Neural Networks in Hardware: A Survey

Yihua Liao
Department of Computer Science, University of California, Davis
One Shields Avenue, Davis, CA 95616
liaoy@cs.ucdavis.edu

Abstract

Over the past decade a large variety of hardware has been designed to exploit the inherent parallelism of the artificial neural network models. This paper presents an overview of neural network hardware. Neural network basics, hardware specification and performance evaluation are introduced. Major categories of neural network architectures are reviewed. Two examples of neurohardware, CNAPS and SYNAPSE-1, and some real-world applications of neural network hardware, are described in detail. The challenges and future of hardware implementation of neural networks are also discussed.

1 Introduction

Neural network hardware has undergone rapid development during the last decade. Unlike the conventional von-Neumann architecture that is sequential in nature, artificial neural networks (ANNs) profit from massively parallel processing. A large variety of hardware has been designed to exploit the inherent parallelism of the neural network models. Despite the tremendous growth in the digital computing power of general-purpose processors, neural network hardware has been found to be promising in some specialized applications, such as image processing, speech synthesis and analysis, pattern recognition, high energy physics and so on.

Neural network hardware is usually defined as those devices designed to implement neural architectures and learning algorithms, especially those devices that take advantage of the parallel nature inherent to ANNs. A few surveys of neural network hardware have been published [1-6]. Due to the fast growth and huge diversity of neurohardware, these overviews are either outdated or limited on certain aspects of hardware implementations of artificial neural networks. The purpose of this paper is to present the state of the art in neural network hardware architectures and provide a broad view of principles and practice of hardware implementation of neural networks. Neural network hardware specification and classification, various architectures and design issues, latest development and real world applications are reviewed in detail. The future direction of neural network hardware is also discussed. However, the general purpose massively parallel computers, or neurocomputer designs based on other implementation techniques, such as opto-electronics, electro-chemical and molecular techniques, are not within the scope of this survey.

2 Artificial Neuron Model and Neural Network Structures

The study of artificial neural networks has been inspired in part by the observation that biological learning system are built of very complex webs of interconnected neurons. Typically, the human brain consists of approximately 10^{11} neurons, each with an average of 10^{3} - 10^{4} connections. It is believed that the immense computing power of the brain is the result of the parallel and distributed computing performed by these neurons [7]. The transmission of signals in biological neurons through synapses is a complicated chemical process in which specific transmitter substances are released from the sending side of the synapse. The effect is to raise or lower the electrical potential inside the body of the receiving cell. The neuron fires if the potential reaches a threshold. This is the characteristic that the artificial neuron model proposed by McCulloch and Pitts [8] attempts to reproduce. This neuron model is widely used in artificial neuron networks with some variations (Figure 1).

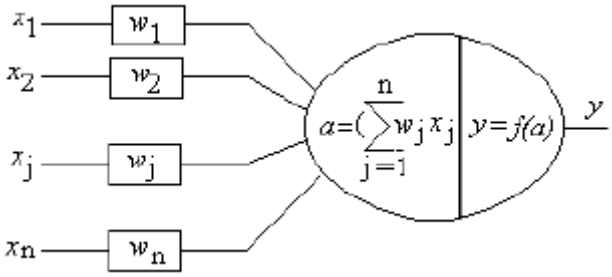

Figure 1: Artificial neuron model

The artificial neuron presented in Figure 1 has N inputs, denoted as x_1, x_2,…, x_n. Each line connecting these inputs to the neuron is assigned a weight, denoted as w_1, w_2,..., w_n, respectively. The action, which determines whether the neuron is to be fired or not, is given by the formula:

$$a = \sum_{j=1}^{n} w_j \, x_j$$

The output of the neuron is a function of its action:

$$y = f(a)$$

Originally the neuron output function *f(a)* proposed in McCulloch-Pitts model was a threshold function. However, linear, ramp and sigmoid functions are also widely used today.

An ANN system consists of a number of artificial neurons and a huge number of interconnections among them. According to the structure of the connections, two different classes of neural network architectures are identified [9](Figure 2).

In layered neural networks, the neurons are organized in the form of layers. The neurons in one layer get input from the previous layer and feed their output to the next layer. This type of network is called *feedforward neural network*. The first and last layers are *input layer* and *output layer* respectively, and the layers that are not input or output are called {\it hidden layers}. Networks with one or more hidden layers are called *multi-layer networks*. *Multi-Layer perceptron* is a well-known feedforward layered

neural network, on which the Backpropagation learning algorithm [10] is implemented.

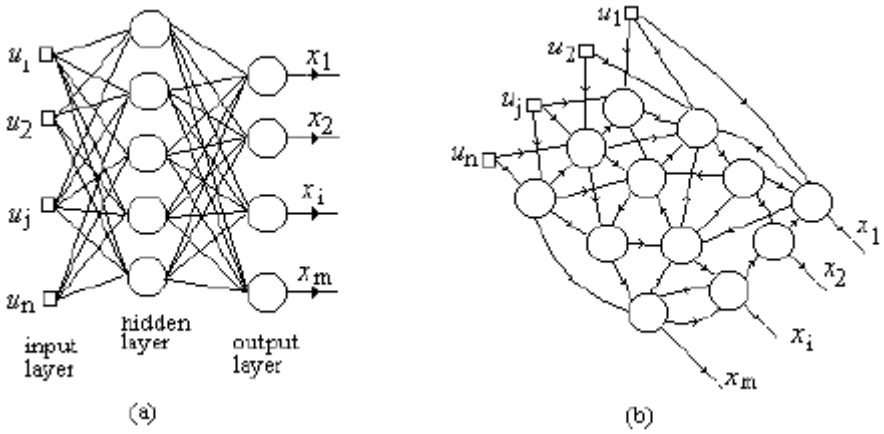

Figure 2: (a) Layered feed forward neural network. (b) Recurrent neural network.

The structure, where connections to the neurons are to the same layer or the previous layers, shown in Figure 2 (b), is called recurrent neural network. Hopfield Neural Network [11] is an example of widely used recurrent networks.

