Chapter 4

Neuromorphic Computing

4.1 Introduction

The desire to build fast and efficient computers has fueled years of research in computing and the Von Neumann architecture, describe in figure 4.1, has become a staple to meet this end. In this architecture, there is a clear line drawn between the processing unit and the memory unit. For a process to take place, information is retrieved from or stored in the memory unit. A major bottleneck to processing speed here is the channel between the memory and processing unit. The efficiency and processing speeds of computers are tuned by dimensional scaling of the transistors that build them up. However, with the transistors approaching their physical limits (Moore's law [57]), scientists are recognising a need to shift to an alternate paradigm. This is especially necessary for deep learning tasks that make use of immense neural networks for the purpose of classification and recognition, where power consumption is very high on conventional hardware [58]. Neuromorphic computing is an alternate approach to computing, where the architecture tries to mimic the structure and function of the human brain. In late 1980s, the term *neuromorphic* was coined to mean organisation of silicon based analog circuits that behave like biological networks of neurons. In recent times, the term has come to be associated with the implementation of neural networks on non-Von Neumann type architecture.

Similar to the vast inter-connections of neuron in the human brain, neuromorphic processors are comprised of densely inter-connected simple analog components. Their organisation grants them inherent parallelism which is a desirable feature for superior computing speeds. Unlike traditional computers, the memory and processing units are co-located which allows these processors to

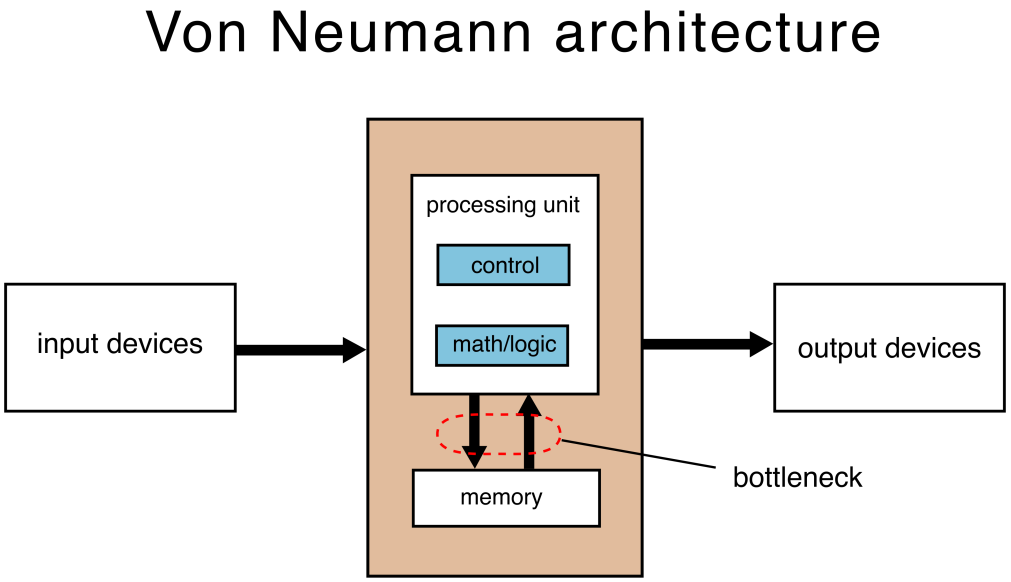


FIGURE 4.1: Von Neumann architecture

overcome the Von Neumann bottleneck. The primary motivation behind the new paradigm is lower power consumption, this influences the choice of analog components used in a processor as well as algorithmic design that is implemented. Owing to its similarity with the brain, this architecture is naturally more suited for cognition tasks than conventional architecture.

The following sections introduce the biological units that make up the human brain and their analog counterparts. Spiking neural networks are also briefly discussed. The discussions in this chapter follow from [59]

4.1.1 Neurons and synapses

Neurons are the fundamental processing unit of the central nervous systems. They are biological cells that can form a dense and intricate networks of several kilometers packed into a cubic centimeter. Though there are a variety of neurons present in the brain, for the purpose of neuromorphic computation we are interested only in ideal spiking neurons, since these are the only kind that take part in spike-based information exchange and processing [59]. The input and outputs to and from a spiking neuron are in the form of electrical pulses called action potentials or spikes. The profile of the action potential is irrelevant to information transfer; information is carried in the number and timing of these spikes. A structure of the spiking

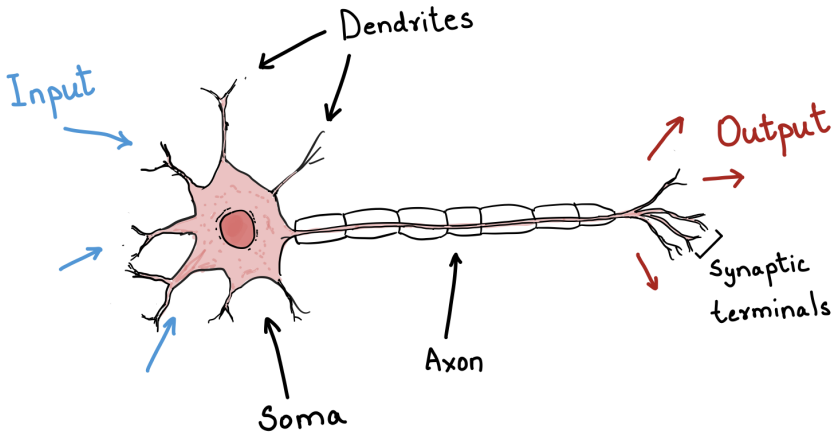


FIGURE 4.2: Structure of a biological Neuron

neuron, shown in figure 4.2, can be divided into three parts: the dendrites, the soma and the axon. The dendrites function as input channels bringing in signals from other neurons. The soma behaves as the processing unit of the cell, *integrating* or summing up the signal it receives. The axon acts as output channels to other neurons. The generation of a pulse in a neuron depends on the voltage inside and outside the cell membrane of the soma and the axon. These voltages are governed by Na^+ and K^+ channels in the cell that open or close depending on the concentration in the cell. When the membrane potential, the difference between the voltage outside and inside the membrane, crosses a given threshold, a spike is generated. A junction between the dendrites and an axon of two neurons is referred to as a synapse. In literature, an input neuron is typically referred to as a *pre-synaptic* neuron and output neuron as the *post-synaptic* neuron. This terminology shall be used here onward. The most common type of junction between neurons is a chemical synapse, where the axon of the pre-synaptic neuron and the dendrites of the post-synaptic neurons are very closely spaced, leaving a tiny gap between them. This gap is called a synaptic cleft, shown in figure 4.3. At the conception of a pre-synaptic spike, a complex chain of bio-chemical reactions take place at the synapse that results in the release of certain chemicals called neurotransmitters from the the pre-synaptic neuron into the synaptic cleft. On the post-synaptic side, the transmitters are received by special receptors that direct the opening of the aforementioned ion channels and eventually lead to spiking of the neuron. The ability of a post-neuron to fire is greatly dependent on the properties of the synapses that links it to pre-neurons. The synaptic weight determines how much of the signal is allowed to pass through to the post-synaptic neuron.

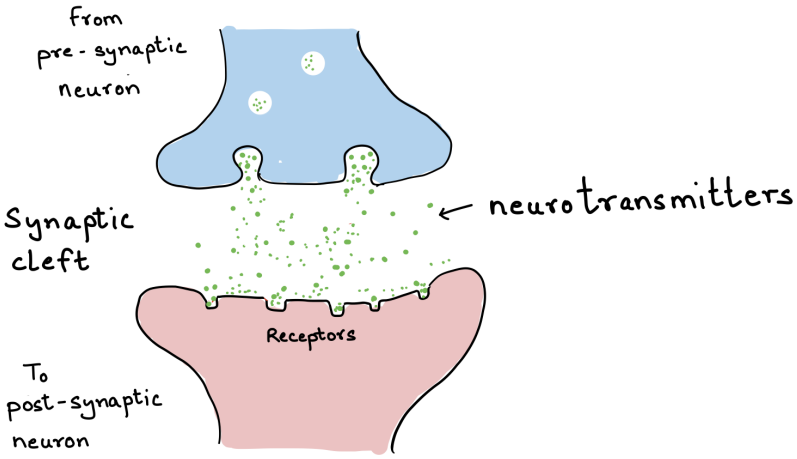


FIGURE 4.3: Synapse and synaptic cleft

Additionally, synapses can either be excitatory or inhibitory in nature. The function of an excitatory synapse is to increase the membrane potential of the post-synaptic neuron while that of an inhibitory synapse is to decrease it. Biologically, there are many factors that determine whether a synapse behaves as an excitatory or inhibitory synapse. These include: the type of neurotransmitters that are channeled, the type of receptors that receive these chemicals as well as the concentration of these chemicals. It is possible for a synapse to switch its function depending on these factors. It has been observed that excitatory synapses are always greater in number than inhibitory synapses [60].

