1

Overview

Learning never exhausts the mind.

Leonardo da Vinci

1.1 History of Neural Networks

Even though the modern von Neumann architecture-based processors are able to conduct logic and scientific computations at an extremely fast speed, they perform poorly on many tasks that are common to human beings, such as image recognition, video motion detection, and natural language processing. Aiming to emulate the capability of human brains, a non-Boolean paradigm of computation, called the neural network, was developed since the early 1950s and evolved slowly over many decades. So far, at least three major forms of neural networks have been presented in the literature, as shown in Figure 1.1.

The simplest neural network is the perceptron, where hand-crafted features are employed as input to the network. Outputs of the perceptron are binary numbers obtained through hard thresholding. Therefore, the perceptron can be conveniently used for classification problems where inputs are linearly separable. The second type of neural network is sometimes called a multilayer perceptron (MLP). Nevertheless, the "perceptrons" in an MLP are different from the simple perceptrons in the earlier neural network. In an MLP, a non-linear activation function is associated with each neuron. Popular choices for the non-linear activation function are the sigmoid function, the hyperbolic tangent function, and the rectifier function. The output of each neuron is a continuous variable instead of a binary state. The MLP is widely adopted by the machine learning community, as it can be easily implemented on general-purpose processors. This type of neural network is so popular that the phrase "artificial neural network" (ANN) is often used to specify it exclusively, even though the word ANN should have been referred to any other neural network besides biological neural networks. ANN is the backbone for the concept of a widely popular mode of learning, called deep learning. A less well-known type of neural network is called a spiking neural network (SNN). Compared to the previous two types of neural networks, SNN resembles more to a biological neural network in the sense that spikes are used to transport information. It is believed that SNNs are more powerful and advanced than ANNs, as the dynamics of an SNN is much more complicated and the information carried by an SNN could be much richer.

Learning in Energy-Efficient Neuromorphic Computing: Algorithm and Architecture Co-Design,
First Edition. Nan Zheng and Pinaki Mazumder.

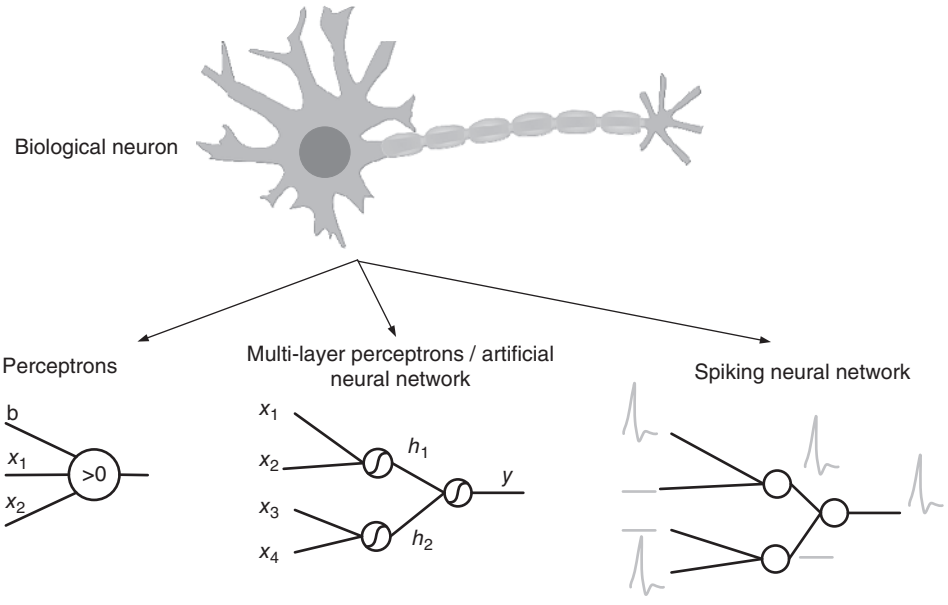

Figure 1.1 The development of neural networks over time. One of the earliest neural networks, called perceptron, is similar to a linear classifier. The type of neural network that is widely used nowadays is referred as an artificial neural network in this book. This kind of neural network uses real numbers to carry information. The spiking neural network is another type of neural network that has been gaining popularity in recent years. A spiking neural network uses spikes to represent information.