Kohonen's selforganizing map (SOM) is another well-known neural network paradigm introduced by Kohonen[12]. Many other ANN learning algorithms have been proposed, including algorithms for more specialized tasks.

ANN models have been proved to be successful in a number of applications, including text to speech conversion [13], protein structure analysis, autonomous navigation, game playing, image and signal processing, intelligent vision, pattern recognition, etc. These artificial models rely heavily on highly interconnected computational units functioning in parallel.

3 Hardware versus Software

A significant amount of work has been done in developing simulation environments for ANNs on sequential machines. An overview of sequential ANN simulators can be found in [14]. The performance of conventional von-Neuman processors, for example, the Intel Pentium series, continues to improve dramatically. When the particular task at hand does not require super fast speed, most designers of neural network solutions find a software implementation on a PC or workstation with no special hardware add-ons a satisfactory solution. However, even the fastest sequential processor cannot provide real-time response and learning for networks with large numbers of neurons and synapses. Parallel processing with multiple simple processing elements (PEs), on the other hand, can provide tremendous speedups. Some specialized applications have motivated the use of hardware neural networks. For example, cheap dedicated devices, such as those for speech recognition in consumer products, and analog neuromorphic devices, such as silicon retinas, which directly implement the

desired functions.

When implemented in hardware, neural networks can take full advantage of their inherent parallelism and run orders of magnitude faster than software simulations. Section 7 will present some real-world applications of neural network hardware.

In general, neural network hardware designers have taken two different approaches. One is to build a general, but probably expensive, system that can be re-programmed for many kinds of tasks, such as Adaptive Solutions CNAPS [15]. Another approach is to build a specialized but cheap chip to do one thing very quickly and efficiently, such as IBM ZISC [16].

4 Block Representation and Specification

Over the past decade a huge diversity of hardware for ANNs has been designed. Figure 3 presents a block level architectural representation for almost all neuro-chips and neurocomputer processing elements [17].

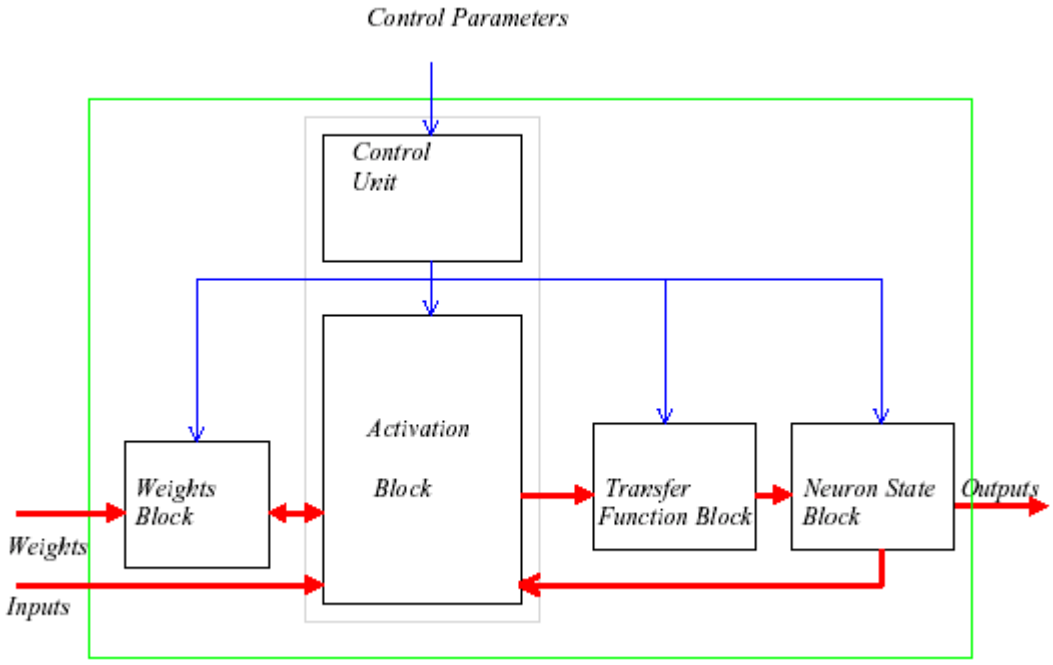

Figure 3: Block level representation for neuro-chips and neurocomputer processing elements after [17].

The activation Block in Figure 3, which performs the multiplications $w_j x_j$ and the summation of these multiplied terms as in Equation (1), is always on the neuro-chip (or the processing element of the neurocomputer). Other blocks, i.e., the Neuron State Block, Weights Block and Transfer Function Block may be on the chip or off the chip, and some of these functions may be performed by a host computer. The data flow between these blocks is controlled by the Control Unit that is always on the

chip. The control parameters are used for controlling the hardware by a host.

The data flow is such that the weights from the Weights Block and the inputs from outside or from the outputs are multiplied and the products are summed in the Activation Block, then the outputs are obtained in the Neuron State Block from the transferred sum of the products. Neuron states and weights can be stored in digital or analog form. Weights can be loaded statically, only once, before the activation computation, or they can be updated dynamically by the host or the Activation Block in the learning phase while activation steps are being performed.

For multi-layer perceptron and Hopfield network (such as [18]) the transfer function may be a threshold, linear, ramp, or sigmoid function. For Kohonen network (for example [19]), what is computed by the Activation Block corresponds to the Euclidean distance between input and weight vectors, and the Activation Block, and the Transfer Function block (in cooperation with the Activation Block) implement the operation of finding the minimal one among all Euclidean distances between input and weight vectors and determining the indexes that denotes the neuron in the Self Organizing Feature Map where the minimum occurred.