Mathematically, in a biological network, the activity of a post-synaptic neuron, S_j is given as

$$S_j(t+1) \propto h\Big(\sum_i w_{ij} S_i(t)\Big) \,, \tag{4.1.1}$$

where w_{ij} signifies the weight of the synapse and h is an arbitrary function. The weight of a synapse indicates its behaviour:

$$w_{ij} = \begin{cases} +ve & \text{excitatory} \\ -ve & \text{inhibitory} \\ 0 & \text{inactive or absent} \end{cases} . \tag{4.1.2}$$

In practice, neurons can be modelled by a set of nonlinear differential equations that describe how the action potentials are propagated. Hodgkin and Huxley [61] proposed a neuron model

that accurately predicted the shape of action potentials generated by a neuron. This gave exceptional insight into the electrical events underlying the action potential. The equations they proposed describe the dynamics of the various ionic channels and leakage channels that contribute to the membrane potential. Inspired by their formulation, simpler models have since been proposed that allow for faster and more efficient simulation of neuronal dynamics. These include models such as the Izhikevich model [62], the leaky integrate-and-fire (LIF) model [63] and the FitzHugh-Nagumo model [64]. In this work, the leaky integrate-and-fire model has been used for its simplicity in computer simulation and hardware modelling. It is modelled as a resistance, battery and capacitor in parallel with a pathway for an input current as shown in figure 4.4. When the cell membrane receives a current, the charge it receives with it is stored within the membrane and the membrane potential rises. A capacitor is employed to replicate this behaviour. To account for charge leakage in the cell membrane over time, a resistance is defined. When the input signal ceases, the cell membrane drops down to a resting potential characterised by the battery in the circuit. Formally, the injected current $I(t)$ at a time t can be written as:

$$I(t) = \frac{u(t) - u_r}{R} + C\frac{du}{dt}\,, \tag{4.1.3}$$

where $u(t)$ is the time dependent potential across the circuit, R and C are the resistance and capacitance in the circuit as shown in fig 4.4 . The term, $u(t)$ corresponds to the membrane potential of the circuit. u_r is the voltage across the battery in the circuit. The equation can be rewritten as:

$$RC\frac{du}{dt} = -(u(t) - u_r) + RI(t) \tag{4.1.4}$$

$$\tau_m\frac{du}{dt} = -(u(t) - u_r) + RI(t)\,, \tag{4.1.5}$$

where $\tau_m = RC$ is called the time constant of the decay. u_r behaves like resting potential, i.e. the membrane potential assumes the value of $u(r)$ when there is no stimulus. The neuron *spikes* when the membrane potential crosses a specified threshold voltage. Immediately after a spike, the neuron briefly enters a *refractory period* where it does not respond to any external inputs. The spiking of such a neuron is displayed in fig 4.5. The threshold voltage and refractory period are not explicitly present in the equations; they are specified separately in hardware, where these parameters can be introduced by the addition of capacitors and transistors. In literature, models for LIF neurons proposed that are based on floating-gate integrator circuits [65].

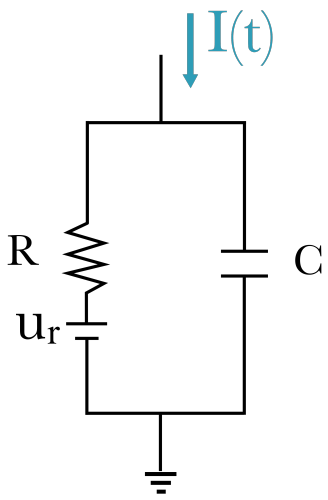


FIGURE 4.4: Leaky integrate-and-fire circuit model

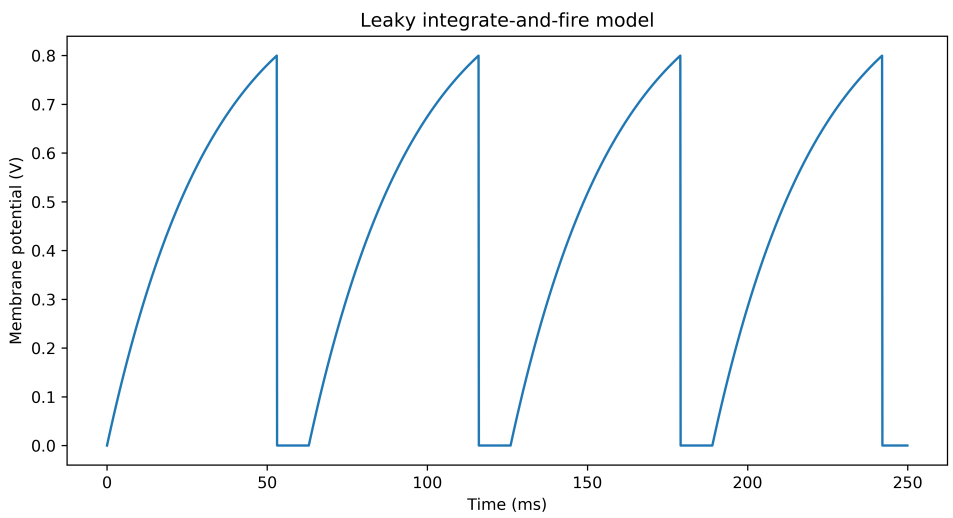


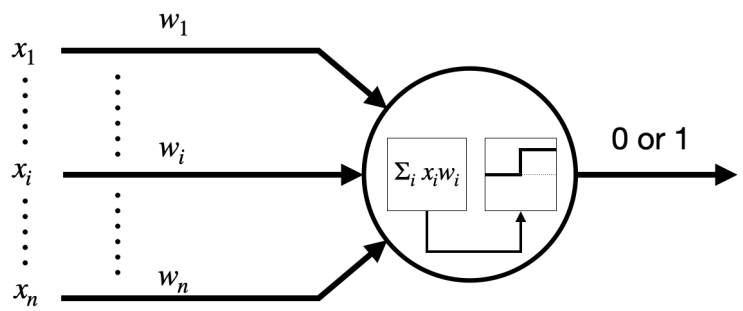
FIGURE 4.5 Leaky integrate-and-fire firing. The neuron has a threshold voltage $v_{th} = 0.8$ V, time constant $\tau_m = 33$ ms and a refractory period $t_r = 5$ ms. The neuron was simulated using *Brian2* [66].

4.2 Neural Networks

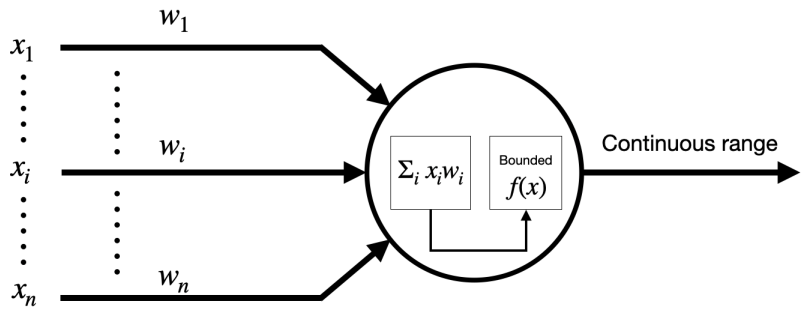
Based on the neuronal functions, neural networks are categorised into three generations [67], shown in figure 4.6. The earliest generation refers to as McCulloch Pitts neurons that function based on a thresholding operation. A neuron, called a linear threshold unit (LTU), gives a binary '1' if the weighted sum of all the inputs is above a certain threshold and '0' if it is not. They are also called perceptrons. In the second generation of neural networks, the processing of inputs at a neuron is based on an *activation function*. In contrast to the first generation, these allow for a continuous set of possible output values. An example of an activation function is the sigmoid function. The most important feature of the second generation is the compatibility of learning algorithms that rely on gradient descent and back-propagation. Modern deep learning networks are based on the second generation. With great advancement in field of neuroscience, a third generation of spiking neural networks (SNNs) have emerged that show great promise on the efficiency front. These networks use spiking neurons similar to the ones mentioned in the previous section. Information is exchanged through spikes which can essentially be construed as binary signals. The distinction between the third generation and the previous generation of networks lies in how information is encoded. The former utilise the amplitude of the signals while the latter focuses more on the timing of the signals. A neuron in an SNN is active only when it is receiving or emitting signals. The neuron does not consume any power while it is "silent". This make them more energy efficient in comparison to the previous generations.

Spiking neural networks are of three kinds, topologically 4.7: (i) feed-forward; (ii) recurrent or (iii) hybrid (a combination of feed forward and recurrent). In feed-forward networks, information flows downstream or in just one direction. The networks are comprised of layers of neurons stacked one after the other with an input layer at one end and an output layer at the other end. Such networks are suitable for low-level sensory systems such as vision. Therefore, they can be used for tasks involving pattern recognition and classification. Recurrent neural networks consist of neurons or populations of neurons that interact through feedback connections. Such networks demonstrate dynamic temporal behaviour. In the human brain, it is believed that such networks play a role in the formation of associative memory and complex decision making. These networks are much harder to train than feed-forward networks. Hybrid networks are a composition of networks that are feed forward and those that are recurrent. Reservoir computing [68] is a widely studied concept that make use of hybrid networks.

