Chapter 1

Introduction

"As engineers, we would be foolish to ignore the lessons of a billion years of evolution."

- Carver Mead, *1993*

Advances in computing power and machine learning have endowed computers with rapidly growing performance in cognitive tasks such as recognising objects [Deng et al., 2009] and playing GO [Silver et al., 2016]; these tasks were once dominated by human intelligence and solved by biological neurons in the brain. However, humans and many other animals still outperform computers in practical tasks, such as vision, and in terms of size and energy cost by several orders of magnitude. For instance, AlphaGO [Silver et al., 2016] has a power consumption of 1 MW on its $1,920$ CPUs and 280 GPUs when playing the game with one of the best human players whose brain is rated at 20 W [Drubach, 2000]. Although we are still far from understanding the brain thoroughly, it is believed that the performance gap between computation in the biological nervous system and in a computer lies in the nature of the fundamental computing units and how they compute. Typical computers employ Boolean logic and deterministic digital operations usually based on synchronous clocks while nervous systems employ parallel, distributed, event-driven, stochastically unreliable components [Indiveri et al., 2009]: neurons. These impressive disparities in cognitive capabilities and energy consumption drive research into biologically-plausible spiking neurons and brain inspired computers, known as Neuromorphic Engineering (NE).

NE was proposed by Carver Mead in the late 1980s [Mead, 1989] to build analogue

circuits which mimic biological neural cells and the architecture of the nervous system using Very-Large-Scale Integration (VLSI) technology. With the ultimate goal of equipping neuromorphic machines with genuine intelligence, the objectives of NE can be summarised as follows [Furber and Temple, 2007]:

- brain modelling: for neuroscientists to understand the brain by modelling and simulating the activities of biological neurons;
- neuromorphic computing: for engineers to build brain-like machines by applying biological principles to computers.

The aims complement each other; building a biologically inspired computer requires a better understanding of the brain, and simulating brain activities at large scale and in real time is feasible only on massively-parallel neuromorphic hardware.

Spiking Neural Networks (SNNs), comprised of spiking neurons, hold the key to address the dual aims of understanding brain functions and building brain-like machines. The spiking neuron mathematically models the dynamics of a single neuron with biological realism and the network describes the architecture of the neural connections and the information transmission among them; readers can refer to Chapter 2 for more detail. Therefore, neuroscientists are able to reproduce the recorded neural dynamics and activities from *in-vivo/vitro* experiments to verify their models and measure the progress of brain understanding, while computer engineers can focus on the hardware implementations of the spiking neurons and the interconnections between them to build energy-efficient neuromorphic hardware.

Over the last decade, considerable development has taken place in NE where simulations of massive SNNs [Markram, 2006; Ananthanarayanan et al., 2009] have proved to be significantly useful in understanding the brain, and large-scale neuromorphic platforms have been launched to simulate SNNs in hardware. These neuromorphic computers develop into energy-efficient systems by implementing neurons, synapses and neuronal communications on analogue circuits [Schemmel et al., 2010; Benjamin et al., 2014; Yu et al., 2012; Moradi et al., 2017] or exploiting parallel low-power microprocessors on digital hardware [Furber et al., 2014; Merolla et al., 2014]. Thus, the neuromorphic hardware systems have successfully demonstrated decreased energy cost of SNN simulations on supercomputers [De Garis et al., 2010; Sharp et al., 2012].

However, the SNN simulations only reconstruct the network behaviours and neural dynamics of some subsystem of the brain, 'but without precisely functionally simulating that subsystem' [De Garis et al., 2010]. In other words, the SNNs are able to repeat the firing activities of groups of neurons, however, not capable of simulating or understanding the functions of these activities. Therefore, this type of SNN simulation can be used to guide neuroscience and the development of neuromorphic hardware systems, but is not directly useful for solving cognitive tasks. Recent SNN applications [Bill and Legenstein, 2014; Diehl et al., 2015a] in Artificial Intelligence (AI) tasks, summarised in Chapter 6, typically comprise only two neural layers and exploit biologically-plausible learning rules, e.g. Spike-Timing-Dependent Plasticity (STDP), and/or Winner-Take-All (WTA) circuits on the synaptic connections. These two-layered SNN models are considered to be 'reactive' since the output neurons simply react to the sensory input. Consequently, such SNNs cannot perform sophisticated effective cognition as can the brain; thus programming these neuromorphic machines to be competent in cognitive applications still remains unsolved. Indiveri et al. [2009] argued that the next substantial challenge of NE is to make these brain-like computers effectively cognitive, also known as 'Neuromorphic Cognition'.

STDP as a learning mechanism based on biological observations has been implemented to be equivalent to a stochastic version of powerful machine learning algorithms, such as Expectation Maximisation [Nessler et al., 2013], Contrastive Divergence [Neftci et al., 2013], Markov Chain Monte Carlo [Buesing et al., 2011] and Gradient Descent [O'Connor and Welling, 2016]. However, in practice, there have been two significant problems prohibiting the SNN from becoming as 'intelligent' as its non-spiking counterpart, the Artificial Neural Network (ANN). Firstly, Deep Learning research has made great achievements in the field of ANNs and dominated state-of-the-art solutions for AI engineering tasks, e.g. exceeding human-level performance on image classification [He et al., 2015], see Chapter 3 for more examples. However, the fundamental differences in data representation and neural computation between spiking and artificial neurons make it difficult to transform ANN models into SNN algorithms, see Chapter 2 for more detail. Secondly, the computational cost for simulating large SNNs of size comparable to commonly-used deep ANNs was considered to be infeasible, though this has gradually been solved by NE.

With the neuromorphic platforms ready for massive SNN simulations, this, therefore, is the main research problem: to improve the cognitive performance of SNNs to catch up with that of ANNs. Hence, researchers turn to Deep Learning to build 'smarter' SNNs. Initial studies have shown that SNNs can be trained by first training an equivalent deep ANN and then transferring the tuned weights to the SNN; this method is called 'off-line' training, since it does not take place on SNNs directly, but rather on ANNs instead. Chapter 4 discusses these 'off-line' training models in detail, and proposes a simple, generalised, off-line SNN training method to overcome the problems of poor modelling accuracy and high computational complexity of the existing methods [Jug et al., 2012; Hunsberger and Eliasmith, 2015; Diehl et al., 2015b]. To embed the biologically-plausible learning rules into deep SNN training, researchers take an extra step to 'on-line' methods where Deep Learning modules can be trained purely on SNNs in an event-driven manner, see Chapter 5. Previous work [Neil, 2013; Neftci et al., 2013; Burbank, 2015] has failed to provide SNNs with recognition accuracy equivalent to ANNs due to the lack of model formalisation and accurate parameter settings. We continue the inspiring work on these biologically-plausible 'on-line' training methods and propose a formalised method to train multi-layered Deep Learning modules on SNNs.

To provide meaningful comparisons between these proposed SNN models and other existing methods within the rapidly advancing field of NE, we propose a large dataset of spike-based images/videos to unify data resources for objective comparisons; and a corresponding evaluation methodology to estimate the overall performance of SNN models and their hardware implementations in Chapter 6. Moreover, we transform one of the common datasets widely used in Computer Vision into spike-based dataset to enable meaningful comparisons between SNNs and conventional machine learning methods.

1.1 Motivation and Aims

NE has led to the development of biologically-inspired computer architectures whose long-term goal is to approach the performance of the human brain in terms of energy efficiency and cognitive capabilities. Although there are a number of neuromorphic

platforms available for large-scale SNN simulations, the problem of programming these brain-like machines to be competent in cognitive applications still remains unsolved. On the other hand, Deep Learning has emerged in ANN research to dominate state-of-the-art solutions for cognitive tasks. Thus the main research problem emerges of understanding how to operate and train biologically-plausible SNNs to close the gap in cognitive capabilities between SNNs and ANNs on AI tasks.

Enabling this massively-parallel neuromorphic hardware to deliver state-of-the-art performance on AI tasks will be a big step towards Neuromorphic Cognition. It will contribute to the ultimate goal of equipping brain-inspired computers with Human brain levels of energy efficiency and cognitive capability.

