs and dedicated hardware (we will address this issue in Section 3.2); the second is a decrease of the initial enthusiasm for artificial neural networks and their appication to real–world problems. The loss of interest for the field of artificial neural networks derives both from the maturity reached by the field and the fact that many open problems of the most famous algorithms (e.g. back-propagation) has not been solved (yet): after more than fifteen years of research, some issues like, for example, generalization properties, architecture selection, etc. prevents an effective applications to many problems. We will return on this concept in Section 4.

Despite these difficulties, the research on dedicated hardware has been going on for the last years and some new proposals and solutions are starting to

[1] We omit here the references due to space constraints. The interested reader can find more information on this subject by pointing to the Web page mantained by C.S.Lindsey at CERN: http://www1.cern.ch/NeuralNets/nnwInHepHard.html.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

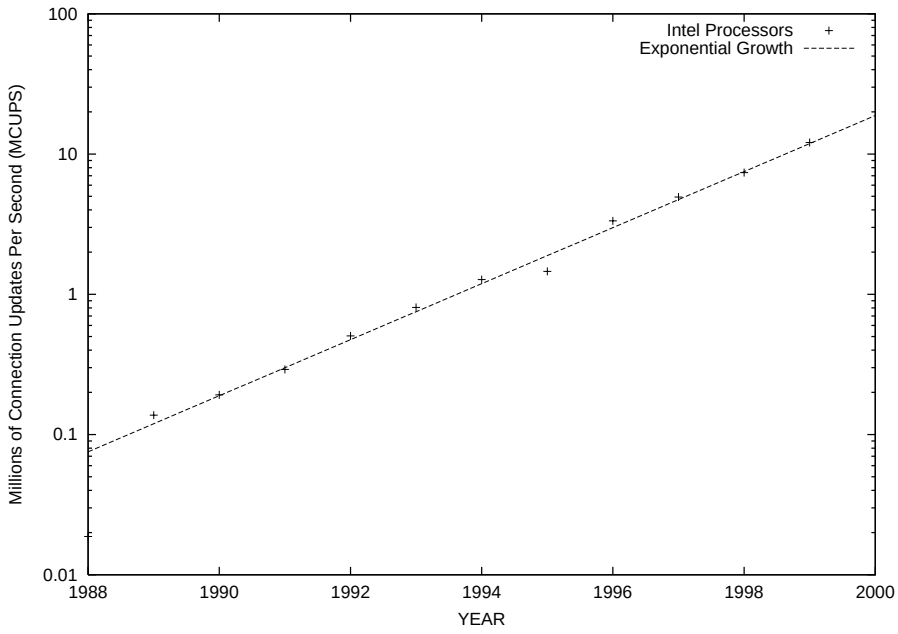

Figure 1: Performance growth of general–purpose Intel microprocessors.

emerge in the literature [11].

3.2 Microprocessors vs. dedicated digital implementations

According to the Moore's Law [5], the number of transistors on a chip doubles every 18 months.

This continuous increase of device density has a direct consequence on the peformance of microprocessors. To confirm this hypothesis, we executed a simple backpropagation algorithm on almost the entire family of Intel microprocessors: from the 80286 (1988) to the Pentium III (1999)[2].

If we plot the computing power, in MCUPS (Millions of Connection Updates per Second), with respect to the year of marketing of each processor, we obtain the graph showed in Fig. 1. As can be easily seen, the computing power of general–purpose μPs, for this particular neural application, has been growing exponentially for the last 12 years. As a further remark, we stress the fact that no compiler improvements during those years nor special instruction sets (e.g. multimedia instructions, beginning from the Pentium MMX series) were

[2]The choice of Intel processors derives from the observation that they are one of the most longstanding and successful microprocessor family. Similar considerations apply to other μPs as well.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

taken in account in our experiment: their use could even increase the slope of the exponential growth. In fact, most supervised and unsupervised learning algorithms make use of three simple operations: additions, multiplications and some nonlinear function computation that, as pointed out by several authors (see for example [3] for some references), can be carried on at low precision. An obvious consequence is that fixed–point math (therefore simpler adders and mutlipliers) can be used for computation, a fact that was exploited by most of the dedicated hardware mentioned in the previous section. (Un)fortunately, fixed–point math is also the basic computational paradigm for multimedia, therefore general–purpose microprocessors have started to include several efficient SIMD (Single Instruction Multiple Data) operations in their instruction sets (now being extended to double precision floating–point on the Pentium IV).

These two aspects have therefore greatly limited the diffusion of dedicated digital hardware in favor of more flexible general–purpose μPs.

4 Discussion and applications

Despite the many (potential) advantages of the analog VLSI implementation of ANNs, some open issues still prevent from their widespread adoption.

