Brain-inspired computation using Spiking Neural Networks

Viktor Iliev
Supervisor: Haibin Zhao

Karlsruher Institut für Technologie, 76131 Karlsruhe, Germany
uqder@student.kit.edu

Abstract. The last two decades have experienced massive growth in the capabilities of neural networks. Artificial neural networks (ANNs) have evolved from simple first generation multi-layer perceptrons (MLPs), capable of computing basic functions to the many state of the art approaches used in the second generation deep neural networks (DNNs). Although outstanding in tasks like image recognition, speech processing, analytics and many more, deep neural networks are very recourse intensive in terms of energy consumption, training data and computational cost. This makes them unsuitable for use in autonomous systems like robots and self-driving vehicles, where energy availability is sparse and computational efficiency is demanded. In recent years spiking neural networks (SNNs), also known as third generation neural networks, have emerged as a promising solution to those problems. This work gives a brief overview of the biological structure of the brain, most studied and commonly used spike-based neuron and synapse models, as well as spike-based information coding techniques. At last, training algorithms for spike-based neuron models will be discussed and some state of the art neuromorphic hardware, software resources and applications will be briefly presented.

Keywords: spiking neural networks, neuromorphic computation, biologically plausible network, neuromorphic hardware

1 Introduction

The brain is the central and the most complex organ in almost every living organism, yet by far the least understood. Despite having much slower "clock rate" than a modern computer, it is still capable of processing massive amounts of information and solving complex computational problems. Comparing the characteristics of the human brain with those of a modern computer chip, certain downsides become clearly visible:

- An action potential is propagated trough the brain at speeds of about 1 m/s [10]. In some rare cases the propagation speeds can reach 10 to 100 m/s, which amounts to 36 to 360 km/h. For comparison electromagnetic waves are propagated trough copper wire at speeds of around 200 000 km/h.

- The refractory period of the neuron does not allow it to exceed "clock speeds" of approximately 1 kHz. This frequency is more then doubled in every modern computer.
- Nerve cells have switching speeds of several milliseconds [10]. For comparison, computer chip transistors have switching speeds in nanoseconds.

Regardless of these deficiencies, the brain is yet unmatched by any other device in certain tasks. For example the brain's ability for locomotive control or visual and auditory recognition is far greater than any computer can achieve. Further more, it is able to accomplish all these tasks consuming around 20 Watts of power, the average for the human brain.

These capabilities arise an obvious question: What makes the brain so much better in these domains, despite its obvious shortcomings in terms of raw computational power? Although there is no clear answer, one possibility can be the massive parallel structure. While a modern high-end computer chip possesses a couple of dozens of cores, the brain has billions of nerve cells capable of processing information independently form one another, providing a high degree of speed and flexibility.

2 History

The beginning of neuromorphic computing is often traced back to the published by W. McCulloch and W. Pittis in the 1943 edition of the "Bulletin of Mathematical Biology" article "A logical calculus of the ideas immanent in nervous activity" [20]. In it the authors show that, in principle, simple types of neural networks could compute any arithmetic or logical function. Later, in 1957 and 1958 F. Rosenblatt and C. Wighman developed the Mark 1 perceptron, the first true neuro-computer. Also known as the first generation of neural networks, perceptrons use a nonlinear threshold function to complete simple tasks like classification. Several other neural processing elements are developed in the next couple of years, but despite the interest, scientists had run out of good ideas and the field stalled until the early 1980s when researchers became bold enough to explore more deeply the development of neuro-computers and neural networks applications. In 1982 and 1984 John Hopfield, a world renowned physicist, publishes his two highly influential papers: "Neural networks and physical systems with emergent collective computational abilities" [12] and "Neurons with graded response have collective computational properties like those of two-state neurons" [13]. Those two papers together with his many lectures persuaded many scientists and mathematicians to join the field of neural networks. Some 20 years later B. DasGupta and G. Schnitger introduce a more complex, sigmoid activation function in their 1996 publication "Analog versus Discrete Neural Networks" [5]. These artificial neural networks are known as the second generation. However in recent years there is a growing interest in the so called third generation of neural networks, the spiking neural networks. Spiking neuron models offer a much more realistic representation of neural activity, taking into consideration the electrical properties of the cell.