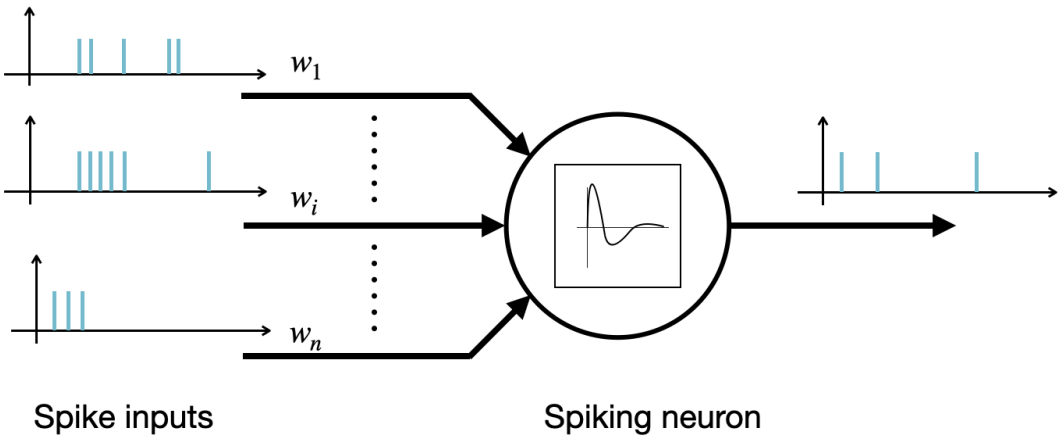
The following section briefly discusses some fundamental learning algorithms for SNNs.



(A) First generation neuron or a perceptron. neuron

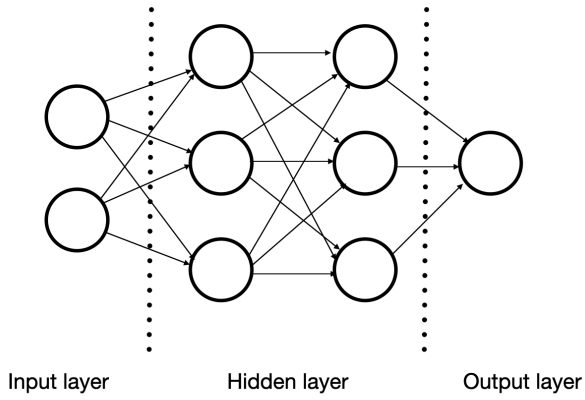


(B) Second generation neuron with activation function, $f(x)$

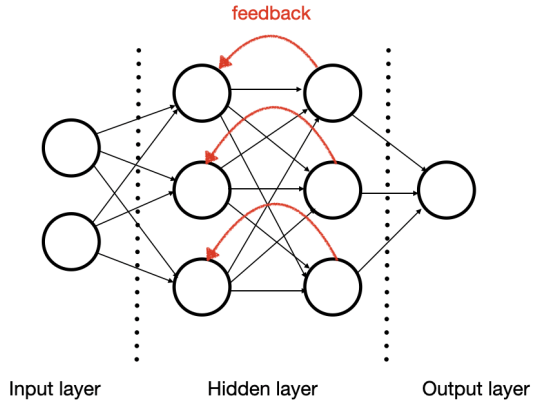


(C) Third generation neuron or a spiking neuron.

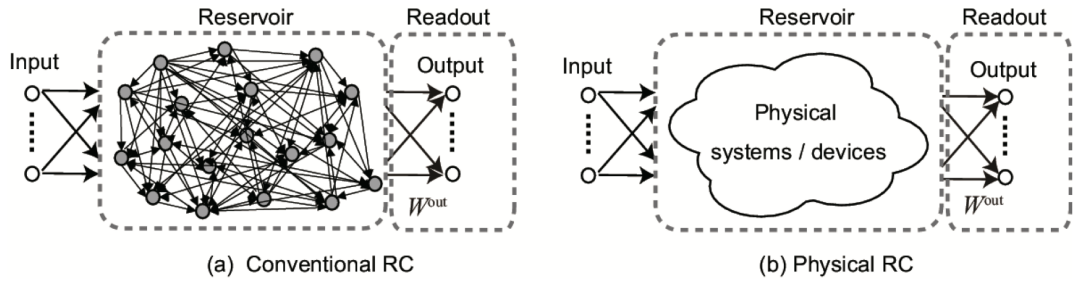
FIGURE 4.6 the figures give a overview of neuronal functions in artificial neural networks. The first generation (A) and the second generation (B) neurons sum (weighted) over all the incoming inputs and output based on a bounded function: a threshold function in (A) and an arbitrary continuous *activation* function in (B). Third generation neurons rely solely on the time and rate of the incoming input or spikes, not on their amplitude.



(A) Feedforward neural network



(B) Recurrent neural Network.



(C) Reservoir computing.

FIGURE 4.7 (a) Feed-forward networks have information transfer in one direction. (b) Recurrent networks have information feedback from layers further downstream. (c) Typically, the network consists of a randomly connected recurrent network or a reservoir and a feed-forward network connected either to the output or the input of the recurrent network. The recurrent network, here, is capable of mapping input signals to a higher dimensional vector spaces due its highly non-linear behaviour. The reservoir is essentially a black box, from which a signal can be readout from and trained on a feed forward networks. In recent years, scientist have proposed physical systems that can be used as reservoirs. Figure adapted from [68]

4.3 Learning in spiking neural networks

A synapse in a network of neurons is categorised by its weight. Synapses in the human brain can change their weights in response to the spiking of pre-synaptic and post-synaptic neurons. The ability to change and retain their synaptic strength is called synaptic *plasticity*. In the context of neural networks, the term *learning* refers to the determining the weights for specific task. A learning rule is a mathematically exact formulation of how the weight changes. If over time, the synaptic strength is continuously increased, it is called *long term potentiation* or LTP for short. On the other hand, if the synaptic weight decreases persistently, it is referred to as *long term depression* or LTD [59].

The algorithms described below are 'unsupervised' learning rules, meaning learning takes place through no additional factors other than the inputs that are fed into the network.

In 1949, Hebb conjectured that if two neuron were causally related in their spiking, then the synaptic connection between them is strengthened [69]. He theorised such a rule would help stabilise neuronal activity and result in associated firings that shows some semblance to learning. Hebb's idea was further extended to include depression when neuron do not show direct causal correlation. The above statement is often summarised as *" neurons that fire together, wire together"*.

The change in weight of a given synapse depends only on features local to that synapse such as its weight $w_{ij}$and the activities of the pre-synaptic neuron, v_i and post-synaptic neuron, v_j. Keeping this in mind, we can write down a general equation for w_{ij}

$$\frac{dw_{ij}}{dt} = c_1(w)v_i + c_2(w)v_j + c_3(w)v_i^2 + c_4(w)v_j^2 + c_5(w)v_iv_j + c_6(w) + \mathcal{O}(v^3)\,. \tag{4.3.1}$$

The coefficients in the equation above can be adjusted in accordance to our learning rule. For example, we can design a rule that depends on the firing rates of the neurons. The weight should be updated only when there is post-synaptic firing. Secondly, when the pre-synaptic firing rate is above a threshold, potentation should occur, otherwise depression takes place. These properties are accommodated in the following equation,

$$\frac{dw_{ij}}{dt} = c_o v_j(v_i - \theta(w_{ij}))\,, \tag{4.3.2}$$

where v_i and v_j are the average firing rates measured in a specified time interval and θ signifies the pre-synaptic firing threshold. This type of plasticity, that is based on the rate of spiking, is also called *spike rate dependent plasticity* or SRDP in short.

A Hebbian rule can also be described based on the exact firing times of the neuron instead of their rates. The simplest such rule is the *pair-based spike dependent plasticity* rule. Let t_{pre} be the time of firing of a pre-synaptic neuron and t_{post} be the time of firing of a post-synaptic neuron. The update rule is given by the difference Δt between the firing times of pre and

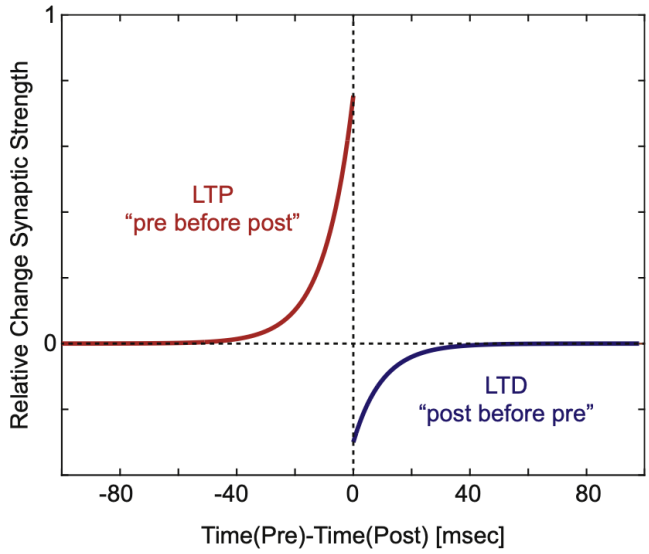


FIGURE 4.8 Change in weight of the synspase through spike-timing-dependent plasticity. Adapted from [70]

post-neurons i.e $\Delta t = t_{post} - t_{pre}$. Formally the rule can be written down as:

$$\Delta w = \begin{cases} +|w_+(\Delta t)| & \Delta t > 0 \\ -|w_-(\Delta t)| & \Delta t < 0 \end{cases} . \tag{4.3.3}$$

In the above equation w_+ and w_- are arbitrary functions of Δt. The rule imports that when a post-firing is followed by a pre-firing, the firings are causally related or $\Delta t > 0$, hence the synapse experiences potentiation. Similarly, when a post-firing is preceded by a pre-firing, the firings are not causally related or $\Delta t < 0$ resulting in a depression. It is important to note that an update is in effect only when both the pre and post-neurons fire. A common choice for the functions w_+ and w_- is

$$w_+ = A_+(w)\exp(-\Delta t/\tau_+) \tag{4.3.4}$$

$$w_+ = A_-(w)\exp(\Delta t/\tau_-) \, , \tag{4.3.5}$$

where A_+ and A_- are weight dependent function, τ_+ and τ_- are decay time constants.