Neural network hardware is usually specified by the number of artificial neurons, or processing elements, and the number of connections between them. The number of neurons and number of connections vary from less than 10 to 10^6. Another important parameter is the precision by which the arithmetical units perform the basic operations. The precision is mostly limited to 16-bit fixed point for the weights of a neural network and to 8-bit fixed point for the neuron outputs. In case of the multi-layer perceptron and the standard error backpropagation learning algorithm this precision was shown to be sufficient in most cases [20].However Kohonen's SOM algorithm can learn very well with only 6-bit weights [21]. Recurrent neural networks may require an arithmetical precision of more than 16 bits [22].

The traditional approach for quantifying neural network hardware performance is to measure the number of multiply and accumulate operations performed in the unit time (measured in MCPS or Millions of Connections Per Second) and the rate of Weight updates (measured in MCUPS or Millions of Connection Update Per Second). These two measurements somewhat correspond to the MIPS or the MFLOPS measured on traditional systems. They only serve as indications and have to be compared with care since the implementations differ in precision and size.

Due to the lack of widely available and portable software, no serious effort has been made to develop a comprehensive benchmark suit for neural network hardware. The NETtalk network [13], which translates text to phonemes, is often used for the learning and recall phase of backpropagation networks. Other hardware benchmark proposals have been made in [2] and [23].

5 Classification of Neural Network Hardware

Neural network hardware ranges from single stand-alone neurochips to full-fledged neurocomputers. A variety of attributes have been used to classify neural network hardware, such as system architecture, degree of parallelism, inter-processor communication network, general purpose or special purpose device, on-chip or off-chip learning, and so on. Neural network hardware can be categorized into 4 classes by the degree of parallelism: coarse-grained, medium-grained, fine-grained and massive parallelism [24]. The number of processing elements yields the degree of parallelism of a system. The more parallel units there are, the faster data is processed. However, parallelism is expensive in terms of chip area or chip count. Therefore highly parallel systems usually employ simpler processing elements. The parallelism can be rated from only a few processing elements referred to as coarse-grained up to almost a one-to-one implementation of neural processing nodes called massive. There are no definite borders between these different categories.

Parallel processing elements only speed up the computation when they do not run idle. Thus, for the system performance it is crucial that the inter-processor communication network provides the processing elements with sufficient data. Broadcast bus, linear array, systolic ring, crossbar and bidimensional mesh are the most frequently encountered communication networks of ANN systems [24].

Here we follow the scheme proposed in [5] and group neural network hardware into four main categories as shown in Figure 4.

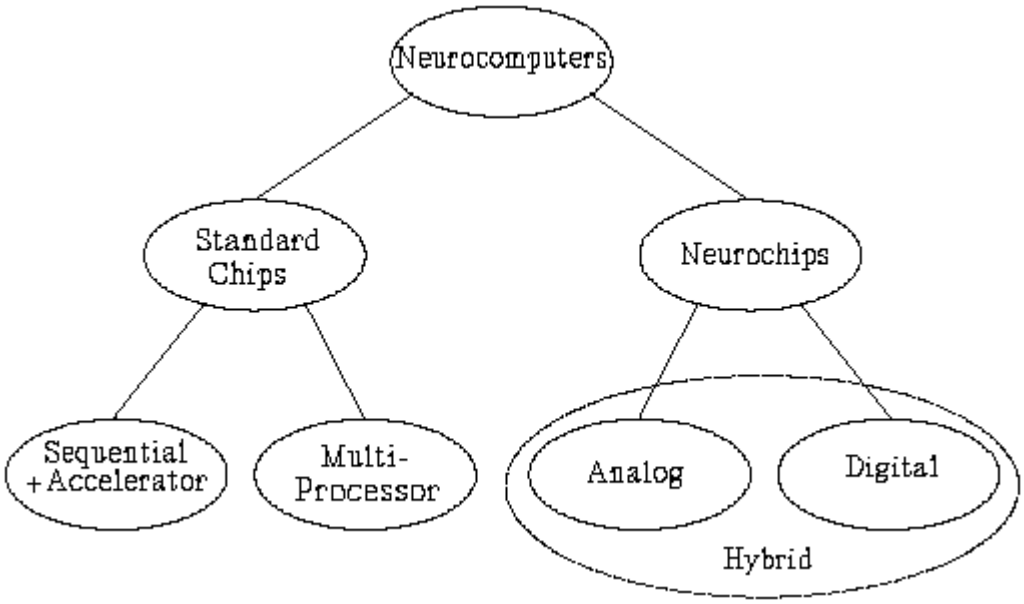

Figure 4: Neural network hardware categories after [5].

The first two main categories consist of neurocomputers based on standard ICs. They consist of

Accelerator boards which speed up a conventional computer like a PC or workstation, and parallel multiprocessor systems, which mostly run stand alone and can be monitored by a host computer. The other main categories are neurochips built from dedicated neural ASICs (Application Specific Integrated Circuits). These neurochips can be digital, analog, or hybrid. The rest of this section will look at each of these categories and discuss their advantages and disadvantages.

5.1 Accelerator Boards

Accelerator boards are the most frequently used neural commercial hardware, because they are relatively cheap, widely available, simple to connect to the PC or workstation, and typically provided with user-friendly software tools. They reside in the expansion slots and are used to speed up the neural network computations. The speed-up that can be achieved is at about one order of magnitude compared to sequential implementations. Accelerator boards are usually based on neural network chips but some just use fast digital signal processors (DSP) that do very fast multiple-accumulate operations. A drawback of accelerator boards is that they are specialized for certain tasks, and thus lack flexibility and do not offer many possibilities for setting up novel paradigms.

A good example of accelerator boards is IBM ZISC ISA and PCI Cards. The ZISC036 chip was developed at the IBM Essonnes Lab [16]. A single ZISC036 holds 36 neurons, or prototypes, to implement an RBF network trained with the RCE (or ROI) algorithm. The ISA card holds to 16 ZISC036 chips, giving 576 prototype neurons. The PCI card holds up to 19 chips for 684 prototypes. PCI card can process 165,000 patterns/sec, where patterns are 64 8-bit element vectors.

Other accelerator systems include SAIC SIGMA-1 [25], Neuro Turbo [26], HNC [27], etc.