1.2 Thesis Statement and Hypotheses

Although fundamental differences in input/output representation and neural computation exist between spiking and conventional artificial neurons, the cognitive capability of SNNs can be improved to catch up with that of ANNs by embedding Deep Learning techniques in training SNNs. Deep Learning has not only successfully equipped ANNs with better-than-human performance on AI tasks, but also studies have proved the equivalent learning capability of SNNs, and neuromorphic hardware is ready for operating large-scale deep SNNs.

According to the thesis statement, the hypotheses are defined as follows:

- Deep SNNs can be successfully and simply trained off-line where the training takes place on equivalent ANNs and the tuned weights then transferred back to the SNNs, thus making them as competent as ANNs in cognitive tasks.

- Unsupervised Deep Learning modules can be trained on-line on SNNs with biologically-plausible synaptic plasticity to demonstrate a learning capability equivalent to ANNs.

- A new set of spike-based vision datasets can provide resources and corresponding evaluation methodology to support objective comparisons and measure progress within the rapidly advancing field of NE.

1.3 Contributions

The primary achievement of the work described in this thesis is the training of deep SNNs, both off-line and on-line, which closes the gap in cognitive capability between SNNs and ANNs. Other achievements contribute to the performance evaluation of SNN models and their hardware implementations. The contributions are:

- **A generalised and simple method for off-line SNN training.**

  The core elements of the training methods are a pair of novel activation functions used in ANNs: Noisy Softplus (NSP) and the Parametric Activation Function (PAF). NSP successfully models the firing activities of biologically-plausible spiking neurons with conventional activation functions of abstract values; and PAF maps these numerical values to concrete physical units in SNNs: current in nA and firing rates in Hz. The proposed activation functions solve the problem of the fundamental differences in data representation and neural computations between ANNs and SNNs, thus tackle the difficulties of transforming ANN models to SNNs. Moreover, they address the problems of inaccurate modelling and high computational complexity of existing approaches.

  This off-line training method consists of three simple steps: firstly, estimate parameter p for the PAF, $y = p \times f(x)$, using the proposed activation function NSP; secondly, use a PAF version of conventional activation functions, e.g. Rectified Linear Unit (ReLU), for ANN training; thirdly, the tuned weights can be transferred directly into the SNN without any further transformation. This method involves the least computational complexity while performing most effectively among existing algorithms.

  NSP is described in Chapter 4 and was published and presented at the International Conference on Neural Information Processing (ICONIP 2016); the work of generalised SNN training using PAF will be submitted to the IEEE Transactions on Neural Networks and Learning Systems (INNLS).

- **An on-line unsupervised learning algorithm working purely on event-based STDP for training spiking Autoencoders (AEs) and Restricted Boltzmann Machines (RBMs).**

Multiplying two numerical values, which is the core operation in the algorithms for training the Deep Learning modules of AEs and RBMs, can be represented with rate multiplication of a pair of rate-coded spike trains. The proposed formalised Spike-based Rate Multiplication (SRM) method transforms the product of rates to the number of coincident spikes emitted from a pair of connected spiking neurons, and the simultaneous events can be captured by the change of the synaptic efficacy using the biologically-plausible learning rule: STDP.

The SRM successfully tackles the problem of translating the weight tuning from numerical computations to event-based, biologically-plausible learning rules in SNNs. In addition, the numerical analysis of the proposed algorithm accurately estimates the parameters, thus closely mimicking the learning behaviour of the AE and RBM modules, and improves the learning performance compared to existing methods. Moreover, we propose solutions to the problem of continuous performance drop caused by correlated spike trains. Thus, spiking AEs and RBMs can be trained with SRM and approach the same, sometimes even superior, classification and reconstruction capabilities compared to their equivalent non-spiking models.

This work comprises Chapter 5. A paper on these findings is in preparation for submission to the Journal of Neural Computation.

- **A dataset and the corresponding evaluation methodology for comparisons of SNN models and their hardware implementations.**

  To objectively compare these proposed SNN models with other existing methods, we propose a Neuromorphic Vision dataset NE15-MNIST which is comprised of spike-encoded images/videos based on a standard computer vision benchmark, the MNIST [LeCun et al., 1998] dataset. The unified dataset satisfies the requirement for quantitatively measuring progress within the rapidly advancing field of NE and provides resources to support objective comparisons between researchers. In addition, a complementary evaluation methodology is presented to estimate the overall performance of SNN models and their hardware implementations, since new concerns relating to energy efficiency and recognition latency emerge in SNNs run on NE platforms.

We also present a potential benchmark system which is evaluated using the Poissonian subset of the NE15-MNIST dataset. It provides a baseline for further comparisons with upcoming SNN models.

The dataset was generated with the help of Garibaldi Pineda-García and Teresa Serrano-Gotarredona. This work comprises Chapter 6 and was published as a journal paper in Frontiers in Neuromorphic Engineering.

1.4 Papers and Workshops

1.4.1 Papers

Much of the work contributed to solving the main research problem of this thesis has either been published or is in the process of submission for publication.

- **Q. Liu**, and S. Furber, **Noisy Softplus: A Biology Inspired Activation Function**, International Conference on Neural Information Processing (ICONIP 2016). This paper [Liu and Furber, 2016] introduces the novel activation function, NSP, which solves the problem of accurately modelling the response firing activity of spiking neurons using conventional abstract activation functions. This paper comprises the first half of Chapter 4.

- **Q. Liu**, Y. Chen, G. García, and S. Furber, **Generalised Training of Spiking Neural Networks**, (to be submitted to INNLS). This paper extends the work of the NSP to solve the problem of mapping abstract numerical values of activation functions to concrete physical units in spiking neurons using PAF, and successfully formalises a simple off-line SNN training method which is also generalised to ReLU-like activation functions. The paper presents the work described in the rest of Chapter 4.

- **Q. Liu**, and S. Furber, **Spike-based Rate Multiplication for On-line SNN Training** (to be submitted to Neural Computation). This paper mainly comprises the work of Chapter 5, which proposes a method for on-line unsupervised training of SNNs equivalent to the conventional Deep Learning techniques: AEs and RBMs.

- **Q. Liu**, G. García, E. Stromatias, T. Gotarredona, and S. Furber, **Benchmarking Spike-Based Visual Recognition: A Dataset and Evaluation**, Frontiers in Neuromorphic Engineering. The work presented in this paper [Liu et al., 2016] mainly comprises the spike-based dataset **NE15-MNIST** and its corresponding evaluation method for Neuromorphic Vision proposed in Chapter 6. In addition, the paper also includes the contributions of the co-authors: the detailed description of a subset of this database and a case study as an example to validate the dataset and its evaluation.

Other publications build up the neuromorphic hardware system for complete event-based visual and auditory processing, providing deep spiking neural networks a valid hardware platform for running applications in biological real time.

- **Q. Liu**, and S. Furber, **Real-Time Recognition of Dynamic Hand Postures on a Neuromorphic System**, International Conference on Artificial Neural Networks (ICANN 2015). We develop an object recognition system operating in real-time on a complete neuromorphic platform in an absolute spike-based fashion. This paper paves the way for further study with solid proof of the capability of a real-time cognitive application built on a neuromorphic platform. In Chapter 2, we introduce this system as an existing vision-based neuromorphic hardware platform which comprises a Dynamic Vision Sensor (DVS) as the front-end and a massive-parallel SNN hardware simulator as the back-end.

- **Q. Liu**, C. Patterson, S. Furber, Z. Huang, Y. Hou and H. Zhang, **Modeling Populations of Spiking Neurons for Fine Timing Sound Localization**, International Joint Conference on Neural Networks (IJCNN 2013). This paper [Liu et al., 2013] presents a model of sound localisation to solve the problem of coarse time resolution of SNN simulations. Such an auditory processing system can be implemented on a similar neuromorphic hardware platform described above, which uses a silicon cochlea as the input (see Chapter 2).

- G. García, P. Camilleri, **Q. Liu**, and S. Furber, **pyDVS: An Extensible, Real-time Dynamic Vision Sensor Emulator using Off-the-Shelf Hardware**, The 2016 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016). This paper [Garibaldi et al., 2016] proposes a visual input system inspired

by the behaviour of a DVS but using a conventional digital camera as a sensor and a PC to encode the images (see Chapter 2).

1.4.2 Workshops

The author participated in workshops organised by the NE community, 1) to establish and contribute to collaborations on mutual interests; 2) to catch up with cutting-edge research and collect inspiration; and 3) to discuss the author's own findings with key researchers in the field.