For completeness, the rule can be expressed in the form of eq. (4.3.1), if we introduce activities S_i and S_j for pre-synaptic and post-synaptic neurons,

$$S_i = \sum_f \delta(t - t_i^f)$$
$$S_j = \sum_f \delta(t - t_j^f)\,,$$

with t_i and t_j indicating the time of firing and f, the firing index. The Hebbian rule is then,

$$\frac{dw_{ij}}{dt} = S_j(t)\int_0^\infty \mathrm{d}s\; w_+(s)S_i(t-s) + S_i(t)\int_0^\infty \mathrm{d}s\; w_-(s)S_j(t-s)\,. \tag{4.3.6}$$

In summary, we described two elementary learning rules inspired by Hebb's formulation of temporally correlated spiking amongst synaptic neurons. Emulating these learning rules on our proposed memristor i.e. Co/Nb:STO diode will be discussed in the next chapter.

Chapter 5

Hardware implementation

5.1 Introduction

In the previous chapter, the neuronal structure and function of the human brain were discussed. From the discussion, we gain insight into how to design neuromorphic hardware that takes inspiration from it. Firstly, such hardware should entail localised memory and processing units that are in close proximity to each other, similar to the vast network of neurons and the synaptic connections between them in the human brain. Secondly, the hardware should comprise of simple analog components with inherent physical features that allow them to simulate the complex dynamics of a biological neuron and synapse.

In this section, we address how Co/Nb:STO devices may be used in as synapses in neuromorphic hardware. As has been discussed earlier, Nb-doped STO Schottky diodes exhibit resistive switching which can be used to mimic synaptic plasticity.

5.2 Crossbar arrangement

In hardware, memory devices are commonly laid out in a crossbar array [71] to achieve a high packing density of its components. A crossbar consists of of two parallel planes of wires closely spaced together. At the crossing of each pair of wires, a memory device is placed that forms a connection, as seen in figure 5.1. The purpose of the upper and lower planes of wire is to supply a voltage to the specific memory device either for writing or reading. However, there is a major disadvantage to the crossbar architecture, namely the sneak path problem. Whenever

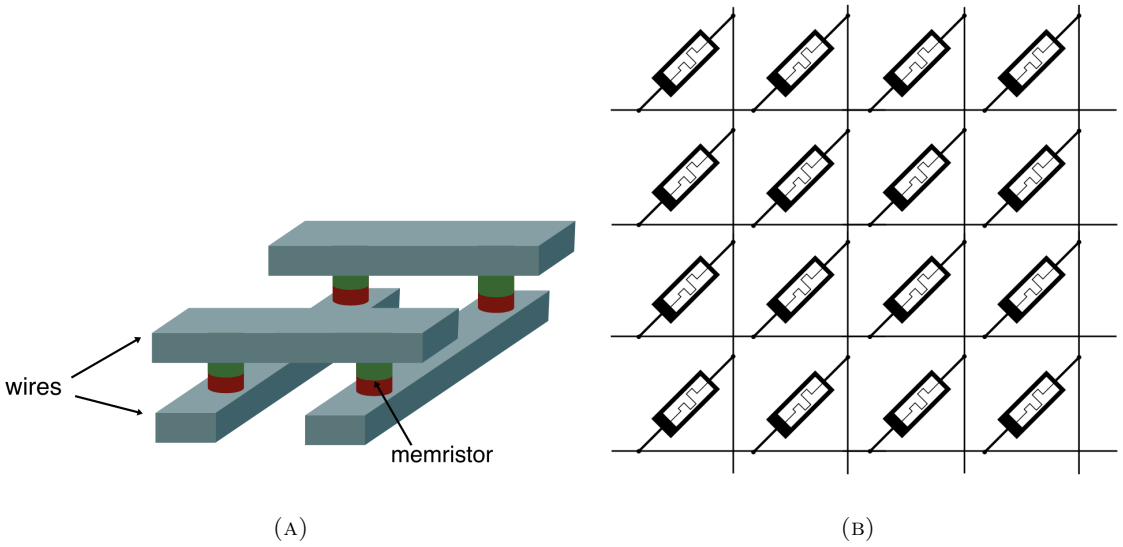


FIGURE 5.1 Schematic diagram of a crossbar arrangement. (A) A four cell arrangement. (B) Crossbar with a memristor circuit element connect upper and lower wires.

an operation is carried out on a given cell, the cells belonging to the row and column of the selected cell, also become biased. These cells are called *half-selected* cells. The remaining cells are called *unselected* cells. During a read or write operation, the current may take a spurious path through the array, depending on the states of the cell, and, hence, may unpredictably change the state of the half-selected and the unselected cells (see figure 5.2). These issues not only lead to unreliability in reading and writing, but may consume unnecessary power as well. To counter these problem ways relating to device characteristic, circuit characteristics, coding and biasing schemes have been proposed in literature [72, 17, 73].

One popular approach is the $V/3$ biasing scheme [17]. Suppose that one of the row wires is energised to V volts while one of the columns is grounded. At the intersection of these wire lies the selected cell. The remaining columns are then set to a voltage of $2V/3$ and the remaining rows to $V/3$. This scheme, hence, biases the un-selected and half-selected cells at $V/3$ volts. Proper execution of this scheme is highly device-dependent. For devices with very low thresholds for switching, such a scheme may not be possible.

Another approach is transistor-gating [72]. In this approach, one terminal of the memristor is connected to the drain or source of the transistor. Collectively, this configuration is called 1T1R, referring to the cell consisting of 1 transistor and 1 memristor. The key feature of this configuration is that the memristor is read/written, only if the the transistor gate receives an ON signal. Hence, the target cell is isolated which, in turn, minimises power consumption due to

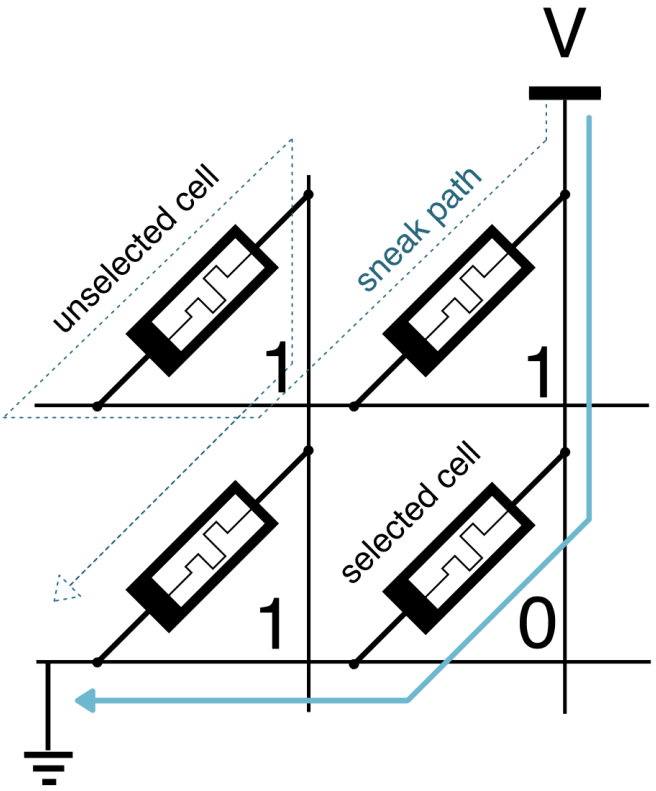


FIGURE 5.2 Sneak path problem. During a read or write process of corner cell in HRS (memory state 0) may occur erroneously because it is surrounded by cells in LRS (memory state 1).

sneak path and improves the reliability of reading and writing operations. One drawback of this design is that the packing density of the array is reduced. Recently, there has been literature suggesting the use of 1S1R cells, where the 'S' refers to a *selector* device [74, 75, 17]. The selector is a two terminal device that displays a highly non-linear response to voltage bias. Its response is such that in a selector-memristor series combination, current flows when the magnitude of voltage drop is sufficiently high. The combination results in a threshold for switching. The high non-linearity improves reliability in a passive read/write bias scheme. A 1S1R can also be coupled with the $V/3$ biasing scheme for improved operations. Proposed selector devices include metal-oxide Schottky barriers, threshold switches, ovonic threshold switching devices, metal insulator transitions devices amongst many others [17].

In the following sections, we discuss a 1T1R configuration using Nb-doped $SrTiO_3$ devices along with a simulation of such a configurations using Ardunio microcontrollers.