- The high costs of design, development and test.
- Large size networks cannot be exhaustively simulated at circuit level due to the high computational burden of available circuit simulators (e.g. SPICE). Nevertheless, the nowadays availability of reliable analog description languages (e.g. analog VHDL) can be a viable way to overcome this drawback.
- An effective and reliable circuit implementation approach for the long term storage of weights; many attempts have been done in the past to design and implement analog memories for VLSI neurocomputing though no reliable and efficient implementation approach emerged. Anyway, reliable CMOS compatible non volatile technologies [22] represent an efficient solution.
- Reliable learning algorithms which can effectively cope with hardware imperfections and non-idealities. A possible solution is the adoption of probabilistic learning algorithms [12].
- The pressing trade-off between silicon area (i.e. cost and size) and performance (i.e. speed, accuracy, power consumption, etc.).
- The still not satisfactory degree of scalability and/or programmability (versus cost and reusability).

Anyway, the major open issue is to identify a real, industrial and/or consumer application which can justify the high design and development costs.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

Given the low power, massive parallelism and adaptativity achievable by analog neural network implementations, potential applications are those where the analog neural network operates directly on raw data coming from sensors or from the field:

- Sensors fusion and linearisation [12, 19] and low level sensory processing tasks (see above).
- Biomedical applications (e.g. implantable devices for cardiovascular deseases [15, 12]) are another interesting field of application which can successfully exploit the low power consumption of analog NNs.
- Telecommunication systems in particular high speed data communications e.g. adaptive equalization of non-linear digital communication channels (see for instance [23, 20]. In this context, the adaptation and classification capabilities of ANNs can effectively cope with the higher data rate (with respect to the equivalent digital solution) at a given power consumption, of analog integrated filters (see for instance [21, 32]).

Regarding digital hardware, and given the premises of previous sections, which are the perspectives in artificial neural networks using this technolgy?

We believe that at least three issues will lead the reserch in this field for the coming years:

- New neural computing paradigms: while the first generation of artificial neural networks was inspired to biological networks, the current research on learning from data has developed new architectures (e.g. Support Vector Machines, Kernel Methods, Gaussian Processes [28]) based on solid statistical foundations [30]. These new architectures and algorithms can provide new insights on supervised and unsupervised learning and new ground for fruitful research.
- New parallel processing requirements: from a computational point of view, these neural architectures are very demanding even though the learning algorithms are simpler than, for example, the well–known BP. The number of operations per learning step grows quadratically with the number of examples and therefore large–scale parallel processing is the obvious method to deal with them [2]. Furthermore, the growing interest in resampling techniques (e.g. Bootstrap) favor the shift toward simple but very copmputer intensive algorithm [17]. At the same time, the digital VLSI technology is providing very flexible and powerful devices (e.g. FPGAs – Field Programmable Gate Arrays) for implementing parallel computation (albeit with low precision arithmetic) that can easily surpass μPs and DSPs as far as computing power is concerned.
- Successful applications: new neural architectures, despite being in their infancy, are starting to show very interesting results in many real–world applications [4, 24].

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 45-56

References

[1] J. Alspector, R. Meir, B. Yuhas, A. Jayakumar, and D. Lippe, *A Parallel Gradient Descent Method for Learning in Analog VLSI Neural Networks* in Advances in Neural Information Processing Systems 5 (NIPS5), 1993, pp. 836–844.

[2] D. Anguita, A. Boni, S. Ridella, *Digital VLSI algorithms and architectures for Support Vector Machines*, Int. J. of Neural Systems, Vol. 10, No. 3, 2000, pp. 159–170.

[3] D. Anguita, S. Ridella, and S. Rovetta, *Worst Case Analysis of Weight Inaccuracy Effects in Multilayer Perceptrons* IEEE Trans. on Neural Networks, Vol. 10, No. 2, 1999, pp. 414–418.

[4] A. Boni, and F. Bardi, Intelligent Hardware for Identification and Control of Non-Linear Systems with SVM. accepted for presentation at ESANN2001, Bruges (Belgium) April 2001.

[5] P.K. Bondyopadhyay, *Moore's Law governs the silicon revolution*, Proceedings of the IEEE, Vol. 86, 1998, pp. 78–81.

[6] G. Cauwenberghs, *An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory*, IEEE Transaction on Neural Networks, Vol. 7, No. 2, 1996, pp. 346–361.

[7] G. Cauwenberghs, *A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization*, in Advances in Neural Information Processing Systems 5 (NIPS5), 1993, pp. 244–251.

[8] G. Cauwenberghs and M. Bayoumi, Eds., *Learning on Silicon - Adaptive VLSI Neural Systems*, Kluwer Academic Publishers, 1999.