- *Capo Caccia Cognitive Neuromorphic Engineering Workshop 2012.*

  Contributed to successful connections of SpiNNaker to neuromorphic sensors[1]. This formed the hardware platform for real-time SNN applications processing event-based sensor data.

- *Telluride Neuromorphic Cognition Engineering Workshop 2013.*

  Developed a real-time sound localisation system on the neuromorphic platform as a main contributor[2]. The work led to the publication of a journal paper [Lagorce et al., 2015].

- *Capo Caccia Cognitive Neuromorphic Engineering Workshop 2014.*

  Developed the real-time neural activity visualiser for the project of 'Integrated Neurorobotics for Real-World Cognitive Behaviour' [3].

- *Capo Caccia Cognitive Neuromorphic Engineering Workshop 2015.*

  Inspired by the projects on Deep Learning in the workshop[4], the author later proposed the off-line SNN training method and the unsupervised on-line learning algorithm of deep SNNs, and led the discussion of benchmarking neuromorphic vision in the workshop [5].

[1] `https://capocaccia.ethz.ch/capo/wiki/2012/csnQian`
[2] `http://neuromorphs.net/nm/wiki/sound_localization`
[3] `https://capocaccia.ethz.ch/capo//wiki/2014/integrneurobot14`
[4] `https://capocaccia.ethz.ch/capo//wiki/2015/spikednn15`
[5] `https://capocaccia.ethz.ch/capo//wiki/2015/visionbenchmark15`

1.5 Thesis Structure

The thesis comprises the following seven chapters:

Chapter 1 introduces the origin and the motivation of the research, states the problem, defines the hypotheses and objectives, summarises the contributions and publications, and outlines the thesis.

Chapter 2 illustrates how biological neurons function, transmit signals between them, and are modelled by mathematical abstractions, thus to unveil the special features of spiking neurons that differ from the neurons of ANNs; and introduces SNN simulators both in software and in hardware including neuromorphic systems.

Chapter 3 gives an overview of popular architectures and models of Deep Learning and illustrates the mechanism of the Convolutional Networks (ConvNets), the AEs, and the RBMs in detail.

Chapter 4 demonstrates the generalised off-line SNN training method to confirm the first hypothesis that SNNs can be trained off-line and perform equivalently as ANNs in cognitive tasks.

Chapter 5 proposes an STDP-based learning algorithm for training spiking AEs and RBMs on-line; and test the second hypothesis that on-line training is able to improve the cognitive capabilities of SNNs and catch up with ANNs.

Chapter 6 puts forward the spike-based vision dataset and the evaluation methodology and presents a case study as a tentative benchmark running on SpiNNaker to assess the hardware-level performance against software simulators.

Chapter 7 summarises the research, discusses the contributions to the field, points out future directions and concludes the thesis.

1.6 Summary

In brief, Figure 1.1 summarises the introduction and demonstrates the outline of the thesis.

Chapter 1 introduces the aims of NE: modelling the brain, and building brain-like machines. Although progress has been made in both directions, it is still far from achieving the long term goal of Neuromorphic Cognition. With the support of accumulated knowledge of SNNs and the massive neuromorphic SNN simulators (both

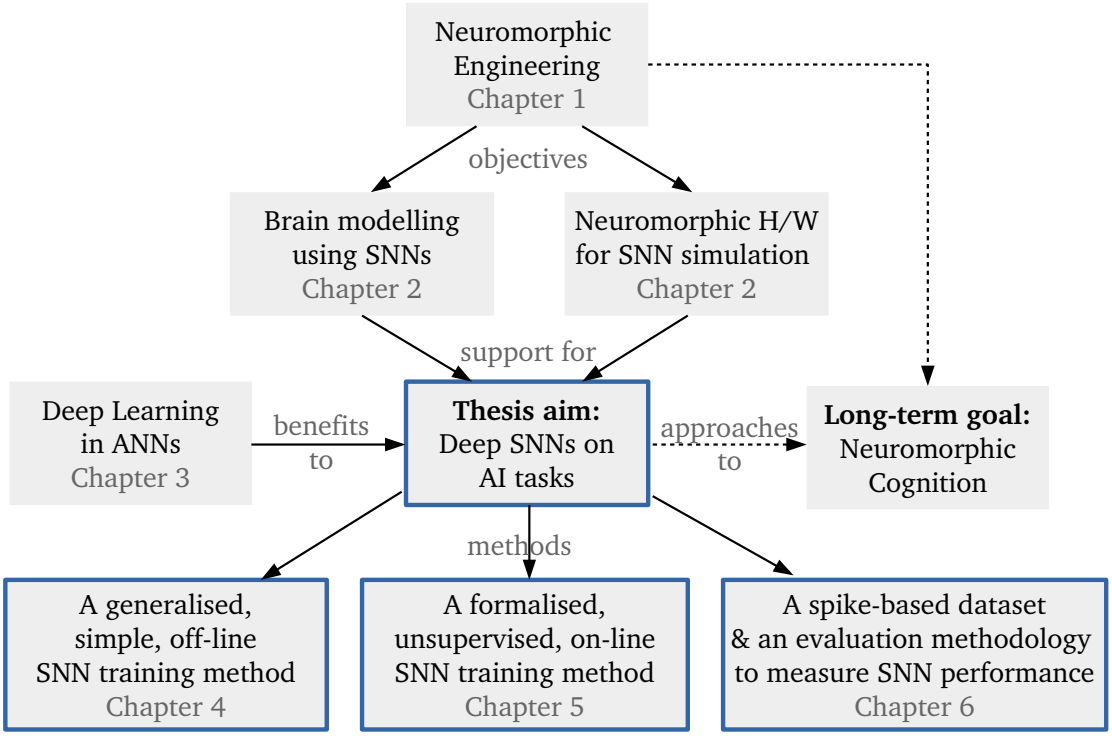


Figure 1.1: The outline of the thesis.

refer to Chapter 2), plus the huge success of Deep Learning in ANNs (see Chapter 3), the research aims at understanding how to operate and train biologically-plausible SNNs to close the gap in cognitive capabilities between SNNs and ANNs on AI tasks, thereby approaching Neuromorphic Cognition.

To achieve the thesis aim, we propose an off-line (Chapter 4) and an on-line (Chapter 5) SNN training method to bring Deep Learning advantages to SNNs, and provide a spike-based dataset and its corresponding evaluation methodology to measure the performance of SNN models and the neuromorphic hardware platforms in Chapter 6.

Chapter 2

Spiking Neural Networks (SNNs)

The so-called third generation of artificial neural network, the Spiking Neural Network (SNN), is comprised of spiking neurons which mimic the dynamics of biological neural behaviour. In this chapter we will demonstrate the special features of spiking neurons that differ from neurons of conventional Artificial Neural Networks (ANNs); these biologically-plausible neuronal operations are the root of the research problem raised in the thesis: how to operate SNNs to equip them with cognitive capabilities as competent as ANNs. Section 2.1 will illustrate how biological neurons function and transmit signals between them. The way neural dynamics are modelled by mathematical abstractions of spiking neurons is described in Section 2.2 , and finally existing SNN simulators are introduced in Section 2.3.

2.1 Biological Neural Components

At the cellular level, the central nervous system consists of two types of cell: neurons, the elementary processing units, and glial cells, the structural and metabolic supporters. Here we focus on the former, since neurons are the basic elements supporting higher brain functions such as cognition, thought and action. The human brain contains around a hundred billion (10^{11}) such processing units, and up to four orders of magnitude more connections (10^{15}) [Azevedo et al., 2009]. Despite being such a huge and complex system, neurons in the brain manage to send signals rapidly and precisely to other cells through these connections, thanks to their special structures.