3 Neurobiological basics

In order to be able to better understand neural networks and their design one must first become familiar with the basic structure of a nerve cell and the underlying mechanisms for information coding and transportation. The following brief introduction covers those topics as described in Alberts et al. (2022) [4].

3.1 Structure of a neuron

There are different types of nerve cells, depending on the purpose they serve in a living organism. However a basic structure is common for all of them. Figure 1 depicts a neuron with its most important components. The soma is the center component of the cell. It has two extensions going out of it. The short ones are called dendrites and are used for receiving signals. The long extensions are called axons and are used for transporting information to other cells trough the body. Axons are covered in a myelin sheath that improves conductivity and and offers protection. At the end of an axon there are fine branches that connect to other neurons' dendrites through synaptic clefts.

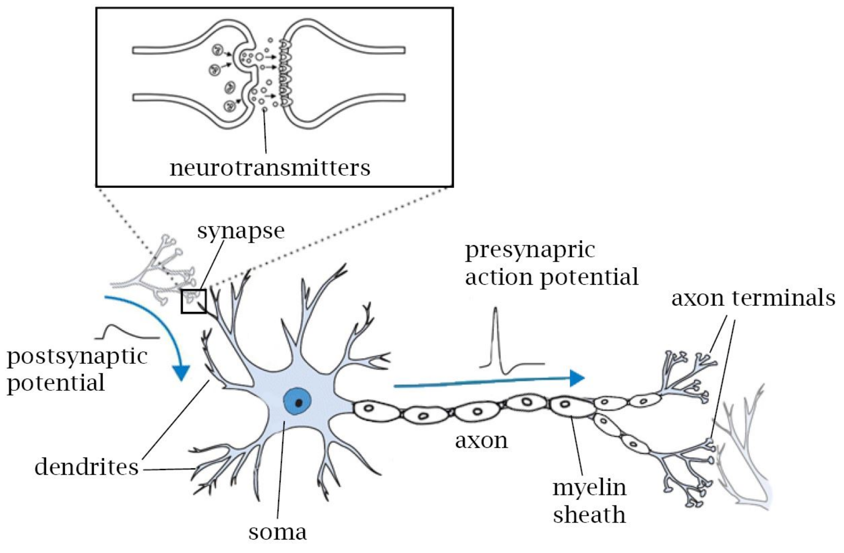


Fig. 1: A typical structure of a biological neuron and synapse. Source: [16]

3.2 Cell membrane

Just like any other cell, neurons are surrounded by a cell membrane. This membrane consists of a lipid bilayer and several types of proteins. The lipid bilayer is a powerful electrical insulator and does not let ions pass through it. However, transportation through proteins ensures that there is a different concentration of ions on both sides of the membrane, thus creating an electrical potential. Other proteins are responsible for creating ion channels which are also crucial for the function of the membrane.

3.3 Ion channels and ion pumps

As already mentioned, two types of ion transportation through the cell membrane take place. Ion channels are responsible for the passive transport of ions. These are electrically active proteins embedded in the cell membrane that permit electrically charged ions to flow through it. The other mechanism involves ion pumps that actively transport ions from one side to the other. Most ion channels are permeable only for certain types of ions. Some channels are voltage gated, which means they can be opened or closed by altering the membrane potential. Others are chemically gated, meaning their states can be switched by chemicals that diffuse through the extracellular fluid. The voltage induced by this ion transportation provides a basis for electrical signal transmission between different parts of the membrane. As discussed above, neurons exchange information through synapses. This is called neurotransmission as it involves the release and reabsorption of a neurotransmitter. The fundamental process that triggers the release of neurotransmitters is the action potential, a propagating electrical signal that is generated by exploiting the electrical excitability of the cell membrane.

3.4 Action potential

Information transportation along the axons of a nerve cell happens through action potentials. When enough ions have passed the cell membrane and it has been depolarized, voltage-gated Na+ channels are opened. The flow of Na+ ions depolarizes the membrane further. This way the membrane potential can quickly change from around 70 mV to about +50 mV [4]. Directly thereafter, the Na+ channels are closed and K+ channels are opened. The outflow of K+ ions brings the membrane potential to its resting level. This is called repolarization. However, in that time period, adjacent Na+ channels have been opened because of depolarization. After an action potential has occurred, there is a transient negative shift, called the refractory period, due to additional K+ currents. This prevents an action potential from travelling back the way it came. This way a wavelike pattern is generated which transports the signal further along the axon. Figure 2 shows a plot of the membrane voltage before, during and after an action potential.

