



A Survey on Hardware Neural Network (HNN)

Raghad Abduljabbar
Abdulhameed

Altinbas University
Istanbul, Turkey
213720323@ogr.altinbas.edu.tr

Abdullahi Abdu Ibrahim

Altinbas University
Istanbul, Turkey
Abdullahi.Ibrahim@altinbas.edu.tr

ABSTRACT

HNN covers academic and commercial examples of Hardware Neural Network (HNN) prototypes to provide a comprehensive overview of AI hardware implementations (ANNs). Although HNN research has been continuing since the 1990s, commercialization has been slow. To assess the field's state, we look at the most important ANN models, hardware design methodologies, and applications. For a model to be useful in a broad variety of applications, it must be mapped onto reliable and energy-efficient hardware. Researchers are researching entire ANN models on chips (digital, analog, hybrid, and FPGA-based) and in the brain. Spiking neural network hardware, cell-based neuron implementations, and programmable FPGA-based solutions are all detailed. Associative neural memory, Additionally, bit-slice and SIMD architectures will be examined for RAM-based parallel digital implementations. Consider the current status of research and forecast the future.

Keywords:

HMM, SIMD, ANN, FPGA.

1. Introduction

1.1 TOPIC DEFINITION

An HNN is a device that implements artificial neural network (ANN) topologies and learning algorithms, taking use of the inherent parallelism of neuronal activity. To train and test neural networks, HNNs, or hardware neural networks, are a kind of artificial neural network (HNN). Energy-efficient neural network hardware capable of fully parallel

processing is required for certain applications, such as streaming video compression. [1] A unique kind of ANN hardware (which may augment or replace software) is required since computers are involved.

There are several advantages to this. Many cellular neural networks (CNNs) have been implemented on VLSI chips, which are capable of running faster than standard DSPs, computers, or even workstations.

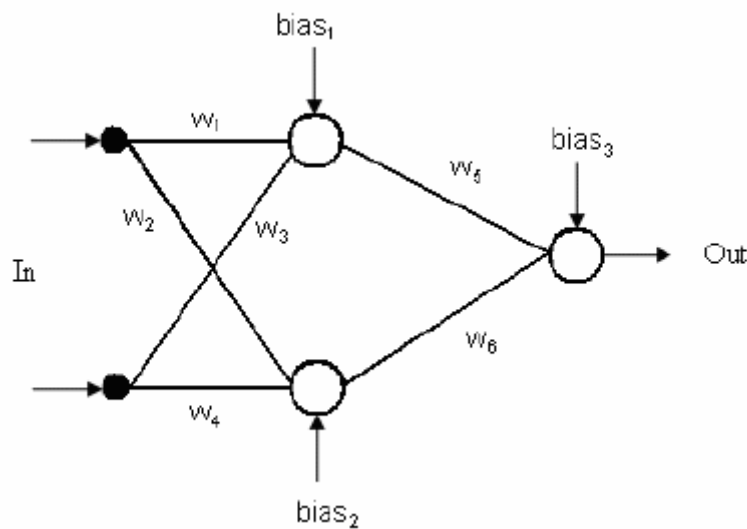


Figure 1: The hardware implementation of an ANN

Limiting the number of components and power needed may lower hardware implementation costs. This is crucial for high-volume, low-cost applications like consumer devices that process images in real-time.

1.2 Problem Statement

Sequential uniprocessor-based applications are prone to faults, which might cause them to stop working. This is a limitation of sequential uniprocessor programs (fail-stop operations). The CPU's design lacks redundancy. Despite the advent of multi-core personal computer processing architectures, effective fault-tolerant solutions are still necessary to keep systems running. Unlike traditional designs, parallel and distributed architectures allow programs to continue functioning even if individual components fail. Parallel hardware solutions give substantial benefits for ANN applications that need high availability or security. It is difficult for (VLSI) HNN designers to map irregular and non-planar network architectures. This involves expensive computations and scattered communication, which is problematic. Hardware restrictions (especially analog components) may induce computational mistakes, preventing learning and leading in inaccurate outputs. Incorrect learning paths increase the number of cycles required to achieve convergence. Design issues occur because to non-linear activation functions. This topic has been explored earlier using various technologies and computer

systems. Real-world applications need more than just an ANN model HNN; they require sensor gathering, pre- and post-processing of inputs and outputs, etc. HNNs are utilized in applications, although not as often as ANNs