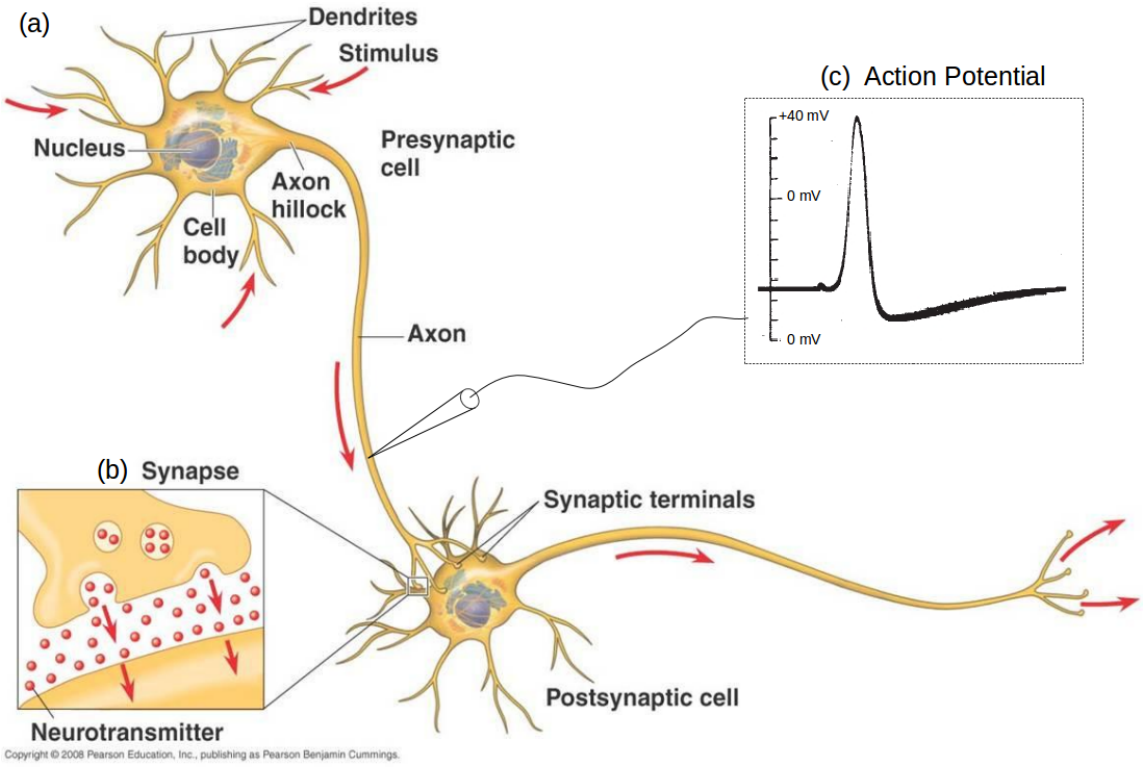


Figure 2.1: Two neurons connected by synapses. A neuron comprises three functional parts: dendrites, the cell body, and the axon. (a) A pre-synaptic cell connects to its post-synaptic cell through synapses (b) [Reece et al., 2011], and the neural signal, the action potential (c) [Hodgkin and Huxley, 1939], propagates in the direction of the red arrows.

2.1.1 Neuron

A typical neuron comprises three functional parts: dendrites, a cell body (soma), and an axon, see Figure 2.1(a). The dendrites of a neuron receive stimuli from other neurons, and transmit the neural signal to the neuron's soma. The soma is the cell body of the neuron, the location of the nucleus, and functions as a non-linear processor which triggers an output signal when the accumulated total input exceeds some threshold. The output signal initiates from the axon hillock where the axon emerges from the soma, and is propagated through the axon to other neurons. Most neurons have only one axon, but may connect to many neurons by branching out axon terminals.

The signal delivery from one neuron to another occurs at the junction between these two neurons, which is called a synapse, see Figure 2.1(b). The configuration can be seen as a pre-synaptic cell which sends the signal, and a post-synaptic cell which receives it.

2.1.2 Neuronal Signals

Neuronal signals propagated among neurons are short electrical pulses, and Figure 2.1(c) shows the original recording of such a so-called **action potential** observed on a squid giant axon. A typical action potential, also known as a '**spike**', is of about 100 mV amplitude and lasts 1-2 ms. Usually, there is a time period immediately after a spike that the neuron is unresponsive to any further stimulus. This minimal time difference between two spikes of a single neuron is the absolute **refractory period** during which no spike can be generated. After the absolute refractory period, it is still difficult but possible to fire a spike during the relative refractory period.

The size and duration of the spikes do not vary much among different species, and maintain the same form as the electrical pulses propagate along the axon as illustrated in Figure 2.1(c). Therefore, the form of an action potential carries little information; it is the frequency and timing of the spikes that encode the messages. A sequence of action potentials generated by a single neuron is called a '**spike train**', which can be viewed as binary events happening in discrete time where 'on' indicates a spiking event within a time step whereas 'off' means none. Information can be encoded in the frequency and timing of these binary events.

The rate coding model states that the spiking rate represents the intensity of a stimulus, e.g. as the stimulus becomes stronger, the frequency of the action potentials also increases. An example of the tuning curve of a V1 (visual area one of the visual cortex) simple cell responding to different stimulus orientation is shown in Figure 2.2. As the stimulus becomes more aligned to the preferred orientation (0°) of the neuron, the firing rate increases.

Rate coding works well when the stimulus is changing slowly and the observation time period is long enough to estimate the firing rate. However, in practice the stimulus, e.g. visual sensory input, varies on a fast time scale and the neurons respond within a short reaction time. Thus, temporal coding encodes information in the precise timing of spikes which is considered to be a candidate for the encoding of a fast changing stimulus.

Sound localisation requires temporal coding at sub-millisecond precision, which is a good example of one of the temporal coding schemes, phase locking. Figure 2.3 shows phase-locked spike trains generated by Inner Hair Cells in the cochlea. Phase

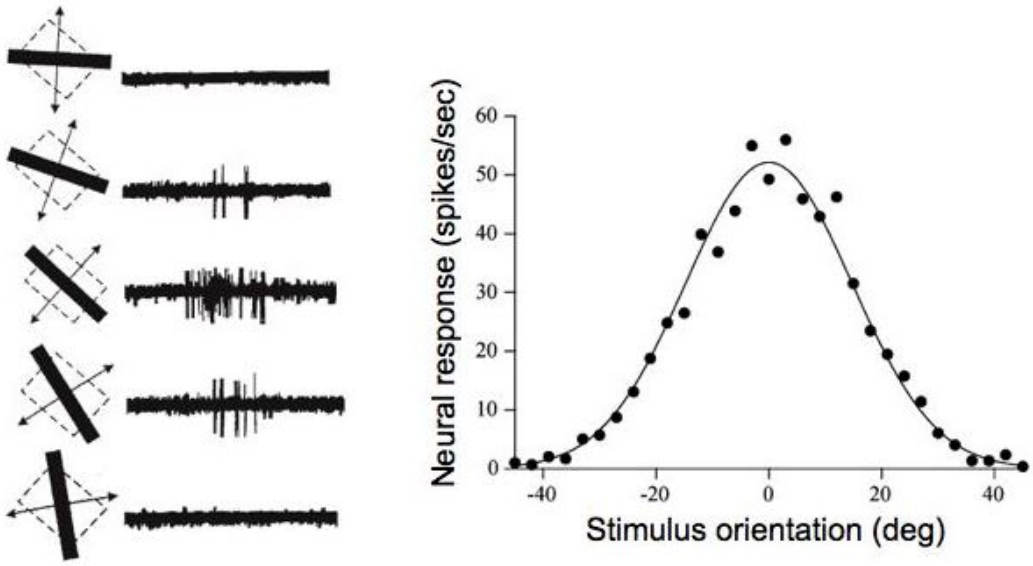


Figure 2.2: Example of rate coding: spike trains for different stimulus orientation (left) of a V1 simple cell of a cat, and the tuning curve (firing rate against stimulus orientation) of the neuron (right) [Hubel and Wiesel, 1962]. The square indicates the visual receptive field of the neuron, and a bar is placed at different orientations and moves to the direction perpendicular to its orientation. As the stimulus becomes more aligned to the preferred orientation (0°) of the neuron, the firing rate increases.

locking forms the basis of detecting time differences of binaural sound inputs.

Time-to-first-spike encodes the information according to the intensity of a stimulus where a spike shortly after a reference signal indicates a strong stimulation and a later action potential is interpreted as a weaker input. The tactile afferent information generated by forcing fingertips from various directions is encoded in such a time-to-first-spike coding scheme [Johansson and Birznieks, 2004]. Synchrony coding also can be found in the brain, where neurons representing the same 'concept' always fire at the same time [Von Der Malsburg, 1994], for example in object recognition [Gray and Singer, 1989]. Established from the context of fast object recognition, rank-order coding was proposed where the precise time of spikes is discarded, but rather uses the relative order of spikes among a group of neurons [Gautrais and Thorpe, 1998].