4 Neuron models

Developing a mathematical models to describe the neurons' activity has been a subject of scientific research for decades. As mentioned in the beginning, first and

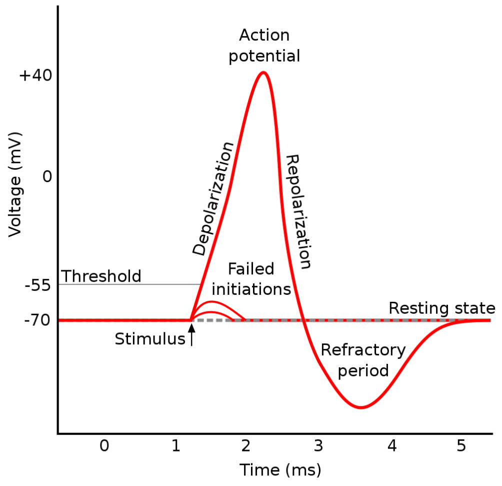


Fig. 2: Voltage plot of am action potential. Source: Wikipidia

second generation neural networks use a value non-linearity activation function. However spiking neural networks use a different kind of activation function: spiking non-linearity function (see Fig. 3). Furthermore inputs and outputs are produced in the form of spikes, essentially 1 or 0 events. This ensures that a neuron is only active whenever it processes a spike, greatly reducing the computational cost and power consumption of the network.

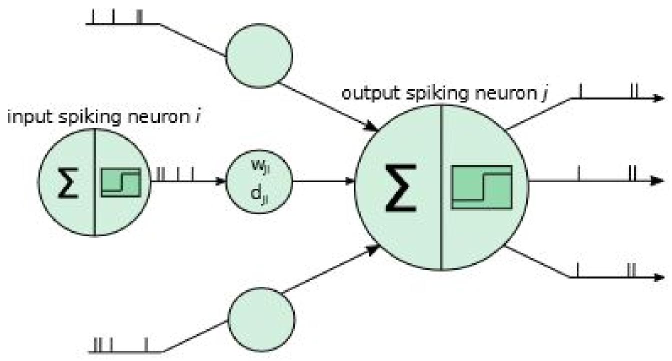


Fig. 3: Scheme for SNNs. Source: [15]

4.1 Hodgkin-Huxley model

The Hodgkin-Huxley model is one of the earliest and most popular neuron models. It represents the neural membrane as an equivalent electrical circuit as shown in Fig 4a. There are two variable conductances for the potassium and the sodium currents, as well as a constant conductance corresponding to the leakage current and a constant membrane capacity. The model was developed according to real data and is therefore highly realistic and able to represent the electrical properties of a real neuron. However the properties of the cell are described by four differential equations as shown below [3]:

$$I = C_M * \frac{dV}{dt} + g_K * n^4 * (V - V_K) + g_{Na} * m^3 * h * (V - V_{Na}) + g_l * (V - V_l) \quad (1)$$

where:

$$\frac{dx}{dt} = \alpha_x * (1 - x) - \beta_x * x \quad \text{for} \quad x \in n, m, h \quad (2)$$

and

$$\alpha_n = \frac{0,01 * (V + 10)}{exp(\frac{V+10}{10}) - 1} \quad (3)$$

$$\beta_n = 0.125 * exp(\frac{V}{80}) \quad (4)$$

$$\alpha_m = \frac{0,1 * (V + 25)}{exp(\frac{V+25}{10}) - 1} \quad (5)$$

$$\beta_m = 4 * exp(\frac{V}{18}) \quad (6)$$

$$\alpha_h = 0.07 * exp(\frac{V}{20}) \quad (7)$$

$$\beta_h = \frac{1}{exp(\frac{V+30}{10}) + 1} \quad (8)$$

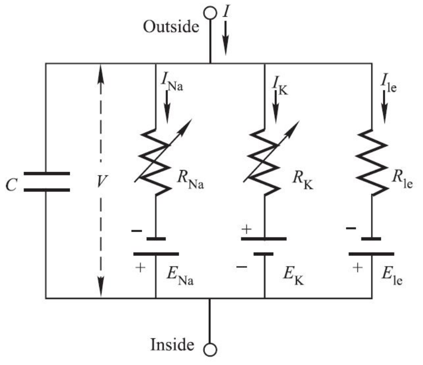
This makes it very computational heavy and unsuitable for large scale neural network simulations. Since it is highly accurate, the Hodgkin-Huxley model is used for single neuron simulations or as benchmark for other, more simple models.