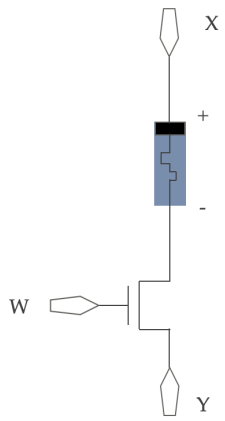


FIGURE 5.3 Circuit representation of 1T1R. W is the gate voltage. X and Y are connected to crossbar wires

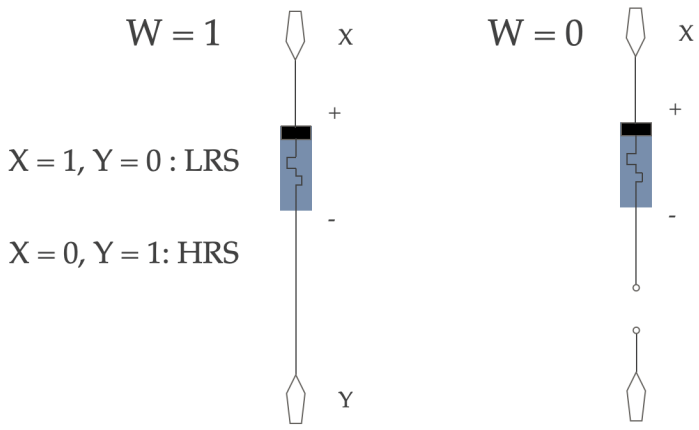


FIGURE 5.4 Functioning of 1T1R cell. When a gate voltage is supplied to the transistor, W = 1, the memristor can either be written to or read from. When W = 0, the memristor is electrically disconnected from the crossbar, hence, no read or write operation takes place.

5.3 1T1R configuration

As described earlier, the 1T1R configuration consists of a memristor and a transistor combination in series as is described in figure 5.3. As a whole, it forms a three terminal device. The gate terminal is called the *write line* and has been denoted by a W in the Figure below. Only when the write line is ON or there is a voltage at the gate, can the state of the device be changed (figure 5.4). The device is used as binary memory here, meaning it can represent a '1' (LRS) or a '0' (HRS) in memory. In the Co/Nb:STO devices, we observed forward biasing results in a low resistance state and negative biasing results in a high resistance state.

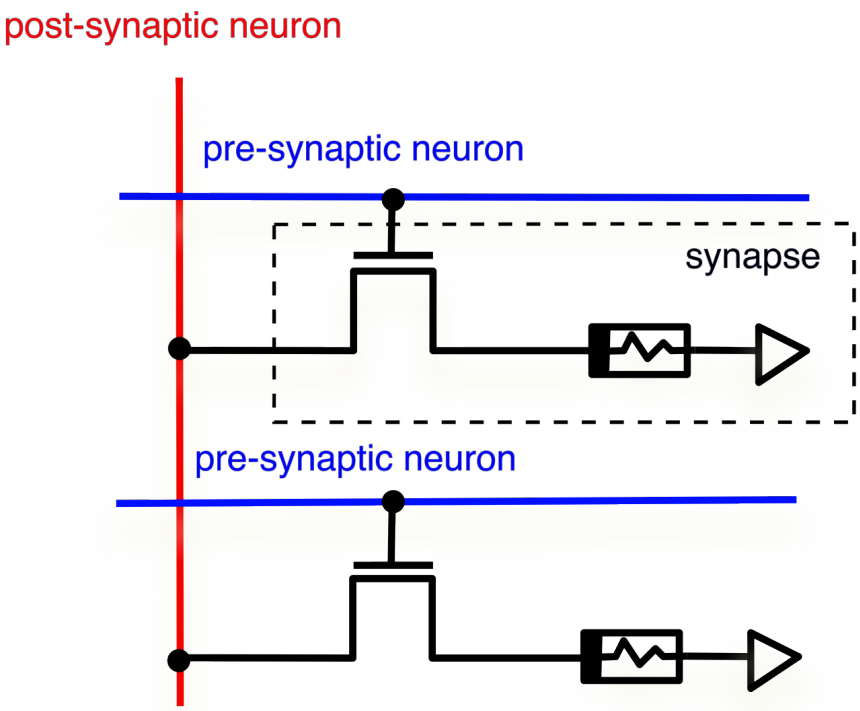


FIGURE 5.5 2x1 1T1R crossbar that has been simulated using microcontrollers.

We have used a crossbar arrangement of 1T1R to simulate an SNN, as shown in figure 5.5. In the network we have two pre-synaptic neuron and one post-synaptic neuron. In figure 5.5, the red line is connected to a post-synaptic neuron and the blue lines are connected to pre-synaptic neurons. The ground in the figure leads into the post-synaptic neuron. Note that the crossbar arrangement only allows the device to be read or written when a pre-synaptic neuron fire.

For the learning rule, we make use of a simplified version of STDP. This rule has been adapted from [76]. Qualitatively, the learning rule as shown in figure 5.6, is:

$$\Delta w = \begin{cases} +w_+(w) & t_0 > \Delta t > 0 \\ -w_-(w) & -t_0 < \Delta t < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.3.1}$$

where, w_+ and w_- are positive weight dependent constants; they do not have any dependence on time and Δt is the difference in spiking times of post-synaptic neuron and pre-synaptic neuron. Additionally, the weight is updated only if $|\Delta t| < t_0$, where t_0 is a constant pre-specified value. For a given weight, we would get an STDP curve as in figure 5.6

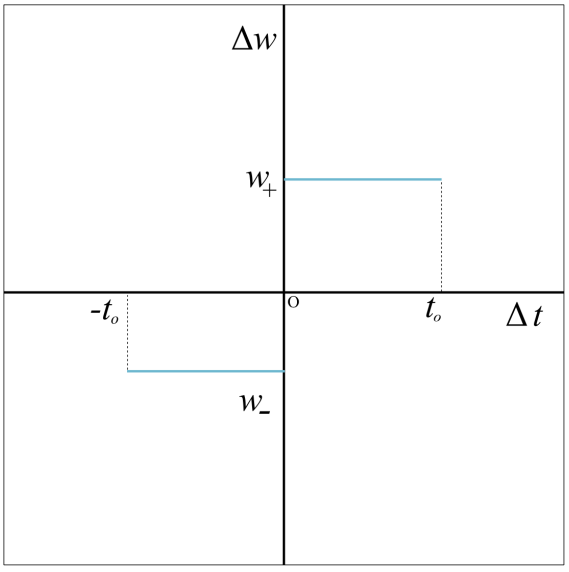


FIGURE 5.6 STDP rule simulated on microcontrollers. The change in weight is constant in the depression range, $\Delta t < t_0$ and potentiation range $\Delta t > t_0$.

This rule can easily be simulated by controlling the read and write voltages supplied to the crossbar. At all time the post-synaptic neuron sends a read voltage, while the pre-synaptic neuron is silent. In our simulation, we have used a negative voltage for reading. From the current voltage measurements in chapter 4, we see a value between -0.5 V and -1 V is sufficiently small to not affect the state of the device as well as large enough to be coherently detected. When a pre-synaptic neuron fires, a voltage pulse high enough to switch on the transistor is sourced at the corresponding pre-synaptic line. The voltage is sourced precisely for t_0 seconds (this t_0 is the same as the one in eq.(5.3.1)). When the transistor is turned on, the post-synaptic neuron can detect the current produced through the read voltage. If the resulting current causes the post-synaptic neuron to fire, the post-synaptic line send a feedback voltage. The feedback voltage consists of SET and a RESET pulse spaced at least t_0 seconds apart. If the SET pulse overlaps with pre-synaptic firing window, the memristor is potentiated, but if the RESET pulse overlaps, depression occurs. The SET pulse results in forward biasing of the device, while the RESET pulse results in reverse biasing. The pulse profile is depicted in figure 5.7. It is important to note that the refractory period of each neuron is set to be greater than t_0, so that there is no interference between outputs and feedback signals.

For the simulation, we use five Ardunio [77] microcontrollers. Two of these serve as 1T1R cells. Out of the remaining, two microcontrollers are used as pre-synaptic neurons, while the last one serves as a post-synaptic neurons. The pre and the post block upload consists of a leaky

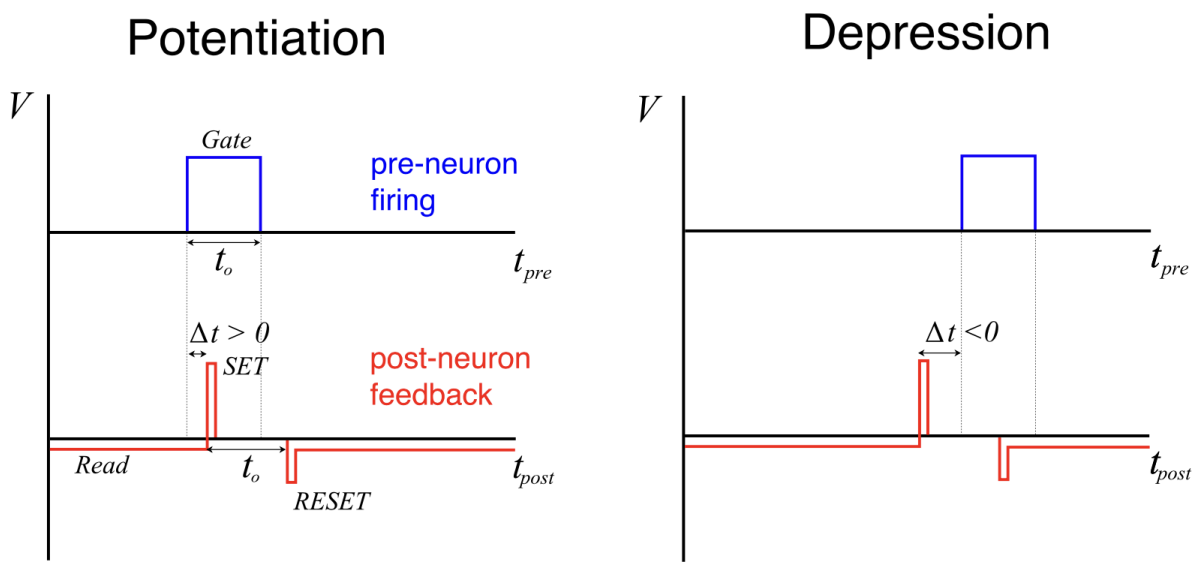


FIGURE 5.7: Potentiation and depression protocol.

integrate-and-fire neuron model along with protocols for the feedback signal. The synapse block is a four terminal block, instead of a three terminal block. Since, the microcontroller can only input/output voltage signals, the fourth line acts as a current line, where the current is encoded as voltage. The weight change in the microcontroller occurs as a small constant value addition (subtraction) to (from) the present weight. This is not representative of how the resistance of the Co/Nb:STO device changes on applied voltage but the code can be easily tweaked to include a more accurate model. Figure 5.8 shows the schematic diagram of the connections made. The program code can be found in the Appendix B.