2.1.3 Signal Transmission

The spike, as an electrical signal, propagates to another neuron through the junction between these two neurons, a chemical synapse. The axon terminal of a pre-synaptic neuron approaches very close (within about 20 nm) to the dendrites (or cell body) of

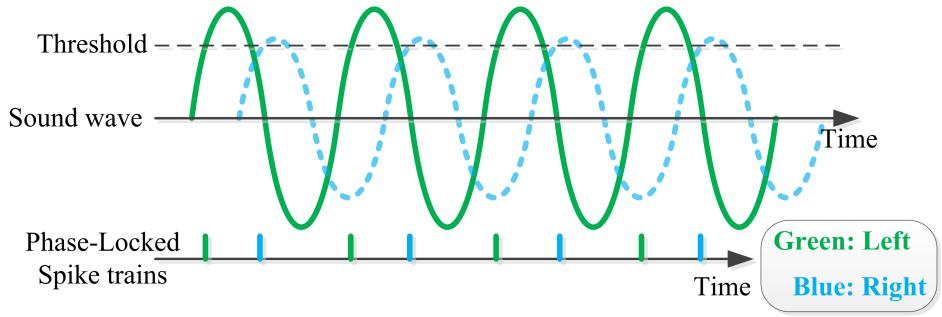


Figure 2.3: Example of temporal coding: phase-locked spike trains generated by simulated Inner Hair Cells in the cochlea [Liu et al., 2013]. A sound source generates a sine wave of a certain frequency and conducts to two ears with a time difference and different amplitude due to the angle and distance of the sound source to the head. Two spike trains respond to different phases manipulated by a threshold of the sound waves. Sound localisation can be resolved by calculating the time difference and/or level difference of these sound waves which are encoded in the spike trains.

a post-synaptic neuron. The tiny space between neurons at a synapse is called the synaptic cleft, which is illustrated in Figure 2.1(b). At such a chemical synapse, the action potential generated by the pre-synaptic neuron triggers chemical neurotransmitter molecules to be released into the synaptic cleft, and once the post-synaptic neuron detects these neurotransmitters it opens specific ion channels to allow electrical current in. Hence, synapses complete the transformation from electrical signal to chemical molecules and then back to ion influx. The amount of neurotransmitter determines the strength of the current flow into the post-synaptic neuron. Thanks to synaptic plasticity, changes of chemical synapses enable modulations of the synaptic efficacy, and form the neuronal correlation of learning and memory.

2.2 Modelling Spiking Neurons

2.2.1 Neural Dynamics

The effect of an ion influx on the post-synaptic neuron caused by spike transmission is a change of potential difference between the interior and exterior of the cell body, which is called the **membrane potential**. The membrane potential of a post-synaptic neuron stays at a **resting potential** in the absence of an input. As soon as a spike arrives, the membrane potential will be either depolarised (increased) or hyper-polarised (decreased) according to the type of synapse, and go back to the resting potential

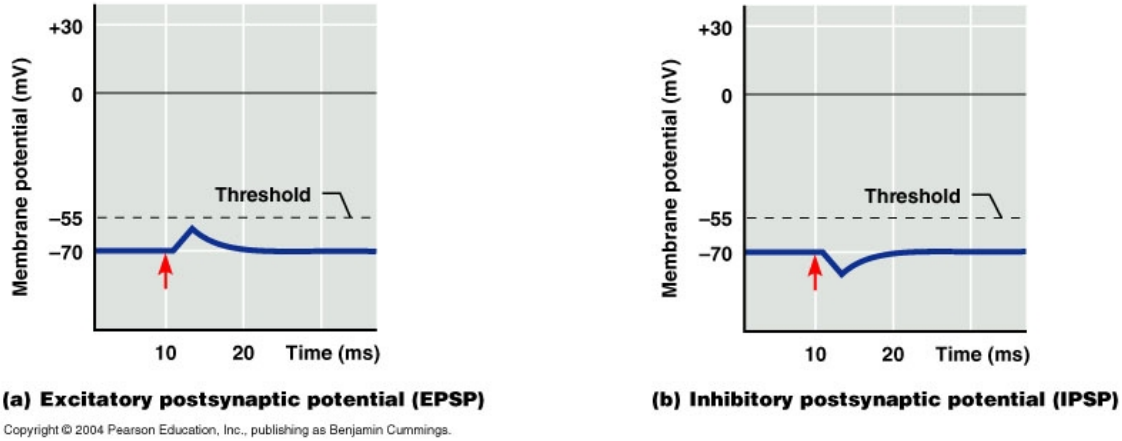


Figure 2.4: Post-synaptic potential driven by a spike, where the red arrow represents a spike arriving at the neuron [Marieb and Hoehn, 2007].

driven by membrane leakage. The state of the membrane potential change caused by a single spike is called the Post-Synaptic Potential (**PSP**). Thus, a spike transmitted by an excitatory synapse triggers a positive PSP, called an Excitatory Post-Synaptic Potential (**EPSP**), see Figure 2.4(a); a negative change, an Inhibitory Post-Synaptic Potential (**IPSP**), is driven by an inhibitory synaptic event and is shown in Figure 2.4(b). Spikes arriving at different synapses at the same post-synaptic neuron have PSPs of different amplitudes according to the synaptic efficacy.

Multiple PSPs have an accumulative effect on the membrane potentials both in

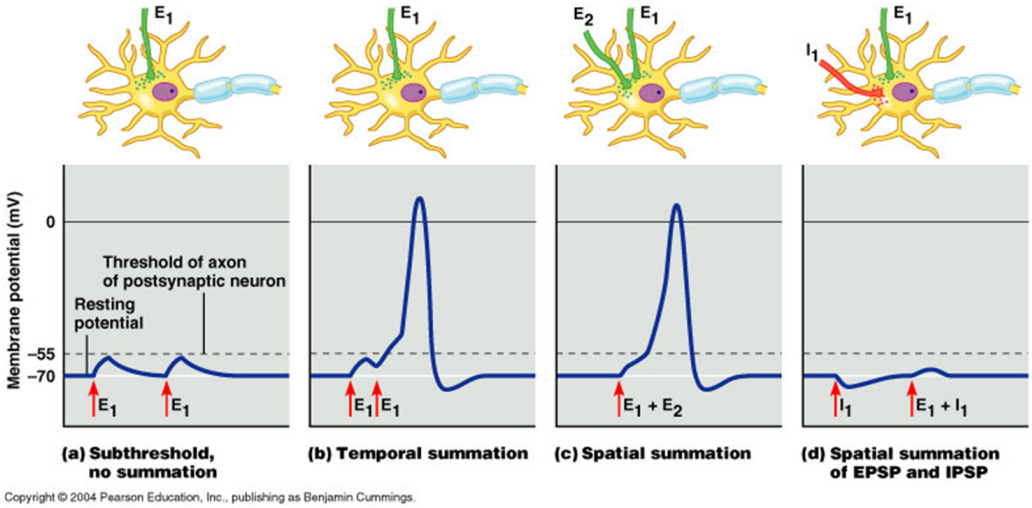


Figure 2.5: Summation of post-synaptic potentials [Reece et al., 2011]. (a) Single EPSPs are usually not strong enough to trigger an action potential without summation. (b) Temporal summation of two EPSPs of the same synapse generates an action potential. (c) Spatial summation of two EPSPs of two synapses generates an action potential. (d) Spatio-temporal summation of both EPSP and IPSPs.

temporal and spatial terms. The accumulation performs a simple summation of PSPs until the membrane potential reaches a **threshold**, when an action potential is generated at the post-synaptic neuron. Figure 2.5 illustrates temporal and spatial summations of PSPs under different circumstances. The temporal summation refers to the accumulated effect of a single synapse where the spatial one integrates the PSPs triggered by multiple synapses.