4.2 Leaky Integrate-and-Fire model

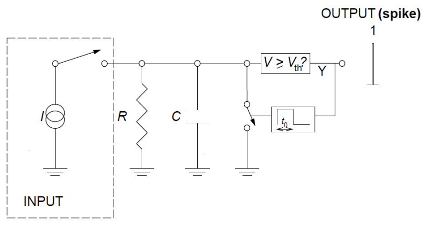
In contrast to the Hodgkin-Huxley model the leaky integrate-and-fire model (LIF) can be described with a single differential equation:

$$C * \frac{dV}{dt} = -g_L * (V - E_L) + I \tag{9}$$

C represents membrane capacity, V membrane voltage, $g_[L]$ is the electrical conductance for the leakage current, E_L is the resting potential and I is an applied electrical current. Since there are no separate models for different ionic currents, the leakage potential is the same as the resting one. Burkitt (2006) [6] offers a more detailed explanation of the model. Once a neuron has spiked, it returns to a so-called "reset potential" which is usually smaller than its resting potential. Following that, the neuron is inactivated for a period of time called the "refractory period". After this amount of time has passed, the neuron assumes its normal behaviour until another spike is released. While the leaky integrate-and-fire model cannot be directly derived from the Hodgkin-Huxley equations, it still offers a good balance between conceptual simplicity and biological accuracy and has played an important role in understanding the dynamics of whole networks. An equivalent electrical circuit of the neuron can be seen in Fig 4b



(a) Equivalent electrical circuit of the Hodgkin-Huxley model. Source: [21]



(b) Equivalent electrical circuit of the LIF model. Source: [8]

Fig. 4: Electrical circuits of Hodgkin-Huxley and LIF models

4.3 Adaptive Exponential Integrate-and-Fire model

The adaptive exponential integrate-and-fire model (AdEx) makes two main modifications to the already discussed LIF model. Firstly it introduces a more realistic smooth spike initiation zone in contrast to the strictly defined threshold potential. This means that a spike is not immediately initiated when the threshold voltage is reached. The second extension is the introduction of an adaptation variable w. Thus the equation takes the following form [17]:

$$C_m * \frac{dV_m}{dt} = -g_L * (v_m - v_r) + g_L * \Delta_T * exp(\frac{V - V_T}{\Delta_T}) - w + I \tag{10}$$

Δ_T is a slope factor for the exponential function. The adaptation current w is defined by:

$$\tau_w * \frac{dw}{dt} = a * (V - E_L) - w \tag{11}$$

where τ_w is the time constant and a represents the level of subthreshold adaptation. At each firing time, the variable w is increased by an amount b, which accounts for spike-triggered adaptation. The current w is subtracted in the potential equation, causing slower rise in the potential when adaptation occurs.

4.4 Izhikevich model

Another model has been proposed by E. Izhikevich in 2003 combining the biological plausibility of the Hodgkin-Huxley model with the computational efficiency of the LIF neuron. The model is described by a system of two differential equation [14]:

$$C_m * \frac{dV_m}{dt} = k * (v_m - v_r) * (v_m - v_t) - u - I(t) \tag{12}$$

$$\frac{du(t)}{dt} = a * (*b * (v_m - v_r) - u) \tag{13}$$

with the auxiliary after-spike resetting

$$\text{if} \quad v_m \geq v_\theta \quad \text{than} \quad v_m \leftarrow c, \quad u \leftarrow u + d \tag{14}$$

where v_m represents the membrane potential, u is the recovery potential, C_m is the membrane capacitance, v_r is the resting membrane potential and v_t is the instantaneous threshold potential. The parameters k and b can be determent when one knows both the rheobase and the input resistance of a neuron. The recovery time constant is a, the spike cutoff value is v_θ and the voltage reset value is c. The parameter d describes the total amount of outward minus inward currents activated during the spike and affecting the after-spike behavior.