1.2 Neural Networks in Software

1.2.1 Artificial Neural Network

Tremendous advancements have occurred in the late 1980s and early 1990s for neural networks constructed in software. One powerful technique that significantly propelled the development of ANNs was the invention of backpropagation [1]. It turned out that backpropagation was very efficient and effective in training multilayer neural networks. It was the backpropagation algorithm that enabled neural networks to solve numerous real-life problems, such as image recognition [2, 3], control [4, 5], and prediction [6, 7].

In the late 1990s, it was found that other machine-learning tools, such as support vector machines (SVMs) and even much simpler linear classifiers, were able to achieve comparable and even better performances in classification tasks, which was one of the most important applications of neural networks at that time. In addition, it was observed that training of neural networks was often stuck at local minima, and consequently failed to converge to the global minimum point. Furthermore, it was generally believed that one hidden layer was enough for neural networks, as more hidden layers did not improve the performance remarkably. Since then, research interest in neural networks started to decline in the computational intelligence community.

Interest in neural networks was revived around 2006 as researchers demonstrated that a deep feedforward neural network was able to achieve outstanding classification

accuracy with proper unsupervised pretraining [8, 9]. Despite its success, the deep neural network was not fully recognized by the computer vision and machine learning community until 2012 when astonishing results were achieved by AlexNet, a deep convolutional neural network (CNN) [10]. Since then, deep learning has emerged as the mainstream method in various tasks such as image recognition and audio recognition.

1.2.2 Spiking Neural Network

As another type of important neural network, SNNs did not receive much attention in comparison to the widely used ANNs. Interest in SNNs mainly came from the neuroscience community. Despite being less popular, many researchers believe that SNNs have a more powerful computational capability compared to their ANN counterparts, thanks to the spatiotemporal patterns used to carry information in SNNs. Even though SNNs are potentially more advanced, there are difficulties in harnessing the power of SNNs. Dynamics of an SNN is much more complicated in comparison to that of an ANN, which makes the purely analytical approach intractable. Furthermore, it is considerably harder to implement event-driven SNNs efficiently on a conventional general-purpose processor. This is also one of the main reasons that SNNs are not as popular as ANNs in the computational intelligence community.

Over the past decades, there were numerous efforts from both the computational intelligence community and the neuroscience community to develop learning algorithms for SNNs. Spike-timing-dependent plasticity (STDP), which was first observed in biological experiments, was proposed as an empirically successful learning rule for unsupervised learning [11–14]. In a typical STDP protocol, synaptic weight updates according to the relative order and the difference between the presynaptic and postsynaptic spike timings. Unsupervised learning is useful in discovering the underlying structure of data, yet it is not as powerful as supervised learning in many real-life applications, at least at the current stage.

1.3 Need for Neuromorphic Hardware

The development of hardware-based neural network or neuromorphic hardware started along with their software counterpart. There was a period of time (late 1980s to early 1990s) when many neuromorphic chips and hardware systems were introduced [15–18]. Later on, after finding out that the performance of neural networks was hard to keep apace with digital computers due to the inadequate level of integration of synapses and neurons, hardware research in neural networks took a back seat while Boolean computing advanced by leaps and bounds, leveraging the scaling and Moore's Law. Around 2006 when the breakthrough was made in the field of deep learning, research interests in hardware implementation of neural networks also revived. The possibilities of deploying neuromorphic computing in real-life applications were explored, as the development of the conventional von Neumann architecture-based computing slowed down because of the looming end of Moore's law.

Electronic computing devices have evolved for several decades, as shown in Figure 1.2. Two trends can be observed from the figure. The first trend is that computing devices are becoming smaller and cheaper. Indeed, partially driven by Moore's law, the sizes and