The neural dynamics of the membrane potentials, PSPs, and spike trains are all time dependent, while the neurons of ANNs, e.g. sigmoid units, only cope with numerical values representing spiking rate, without timing information, see Figure 2.6. A regular artificial neuron (Figure 2.6(a)) comprises a weighted summation of input data, $\sum x_i w_i$, and an activation function, f, applied to the sum. Usually, a bias is included in the weighted summation which increases the expression ability of a neuron. However, in this thesis we remove biases of both ANNs and SNNs to simplify the neural models and to reduce the number of parameters. Nevertheless, our experimental results show that the performance almost keeps the same when solving a relatively simple task, the MNIST. Thus the inputs of a spiking neuron (Figure 2.6(b)) are spike trains generated by pre-synaptic neurons, which create PSPs on the post-synaptic neuron and trigger a spike train as the output of this spiking neuron. These fundamental differences in input/output representation and neural computation lead to special model descriptions of spiking neurons (illustrated in the next section), and raise the difficulties of transforming ANN models to spiking neurons. Hence, this research aims to address the problem of how to operate and train biologically-plausible SNNs to be as competent in cognitive tasks as are ANNs.

2.2.2 Neuron Models

The keys to modelling a spiking neuron are:

- to mathematically formalise the evolution of the membrane potential;
- to state a mechanism of spike generation.

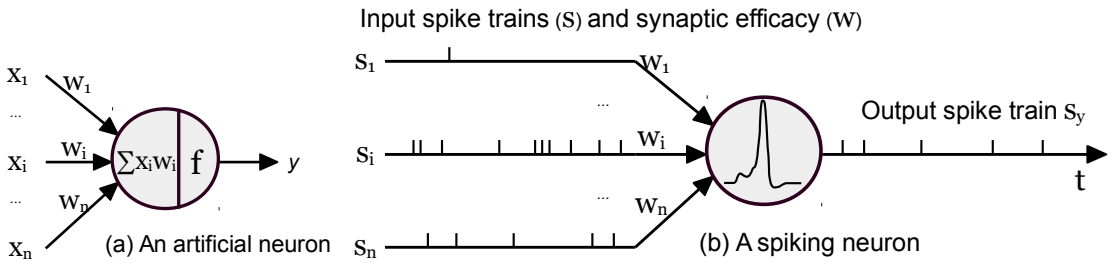


Figure 2.6: Comparisons of processing mechanisms of an artificial and a spiking neuron. (a) An artificial neuron takes numerical values of vector $\mathbf{x}$ as input, works as a weighted summation followed by an activation function f. (b) Spike trains flow into a spiking neuron as input stimuli, trigger linearly summed PSPs through synapses with different synaptic efficacy $\mathbf{w}$, and the post-synaptic neuron generates output spikes when the membrane potential reaches some threshold.

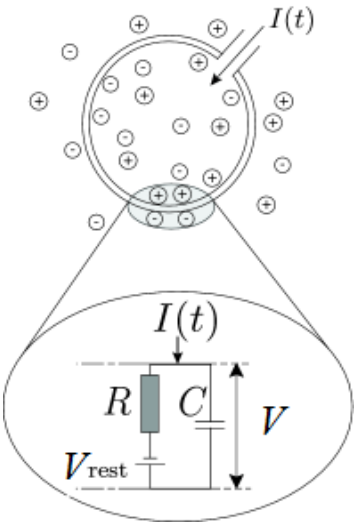


Figure 2.7: The cell membrane acts like a Resistor-Capacitor (RC) circuit [Gerstner et al., 2014]. Input current $I(t)$ flows into a neuron which charges the capacitor C and leaks through the resistance R in line with a battery V_{rest}.

Leaky Integrate-and-Fire (LIF) Model

The evolution of membrane potential V can be simplified to a Resistor-Capacitor (RC) circuit which consists of a membrane capacitance C_m and a membrane resistance R_m, both driven by an input current flow I, see Figure 2.7. In the resting state without any input, the membrane potential V stays at the same potential as the battery V_{rest}. When current flows into the neuron, it will charge the capacitor with current $I_C(t)$ and discharge through the resistance with current $I_R(t)$. When the input current stops the

capacitive charge will decay back to V_{rest} by leaking through the resistance:

$$\begin{aligned} I(t) &= I_R(t) + I_C(t) \\ &= \frac{V - V_{rest}}{R_m} + C_m \frac{\mathrm{d}V}{\mathrm{d}t} \quad . \end{aligned} \tag{2.1}$$

The standard form of the LIF model describes the sub-threshold membrane potential evolution as follows:

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - V_{rest}) + R_m I(t) \ , \tag{2.2}$$

where $\tau_m = C_m R_m$ is called the membrane time constant, and as soon as the membrane potential reaches the threshold V_{thresh}, it is set to a reset potential V_{reset}, which is usually lower than V_{rest}:

$$V = V_{reset} \ . \tag{2.3}$$

The simple LIF model uses: (1) a linear differential equation to describe the evolution of membrane potential; and (2) a threshold to generate a spike.

Hodgkin-Huxley Model

The Hodgkin-Huxley model [Hodgkin and Huxley, 1952] is the Nobel Prize winning model that explains the ionic mechanisms generating and transmitting action potentials in the squid giant axon. The current $I_R(t)$ which flows through the membrane resistance is determined by three ion channels: a leak channel with conductance g_L, the sodium channel with conductance g_{Na} and the potassium channel with conductance g_K. The currents which flow through these channels are all proportional to the difference between the membrane potential and the reversal potentials of the channels: $V - E_L$, $V - E_{Na}$, and $V - E_K$ respectively. Thus Equation 2.1 is detailed as:

$$\begin{aligned} I(t) &= I_L(t) + I_{Na}(t) + I_K(t) + I_C(t) \\ &= g_L(V - E_L) + g_{Na} m^3 h (V - E_{Na}) + g_K n^4 (V - E_K) + C_m \frac{\mathrm{d}V}{\mathrm{d}t} \quad . \end{aligned} \tag{2.4}$$

The Hodgkin-Huxley model can be seen as a non-linear differential equation with four state variables, V, m, h and n that change against time:

$$
\begin{aligned}
C_m \frac{\mathrm{d}V}{\mathrm{d}t} &= I(t) - g_K n^4 (V - E_K) - g_{Na} m^3 h (V - E_{Na}) - g_L (V - E_L) \ , \quad \text{and} \\
\frac{\mathrm{d}m}{\mathrm{d}t} &= \alpha_m(V)(1 - m) - \beta_m(V) m \ , \\
\frac{\mathrm{d}n}{\mathrm{d}t} &= \alpha_n(V)(1 - n) - \beta_n(V) n \ , \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \alpha_h(V)(1 - h) - \beta_h(V) h \ ,
\end{aligned}
\tag{2.5}
$$

where $\alpha(V)$ and $\beta(V)$ are empirical functions of membrane potential.

With regard to the mechanism of spike initiation, the most significant property of the Hodgkin-Huxley model is that the model is able to generate action potentials with the changes of those dynamic internal variables alone.

The Hodgkin-Huxley equations provide a detailed, quantitative, and reasonably accurate mathematical model explaining the evolution of the membrane potential and the action potential [Byrne et al., 2014]. However, its numerical complexity and highly non-linear characteristics prohibit it from being intuitively understood and make large-scale simulations too expensive. Therefore, neural model selection should take account of objectives, degree of detail and computational power.

Izhikevich Model

The Izhikevich model was proposed to solve the problems of computational complexity of the Hodgkin-Huxley model and the insufficient capability of LIF model to reproduce the complex dynamics of cortical neurons [Izhikevich, 2003]. Thus the model can be employed to simulate large-scale brain models comprising real biological neurons.

The membrane potential evolves in accordance with a pair of differential equations:

$$
\begin{aligned}
\frac{\mathrm{d}v}{\mathrm{d}t} &= 0.04v^2 + 5v + 140 - u - I(t) \ , \\
\frac{\mathrm{d}u}{\mathrm{d}t} &= a(bv - u) \ ,
\end{aligned}
\tag{2.6}
$$

where v is the membrane potential and u represents the membrane recovery which negatively feeds back to v.

In terms of spike generation, the initiation part of an activation potential is produced by the equations, but a resetting scheme is needed:

$$\begin{cases} v = c \\ u = u + d \end{cases} \quad \text{when } v \geq 30. \tag{2.7}$$

Parameters a, b, c, and d are constant, which can be configured to reproduce various neural dynamics of real biological neurons [Izhikevich, 2004].