5 Synapse models

Until now the neuron models have been described as independent cells. In reality, multiple cells are interconnected trough synapses. The properties of these synaptic connections can heavily influence the behavior of the neural network. Moreover attributes like synaptic plasticity are involved in the learning mechanisms. The following is a brief overview of the two most common synaptic models: The current-based synapse and the conductance-based synapse as described by Burkitt(2006) [6].

5.1 Current-based synapses

The current-based model is the simpler option of representing synapses. The synaptic input is described by the equation:

$$I_S(t) = C_M * \sum_{k=1}^{N} w_k * S_K(t) \tag{15}$$

C_M represents the synaptic membrane capacity, N the number of input synapses and w_k is the change a single incoming spike would cause. The incoming spikes at synapse k are modeled by S_k:

$$S_K(t) = \sum_{t_k} \delta(t - t_k) \tag{16}$$

where δ is the Dirac function. One important property of the current-based synapses is that the input current I_S does not depend on the membrane potential value. In other words, the membrane potential changes with a constant value when receiving incoming spikes.

5.2 Conductance-based synapses

The conductance-based synapse model gives a more biologically accurate solution for the synaptic current. In this case, the current generated by the incoming spikes depends on the membrane potential at the time of the spike. The synaptic current is described by the following equation:

$$I_S = (E_{ex} - V(t)) * \sum_{k=1}^{N_E} g_{E,k} * S_{E,k}(t) + (E_{in} - V(t)) * \sum_{k=1}^{N_I} g_{I,k} * S_{I,k}(t) \tag{17}$$

N_E and N_I are the number of excitatory and inhibitory input synapses, $g_{E,k}$ and $g_{I,k}$ the synaptic conductances of the excitatory and inhibitory synapses, also known as weights. Similarly to the current-base synapses, conductance based synapses also use the variables $S_{E,k}$ and $S_{I,k}$ to model incoming spikes. E_{ex} and E_{in} are the (constant) reversal potentials ($E_{in} < V_{reset} < V_{th} < E_{ex}$), named that way because the direction of the current flow changes when the membrane potential passes trough the corresponding reversal potential. These potentials introduce a non-linearity in the summation of the incoming synaptic inputs. In other words, the closer the membrane potential is to the reversal potential, the less it is changed by subsequent incoming spikes, corresponding to the behavior observed in living organisms.

6 Neural information coding

Another important aspect of the brain functionality is the way information is encoded when being transported. Although there is no definitive answer to this question, several theories about information coding in the brain exist. The following is an overview of the most popular suggestions. More detailed reviews can be found in Gerstner and Kistles [19]

6.1 Rate coding

The simplest and most straightforward explanation of how information coding in the brain takes place is that data is not contained in particular spikes but in the spike rate over a window of time. Depending on the different averaging schemes, there are several ways to define the firing rate, such as an average over time, an average over a population of neurons or an average over several repetitions. Since the last option does not account for real-time information processing, the following section will concentrate on the first two suggestions.

6.1.1 Rate as spike count over time The most commonly used definition of a spiking rate refers to a temporal average. This means counting the amount of spikes s a single neuron fires in a time period T and getting the rate as a quotient of the two terms ($r = \frac{s}{T}$). The rate is measured in Hertz (Hz) and the choice of time interval is up to the experimenter, but usually depends on the type of neuron and the stimulus applied. Common time intervals range from 100ms to 500ms. This type of information encoding however is not biologically plausible. The fast reaction time the brain can achieve would not be reached if it first had to wait for the time interval to pass before issuing a command.

6.1.2 Rate as population activity The number of neurons in the brain is enormous and often population of neurons have similar properties and respond to the same stimuli. In an idealized situation, one can assume that these populations consist of neurons with exactly identical properties and the same pattern of input and output connections. If all neurons of a population m are connected to all neurons of a population n then it would not be necessary to wait for a long time interval before measuring the firing rate. One can instead just calculate the proportion of firing neurons in the presynaptic population m in a short time interval Δ_T. This way, much less time would be necessary to convert an input into an output. An obvious flaw of this method is that it assumes the total homogeneity for neuron parameters and connection patterns, which is not biologically plausible.