5.2 Neurocomputers Built from General Purpose Processors

General-purpose processors offer enough programmability for the implementation of neural functions. These implementations will of course never be maximally efficient. But because of their wide availability and relatively low prices, a number of neurocomputers have been assembled from general-purpose chips. Implementations range from architectures of simple, low-cost elements (for example, the BSP400 [28] and COKOS [29]) to architectures with rather sophisticated processors like transputers, which are unique for their parallel I/O lines [30], or DSPs, which were primarily developed for correlators and discrete Fourier transforms [31]. Much experience has been gained from these implementations, which can be useful for the design of "true" neurocomputers, i.e., dedicated neurocomputers completely built from special purpose elements like neurochips. For instance, in many cases the sigmoid function forms the most computationally expensive part of the neural calculation. A solution for this can be found in using a look-up table rather than calculating the function [32]. Finding an interconnection strategy for large numbers of processors has turned out to be another non-trivial problem. Fortunately, much knowledge about the architectures of these massively

parallel computers can be directly applied in the design of neural architectures.

The RAP (Ring Array Processor) [33] is an example of neurocomputers built from general-purpose processors. It was developed at the ICSI (International Computer Science Institute, Berkeley, CA) and has been used as an essential component in the development of connectionist algorithms for speech recognition since 1990. Implementations consist of 4 to 40 Texas Instruments TITMS320C30 floating point DSPs containing 256 Kbytes of fast static RAM and 4 Mbytes of dynamic RAM each. These chips are connected via a ring of Xilinx programmable gate arrays (PGAs), each implementing a simple two-register data pipeline. Additionally each board has a VME bus interface logic, which allows it to connect to a host computer. The software support of RAP contains a workstation based command interpreter, tools for the standard C environment and a library of matrix and vector routines. A single board can perform 57 MCPS when computing a multi-layer perceptron network in forward operation, and 13.2 MCPS with backpropagation training.

5.3 Neurochips

For neurocomputers in Section 5.2 the neural functions are programmed on general-purpose processors. Dedicated circuits are devised in special purpose chips for the neural functions. This will speed up the neural iteration time by about 2 orders of magnitude compared to general-purpose processor implementations. Several implementation technologies can be chosen for the design of neurochips. The main distinction lies in choice of a fully digital, fully analog, or hybrid design. Direct implementation in circuits in many cases alters the exact functioning of the original (simulated or analyzed) computational elements. This is mainly due to limited precision. The influence of this limited precision is of great importance to the proper functioning of the original paradigm. In order to build large-scale implementations, many neurochips have to be interconnected. Some chips are therefore supplied with special communication channels. Other neurochips are to be interconnected by specially designed communication elements.

5.3.1 Digital Neurochips

Digital Neural ASICs are the most powerful and mature neurochips. Digital techniques offer high computational precision, high reliability, and high programmability. Furthermore, powerful design tools are available for digital full- and semi-custom design. Disadvantages are the relatively large circuit size compared to analog implementations. Synaptic weights can be stored on or off chip. This choice is determined by the trade-off between speed and size.

Section 6 will discuss two well-known digital Neurochips, CNAPS [15] and SYNAPSE-1 [34], in much detail. Unlike CNAPS and the SYNAPSE which were designed for a wide range of neural network algorithms, the NESPINN (Neurocomputer for Spiking Neural Networks), designed at the Institute of Microelectronics of the Technical University of Berlin, is optimized more strictly to a

certain class of neural networks: spiking neural networks. Spiking neural networks model neurons on a level relating more closely to biology. They do not only incorporate synaptic weighting, postsynaptic summation, static threshold and saturation, but also computation of membrane potentials, synaptic time delays and dynamical thresholds. One NESPINN-Board is designed to compute about 10^5 programmable neurons in real-time [35].

5.3.2 Analog Neurochips

Analog electronics have some interesting characteristics that can directly be used for neural network implementation. Operational amplifiers (Opamps), for instance, are easily built from single transistors and automatically perform neuron-like functions, such as integration and sigmoid transfer. These otherwise computationally intensive calculations are automatically performed by physical processes such as summing of currents or charges. Analog electronics are very compact and offer high speed at low energy dissipation. With current state-of-the-art micro electronics, simple neural (non-learning) associative memory chips with more than 1000 neurons and 1000 inputs each can be integrated on a single chip performing about 100 GCPS.

Disadvantages of analog technology are the susceptibility to noise and process-parameter variations that limit computational precision and make it harder to understand what exactly is computed. Chips built according to the same design will never function in exactly the same way.

Apart from the difficulties involved in designing analog circuits, the problem of representing adaptable weights is limiting the applicability of analog circuits. Weights can for instance be represented by resistors, but these are not adaptable after the production of the chips. Chips with fixed weights can only be used in the recall phase. Implementation techniques that do allow for adaptable weights are: capacitors, floating gate transistors, charge coupled devices (CCDs), etc [1]. The main problems with these techniques arise from process-parameter variations across the chip, limited storage times (volatility), and lack of compatibility with standard VLSI processing technology. The weight sets for these train-able chips are obtained by training on a remote system (PC or workstation) and are then downloaded onto the chip. Then another short learning phase can be carried out in the chip used for the forward phase, and the remote system updates the weights until the network stabilizes. This yields a weight matrix that is adjusted to compensate for the inevitable disparities in analog computations due to process variance. This ``chip in loop'' method has been used for Intel's analog ETANN chip [36]. It should be clear that these chips are suited for many different applications, but do not allow for on-board training.

In order to get the benefits of fast analog implementation and the adaptability properties of neural networks, one has to implement learning mechanisms on the chip. Only then can the adaptive real-time aspects of neural networks be fully exploited. However, the implementation of most learning rules into

analog VLSI turns out to be very hard. One of the problems in multi-layered networks is that the target values of the hidden nodes are not defined. The backpropagation method gets around this by passing error signals recursively backwards from the output layer, estimating the effect of intermediate weight changes on each error signal via a relatively complex backwards pass. Information is non-local, which renders extra difficulties for implementation. In order to overcome these difficulties many research groups are investigating learning methods that better suit implementation in analog circuits. Most proposed methods use the so-called weight perturbation technique that only requires a feed forward phase. These methods have proved to be quite successful [37, 38].