1.2 Motivation

High-energy physics experiments (using Adaptive Solutions CNAPS boards for online data filtering and Level II triggers in the H1 electron-proton collision experiment) and robotics are just a few examples. These new technologies have raised the requirement for fast surveys. Surveys have sprung up in the past, but nothing has stuck. On to the polling findings, which we shall discuss later. Glesner and Poechmueller (1989, 1992) used VLSI technology to create ANN models (1994). They provide one of the first complete overviews of the topic, including both electronic approaches and commercially available equipment. On the other hand, Heemskerk (1995) describes neurochips developed by industry and research. (1996) studied two basic parallel system designs: conventional digital components and bespoke processors.

2. Related Works

In a brief ANN issue, the researchers found that training times on two widely available computers were significantly slower or marginally faster than on a serial workstation. Aybay et al criteriasimplifies.'s the analysis of digital neurocomputers and neurochips (1996). For example, quantization and associated

weight discretization's, analog nonuniformities, and nonideal responses are discussed in Moerland and Fiesler (1997). Moerland and Fiesler (1997) offer a friendly learning method that accepts imperfect responses. Sundararajan and Saratchandran (1998) describe in detail the parallel implementation components of BP neural networks, ART neural networks, RNNs, and MIMD (multiple instruction multiple data) using MPI interface (1998). This book is divided into sections that each concentrate on a particular subject, such as network parallelism and training set parallelism in BP-based neural networks. Burr's methods allow for early prediction of HNN chip size, performance, and power consumption (1992,1991). It can also predict future capacity and performance of neural networks. Hammerstrom has been investigating digital neural networks (ANNs) since the late 1990s (1998). In Reyneri's 2002 comments, different existing modulations are compared for precision, response time, power consumption, and energy demands. Zhu and Sutton (2003) discussed FPGA-based neural network design difficulties. Assemblage and simulation, density augmentation, and topological flexibility are the reconfiguration aims discussed in this section (integer, floating point, and bit stream arithmetic). Many key areas of hardware implementation technology have been identified and investigated, including the use of co-design approaches for hardware and software. The recent investigation [3] by Diasa et al. utilizing commercial gear was one of the most extensive (2004). Spiking Neural Networks may now be built on FPGAs, following Schrauwen and D'Haene's work (2005). (SNN). Maguire et al. (2007a) analyze FPGA-based SNN models, emphasizing significant difficulties. Bartolozzi and Indiveri explain their findings in a hardware comparison of spiking synaptic models (2007). Smith analyzes digital and analog VLSI implementation choices for time-varying neural networks (2006) Hammerstrom and Waser [4] present an engaging historical critique of digital, analog, and HNN approaches across many decades (2008). Indiveri et al. also examine possible issues that may occur in the future when cognitive skills are brought to

these systems. Specialized volumes on a variety of HNN topics are now available and gaining popularity. Austin created a RAM-based HNN library (1998). Also, Ormoindi and Rajapakse published a book on FPGA-based artificial neural networks (ANNs) (2006). There are several examples and lessons learned from a large-scale FPGA-based ANN implementation in this book. In a curated collection, Valle outlines the many approaches to developing smart adaptable devices. There are various reviews and edited publications on the subject, although most of them are either outdated or concentrate in one area of HNN research. There are several ways and concepts for HNN design in the literature and commercial uses. For the previous two decades, this work has attempted to provide an overview of HNN models, hardware design techniques and applications. Our research includes a number of notable works that have appeared positively in the literature. For example, perturbation learning (1992) and constructive learning (1993) are not included in this review, as are cascade error projection (1995, 2000), local learning (2004) (including spike-based Hebbian learning) and local learning (2004). (2007, 2008) The research focuses on MLP with back propagation [5] (1994) and radial basis function networks (19) (1994, 2000) as well as HNN architectures (1994,1996).