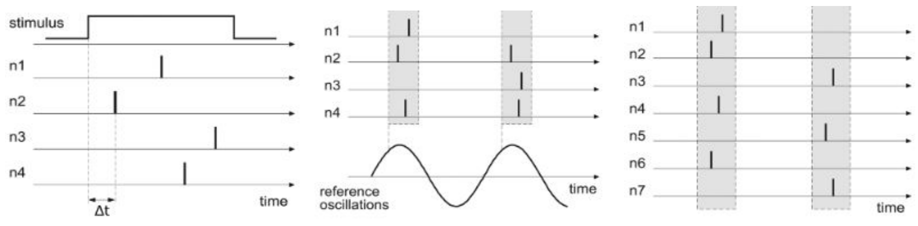
6.2 Spike timing based coding

In this section three of the most popular coding strategies based on spike timing will be discussed .

6.2.1 Time to first spike Time to first spike model encodes neural information as the latency Δt between the beginning of the stimulus t_0 and the time of the first spike t_f in the neural response (Figure 5a). A neuron firing quickly after receiving an input might signify a strong stimulation, whereas longer time periods before the first spike might be associated with a weaker signal. In an idealized version of this scenario, each neuron would only need to fire a single spike, since the timing of the first spike contains all the information, thus making all following spikes irrelevant.

6.2.2 Phase When the input signal is not a single event but a periodic function in time, emitted spikes can be referred to the reference time point in a periodic signal. In this way neuronal spike trains can encode information in the phase of a pulse with respect to the background oscillations (Figure 5b). In case the input does not change over the period of the signal, the same pattern repeats periodically.

6.2.3 Correlation and synchrony This model is based on the assumption that neurons that encode bits of information related to the same object will fire synchronously (Figure 5c). This behavior has been observed in several experiments. For example neurons in the visual cortex tend to synchronize their ”firing” with precision in milliseconds when activated by a single contour, while failing to do so when activated by different contours moving in different directions [9].



(a) Time to first spike (b) Phase coding (c) Coding by synchrony

Fig. 5: Spike based information coding strategies. Source: [9]

7 Training algorithms for SNNs

The training of spiking neural networks poses unique challenges, as the spike-based information representation and propagation mechanisms are different from those in ANNs. Some of the most common training algorithms for SNNs can be classified into categories similar to learning in traditional neural networks: supervised learning, unsupervised learning, reinforcement learning.

7.1 Unsupervised learning

Most unsupervised learning algorithms for SNNs are based on some form of spike-timing-dependent plasticity (STDP). STDP is a Hebbian learning rule which states, that synapse's strength between two neurons depends on the relative timings of the pre- and post-synaptic spikes. If a pre-synaptic (input) neuron fires briefly before a post-synaptic (output) neuron, than the connection between them is strengthened and vice versa, if the pre-synaptic neuron fires briefly after the post-synaptic neuron, then the connection between them is weakened[7]. The STDP learning rule can be expressed with the following equation:

$$\mathit{\Delta}_w = \begin{cases} A * exp(\frac{-(|t_{pre}-t_{post}|)}{\tau}) & \text{if} \quad t_{pre}-t_{post} \leq 0 \quad \text{and} \quad A > 0 \\ B * exp(\frac{-(|t_{pre}-t_{post}|)}{\tau}) & \text{if} \quad t_{pre}-t_{post} \geq 0 \quad \text{and} \quad B < 0 \end{cases} \tag{18}$$

where δ_w is the weight change, A and B are constant parameters, τ is the timing window constant (usually 10 ms) and t_{pre} and t_{post} are the absolute timing of the pre- and post-synaptic spikes.

7.2 Supervised learning

Similar to ANNs many backpropagation and gradient descent based algorithms for SNNs have been developed. The idea behind the process is to minimize the error between desired output and actual output spike train using labeled data. Despite supervised learning methods for SNNs to have the ability to reach accuracy similar to ANNs, one major factor limits their development. Since spikes are discrete in time, their activation function can not be derived[7]. Thus major task by the development of supervised learning methods consists in finding an differentiable, real-valued approximation function to the activation function of the spiking neurons. Some of the most commonly used algorithms include SpikeProp, SLAYER, Tempotron, Remote supervised method (ReSuMe), Chronotron, Spike pattern association neuron (SPAN), Precise-spike-driven (PSD) synaptic plasticity and others.