Although analog chips will never reach the flexibility attainable with digital chips, their speed and compactness make them very attractive for neural network research, especially when they adopt the adaptive properties of the original neural network paradigms. A final promising advantage is that they more directly interface with the real, analog world, whereas digital implementations will always require fast analog-to-digital converters to read in world information and digital-to-analog converters to put their data back into the world.

Besides the Intel ETANN chip, other fully analog chips include [39], [40], etc.

5.3.3 Hybrid Neurochips

Both digital and analog techniques offer unique advantages, as was discussed in the former sections but they also have drawbacks with regard to their suitability for neural network implementations. The main shortcomings of digital techniques are the relative slowness of computation and the large amount of silicon and power that is required for multiplication circuits. Shortcomings of analog techniques are, for instance, the sensitivity to noise and susceptibility to interference and process variations. The right mixture of analog and digital techniques for the implementation of these processes will be very advantageous. In order to gain advantages of both techniques, and avoid the major drawbacks, several research groups have implemented hybrid systems.

The ANNA (Analog Neural Network Arithmetic and Logic Unit) chip was designed at AT&T Bell Labs. It can be used for a wide variety of neural network architectures (see [41] for an OCR application) but is optimized for locally connected, weight-sharing networks and time-delay neural networks (TDNNs). Synaptic weights are trained off chip, quantized to the chip's resolution, and then downloaded into the chip's weight memory. They are represented by voltages. The interface to the chip is purely digital with two on-chip DACs converting the 6-bit digital weight values into the appropriate voltages. The system board for the ANNA chip is provided by a floating point DSP-32C for the learning process and calculation of the output layer of the backpropagation network. The ANNA chip comprises 4096 synapses and 8 linear neurons, and can handle up to 256 neural state inputs. Performance: 5000 MCPS (peak), 1000 to 2000 MCPS (average).

The Epsilon [42] (Edinburgh Pulse Stream Implementation of a Learning Oriented Network) chip is a hybrid neurochip that uses pulse coding techniques. In pulse coding techniques, the analog neural states are represented as sequences of pulses. This offers a number of advantages with regard to power consumption, calculations and their propagation.The Epsilon chip consists of 30 nodes and 3600 synaptic weights, and can be used both as a ``save'' accelerator to a conventional computer and as an ``autonomous'' processor. With this chip it has been shown that it is possible to implement robust and reliable networks using the pulse stream technique. Performance can achieve 360 MCPS.

A more recent neurochip that uses pulse stream technique is The PDM (Pulse Density Modulating) digital neural network system [43]. It is a neural network hardware that can simulate feedback and feedforward neural networks in a fully parallel and continuous manner. Analog output from each neuron is transmitted by a pulse stream whose frequency is proportional to the output. In total, there are 1,008 neurons and 1,028,160 synapses in the system.

6 Case Studies

6.1 CNAPS

One of the most well known commercially available neurocomputers is the CNAPS (Connected Network of Adaptive Processors) [15] from Adaptive Solutions. The basic building block of the CNAPS system is the neurochip N6400. As shown in Figure 5, the N6400 itself consists of 64 processing elements (referred to as processing nodes PN) that are connected by a broadcast bus in a SIMD (Single Instruction Multiple Data) mode. Two 8-bit buses allow the broadcasting of input and output data to all PNs.

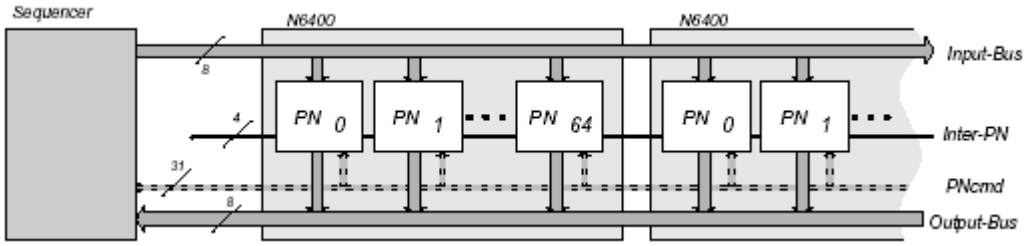

Figure5: SIMD-Architecture of the CNAPS

One of the big advantages of the CNAPS architecture is the scalability of the system: due to the broadcast bus, inter-processor communication and the SIMD mode, additional N6400 chips can be easily added as depicted in Figure 5. The standard CNAPS system consists of a common sequencer chip and four processor chips (systems with up to eight chips and altogether 1064 PNs are available). The regularity of the broadcast bus structure is exploited also in another way in the CNAPS system: the N6400 die measures about one square inch with more than 13 million transistors integrated. The yield is kept at an acceptable level by introducing redundancies and reconfiguring faulty elements after

fabrication. Out of 80 PNs integrated 64 PNs are used after test and reconfiguration, resulting in a 90% yield.

The PNs are designed like simple DSPs including fixed-point adder and multiplier. Each PN is equipped with 4-KByte local on-chip SRAM that needs to hold the weights. The size of the local memory is the bottleneck for large networks: once the connectivity cannot be stored locally anymore a communication via the broadcast bus becomes necessary. Of course the system performance drops dramatically when 64 PNs try to communicate over two 8bit buses. Since the two data buses do not allow an efficient communication between PNs, networks must be mapped n-parallel onto the CNAPS. When employing backpropagation learning each processing node has to store not only the weight matrix but also the inverse as well. In that case the size of networks the CNAPS architecture can handle is smaller.

However, the versatile character of the PN provides programmability for a broad range of algorithms: the possibility of implementing several algorithms including backpropagation and Kohonen self-organizing feature maps as well as image processing algorithms. Also, for convenient programming CNAPS tools include a C-compiler with extensions to take full advantage of the parallel architecture. According to the previously mentioned taxonomy of ANN hardware, one would consider the complete CNAPS system a neurocomputer built of neurochips. However, the N6400 has also been used to build accelerator boards.