2.2.3 Synapse Model

Applying spiking neuron models to synaptic spike transmission, we can use two types of synapse: current-based and conductance-based models. Thus the synaptic efficacy w determines either the input current intensity flowing through the synapse:

$$I(t) = w(t) \ , \tag{2.8}$$

or the electrical conductance g_{syn} of the ion channel:

$$I(t) = g_{syn}(V - E_{syn}) = w(t)(V - E_{syn}) \ , \tag{2.9}$$

where E_{syn} indicates the reversal potential of a synapse. Both equations identify the strength of a synaptic current, thus simply adding up all synaptic currents on the same post-synaptic neuron represents the external current $I(t)$ for all the neuron models stated in Section 2.2.2.

The current flow usually has a much longer time constant than an action potential and decays over time, thus a simple exponential decay is able to model the decaying synaptic efficacy. Assuming spikes are delivered at time $t = t_0, t_1, ..., t_n$, the initial synaptic weight is set to w_0 and τ_{syn} is the synaptic time constant, the decaying synaptic current or the conductance can be described as:

$$w(t) = \sum_k w_0 e^{-(t-t_k)/\tau_{syn}} \ . \tag{2.10}$$

In this thesis we mostly employ LIF neurons and a current-based synapse model with decaying synaptic efficacy, due to its simple mathematical expression, low numerical complexity and high-level abstraction hiding much of the detailed neural dynamics. Therefore, at the initial stage of merging artificial Deep Learning with biologically-plausible SNNs, we can (1) target standard LIF neurons which are supported by most

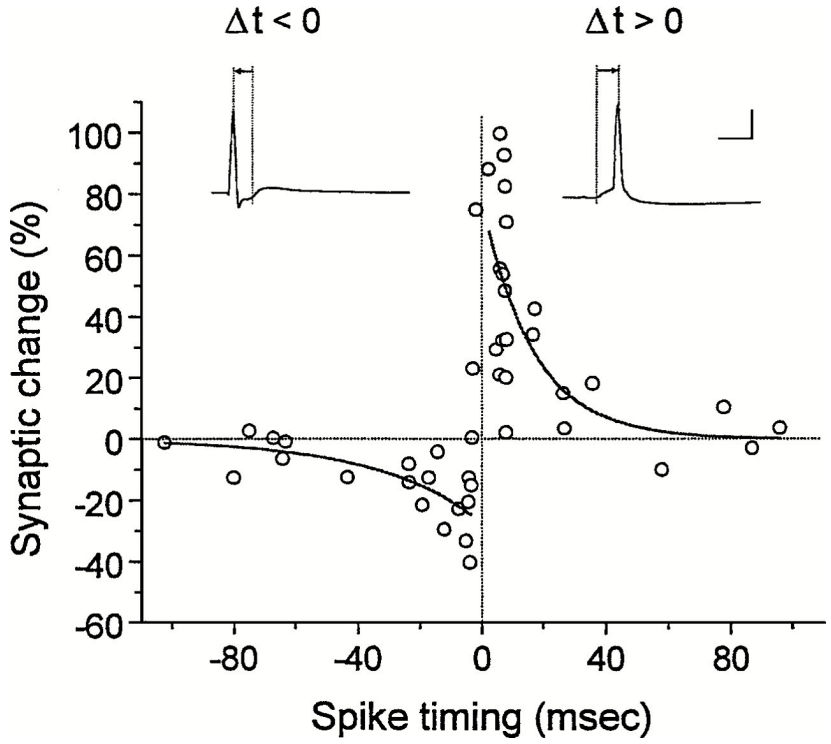


Figure 2.8: Spike-Timing-Dependent Plasticity (STDP) [Bi and Poo, 2001]. The circles record the synaptic weight change of real biological observations on 60 pairs of hippocampal neurons. Curves of exponential decays against relative timing of pre- and post-synaptic spikes fit well to the real biological data.

of the neuromorphic hardware systems; (2) simulate large-scale SNNs with deep architectures without a tight limitation on computational power; and (3) have fewer parameters thus resulting in a simplified problem.

2.2.4 Synaptic Plasticity

As mentioned in Section 2.1.3, synaptic plasticity provides the neuronal level of learning and the memory of the brain. Biological observations have provided evidence that modulations of the synaptic efficacy depend on the relative timing of the pre- and post-synaptic spikes [Bi and Poo, 1998]. This mechanism is known as Spike-Timing-Dependent Plasticity (**STDP**) [Song et al., 2000], and the standard STDP learning window is illustrated in Figure 2.8:

$$\Delta w = \begin{cases} A_+ e^{-\Delta t/\tau_+}, & \text{when } \Delta t \geq 0 \ , \\ -A_- e^{\Delta t/\tau_-}, & \text{when } \Delta t < 0 \ . \end{cases} \tag{2.11}$$

The synaptic weight is potentiated when a post-synaptic spike fires later than a pre-synaptic spike, and the amplitude of such a potentiation is determined by the curve of

exponential decay with a time constant τ_+ and an initial quantity A_+; however, when a post-synaptic spike is generated before a pre-synaptic one, synaptic depression will occur according to the exponential decay defined by τ_- and $-A_-$.

Besides the standard STDP model [Song et al., 2000], also known as the additive model, stated in Equation 2.11, variations of the STDP learning rule have been proposed to satisfy different learning speeds and classification accuracy. Multiplicative STDP [Morrison et al., 2008] is an alternative model where the weight change is not only dependent on the relative timing, but also on the current synaptic efficacy:

$$\begin{cases} A_+(w) = (w_{max} - w)\eta_+ \quad , \\ A_-(w) = w\eta_- \quad , \end{cases} \tag{2.12}$$

where A_+ and A_- in Equation 2.11 become functions of variables w, η_+ and η_- define the learning rate of the weight potentiation and depression, and w_{max} is the maximum synaptic efficacy while 0 is the minimum. We use the multiplicative STDP for SNN training in Section 6.5. It performs better than the additive STDP when used in such a classification task, since the weight are much more distributed in the working range.

In Chapter 5 we exploit a much simplified version of asymmetric rectangular STDP where the weight change is just a constant within an STDP window, τ_{win}:

$$\Delta w = \begin{cases} \eta \ , & \text{when} \quad 0 \leq \Delta t \leq \tau_{win} \\ 0 \ , & \text{otherwise} \end{cases} \tag{2.13}$$

We choose this simple algorithm for the straightforward linear conversion of the number of coincident spikes to the weight update.

2.3 Simulating Networks of Spiking Neurons

In the previous section, we described neural dynamics as mathematical models at the neuronal level. However, it is challenging to simulate a large SNN with a high volume of synaptic connections, even using simple models such as LIF, because of the high event rate (10^4 synaptic events per second per single neuron on average). Addressing this problem, existing solutions vary from software simulators to neuromorphic hardware.

2.3.1 Software Simulators

Existing approaches to software simulation can be seen as: 'clock-driven' where the neural state is updated with some fixed time resolution, or 'event-driven' where the membrane potential is only modified when a spike arrives. The synchronous 'clock-driven' method uses numerical integration for solving the Ordinary Differential Equations (ODEs), that describe the evolution functions of the membrane potential with respect to time. However, with updates only on the time clocks (usually at 1 ms resolution), the non-linear differential equations can only be approximated rather than solved, and the spike times lose precision since they are bound to discrete time steps. 'Event-driven' approaches, in comparison, are accurate since they use explicit solutions of the ODEs and the spike arrival time is not rounded to time bins. Unfortunately, apart from LIF neurons, all the other models described in Section 2.2.2 are analytically unsolvable. The high synaptic event rate (10^4 Hz per neuron) takes no advantage of computational efficiency using this asynchronous approach. Therefore, most of the popular software simulators use a 'hybrid' solution, including NEST [Gewaltig and Diesmann, 2007] and Brian [Goodman and Brette, 2008], where neural state is updated synchronously, but the synapse operates in an event-based way.

Another software tool, PyNN [Davison et al., 2008], is a description language for building SNNs; it abstracts away the detail of various simulators and provides unified APIs for any simulator that supports it. Consequently, neuroscientists and SNN designers do not need to learn different 'languages' for specific simulators, and the models written in PyNN are supposed to run freely on the supporting simulators.