7.3 Reinforcement learning

Reinforcement learning (RL) refers to the process of adapting the parameters of a SNN based on external reward signal that depends on the predictions generated by the network. Desired actions are reinforced with positive rewards, whereas undesired actions are punished by negative reward signals[9]. Despite not receiving much attention in the field of SNNs, there are some well-known RL learning methods, one of which is the reward-modulated STDP[18].

8 Hardware

First attempts of neuromorphic hardware designs were mostly done by researchers for research purposes, but after noticing the huge potential of the brain-inspired models, lately corporations have started developing their own chips, some in partnership with research institutes. This section presents some of the most significant projects involving spiking neural models.

8.1 Neurogrid and Braindrop

Developed at Stanford University, Neurogrid and Braindrop are two separate neuromorphic hardware designs. Both designs are a mixed-analog-digital systems, but differ in their applications and hardware components. Neurogrid is able to simulate real-time biological brain activity with millions of neurons and billions of synaptic connections[2]. The Neurogrid system is build out of 16 Neurocore chips, placed on a board and arranged into a tree like structure. Every Neurocore chip is made up of analog neurons placed inside a 256x256 array fabricated in a 180-nm CMOS.

On the other hand, Braindrop is designed to be programmed at a high level of abstraction. It uses Neural Engineering Framework(NEF) that acts as a neural compiler: given the specific properties of the neurons, the values to be represented, and the functions to be computed, it solves for the connection weights between components that will perform the desired functions. Braindrop is fabricated in 28-nm FDSOI process and integrates 4,096 neurons onto a single Braindrop chip [2].

8.2 BrainScale and SpiNNaker

These two neuromorphic systems come from the Human Brain Project(HBP). Developed by a collaboration between many research groups BrainScaleS is a mixed-analog-digital waferscale neuromorphic hardware system. It utilizes an entire 20 cm wafer for a very-large-scale neuromorphic system with 40 million synapses and up to 180 thousand neurons. A structure called the Analog Network Core (ANC) is used to implement the neuron circuits and their synapses. The ANC itself is fabricated using 180-nm CMOS to create a High Input Count Analog Neural Network (HICANN). A single wafer is able to fit up to 352 of these HICANN chips[2].

SpiNNaker is the second meuromorphic system part of the Human Brain Project. In contrast to BrainScale, it does not use custom neuron circuitry and instead simulates the brain in real-time by connecting together over one million ARM processors. A SpiNNaker machine is a 2-D array of chip multiprocessors (CMPs) where each CMP contains 18 ARM968 processor cores. Sixteen processors are used for simulation, one processor for administration, and one processor is a backup in case of a processor failure. A single SpiNNaker board contains 48 of these CMPs[2].

8.3 TrueNorth

TrueNorth is a neuromorphic platform developed by IBM. One TrueNorth chip is made up of 4096 neurosynaptic cores, fabricated in IBM's 45-nm SOI process. Each core implements 256 digital integrate-and-firere neurons with 1024 axonal circuits for input connectivity organized as an Static Random Access Memory (SRAM) crossbar. Thus one TrueNorth chip contains 5.4 billion transistors and implements 1 million digital neurons and 256 million synapses[2].

8.4 Loihi

Loihi is Intel's implementation of a digital neuromorphic chip. A single Loihi chip is composed of 128 neuromorphic cores, 3 Lakemont cores which help with advanced learning rules and with managing the other cores and is fabricated in Intel's 14-nm process. The 128 neuromorphic cores implement 30,000 artifcial CUBA leaky-integrate-and-fire neurons and 130 million synapses[2].

9 Modeling and simulation

Many open-source software tools for modeling and simulation of SNNs exist. Two of the most well known such simulators are NEST[11] and PyNN[1].

9.1 NEST

NEST is a computer program for simulating large heterogeneous networks of point neurons or neurons with a small number of compartments. It is best suited for simulations focused on dynamics, size and structure of neural systems rather than exact biophysical properties of single neurons. When building a NEST model, the developer creates one or more model neurons and connects them via synapses to form a network. Several types of neuron models are available. Some of the more commonly used ones include:

- The Hodgkin-Huxley model, a very biologically correct neuron model
- The Leaky Integrate-and-fire model, a simplified neuron model for use in large-scale networks
- The Adaptive Exponential Integrate-and-fire model, an enhanced version of the LIF model

9.2 PyNN

Computational neuroscience has produced a large diversity of software for modelling and simulating large networks of spiking neurons. While this has its benefits, such as cross-reference different implementations and choosing the most appropriate tool for a specific task, a major drawback is that each simulator has its own language, which makes porting models between simulators a difficult task. The simulation framework PyNN is an interface that solves this problem. It enables the user to write scripts in Python and then use them on the various supported simulators. This level of abstraction makes code sharing and reuse much easier while still retaining the possibility of testing models on different simulators. PyNN supports NEST as well as several other simulators and even neuromorphic hardware such as the SpiNNaker board.