6.2 SYNAPSE-1

Siemens' MA-16 neurochip is the basic building block for the neurocomputer SYNAPSE-1 (Synthesis of Neural Algorithms on a Parallel Systolic Engine). MA-16 is designed for fast 4x4 matrix operations with 16-bit fixed-point precision [34]. Multiple MA-16 chips can be cascaded to form systolic arrays. This way inputs and outputs are passed from one MA-16 chip to another in a pipelined manner ensuring an optimal throughput as shown in Figure 6.

The SYNAPSE-1 consists of eight MA-16 chips connected in two parallel rings controlled by two Motorola MC68040 processors. Weights are stored in an off-chip DRAM that amounts to 128 MByte and can be further expanded up to 512 MByte. The neural network is mapped sp-parallel for the forward phase and np-parallel for the learning phase. The neuron transfer functions are calculated off-chip using look-up tables. Especially the high capacity of the on-line weight memory qualifies the SYNAPSE-1 for complex applications. Like the CNAPS, SYNAPSE-1 is not dedicated to specific algorithms. Several networks have been mapped onto SYNAPSE-1, e.g. backpropagation and Hopfield networks. In opposite to the simple SIMD architecture of the CNAPS, programming the SYNAPSE-1 is difficult. The fairly complex processing elements and the 2-dimensional structure of the systolic array hinder a straightforward programming even though a neural Algorithmic Programming Language

is available.

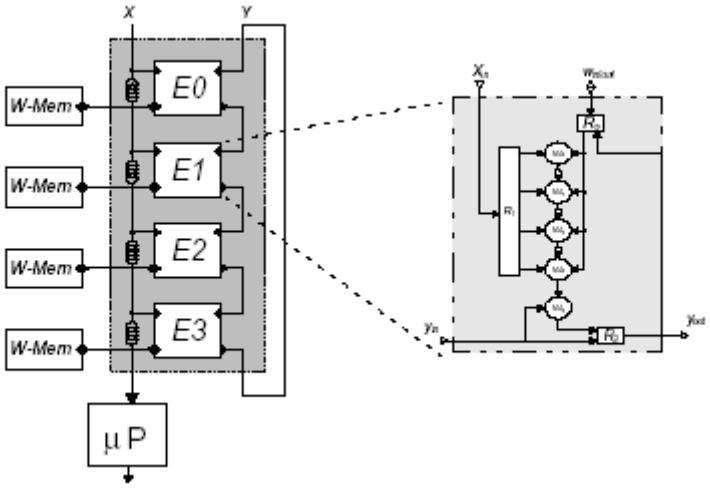

Figure 6: Scalar product chain of the MA-16 chip

7 Neural Network Hardware Applications

Neural network hardware is appearing in ever increasing numbers of real world applications and are making real money. This section illustrates its applications in Optical Character Recognition (OCR), speech recognition, neuromorphic systems and high energy physics.

7.1 OCR

OCR has become one of the biggest commercial applications of neural networks. Nowadays a purchase of a new scanner typically includes a commercial OCR program. To turn a picture of text into a text file, a dozen or more steps must be completed successfully by the OCR program, including cleaning up image, segmenting characters, extracting features, classifying and verifying characters, and so on. Most OCR programs choose to accomplish one or more of these steps with ANNs while using for other steps other techniques such as conventional AI (If-Then rules), statistical models, hidden Markov models, etc.

OCR neural network hardware illustrates two extremes: for high throughput, special high performance hardware is required; for consumer products a cheap dedicated chip would be needed. Adaptive Solutions form and image capture systems [44] exemplifies the first case. A large and elaborate high end OCR system is designed for high speed and high volume processing of forms. Ligature Ltd. OCR-on-a-Chip [45] illustrates the second case. OCR-on-a-Chip is a powerful OCR tool that provides reading capabilities to any machine or integrated system without the need for a PC or extensive memory. The first product on the market featuring Ligature's OCR-on-a-Chip technology is Wizcom's Quicktionary, a hand-held pen-scanner that uses 64K RAM to scan and translate text clips.

7.2 Speech Recognition

Sensory Inc. has specialized in neural chips for speech recognition [46]. The chips cost only a few dollars. The chips recognize a limited vocabulary, e.g. 10-100 words, can be either speaker independent or dependent. They are intended for consumer applications such as cell phones, toys, etc. They involve preprocessing of the raw acoustic signal into a rate and distortion-independent representation that is fed into the neural network. The neural network is structured to perform nonlinear Bayesian classification. Training data consists of a large corpus of 300-600 voice samples representative of potential application users.

7.3 Neuromorphic Hardware

NeuroMorphic refers to systems that closely follow the structure and functions of biological neural systems, such as: silicon retinas and analog cochlears [47]. Such devices are mostly analog, particularly at the front-end sensor stage. One commercially successful product is Synaptics Touchpad [48]. It is a small, touch-sensitive pad that senses the position of a person's finger on its surface to provide screen navigation, cursor movement, and a platform for interactive input. Synaptics Touchpad uses ideas from retina and touch research, especially the way that a neuron's output is influenced by its connections to other neurons nearby. The Synaptics TouchPad can be used in a wide variety of applications that require a thin, robust, accurate, and easy to use input and navigation device. A neuromorphic device like the touchpad does more of the front end processing with analog circuits before the conversion to digital and so reduces the bandwidth required.

7.4 High Energy Physics Online Filter

High energy physics experiments involve the collision of sub-atomic particles, such as protons with electrons, in particle accelerators. The particles emitted in the debris are detected in enormous sensors that surround the collision region. Most collisions are glancing ones and usually do not produce anything interesting. Collision rates can exceed 100s of MHz rates so sophisticated online filters must reject most of the "events" and only record those likely to be of interest. Hardware neural networks have been used to classify patterns in less than 10ms [49].