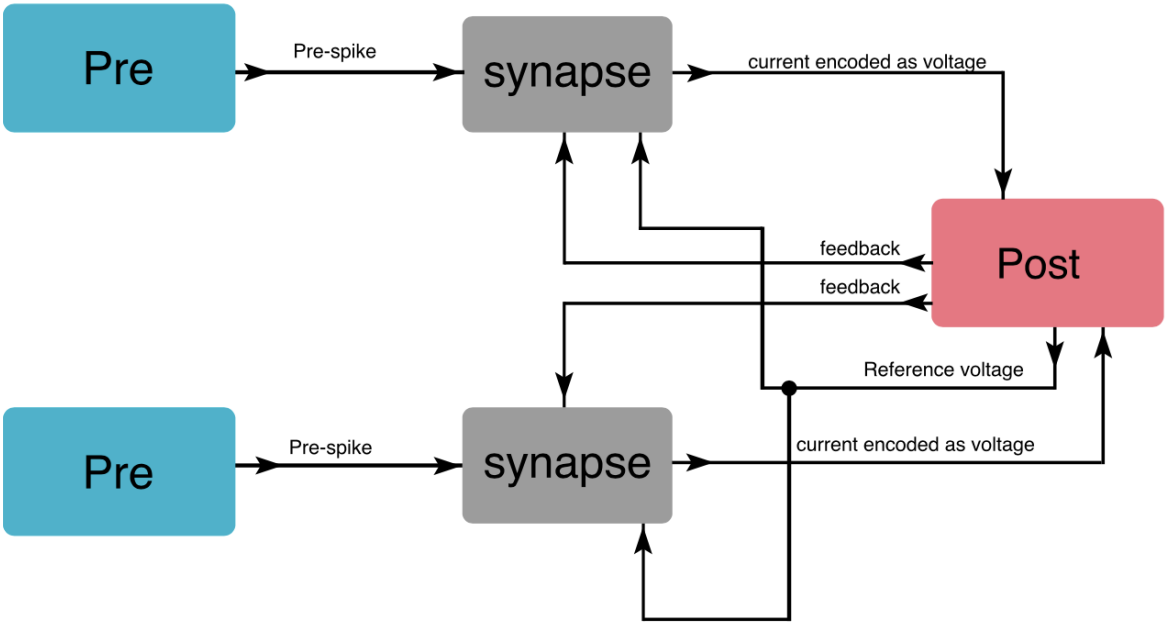


FIGURE 5.8 Schematic diagrams of simulation circuit. Each block represents an Ardunio Uno microcontroller. The input and outputs to a block are indicated by an arrow. The voltage bias in the synapse block is the difference between the voltage signals in the feedback and reference lines. The current output from synapses is encoded as a voltage signal.

Chapter 6

Conclusion

In this work, we look at Nb-doped $SrTiO_3$ Schottky diodes as prospective devices for neuromorphic architecture. Such devices display resistive switching characteristics which allow them to be used as electronic analogues of synapses in hardware. The resistive switching in these devices is categorised as interface-type and is most likely explained by an interplay between charge trapping/de-trapping at interface defects and mobile oxygen vacancies. The device is set in a low resistance state when a positive voltage is supplied to it, while it is set in a high resistance state when a negative voltage is supplied. The device shows multi-level switching: the resistance of the device can raised by systematically increasing the negative voltage. We explore the scalability of these devices by fabricating devices of three different areas. We saw that there was little device to device variation between devices of different areas which comments on reproducibility of the devices. Surprisingly, we observe an unusual trend of increasing hysteretic behaviour with decreasing device size.

In the second part of the thesis, we delve into neuromorphic computing from a software perspective to gain insights into how the Nb-$SrTiO_3$ Schottky diode can be incorporated into hardware. In hardware, the neurons, which are the processing units, receive and transmit information in the form of spikes whereas, synapses, which acts as the memory units, keep account of weights which dictate how much information flows through them. The behaviour of synapses can superficially be emulated by Nb-$SrTiO_3$ Schottky diodes, where the weight of synapse is analogous to the resistance of the memristor. The software part of neuromorphic computing comprises of spiking neural networks: these are neural networks that are mimic the human brain in their functioning. The weights in the network are determined by a learning rule. In our case,

we look at a pair-spike-timing dependent plasticity (STDP) learning rule, which changes the weight of a synapse according to spike timings of the neurons that it is connected to. The rule was simulated on a 2 input 1 output spiking neural network using Ardunio microcontrollers. In practice, synapses are connected in a crossbar arrangement on chip. Crossbar arrangements, though efficient memory arrangements, encounter the sneak path problem where the topology and states of the memory units can result in erroneous reading and writing operations. One proposed solution is the 1-transistor 1-memristor (1T1M) configuration which allow for the possibility of electrical isolation of memory units from one another. In the aforementioned simulation, five microcontrollers were used : three microcontrollers as neurons and two as 1T1M synapses.

Outlook

This work has opened up new questions in field of device physics and in neuromorphic hardware. The most striking discovery was that of the unusual hysteretic trend with decreasing device area. This can possibly be explained by a disparity in the nature of traps in the devices. This is presently being investigated. Secondly, the asymmetry in the reverse bias and forward bias I-V curves indicate that different switching mechanism are at play. The SET process is typically seen to occur on shorter time scales indicating electronic transport while the RESET process occurs on longer time scales indicating ionic transport. This effect needs to be investigated further. In hardware, to counter the sneak path problem, even though the 1T1M configuration serves as a viable solution, it occupies up unnecessary space. One could explore possible selector devices that can be stacked on top of memristors to resolve this issue.

Appendix B

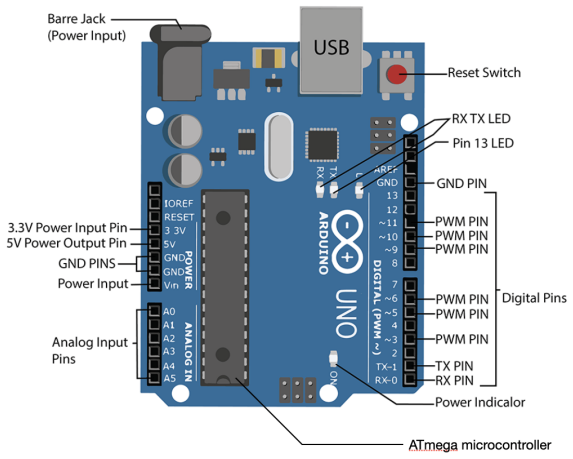
Ardunio Programming

B.1 What is Arduino?

"Arduino is an open-source electronics platform based on easy-to-use hardware and software."

The hardware part - Arduino board, a microcontroller. It is a miniature computer: it has a processor, memory and some kind of input/output devices.

The software part - the Arduino IDE that allows you to easily program an Arduino board. Programs or sketches are written in C++ plus some additional predefined functions. The IDE is available on all major OS.



Processor	Maximum operating frequency is around 20 MHz
Memory	• Flash memory 32 KB : stores code • SRAM 2KB : memory for variables in code • EEPROM 1KB : Additional long-term memory
Input/Output	• Digital input and output pins • Analog Input pins : 10 bit res ADC • Analog output pins : PWM - 8 bit res

FIGURE B.1: Image of an Ardunio Uno microcontroller and hardware specifications.

B.2 Programming in the IDE

The program is written in C++. There are additional functions native to Arduino useful useful in I/O and debugging. There are two main parts to the code:

- `void setup()` - Initialise pins, functions, variables, indicate which libraries to use etc. Run only once.
- `void loop()` - The body of program goes here. The code in here is looped continuously while the board is being powered.