10 Applications

Computer vision and robotics are two of the fields in which SNNs have great potential. Although still not widely used, SNN's utilization in computer vision tasks gradually increases. Main reasons for the limited scope of applications are the complex dynamics of the spiking neurons. Despite some studies indicating SNNs can be used for image classification on large data bases as ImageNet, most of these tasks are performed on smaller data sets such as MNIST and N-MNIST[2]. Neuromorphic video cameras such as Dynamic vision sensors (DNS) work in an event triggered manner, inspired by the biological retina and detect changes in the intensity of the pixels. A spike is generated only if the intensity of a pixel exceeds a threshold. This event triggered character allows DVS cameras to achieve very high temporal resolution with very low bit-rate compared to traditional cameras[21] making them very useful in tasks like object detection or tracking. Furthermore the energy efficiency of such sensors makes them unmet in energy limited environments such as autonomous robots.

References

1. A. P. Davison, D. Brüderle, J.E.J.K.E.M.D.P.L.P.P.Y.: Pynn: a common interface for neuronal network simulators. Frontiers in Neuroinformatics (Jan 2009)
2. A. R. Young, M. E. Dean, J.S.P.G.S.R.: A review of spiking neuromorphic hardware communication systems. IEEE Access pp. 135606–135620 (Sep 2019)
3. A.L. Hodgkin, A.H.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of phisiology p. 500–544 (Aug 1952)
4. B. Alberts, R. Heald, A.J.D.M.R.K.R.P.W.: Molecular biology of the cell. W.W. Norton Company (2022)
5. B. DasGupta, G.S.: Analog versus discrete neural networks. Neural Computation pp. 805–818 (May 1996)
6. Burkitt, A.N.: A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. Biological Cybernetics pp. 1–19 (Apr 2006)

7. D. Nguyen, X. Tran, F.I.: A review of algorithms and hardware implementations for spiking neural networks. Journal of Low Power Electronics and Applications (May 2021)
8. D. Tal, E.L.S.: Computing with the leaky integrate and fire neuron: logarithmic computation and multiplication. Neural Computation pp. 305–318 (Feb 1997)
9. F. Ponulak, A.K.: Introduction to spiking neural networks: Information processing, learning and applications. Acta Neurobiologiae Experimentalis pp. 409–433 (Jan 2011)
10. Gegenfurtner, K.R.: Gehirn und Wahrnehmung: Eine Einführung. Fischer Taschenbuch (2011)
11. Gewaltig, M.O., Diesmann, M.: Nest (neural simulation tool). Scholarpedia
12. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA pp. 2554–2558 (Apr 1982)
13. Hopfield, J.J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. USA pp. 3088–3092 (May 1984)
14. Izhikevich, E.: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. MIT Press (2007)
15. J. Loboa, J.Del Ser, A.B.N.K.: Spiking neural networks and online learning: An overview and perspectives. Neural Networks (Jan 2020)
16. K.Yamazaki, V. Vo-Ho, D.B.N.L.: Spiking neural networks and their applications: A review. Brain Science (Jun 2022)
17. R. Brette, W.G.: Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. Journal of neurophysiology pp. 3637–3642 (Nov 2005)
18. S. Dora, N.K.: Spiking neural networks for computational intelligence: An overview. Big data and cognitive computing (Nov 2021)
19. W. Gerstner, W.K.: Spiking neuron models: Single Neurons, Populations, Plasticity. Cambridge university press (2002)
20. W. McCulloch, W.P.: A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biology pp. 115–133 (1943)
21. Z. Ming, G. Zonghua, P.G.: A survey of neuromorphic computing based on spiking neural networks. Chinese Journal of Electronics pp. 667–888 (Jul 2018)