At the Fermilab Tevatron proton-antiproton collider, the analog Intel ETANN chip, classified energy deposits in a calorimeter as either from electrons or gamma rays for the CDF experiment [50].

8 Discussion

Among the challenges neural network hardware faces today the competition with general-purpose hardware is probably the toughest one: computer architecture is a highly competitive domain that advances at an incredible pace. Neural networks in software have become well-established money making tools in a diverse range of pattern recognition and AI applications. The area of ANN hardware on the other hand is not yet as commercialized as general-purpose hardware. Also neural networks

hardware tends to be more algorithm-specific. This requires a good knowledge about algorithms as well as system design and leads to a high time-to-market. Therefore, general-purpose computers can profit more often from advances in technology and architectural revisions. Also, in many other respects general-purpose hardware seems to be more user-friendly: it is not bound to algorithmic a-priori-assumptions and therefore offers high flexibility. Uniform programming interfaces exist for general-purpose hardware. This can be important not only to get a better start when programming a system, but also to allow reusability when moving on to the next hardware generation.

On the other hand, there are ANN problems, exceeding the computational capabilities of workstations or PCs such as real-time applications, the simulation of large networks or networks employing very complex neuron models. For these applications neurohardware is attractive. Other niche areas for neural hardware are embedded applications of simple, hardwired networks, for example, voice recognition chips, and neuromorphic systems that directly implement a desired function, such as touchpad and silicon retinas. Neurohardware might provide a much better cost-to-performance ratio, lower power consumption and smaller size.

The field of neural network hardware has become maturer since it's ``gold rush" period in late 1980s and early 1990s. Clearly an algorithmic success in artificial neural networks would revive the area of neurohardware. As long as conventional hardware can not provide sufficient performance, there is a need for neural network hardware.

References

[1] Schwartz, T.J., 1990, A Neural Chips Survey, *AI Expert*, 5, 12, 34-39, 1990.

[2] Ienne, P., 1994, *Architecture for Neuro-Computers: Review and Performance Evaluation*, Technical Report no. 93/21, Microcomputing Laboratory, Swiss Federal Institute of Technology, Lausanne, 1994.

[3] Glesner, M. and Pochmuller, W., 1994, *An Overview of Neural Networks in VLSI*, Chapman & Hall, London, 1994.

[4] Lindsey, C. and Lindblad, T., 1994, Review of Hardware Neural Networks: A User's Perspective. *Proceeding of 3rd Workshop on Neural Networks: From Biology to High Energy Physics*, Isola d'Elba, Italy, Sept. 26-30, 1994.

[5] Heemskerk, J. N. H., 1995, Overview of Neural Hardware. *Neurocomputers for Brain-Style Processing. Design, Implementation and Application*, PhD Thesis, Unit of Experimental and Theoretical Psychology, Leiden University, the Netherlands.

[6] Misra, M., 1997, Parallel Environment for Implementing Neural Networks. *Neural Computing Survey*, Vol. 1, 48-60, 1997.

[7] Rumelhart, D. E., McClelland, J. L. and the PDP Research Group, 1986, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vol. 1, MIT Press, Cambridge, Massachusetts, 1986.

[8] McCulloch, W. S. and Pitts, W., 1943, A Logical Calculus of the Ideas Immanet in Nervous Activity. *Bulletin of Mathematical Biophysics*, vol. 5, 115-133, 1943.

[9] Gelenbe, E. and Halici U., 1994, *Lecture Notes on Neural Networks*, METU.

[10] Rumelhart, D. E. and McClelland, J. L., 1986, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition (Vols. 1&2)*. Cambridge, MA: MIT Press.

[11] Hopfield, J. J., 1982, Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences USA*, 79, 2554-2558.

[12] Kohonen, T., 1984, *Selforganization and Associative Memory*, Springer-Verlag.

[13] Sejnowski, T. J. and Rosenberg, C. R., 1987, Parallel Networks That Learn to Pronounce English Text. *Complex Systems*, 1:145-168, 1987.

[14] Murre, J. M. J., 1995, Neurosimulators, In Arbib, M. A. editor, *Handbook of Brain Research and Neural Networks*, MIT Press 1995.

[15] McCartor, H., 1991, A Highly Parallel Digital Architecture for Neural Network Emulation. In Delgado-Frias, J. G. and Moore, W. R. (eds.), *VLSI for Artificial Intelligence and Neural Networks*, 357-366, Plenum Press, New York, 1991.

[16] Lindsey, C. S., Lindblad, Th., Sekniaidze, G., Minerskjold, M., Szekely, S., and Eide, A., 1995, Experience with the IBM ZISC Neural Network Chip. *Proceedings of 3rd Int. Workshop on Software Engineering, Artificial Intelligence, and Expert Systems, for High Energy and Nuclear Physics*, Pisa, Italy, April 3-8, 1995.

[17] Aybay, I., Cetinkaya, S. and Halici, U., 1996, Classification of Neural Network Hardware. *Neural Network World*, IDG Co., Vol. 6 No. 1, 11-29, 1996.

[18] Abramson, D., Smith, K., Logothetis, P. and Duke, D., 1998, FPGA Based Implementation of a Hopfield Neural Network for Solving Constraint Satisfaction Problems. *Proceedings of Workshop on Computational Intelligence of the 24th Euromicro Conference*, Vasteras, Sweden, August 25th-27th, 1998.

[19] Speckmann, H., Thole, P. and Rosenstiel, W., 1993, Hardware Synthesis for Neural Networks from a Behavioral Description with VHDL. Proceedings of International Joint Conference on Neural Networks, Nagoya, 1993.

[20] Holt, J. and Hwang, J., 1993, Finite Precision Error Analysis of the Neural Network Hardware Implementations. *IEEE Trans. on Computers*, 42:281-290, 1993.

[21] Thiran, P., Peiris, V., Heim, P. and Hochet, B., 1994, Quantization Effects in Digitally Behaving Circuit Implementations of Kohonen Networks. *IEEE Trans. on Neural Networks*, 5(3):450-458, 1994.