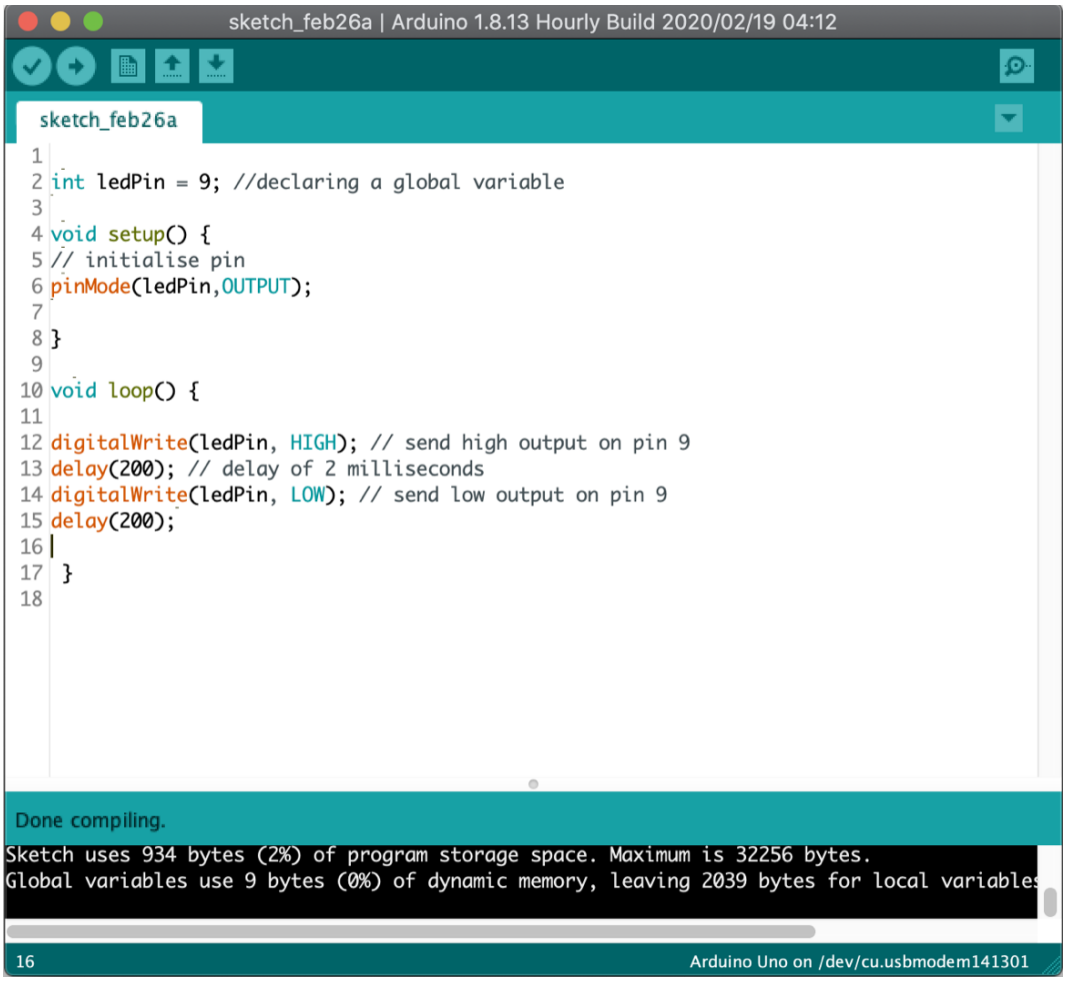


FIGURE B.2: Instance of Ardunio IDE.

B.3 Simulation

The simulation was done four Ardunio Uno board and 1 Nano board. The Ardunio Nano has hardware specifications similar to Uno.

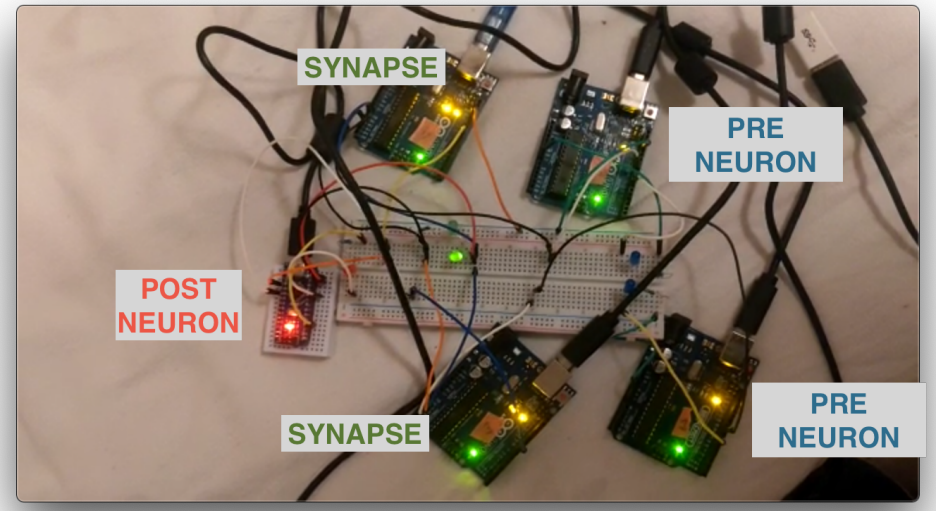


FIGURE B.3: Circuit with microcontrollers

B.3.1 LIF Neuron

```
// LIF Neuron
int drive = 9;
int feedback = 10;
int ledPin = 7;
int cur_input = A0;
int v_input = 11;
float v_th = 1.0;
float v_rest = 0.0;
float dt = 0.1 ;// seconds
int ext_fire = 2;
int d;
float t_rest = 0;
int t_ref = 1000;
float v = 0;
float I = 0;
float vread = 0.5;
float vplus = 3;
```

Bibliography

[1] David Silver et al. "Mastering the game of Go without human knowledge". In: *Nature* 550.7676 (2017), pp. 354–359. DOI: 10.1038/nature24270.

[2] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[3] John Von Neumann. *The computer and the brain*. Yale University Press, 2012.

[4] Edmund T. Rolls and Gustavo Deco. *The noisy brain: stochastic dynamics as a principle of brain function*. Oxford University Press, 2012.

[5] Daniele Ielmini. "Brain-inspired computing with resistive switching memory (RRAM): Devices, synapses and neural networks". In: *Microelectronic Engineering* 190 (2018), pp. 44–53. DOI: 10.1016/j.mee.2018.01.009.

[6] Ting-Chang Chang et al. "Resistance random access memory". In: *Materials Today* 19.5 (2016), pp. 254–264. DOI: 10.1016/j.mattod.2015.11.009.

[7] A. F. Vincent et al. In: *IEEE Transactions on Biomedical Circuits and Systems* ().

[8] S. R. Nandakumar et al. "A phase-change memory model for neuromorphic computing". In: *Journal of Applied Physics* 124.15 (2018), p. 152135. DOI: 10.1063/1.5042408.

[9] Harish Bhaskaran et al. "On-chip phase-change photonic memory and computing". In: *Active Photonic Platforms IX* (2017). DOI: 10.1117/12.2272127.

[10] L. Chua. "Memristor-The missing circuit element". In: *IEEE Transactions on Circuit Theory* 18.5 (1971), pp. 507–519.

[11] L. O. Chua and Sung Mo Kang. "Memristive devices and systems". In: *Proceedings of the IEEE* 64.2 (1976), pp. 209–223.

[12] Dmitri B. Strukov et al. "The missing memristor found". In: *Nature* 453.7191 (2008), pp. 80–83. DOI: 10.1038/nature06932.

[13] J.f. Gibbons and W.e. Beadle. "Switching properties of thin Nio films". In: *Solid-State Electronics* 7.11 (1964), pp. 785–790. DOI: 10.1016/0038-1101(64)90131-5.

[14] K. L. Chopra. "Avalanche-Induced Negative Resistance in Thin Oxide Films". In: *Journal of Applied Physics* 36.1 (1965), pp. 184–187. DOI: 10.1063/1.1713870.

[15] J.g. Simmons and R.r. Verderber. "New thin-film resistive memory". In: *Radio and Electronic Engineer* 34.2 (1967), p. 81. DOI: 10.1049/ree.1967.0069.

[16] F. Argall. "Switching phenomena in titanium oxide thin films". In: *Solid-State Electronics* 11.5 (1968), pp. 535–541. DOI: 10.1016/0038-1101(68)90092-0.

[17] Daniele Ielmini and Rainer Waser. *Resistive switching: from fundamentals of nanoionic redox processes to memristive device applications*. Wiley-VCH Verlag, 2016.

[18] Myoung-Jae Lee et al. "Electrical Manipulation of Nanofilaments in Transition-Metal Oxides for Resistance-Based Memory". In: *Nano Letters* 9.4 (2009), pp. 1476–1481. DOI: 10.1021/nl803387q.

[19] J Joshua Yang et al. "Corrigendum on The mechanism of electroforming of metal oxide memristive switches". In: *Nanotechnology* 21.33 (2010), pp. 339803–339803. DOI: 10.1088/0957-4484/21/33/339803.

[20] Ruth Muenstermann et al. "Coexistence of Filamentary and Homogeneous Resistive Switching in Fe-Doped SrTiO3 Thin-Film Memristive Devices". In: *Advanced Materials* 22.43 (2010), pp. 4819–4822. DOI: 10.1002/adma.201001872.

[21] Akihito Sawa. "Resistive switching in transition metal oxides". In: *Materials Today* 11.6 (2008), pp. 28–36. DOI: 10.1016/s1369-7021(08)70119-6.

[22] M.j. Sanchez. "Mechanism for bipolar resistive switching in transition metal oxides memristive interfaces". In: *SciVee* (2011). DOI: 10.4016/37657.01.