3. Neural Network Chips

These neurochips build HNNs, which may subsequently be used to construct ANNs. This incorporates neural associative memory on a RAM or FPGA chip. There are other neural chips for digital and analog processing. A general-purpose neurochip can build several neural algorithms for one application, whereas a special-purpose neurochip can replicate a single neural algorithm for many applications. A neurochip's activation block is constantly present. This block performs weight x input multiplication and summation. The host computer may perform some of the tasks required by the chip-based blocks such neuron status, weights, and activation functions. Weights may be stored digitally or analogously, and loaded statically or dynamically. The neuronal weights.

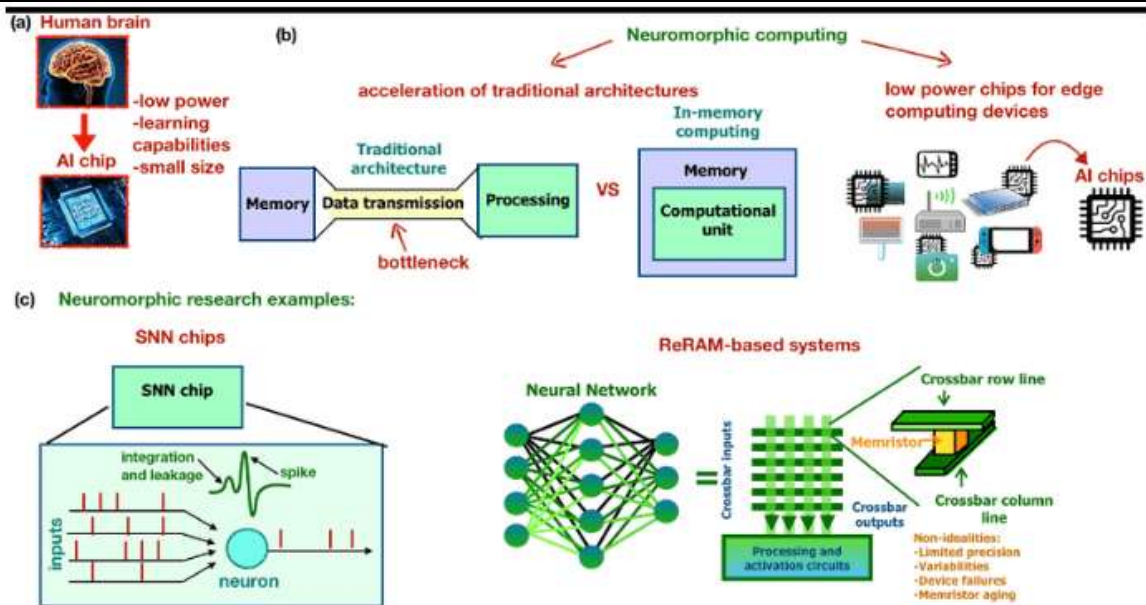


Figure 2: An introduction to hardware-based neural networks on chips. (a) highlights the most important AI chip benchmarks, (b) display and use of computers with in-memory data, (c) conceptualization of SNN, (d) crossba

3.1 Neurochips that are digitally implanted

CMOS technology is widely used in digital chips. Bit-slice processors, SIMD processors, and systolic arrays are all examples of digital chips. There are several benefits to digital technology, including as well-understood manufacturing methods (RAM weight storage), and programmable designs. As the slowest link in the network, synaptic multipliers are the most difficult to bypass. Each module of a processor processes a single bit field of an operand. Their fundamental building blocks (often single neurons) may be used to construct more complex and precise neural networks. One of the earliest commercial HNN devices was the MD1220 Neural Bit Slice from Micro Devices (1990). There are eight neurons with hard-limits thresholds and 16-bit synapses with 1-bit inputs, both of which are quite rare. There is no limit to the amount of data this device can process in synapses since it operates approximately 9 million clock cycles per second. The Philips Lneuro microprocessor from 1992 and the Neuralogix NLX-420 neural processor both use slice designs (1990). Off-chip learning is vital for slicing architectural design. Simultaneous Multi-Data Set Execution (SIMD) is the acronym for this (1991). Programmable systems are required to better fulfill ANN requirements, and most of these solutions adapt SIMD. A PE1 component's