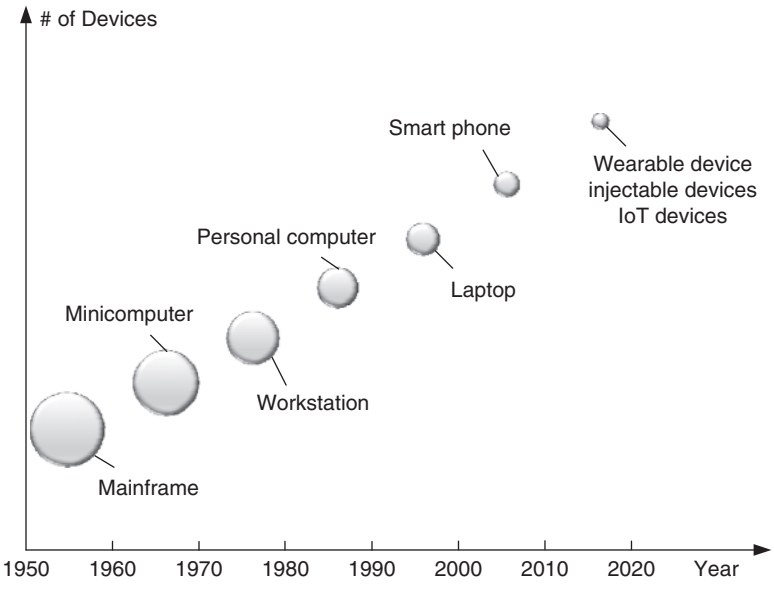

Figure 1.2 History of the development of computing devices. From the early mainframe computer that occupied the entire laboratory to nowadays ubiquitous IoT devices, the sizes of the computing devices have been shrinking over the past few decades, partially driven by Moore's law. The number of devices, on the other hand, has kept growing. As a consequence of increasing portability, more and more devices are powered by batteries today.

prices of consumer electronics are decreasing continuously. The second trend is that both the variety and the amount of information processed by the computing devices are increasing. Nowadays there are many types of sensors, such as motion, temperature, and pressure sensors, in our smart phones and wearable devices that keep gathering data for further processing. Therefore, we are experiencing a transition from the conventional rule-based computing to a data-driven computing.

With more and more low-power sensor devices and platforms being deployed in our everyday life, an enormous amount of data is being collected continuously from these ubiquitous sensors. One dilemma we often encounter is that despite the amount of data we collate, we lack the capability to fully exploit the gathered information. There is a strong need to provide these sensor platforms with built-in intelligence so that they can sense, organize, and utilize the data more wisely. Fortuitously, deep learning has emerged as a powerful tool for solving this problem [8, 19–24]. In fact, machine learning, especially deep learning, has become such a hot technology recently that it has a huge impact on how the commercial world operates. With more and more startups and big companies investigating this field, it is expected that use of artificial intelligence (AI) and machine learning will grow much faster in the near future.

Despite its successes in smaller applications, a deep neural network can only be employed in a real-life application if millions or even billions of synapses can be integrated in the system. Training of such a huge neural network usually takes weeks and burns an excessive amount of energy, even when highly optimized hardware such as graphics processing units (GPUs) are employed and matrix solving are being largely parallelized [20]. In the near future, we will have more and more ultra-low-power

sensor systems for health and environment monitoring [25–27], mobile microrobots that chiefly rely on energy scavenging from the environment [28–31], and over 10 billion internet-of-things (IoT) devices [32]. For all these applications where power consumption is an important consideration, neither power-hungry GPUs nor sending raw data to the cloud computer for further analysis is a viable option. To tackle this difficulty, sustained endeavors from both industry and academia have culminated in developing low-power deep learning accelerators.

Google has built a customized application-specific integrated circuit (ASIC) called a tensor processing unit (TPU) in order to accelerate deep-learning applications in their datacenter [33], while Microsoft has utilized field-programmable gate arrays (FPGAs) in their datacenter for deep learning [34]. The realization of deep learning with FPGAs provides a cost-effective and energy-efficient solution compared to the more conventional GPU-based approach. Intel has unveiled its Nervana chip, which is the neural network processor Intel has developed with the aim to revolutionize AI computing across many different fields. In addition to the efforts made by the industry, an increasing number of papers have been published in recent years to discuss various architectures and design techniques for building energy-efficient ANN accelerators. With the growing popularity of the deep neural network, more innovations are expected in the near future.

Despite its mathematical simplicity, ANN-based learning faces challenges in scalability and power efficiency. To tackle these issues, more and more researchers in the hardware community started working on SNN-based hardware accelerators. This trend is attributed to many unique advantages that SNNs have. The event-triggered nature of an SNN can lead to a very power-efficient computation. The spike-based data encoding also facilitates the communication between neurons, providing good scalability. Nevertheless, building and utilizing specialized spike-based neuromorphic hardware is still in its early stage and there are many difficulties that need to be addressed before the hardware can become meaningfully useful. One main challenge we are encountering is how to train a spike-based neural network properly. After all, it is the learning capability of neural networks that empowers the neuromorphic system with the intelligence that can be exploited by many applications.