Most of the SNN models developed in this thesis are described in PyNN and run on NEST, and some of them are also tested on a hardware simulator, SpiNNaker [Furber et al., 2014], which will be introduced in the following section. In Chapter 5 we develop our own SNN simulator to implement and test a proposed learning algorithm, and the simulator follows the synchronous convention due to its programming simplicity and flexible neural model selection.

2.3.2 Neuromorphic Hardware

Neuromorphic systems can be categorised as analogue, digital, or mixed-mode analogue/digital, depending on how neurons, synapses and spike transmission are implemented. Some analogue implementations exploit sub-threshold transistor dynamics to emulate neurons and synapses directly in hardware [Indiveri et al., 2011] and are more energy-efficient while requiring less area than their digital counterparts [Joubert et al., 2012]. However, the behaviour of analogue circuits is hard to control through the fabrication process due to transistor mismatch [Indiveri et al., 2011; Pedram and Nazarian, 2006; Linares-Barranco et al., 2003], and achievable wiring densities render direct point-to-point connections impractical for large-scale systems.

The majority of mixed-mode analogue/digital neuromorphic platforms, such as the High Input Count Analog Neural Network (HI-CANN) [Schemmel et al., 2010], Neurogrid [Benjamin et al., 2014], and HiAER-IFAT [Yu et al., 2012], use analogue circuits to emulate neurons and digital packet-based technology to communicate spikes using Address-Event Representation (AER) [Lazzaro and Wawrzynek, 1995] protocol. This enables reconfigurable connectivity patterns between the neurons and fulfils the real-time requirement.

Digital neuromorphic platforms such as TrueNorth [Merolla et al., 2014] use digital circuits with finite precision to simulate neurons in an event-driven manner to minimise the active power dissipation. Such systems suffer from limited model flexibility, since neurons and synapses are fabricated directly in hardware with only a small subset of parameters under the control of the researcher. The SpiNNaker many-core neuromorphic architecture [Furber et al., 2014] uses low-power programmable cores and scalable event-driven communications hardware allowing neural and synaptic models to be implemented in software. While software modelling provides great flexibility, digital platforms generally have reduced precision (due to the inherent discretisation) and higher energy consumption when compared to analogue platforms.

2.3.3 Neuromorphic Sensory and Processing Systems

Neuromorphic engineers have successfully produced visual and auditory silicon devices mimicking the biological retina and cochlea, and boosted the applications of spike-based sensory processing in artificial vision and audition.

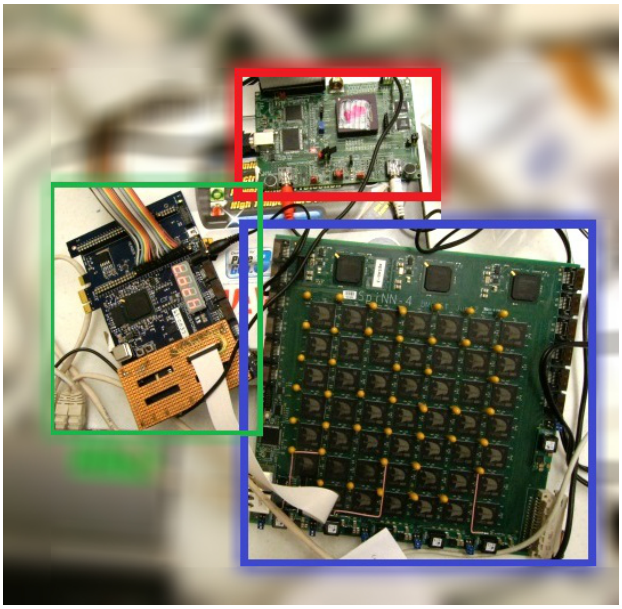
The visual input is captured by a DVS (Dynamic Visual Sensor) silicon retina [Delbruck, 2008; Serrano-Gotarredona and Linares-Barranco, 2013], which is quite different from conventional video cameras. Each pixel generates spikes when its change in brightness reaches a defined threshold; thus, instead of buffering video into frames, the activity of pixels is sent out and processed continuously with time. The level of activity depends on the contrast change; pixels generate spikes faster and more frequently when they are subject to more active change. The sensor is capable of capturing very fast moving objects (e.g., up to 10K rotations per second), which is equivalent to 100K conventional frames per second [Leñero-Bardallo et al., 2011]. However, DVSs on the market are still expensive to purchase, thus we present an extensible behavioural emulator of a DVS using a conventional digital camera, pyDVS [Garibaldi et al., 2016].

The binaural silicon cochlea [Liu et al., 2010] models the elementary functions of the cochlea including the basilar membrane, the Inner Hair Cells (IHCs), and the Spiral Ganglion Cells (SGCs). The input sound wave of each cochlea is filtered by a 64 channel bank of cascaded filters to model the frequency distribution along the basilar membrane. IHCs located at each frequency channel perform approximately as half-wave rectifiers, since they release neurotransmitter only when their stereocilia bend in one direction driven by the basilar membrane. The transformation from mechanical waves to electrical action potentials completes at the SGCs, where four pulse-frequency modulators at each frequency channel act like SGCs and generate spikes with individual thresholds.

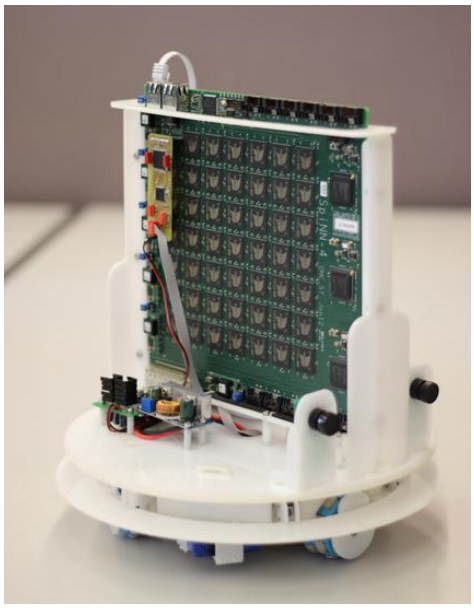
The output spikes from the sensory devices are then communicated to back-end SNN processing using the asynchronous AER protocol. Visual/auditory recognition on such complete neuromorphic hardware systems has emerged in pursuit of efficient energy cost and low sensory latency. However, these hardware SNN simulators either run on FPGAs [Neil and Liu, 2014; Kiselev et al., 2016] or analogue circuits [Qiao et al., 2015] thus are limited in scale.

SpiNNaker provides an ideal platform for real-time visual/auditory processing with

(a) Picture of the neuromorphic visual processing platform. From left to right: a silicon retina, an FPGA board which converts AER packets to SpiNNaker format, and a 48-node SpiNNaker system.



(b) Picture of the neuromorphic auditory processing platform which is similar to (a) in that a silicon cochlea (top) connects to a SpiNNaker board (lower right) through an FPGA (left).



(c) Picture of the omni-directional platform with embedded low-level motor control and elementary sensors: stereo silicon retinas, wheel encoders, and a bump-sensor ring.

Figure 2.9: Three set-ups of neuromorphic sensory and processing hardware platform using SpiNNaker.

large SNN models. Figure 2.9 shows three set-ups of such 'stand-alone' neuromorphic sensory and processing hardware platforms, which can be operated on their own as closed-loop systems (Figure 2.9(c)). The visual system, shown in Figure 2.9(a), ran a 5-layered SNN model [Liu and Furber, 2015] we designed for live gesture recognition, which contained a network of 74,210 neurons and 15,216,512 synapses, and used 290 SpiNNaker cores in parallel and reached 93.0% accuracy. We also successfully implemented a sound localisation model of spiking neurons on an auditory system, see Figure 2.9(b), which could operate with input spikes with sub-millisecond resolution [Lagorce et al., 2015]. These works prove that real-time, large-scale, sensory neuromorphic systems are ready for further study in effective cognition and the genuine intelligence capabilities of such biological-plausible machines.

2.4 Summary

This chapter introduced the structure and behaviour of biological neurons and illustrated how these neural dynamics can be modelled by mathematical abstractions as spiking neurons, which are the basic components of an SNN. The difference between biologically-plausible spiking neuronal operations and rate-based artificial activations holds the key to the research question: how to equip SNNs with cognitive capabilities equivalent to ANNs. Finally, we gave an overview of all the tools and hardware platforms which are used later in the thesis for SNN simulations.