settings are set using an instruction word with horizontal encoding. Conventional ANN applications may utilize it since it lacks address/issue logic and employs a basic instruction decoding approach. Adaptive Solutions' 1990 N64000 CPU has onboard memory for weight storing and an integer multiplier. Kim et al. [6] demonstrated an LSB-based SIMD neural network processor for image processing. There are twenty-four activities that may be performed by the suggested CPU. A PU has a 2K-word Local Memory, a PE, and a ROM. The synaptic multipliers in asynchronous array-based designs are more efficient since each PE performs one step of a computation in parallel with the other PEs. PEs. Siemens' MA-16 (1993) uses 16-bit components to perform multiplication, subtraction, and addition on a 4 by 4 matrix. The outputs and accumulators of the multiplier have a precision of 48 bits each. While neural transmission is handled using off-chip look-up tables, weights are stored in off-chip RAM. A excellent choice for artificial neural networks (ANNs) that need fine-grained parallelism is a synchronous array (systolic array). Administration and interaction with a host system are made more complicated by using this strategy. The vector processor arrays from 1992, the common bus design from 1994, the ring architecture from 1995, and the

TORAN (Twain-One Ring Array Network) from 1996 are examples of systemic high-performance neural network (HNN) systems. (1999). Researchers at the Massachusetts Institute of Technology (MIT) developed the parallelism-capable systolic processor array called SAND (Simple Applicable Neural Device). An RBF and Kohonen feature map may be used to map the neurochip. It was a low-cost alternative at the time, with an input frequency of 50 MHz and a data bit depth of 16 bits. A single SAND chip can do 200 MCPS using four 16-bit multipliers and four 40-bit adders. Because of the difficulty in assigning weights to neurons in neural networks, In the early 1990s, Wang developed an analog recurrent neural network (1992). It required substantial mapping and programming for analog implementation. Hung and Wang (2003) used a one-dimensional systolic array with ring connections to implement the procedure digitally. For realignment, this smaller model made use of FPGA-based components. The assignment issue may be simplified by taking use of the dataset's regularities. In the digital world, there are more HNN designs available. In order for classification algorithms to perform better, it may be necessary to create data ensembles (Bagging, 1996). In certain cases, extra training sets may be generated by selecting and substituting individuals at random. Bagging enhances performance when used with unreliable classifiers. Because even little modifications to the training data may have a significant impact on the final classifier. Bagging ensembles used in a three-dimensional circuit may increase pattern recognition efficiency. In addition to decision trees and threshold logic units (TLUs), the ensemble employs threshold networks (TLUs). Depending on the number of network ensembles or TLU/input pairs per network, we provide a scalable network structure. The learning rule's ability to create a proper equation and the distance needed by a PE are directly related to each other (SOFM). Problem-solving gets more complex as more PEs are involved. To build a learning algorithm, Rueping and colleagues (1994) recommend using the following digital architecture: A large

number of PEs may be packed on a single chip using the Manhattan Distance and a specific adaption factor handling in this design [8]. In terms of size, the circuitry can generate 10 10 maps on a single chip with only 28 pins. Using just binary data and a 50/50 map, a speed of >25 GCPS may be achieved. Dynamic Synapse Neural Networks have been used to study an acoustic sound detection model. DSP Starter Kit TMS320C6713 was used to make the model (a floating-point DSP processor). 90 percent of the time, the new hardware accurately classifies and pinpoints the location of gunfire.

3.2 Analog Neurochips

Intel's ETANN and Synaptic's Silicon Retina are two early analog processors that were created from scratch. Each of the Intel 80170NX's 64 neurons is directly connected to every other one in the system (1990). On this general-purpose neurochip, analog nonvolatile weights are stored using floating gates, while four-quadrant multiplication is performed via Gilbert-multiplier synapses. After the learning phase, ETANN depends on a host computer to ensure that the weights can be downloaded to the chip, rather than allowing on-chip learning. The device's maker claims that it can calculate at a pace of 2 GCPS with an accuracy of 4 bits and a bus size of 64 bits. An additional 10,240 synapses may be customized. Direct pin/bus connections between ETANN chips may be used to create networks of up to 1024 neurons and 81,920 weights. For real-time visual processing, the Mod2 Neurocomputer (1992) made use of ETANN chips. The MBOX II (1994), an analog audio synthesizer equipped with eight ETANN chips, included these ETANN chips later in its existence. It is common to need to adjust the distance between the input vectors and the weights when using ANNs based on competition, such as the Kohonen SOFM. An analog implementation of a SOFM often results in a tiny circuit block that correctly computes the distances between two points. " To determine the distance between two points, two commonly used metrics are Euclidean and Manhattan. Distance computation circuits employing the Euclidean distance function technique have been constructed by academics such as Lanbolt and