1.4 Objectives and Outlines of the Book

Machine learning, especially deep learning, has emerged as an important discipline through which many conventionally difficult problems, such as pattern recognition, decision making, and natural language processing, can be addressed. Nowadays, millions and even billions of neural networks are running in data centers, personal computers and portable devices to perform various tasks. In the future, it is expected that more complex neural networks with larger sizes will be needed. Such a trend demands specialized hardware to accommodate the ever-increasing requirements on power consumption and response time.

In this book, we focus on the topic of how to build energy-efficient hardware for neural networks with a learning capability. This book strives to provide co-design and co-optimization methodologies for building hardware neural networks that can learn to perform various tasks. The book provides a complete picture from high-level algorithms to low-level implementation details. Hardware-friendly algorithms are developed with

the objective to ease implementation in hardware, whereas special hardware architectures are proposed to exploit the unique features of the algorithms. In the following chapters, algorithms and hardware architectures for energy-efficient neural network accelerators are discussed. An overview of the organization of this book is illustrated in Figure 1.3.

In Chapter 2, algorithms for utilizing and training rate-based ANNs are discussed as well as several basic concepts involved in the learning and inference of an ANN. Popular network structures, such as a fully connected neural network, a CNN, and a recurrent neural network, are introduced, and their advantages are discussed. Different types of learning schemes, such as supervised learning, unsupervised learning, and reinforcement learning, are demonstrated, and a concrete case study is provided to show how to employ an ANN in a reinforcement-learning task. Considering many astonishing results achieved by deep learning recently, emphasis is placed on concepts and techniques commonly used in deep learning in this chapter.

In Chapter 3, various options of executing neural networks are introduced, ranging from general-purpose processors to specialized hardware and from digital accelerators to analog accelerators. Hardware realizations of many neural network structures and deep-learning techniques presented in Chapter 2 are discussed in this chapter. Various architecture- and circuit-level techniques and innovations that can help build energy-efficient accelerators are presented for both digital and analog accelerators. A case study of building a low-power accelerator for adaptive dynamic programming with neural networks is discussed in detail to provide a concrete example.

In Chapter 4, fundamental concepts and popular learning algorithms for SNNs are discussed, starting with the basic operational principle of typical SNNs. The similarities and key differences between SNNs and ANNs are identified. Many classic learning algorithms that are capable of training shallow neural networks are first discussed. Inspired by the recent success achieved by deep ANNs, how to extend learning into deep SNNs is also explored. Popular ways of training multilayer SNNs are examined. To demonstrate the feasibility of training deep SNNs, a supervised learning algorithm that exploits spike timings for estimating gradient information needed by backpropagation is presented in great detail.

In Chapter 5, hardware implementations for SNNs are discussed. Several advantages of SNN hardware are highlighted, which serve as the motivation for implementing SNN hardware. A few general-purpose large-scale spiking systems that target simulating biological neural networks or performing cognitive tasks are presented, including both digital systems such as TrueNorth and SpiNNaker and analog systems such as Neurogrid and BrainScaleS. In addition to these large neuromorphic systems, compact customized SNN hardware aiming at accelerating specific tasks with a high energy efficiency is also discussed. To implement the learning algorithm presented in Chapter 4 efficiently in hardware, three design examples are presented. Two of the designs are digital accelerators based on conventional CMOS technology, whereas the third design is an analog system based on an emerging nanotechnology. Through these three design examples, many important aspects of designing SNN hardware are covered.

Chapter 6 concludes this book and provides some thoughts and outlooks concerning the future research directions in the field of neural network hardware.