[22] Strey, A. and Avellana, N., 1996, A New Concept for Parallel Neurocomputer Architectures. *Proceedings of the Euro-Par'96 Conference*, Lyon (France), Springer LNCS 1124, Berlin, 470-477, 1996.

[23] Van Keulen, E., Colak, S., Withagen, H., and Hegt, H., 1994, Neural Network Hardware Performance Criteria. {\it Proceedings of IEEE International Conference on Neural Networks}, 1885-1888, 1994.

[24] Schoenauer, T., Jahnke, A., Roth, U. and Klar, H., 1998, Digital Neurohardware: Principles and Perspectives. *Proceedings of Neuronal Networks in Applications (NN'98)*, Magdeburg, 1998.

[25] Treleaven, P. C., 1989, *Neurocomputers. International Journal of Neurocomputing*, 1, 4-31, 1989.

[26] Arif, A. F., Kuno, S., Iwata., A. and Yoshita, Y., 1993, A Neural Network Accelerator Using Matrix Memory with Broadcast Bus. *Proceedings of the IJCNN-93-Nagoya*, 3050-3053, 1993.

[27] HNC, 1993, High-Performance Parallel Computing. *SIMD Numerical Array Processor*, Data Sheet, San Diego.

[28] Heemskerk, J.N.H., Hoekstra, J., Murre, J.M.J., Kemna, L.H.J.K. and Hudson, P.T.W., 1994, The BSP400: A Modular Neurocomputer. *Microprocessors and Microsystems*, 18, 2, 67-78, 1994.

[29] Speckman, H., Thole, P. and Rosentiel, W., 1993, COKOS: A Coprocessor for Kohonen's Selforganizing Map. Proceedings of the ICANN-93-Amsterdam, London: Springer-Verlag, 1040-1045, 1993.

[30] Foo, S. K., Saratchandran, P. and Sundararajan, N., 1993, Parallel Implementation of Backpropagation on Transputers. *Proceedings of the IJCNN-93-Nagoya*, 3058-3061, 1993.

[31] Onuki, J., Maenosono, T., Shibata, M., Iima, N., Mitsui, H., Yoshida, Y. and Sobne., M., 1993, ANN Accelerator by Parallel Processor Based on DSP. *Proceedings of the IJCNN-93-Nagoya*, 1913-1916, 1993.

[32] Shams, S. and Gaudiot, J., 1992, Efficient Implementation of Neural Networks on the DREAM Machine. *Proceedings of the 11th International Conference on Pattern Recognition*, The Hague, The Netherlands, 204-208, 1992.

[33] Morgan, N., Beck, J., Kohn, P., Bilmes, J., Allman, E. and Beer, J., 1992, The Ring Array Processor: A Multiprocessing Peripheral for Connectionist Applications. *Journal of Parallel and Distributed Computing*, 14, 248-259, 1992.

[34] Ramacher, U., Raab, W., Anlauf, J., Hachmann, U., Beichter, J., Bruls, N., Webeling, M. and Sicheneder, E., 1993, Multiprocessor and Memory Architecture of the Neurocomputers SYNAPSE-1. *Proceedings of the 3rd International Conference on Microelectronics for Neural Networks (Micro Neuro)*, 227-231, 1993.

[35] Jahnke, A., Roth, U. and Klar, H., 1996, A SIMD/Dataflow Architecture for a Neurocomputer for Spike-Processing Neural Networks (NESPINN). *Proceedings of the 6th International Conference on Microelectronics for Neural Networks (Micro Neuro*), 232-237, 1996.

[36] Tam, S., Gupta, B., Castro, H. and Holler, M., 1990, Learning on an Analog VLSI Neural Network Chip. *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, 1990.

[37] Jabri, M. and Flower, B., 1991, Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multi-Layer Networks. *Neural Computation*, 3, 546-565, 1991.

[38] Woodburn, R., Reekie, H.M. and Murray, A.F., 1994, Pulse-Stream Circuits for On-Chip Learning in Analogue VLSI Neural Networks. *Proceedings of the IEEE International Symposium on Circuits and Systems*, London, 103-106, 1994.

[39] Maeda, Y., Hirano, H. and Kanata, Y., 1993, AN Analog Neural Network Circuit with a Learning Rule via Simutaneous Perturbation. *Proceedings of the IJCNN-93-Nagoya*, 853-856, 1993.

[40] Withagen, H., 1994, Implementing Backpropagation with Analog Hardware. *Proceedings of the IEEE ICNN-94-Orlando Florida*, 2015-2017, 1994.

[41] Sackinger, E., Boser, E.B., Bromley, J., LeCun, Y. and Jackel, L.D., 1992, Application of the ANNA Neural Network Chip to High Speed Character Recognition. *IEEE Transactions on Neural Networks*, 3, 3, 498-505, 1992.

[42] Churcher, S., Baxter, D.J., Hamilton, A., Murray, A.F. and Reekie, H.M., 1992, Generic Analog Neural Computation-the Epsilon Chip. Advances in Neural Information Processing Systems: Proceedings of the 1992 Conference}, Denver, Colorado.

[43] Hirai, Y. 1998, A 1,000-Neuron System with One Million 7-bit Physical Interconnections. In Jordan, M.I., Kearns, M.J. and Solla, S.A. eds. *Advances in Neural Information Processing Systems 10*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, pp.705-711, 1998.

[44] Adaptive Solutions, http://www.asi.com/

[45] Ligature Ltd. OCR-on-a-Chip, http://www.ligatureltd.com/products/ocr.html

[46] Sensory Inc., http://www.sensoryinc.com/

[47] Caltech center for Neuromorphic Systems Engineering, http://www.erc.caltech.edu/

[48] Synaptics Touchpad, http://www.synaptics.com/products/touchpad.cfm

[49] Neural Networks in HEP, http://www1.cern.ch/NeuralNets/nnwInHepHard.html.

[50] The Collider Detector at Fermilab, http://www-cdf.fnal.gov/.