Churher (1992) and Churher et al. (1993). (1993). (1993). For calculating the Euclidean distance metric, Churher et al. [9] (1993) constructed extensively utilized circuits in the early 1990s. An analog VLSI version of several Euclidean distance computation circuits was provided by Gopalan and Titus (2003) and may be used as part of a high-density SOFM hardware implementation. In a mixed signal, CMOS feed-forward system, Liu et al. (2002) show the use of on-chip error-reduction technology for real-time adaption. For oscillating operating conditions, MOSIS used Orbit 2m n-well technology to build the device, and weights were stored in capacitors. When used in conjunction with the Random Weight Change (RWC) algorithm, which does not need an intended neural network output to be known in order to calculate error rates, the implemented learning technique is a genetic random search algorithm. In spite of its little stature, the RWC chip was able to effectively suppress unstable oscillations that simulated combustion engine instability during testing. Weight storage remains a problem, which drastically limits the number of applications. Morphological neural networks may be implemented using the discrete analog hardware model of the discrete analog hardware model proposed by Ortiz and Ocasio (2003), rather than multiplication or addition. It has been shown that the inherent quadratic nonlinearity of the synapses affects learning convergence and vector direction optimization by using a MOSFET-based analog signal synapse model developed by Milev and Hristov (2003) in a typical 0.35-micron CMOS manufacturing process. Once this is done, the synapse concept is put into practice on a VLSI chip, which has 2176 synapses and can be used to extract fingerprint features. Mixed-mode analog VLSI is used to build a signal processing circuit for a Continuous-Time Recurrent Neural Network, where state variables are represented by voltages and neural impulses conveyed as currents, according to Brown et al. (2004). Brain signals may be accurately processed across great distances using current, resulting in a scalable and resilient neural signal processing system. On the other hand,

Bayraktaroglu et al. (1999) describe ANNSyS as a machine learning technique that uses an approximated version of on-chip training to build analog neural networks (ANNs). Using a SPICE circuit simulator and an assembler for MOS technology, the synthesis system may be used to create analog neural networks.

3.3 Neurochips with a Mixture of Functions

For maximum system performance, hybrid chips combine digital and analog components. When it comes to determining speed and weight, analog internal processing is the preferred method. Example: The University of Twente Mesa Research Institute developed a hybrid Neuro-Classifer in 1994 that uses five-bit digital weights to reach a feed-forward processing rate of up to 20 GCPS [10.] This device contains 70 analog inputs, six hidden nodes, and one analog output. Even if the final output has no transfer function, several chips may be stacked together to increase the number of hidden units. A matrix-vector multiplier for artificial neural networks (ANNs) is shown by the authors utilizing digitally recorded synaptic strengths (2004). Even though analog operations have accuracy constraints, it was revealed in 1991 and 1994 that combining cortical neurons into populations where each neuron's signal is restored to an appropriate analog value via a collective technique allows them to compute consistently. In order to create cortical amplifier networks with a linear threshold transfer function, Douglas and colleagues suggest a hybrid analog-to-digital CMOS design (1994). For neural co-processing, Romariz et al. suggest a hybrid architecture that makes use of digitally controlled multiplexing of analog multipliers and capacitors to simulate the experience of numerous levels. With the use of a predetermined collection of analog multipliers and capacitors, this system attempts to imitate many levels of experience (analog memory). It's been proven that hybrid architecture can enable on-chip learning (1999). The design of a circuit relies heavily on the use of analog and digital components. The analog ANN unit uses a charge-based circuit architecture to calculate neural functions. A ten-bit vector is sent to each of the twenty

neurons in this layer. Neurons are selected as winners depending on how closely the recorded pixel patterns and current input vectors match. Digital generation may do all of these duties, including error correction, circuit control, and clock generation.