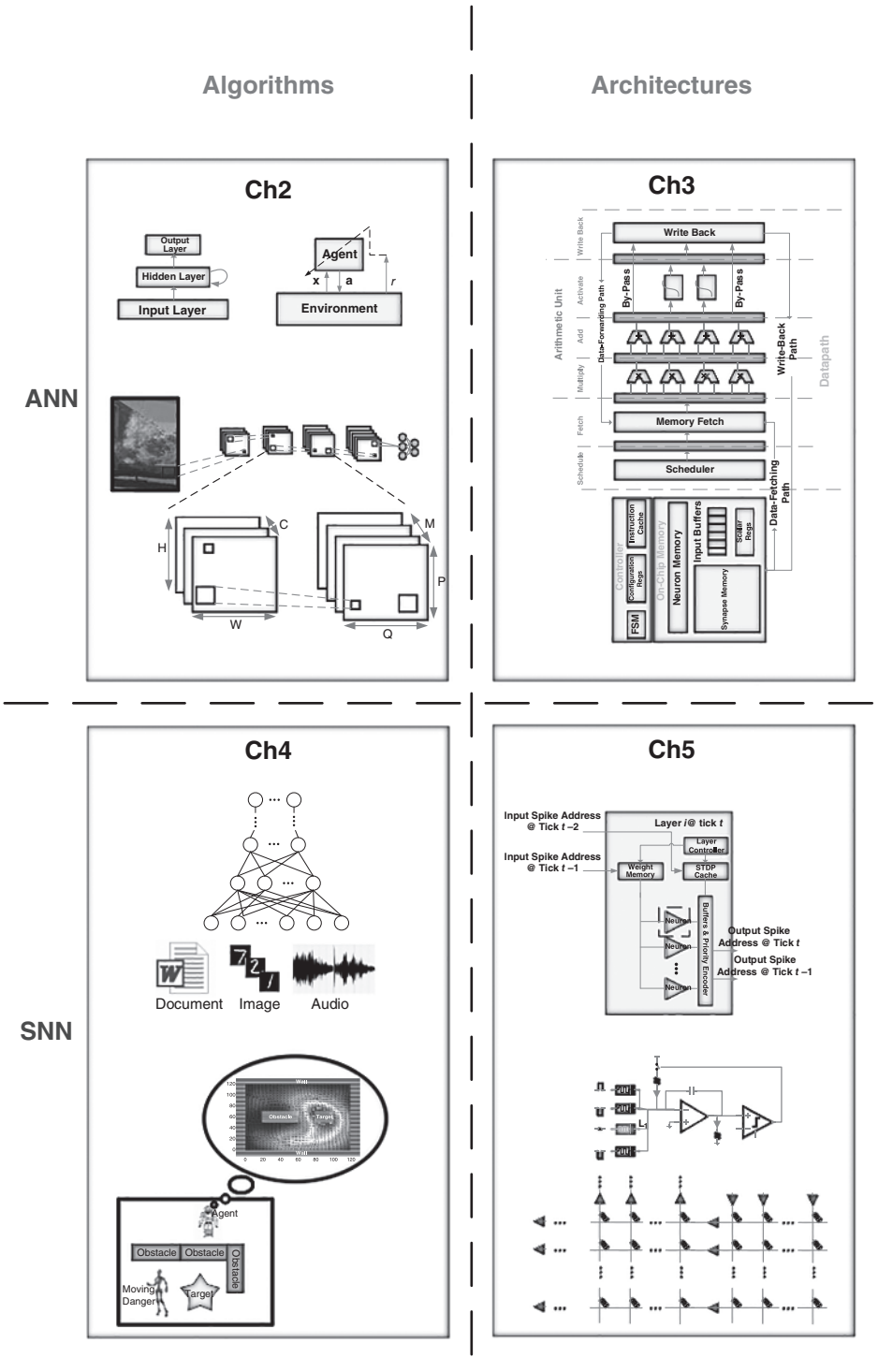

Figure 1.3 Overview of the organization of the book.