[9] G. Cauwenberghs, M. Bayoumi, and E. Sanchez-Sinencio, Eds., *Special Issue on Learning on Silicon*, Analog Integrated Circuits and Signal Processing, Vol. 18, No. 2–3, 1999.

[10] B.K. Dolenko, and H.C. Card, *Tolerance to Analog Hardware of On-Chip Learning in Backpropagation Networks*, IEEE Trans. on Neural Networks, Vol. 6, No. 5, 1995, pp. 1045–1052.

[11] S. Draghici, *Guest Editorial – New Trends in Neural Network Implementations*, Int. J. of Neural Systems, Vol. 10, No. 3, 2000.

[12] P. Fleury, R. Woodburn, and A. Murray, *Matching analogue hardware with applications using the products of experts algorithms*, accepted for presentation at ESANN2001, Bruges (Belgium) April 2001.

[13] G. Frantz, *Digital Signal Processor Trends*, IEEE Micro, Vol. 20, No. 6, 2000, pp. 52–59.

[14] M. Ismail, and T. Fiez, *Analog VLSI Signal and Information Processing*, Mc Graw-Hill International Editors, 1994.

[15] M.A. Jabri, R.F. Coggings, and B.G. Flower, *Adaptive analog VLSI neural systems*, Chapman & Hall, 1996.

[16] M. Jabri and B. Flower, *Weight Perturbation: An Optimal Architecture and learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks*, IEEE Trans. on Neural Networks, Vol. 3, No. 1, 1992, pp. 154–157.

[17] A.K. Jain, R.C. Dubes, and C.C. Chen, *Bootstrap Techniques for Error Estimation*, IEEE Trans. on PAMI, Vol. 9, No. 5, 1987, pp. 628–633.

[18] C. Mead, *Analog VLSI and Neural Systems*, Addison Wesley, Reading Ma, 1989.

[19] N.J. Medrano-Marqus, and B. Martin-del-Brio, *A general method for sensor linearization based neural networks*, IEEE ISCAS 2000, May 28–31, 2000, Geneva, Switzerland, pp. II497–II500.

[20] S.K. Nair and J. Moon, *Data storage channel equalization using neural networks*, IEEE Trans on Neural Networks, vol. 8, No. 5, Sept. 1997, pp. 1037–1048.

[21] J.C. Park, and R. Carley, *High-speed CMOS continuous-time complex graphic equalizer for magnetic recording*, IEEE Journal of Solid State Circuits, Vol. 33, No. 3, March 1998, pp. 427–438.

[22] P. Pavan, R. Bez, P. Olivo, and E. Zanoni, *Flash Memory Cells - an Overview*, Proceedings of the IEEE, Vol. 85, No. 8, 1997, pp. 1248–1271.

[23] C. Petra, et al., *Nonlinear channel equalization for QAM signal constellation using artificial neural networks*, IEEE Trans. on Systems, Man, and Cybernetics, Part B, vol. 29, No. 2, April 1999, pp. 262–270.

[24] T. Poggio, and A. Verri, Eds., *Special Issue on Learning and Vision at the Center for Biological and Computational Learning, Massachusetts Institute of Technology*, Int. J. of Computer Vision, Vol. 38, No. 1, 2000.

[25] A. Rodriguez-Vàzquez, et al., *CMOS Design of Focal Plane Programmable Array Processors*, accepted for presentation at ESANN2001, Bruges (Belgium) April 2001.

[26] R. Sarpeshkar, *Analog Versus Digital: Extrapolating from Electronics to Neurobiology*, Neural Computation, Vol. 10, 1998, pp. 1601–1638.

[27] S. Satyanarayana, Y.P. Tsividis and H.P. Graf, *A Reconfigurable VLSI Neural Network*, IEEE Journal of Solid State Circuits, Vol. 27, No. 1, 1992, pp. 67–81.

[28] B. Shölkopf, C. Burges, and A. Smola, Eds. *Advances in Kernel Methods – Support Vector Learning*, MIT Press 1999.

[29] A. Shoval, D.A. Johns, and W.M. Snelgrove, *Comparison of DC offset Effects in Four LMS Adaptive Algorithms*, IEEE Transaction on Circuits and Systems II, Vol. 42, No. 3, 1995, pp. 176–185.

[30] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[31] E.A. Vittoz, *Analog VLSI Signal Processing: Why, Where, and How ?*, Journal of VLSI Signal Processing, Vol. 8, 1994, pp. 27–44.

[32] X. Wang, and R.R. Spencer, *A low-power 170 MHz Discrete-time analog FIR filter*, IEEE Journal of Solid State Circuits, Vol. 33, No. 3, March 1998, pp. 417–426.