3.4 Implementations Using FPGAs

Hybrid chips combine digital and analog components to enhance the overall performance of the system they're installed in. Speed can be determined using analog internal processing, but weights may be modified using a digital internal processing method instead. Example: Neuro-Classifer developed by the University of Twente Mesa Research Institute in 1994 used five-bit digital weights to reach a feed-forward processing rate of up to 20 GCPS [10.] It has 70 analog inputs, six hidden nodes, as well as a single analog output. Even though the final output has no transfer function, it is possible to stack multiple chips together to increase the number of hidden units. Employing digitally recorded synaptic strengths, the authors explain how to build a matrix-vector multiplier for artificial neural networks (ANNs) (2004). Grouping cortical neurons into populations in which each neuron's signal is restored to an appropriate analog value via a collective technique allows them to compute consistently even when the precision constraints of analog operations are in place, as shown in 1991 and 1994. Cortical amplifier networks with a linear threshold transfer function may be designed using an analog-to-digital CMOS hybrid architecture, as proposed by Douglas and collaborators (1994). Romariz et al. propose a hybrid neural co-processing architecture that uses digitally controlled multiplexing of analog multipliers and capacitors to mimic the experience of several layers. Using a predetermined set of analog multipliers and capacitors, this system replicates several layer experiences (analog memory). A hybrid architecture has shown the feasibility of learning on-chip (1999). Among the most important features of the circuit architecture are analog and digital components. A charge-based circuit architecture is used in the analog ANN unit to calculate neural functions. Neurons in this

layer get a ten-bit vector input from twenty other neurons in the same layer. Based on how closely the stored pixel pattern and the current input vector match, winner-takes-all units pick a single neuron as the winner. Error correction, circuit management, and clock creation are all possible with digital generation.

4. Other Approaches

Szabo and colleagues (2000) propose for using bit-serial distributed arithmetic in a bit-serial/parallel technique to increase the performance of digital filters. Their matrix-vector multiplier approach is developed using optimized CSD (Canonic Signed Digit) encoding and bit-level pattern coincidences. The architecture may be developed and used in neural network design environments using either an FPGA or an ASIC. Both MLPs and neural networks benefit from the proposed matrix multiplier structure (CNNs).

4.1 Applications of Associative Neural Memory Techniques

Threshold operations are used to map between two pattern sets in an Associative Neural Memory, a kind of artificial neural network. This problem was investigated by Palm et al. (1993) using a very simple model of a neural network in which the input, output, and link weights are all binary. These circuits were designed using analog, digital, and mixed signal techniques by Ruckert et al. [14] in 1991 and 2002. NPU (neural processing unit), an I/O coding block, and an on-chip controller are some of the fundamental elements of digital architecture (synchronization, control, and testing instructions). It is calculated that in this circumstance, the rate of learning is 0.48 GCUPS. There are 16 neurons and 16 synapses on the test chip, which is built using 1.2-CMOS technology and the test chip. The number of neurons and inputs may be increased to 4000, with each neuron receiving 16,000 inputs. The addition of additional of these chips might make the overall design more difficult. There's an ANM (Automatic Neural Network) form of the Willshaw et al. (1969) ANM model in which the output pattern is the label of the stored pattern that's most similar to the input pattern (input pattern). On-board training and testing

for high-performance pattern recognition applications are discussed by Justin et al. [15]. (2005). We suggest Hassoun's edited book on ANM models to anybody looking for a convenient source of knowledge (1993).

4.2 Implementations based on RAM

Bledsoe and Browning (1959) first developed a RAM-based neural network (RNN), which is frequently referred to as a weightless NN. To develop lookup tables, random access memory may store neural functions (RAM). Ten times faster than previous models, they may be learned in less than a day utilizing low-cost, easily accessible technology. Look-up tables rather than weights are used for training RNNs, unlike ordinary neural networks. Pattern-recognition systems, such as RNNs, may be employed in a number of scenarios, such as photo recognition, for example. RNNs are extensively covered in the book (1998,1999). Aleksander and colleagues created WISARD, the first general-purpose image recognition system based on RAM circuits (1984). The learning method and hardware implementation of a probabilistic RAM network are both explained in depth (1992). In medical imaging, a SAT (Sum and Threshold) processor was first described by Kennedy and Austin (1994) as a customized hardware version of a binary neural image processor. Advanced Distributed Associative Memory (ADAM) is compatible with the SAT processor when it comes to memory. In order to recognize and extract information from images in a variety of situations, ADAM is a binary-weighted, two-layer neural network. C-NNAP, according to Austin et al., is used to deal with issues related to item identification (1995). MIMD, an array of ADAM processors, will help solve the identification of objects issue in a distributed manner (Cellular Neural Network Associative Processor).