References

1 Werbos, P.J. (1990). Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78 (10): 1550–1560.
2 Rowley, H.A., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (1): 23–38.
3 LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86 (11): 2278–2323.
4 Psaltis, D., Sideris, A., and Yamamura, A.A. (1988). A multilayered neural network controller. *IEEE Control Syst. Mag.* 8 (2): 17–21.
5 Kawato, M., Furukawa, K., and Suzuki, R. (1987). A hierarchical neural network model for control and learning of voluntary movement. *Biol. Cybern.* 57 (3): 169–185.
6 Kimoto, T., Asakawa, K., Yoda, M., and Takeoka, M. (1990). Stock market prediction system with modular neural networks. In: *1990 IJCNN International Joint Conference on Neural Networks*, vol. 1, 1–6. IEEE.
7 Odom, M.D. and Sharda, R. (1990). A neural network model for bankruptcy prediction. In: *1990 IJCNN International Joint Conference on Neural Networks*, vol. 2, 163–168. IEEE.
8 Hinton, G.E. and Osindero, S. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 1554 (7): 1527–1554.
9 Erhan, D., Bengio, Y., Courville, A. et al. (2010). Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* 11 (Feb): 625–660.
10 Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, 1097–1105.
11 Diehl, P.U. and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9: 99.
12 Querlioz, D., Bichler, O., Dollfus, P., and Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* 12 (3): 288–295.
13 Masquelier, T. (2012). Relative spike time coding and STDP-based orientation selectivity in the early visual system in natural continuous and saccadic vision: a computational model. *J. Comput. Neurosci.* 32 (3): 425–441.
14 Masquelier, T. and Thorpe, S.J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* 3 (2): 0247–0257.
15 Duranton, M., Gobert, J., and Sirat, J.A. (1992). Lneuro 1.0: a piece of hardware LEGO for building neural network systems. *IEEE Trans. Neural Networks* 3 (3): 414–422.
16 Eberhardt, S., Duong, T., and Thakoor, A. (1989). Design of parallel hardware neural network systems from custom analog VLSI 'building block' chips. *Int. Jt. Conf. Neural Networks* 3: 183–190.
17 Maeda, Y., Hirano, H., and Kanata, Y. (1995). A learning rule of neural networks via simultaneous perturbation and its hardware implementation. *Neural Networks* 8 (2): 251–259.

18 Mazumder, P. and Jih, Y.-S. (1993). A new built-in self-repair approach to VLSI memory yield enhancement by using neural-type circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 12 (1): 124–136.

19 Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. In: *Advances in Neural Information Processing Systems*, 153–160.

20 Le, Q.V., Ranzato, M.A., Monga, R. et al. (2011). Building high-level features using large scale unsupervised learning. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 8595–8598.

21 LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521 (7553): 436–444.

22 Mnih, V., Kavukcuoglu, K., Silver, D. et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518 (7540): 529–533.

23 Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks* 61: 85–117.

24 Silver, D., Huang, A., Maddison, C.J. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature* 529 (7587): 484–489.

25 Lee, Y., Bang, S., Lee, I. et al. (2013). A modular 1 mm^3 die-stacked sensing platform with low power I^2C inter-die communication and multi-modal energy harvesting. *IEEE J. Solid-State Circuits* 48 (1): 229–243.

26 Chen, Y.P., Jeon, D., Lee, Y. et al. (2015). An injectable 64 nW ECG mixed-signal SoC in 65 nm for arrhythmia monitoring. *IEEE J. Solid-State Circuits* 50 (1): 375–390.

27 Lee, I., Kim, G., Bang, S. et al. (2015). System-on-mud: ultra-low power oceanic sensing platform powered by small-scale benthic microbial fuel cells. *IEEE Trans. Circuits Syst. I Regul. Pap.* 62 (4): 1126–1135.

28 Pérez-Arancibia, N.O., Ma, K.Y., Galloway, K.C. et al. (2011). First controlled vertical flight of a biologically inspired microrobot. *Bioinspiration Biomimetics* 6 (3): 036009.

29 Mazumder, P., Hu, D., Ebong, I. et al. (2016). Digital implementation of a virtual insect trained by spike-timing dependent plasticity. *Integr. VLSI J.* 54: 109–117.

30 Wood, R.J. (2008). The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Trans. Rob.* 24 (2): 341–347.

31 Hu, D., Zhang, X., Xu, Z. et al. (2014). Digital implementation of a spiking neural network (SNN) capable of spike-timing-dependent plasticity (STDP) learning. In: *14th IEEE International Conference on Nanotechnology, IEEE-NANO 2014*, 873–876. IEEE.

32 Friess, P. (2011). *Internet of Things-Global Technological and Societal Trends from Smart Environments and Spaces to Green ICT*. River Publishers.

33 Jouppi, N.P., Young, C., Patil, N. et al. (2017). In-datacenter performance analysis of a tensor processing unit. In: *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12. IEEE.

34 Chung, E., Fowers, J., Ovtcharov, K. et al. (2018). Serving DNNs in real time at datacenter scale with project brainwave. *IEEE Micro* 38 (2): 8–20.