5. Conclusion

As an introduction to the hardware implementation of AI, the following examples of HNN prototypes from academia and industry are presented (ANNs). Research on HNN began in the 1990s, but has yet to be put into use in the real world. ANN models, hardware design

techniques, and applications are examined to determine the current state of the field. In order to be applicable in a broad variety of applications, the model must be mapped onto reliable and energy-efficient hardware. Complete ANN models in the brain and on chips are being studied by researchers (digital, analog, hybrid). FPGA-based versions of neurons and hardware for spiking neural networks are all provided. RAM-based parallel digital implementations, bit slices, and SIMD structures will also be investigated. Associative brain memory will also be examined. Consider the current status of research before making a future forecast.

References

1. Y. S. AbuMostafa and D. Psaltis. Optical neural computers. *Scientific American*, 255:88–95, 1987.
2. Gyorgy Cserey Adam Rak, Balazs Gergely Soos. Stochastic bitstream-based CNN and its implementation on FPGA. *International Journal of Circuit Theory and Applications*, 37(4):587–612, 2002.
3. A. J. Agranat, C. F. Neugebauer, and A. Yariv. A CCD based neural network integrated circuit with 64k analogprogrammable synapses. In *International Joint Conference on Neural Networks (IJCNN)*, pages 551–555, 1990.
4. B. Ahmed, J. C. Anderson, R. J. Douglas, K. A. Martin, and C. Nelson. Polyneuronal innervation of spiny stellate neurons in cat visual cortex. *Comparative Neurology*, 341(1):39–49, 1994.
5. E. Ahmed and K. Priyalal. Algorithmic mapping of feedforward neural networks onto multiple bus systems. *IEEE Transactions on Parallel and Distributed Systems*, 8:130–136, 1997.
6. SF Al-Sarawi, D. Abbott, and PD Franzon. A review of 3-D packaging technology. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, 21(1):2–14, 1998.

7. I. Aleksander, W. V. Thomas, and P. A. Bowden. WISARD: a radical step forward in image recognition. *Sensor Review*, 4(3):120–124, 1984.
8. F. Alibart, S. Pleutin, D. Gu'erin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat, and D. Vuillaume. An Organic Nanoparticle Transistor Behaving as a Biological Spiking Synapse. *Advanced Functional Materials*, 20(2):330–337, 2009.
9. A. P. Almeida and J. E. Franca. Digitally programmable analog building blocks for the implementation of artificial neural networks. *IEEE Transactions on Neural Networks*, 7(2):506–514, 1996.
10. H. Amin, K. M. Curtis, and B. R. Hayes-Gill. Piecewise linear approximation applied to nonlinear function of a neural network. In *IEE Proceedings of Circuits, Devices and Systems*, pages 313–317, 1997.
11. H. Amin, K. M. Curtis, and B. R. Hayes-Gill. Two-ring systolic array network for artificial neural networks. In *IEE Proceedings of Circuits, Devices and Systems*, pages 225–230, 1999.
12. D. Anguita, I. Baturone, and J. Miller, editors. Special issue on hardware implementations of soft computing techniques: *Applied Soft Computing*, volume 4(3), Aug 2004.
13. M. Anguita, FJ Pelayo, I. Rojas, and A. Prieto. Area efficient implementations of fixed-template CNN's. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 45(9):968–973, 1998.
14. A.J. Annema, K. Hoen, and H. Wallinga. Precision requirements for single-layer feed forward neural networks. In *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, pages 145 – 151, Turin, Italy, 1994.
15. Annon. The intelligent flight control: Advanced concept program final report. Technical report, The Boeing Company, 1999.