[23] S. Tsui et al. "Field-induced resistive switching in metal-oxide interfaces". In: *Applied Physics Letters* 85.2 (2004), pp. 317–319. DOI: 10.1063/1.1768305.

[24] B.J. et al. Choi. "Resistive switching mechanism of TiO2 thin films grown by atomic-layer deposition". In: *Journal of Applied Physics* 98.3 (2005), p. 033715. DOI: 10.1063/1.2001146.

[25] Y. B. Nian et al. "Evidence for an Oxygen Diffusion Model for the Electric Pulse Induced Resistance Change Effect in Transition-Metal Oxides". In: *Physical Review Letters* 98.14 (2007). DOI: 10.1103/physrevlett.98.146403.

[26] Seiji Suzuki et al. "Fabrication and characterization of Ba1xKxBiO3/Nb-doped SrTiO3 all-oxide-type Schottky junctions". In: *Journal of Applied Physics* 81.10 (1997), pp. 6830–6836. DOI: 10.1063/1.365242.

[27] Zhen Fan et al. "Resistive switching induced by charge trapping/detrapping: a unified mechanism for colossal electroresistance in certain Nb:SrTiO3-based heterojunctions". In: *Journal of Materials Chemistry C* 5.29 (2017), pp. 7317–7327. DOI: 10.1039/c7tc02197f.

[28] Xi Zou et al. "Charge trapping-detrapping induced resistive switching in Ba0.7Sr0.3TiO3". In: *AIP Advances* 2.3 (2012), p. 032166. DOI: 10.1063/1.4754150.

[29] Evgeny Mikheev et al. "Resistive switching and its suppression in Pt/Nb:SrTiO3 junctions". In: *Nature Communications* 5.1 (2014). DOI: 10.1038/ncomms4990.

[30] Rickard Fors, Sergey I. Khartsev, and Alexander M. Grishin. "Giant resistance switching in metal-insulator-manganite junctions: Evidence for Mott transition". In: *Physical Review B* 71.4 (2005). DOI: 10.1103/physrevb.71.045305.

[31] M. J. et al. Rozenberg. "Nonvolatile Memory with Multilevel Switching: A Basic Model". In: *Physical Review Letters* 92.17 (2004). DOI: 10.1103/physrevlett.92.178302.

[32] R. C. Neville, B. Hoeneisen, and C. A. Mead. "Permittivity of Strontium Titanate". In: *Journal of Applied Physics* 43.5 (1972), pp. 2124–2131. DOI: 10.1063/1.1661463.

[33] S. P. Chen. "Compositional and physical changes on perovskite crystal surfaces". In: *Journal of Materials Research* 13.7 (1998), pp. 1848–1852. DOI: 10.1557/jmr.1998.0262.

[34] A.i. Kingon et al. "High-Permittivity Perovskite Thin Films for Dynamic Random-Access Memories". In: *MRS Bulletin* 21.7 (1996), pp. 46–52. DOI: 10.1557/s0883769400035910.

[35] Yun-Yi Pai et al. "Physics of SrTiO3-based heterostructures and nanostructures: a review". In: *Reports on Progress in Physics* 81.3 (2018), p. 036503. DOI: 10.1088/1361-6633/aa892d.

[36] Kumari Gaurav Rana, Saurabh Roy, and Tamalika Banerjee. "Electron Transport across Oxide Interfaces on the Nanoscale". In: *Oxide Spintronics* (2019), pp. 65–106. DOI: 10.1201/9780429468193-3.

[37] A. Spinelli et al. "Electronic transport in dopedSrTiO3: Conduction mechanisms and potential applications". In: *Physical Review B* 81.15 (2010). DOI: 10.1103/physrevb.81.155110.

[38] Raymond T. Tung. "Recent advances in Schottky barrier concepts". In: *Materials Science and Engineering: R: Reports* 35.1-3 (2001), pp. 1–138. DOI: 10.1016/s0927-796x(01)00037-7.

[39] S. M. Sze and Kwok Kwok. Ng. *Physics of semiconductor devices.* Wiley-Interscience, 2007.

[40] E. H. Rhoderick and R. H. Williams. *Metal-semiconductor contacts.* Clarendon Press, 1988.

[41] Haruhiro Hasegawa and Toshikazu Nishino. "Electrical properties of Au/Nb-doped-SrTiO3contact". In: *Journal of Applied Physics* 69.3 (1991), pp. 1501–1505. DOI: 10.1063/1.347239.

[42] A. Kamerbeek. *PhD Thesis:Charge and spin transport in Nb-doped SrTiO3 using Co/AlOx spin injection contacts.* 2016.

[43] A. S. Goossens, A. Das, and T. Banerjee. "Electric field driven memristive behavior at the Schottky interface of Nb-doped SrTiO3". In: *Journal of Applied Physics* 124.15 (2018), p. 152102. DOI: 10.1063/1.5037965.

[44] El Mostafa Bourim, Yoonjung Kim, and Dong-Wook Kim. "Interface State Effects on Resistive Switching Behaviors of Pt/Nb-Doped SrTiO3Single-Crystal Schottky Junctions". In: *ECS Journal of Solid State Science and Technology* 3.7 (2014). DOI: 10.1149/2.0081407jss.

[45] D. R. Wolters and J. J. Van Der Schoot. "Kinetics of charge trapping in dielectrics". In: *Journal of Applied Physics* 58.2 (1985), pp. 831–837. DOI: 10.1063/1.336152.

[46] T. Fujii et al. "Hysteretic current–voltage characteristics and resistance switching at an epitaxial oxide Schottky junction SrRuO3SrTi0.99Nb0.01O3". In: *Applied Physics Letters* 86.1 (2005), p. 012107. DOI: 10.1063/1.1845598.

[47] X. G. Chen et al. "Comprehensive study of the resistance switching in SrTiO3 and Nb-doped SrTiO3". In: *Applied Physics Letters* 98.12 (2011), p. 122102. DOI: 10.1063/1.3569586.

[48] T. Fujii et al. "Electrical properties and colossal electroresistance of heteroepitaxialSrRuO3SrTi1xNbxO3(0.0002x0.02)Schottky junctions". In: *Physical Review B* 75.16 (2007). DOI: 10.1103/physrevb.75.165101.

[49] Christian Rodenbücher. "Resistive switching phenomena of extended defects in Nb-doped $SrTiO_3$ under influence of external gradients". RWTH Aachen, Diss., 2014. Dr. Jülich: RWTH Aachen, 2014, xiii, 200 pp. ISBN: 978-3-89336-980-5. URL: https://juser.fz-juelich.de/record/155980.

[50] Bin Liu et al. "Composition dependent intrinsic defect structures in SrTiO3". In: *Phys. Chem. Chem. Phys.* 16.29 (2014), pp. 15590–15596.

[51] Shinbuhm Lee et al. "Anomalous effect due to oxygen vacancy accumulation below the electrode in bipolar resistance switching Pt/Nb:SrTiO3 cells". In: *APL Materials* 2.6 (2014), p. 066103. DOI: 10.1063/1.4884215.

[52] W. Jiang et al. "Mobility of oxygen vacancy in SrTiO3 and its implications for oxygen-migration-based resistance switching". In: *Journal of Applied Physics* 110.3 (2011), p. 034509. DOI: 10.1063/1.3622623.

[53] Jinho Park et al. "Role of oxygen vacancies in resistive switching in Pt/Nb-doped SrTiO3". In: *Applied Physics Letters* 105.18 (2014), p. 183103. DOI: 10.1063/1.4901053.

[54] A. Ohtomo and H. Y. Hwang. "Surface depletion in doped SrTiO3 thin films". In: *Applied Physics Letters* 84.10 (2004), pp. 1716–1718. DOI: 10.1063/1.1668329.

[55] N. Shanthi and D. D. Sarma. "Electronic structure of electron doped $SrTiO_3$: $SrTiO_{3-\delta}$ and $Sr_{1-x}La_xTiO_3$". In: *Phys. Rev. B* 57 (4 Jan. 1998), pp. 2153–2158. DOI: 10.1103/PhysRevB.57.2153. URL: https://link.aps.org/doi/10.1103/PhysRevB.57.2153.

[56] Toru Hara. "Electronic structures near surfaces of perovskite type oxides". In: *Materials Chemistry and Physics* 91.2-3 (2005), pp. 243–246. DOI: 10.1016/j.matchemphys.2004.11.032.

[57] G. E. Moore. "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff." In: *IEEE Solid-State Circuits Society Newsletter* 11.3 (2006), pp. 33–35.

[58] Emma Strubell, Ananya Ganesh, and Andrew McCallum. *Energy and Policy Considerations for Deep Learning in NLP*. 2019. arXiv: 1906.02243 [cs.CL].

[59] Wulfram Gerstner et al. *Neuronal dynamics from single neurons to networks and models of cognition*. Cambridge University Press, 2016.

[60] Larry R. Squire. *Encyclopedia of neuroscience*. Elsevier, Acad. Press, 2009.

[61] A. L. Hodgkin and A. F. Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *Bulletin of Mathematical Biology* 52.1-2 (1990), pp. 25–71. DOI: 10.1007/bf02459568.

[62] E.m. Izhikevich. "Simple model of spiking neurons". In: *IEEE Transactions on Neural Networks* 14.6 (2003), pp. 1569–1572. DOI: 10.1109/tnn.2003.820440.

[63] L.f Abbott. "Lapicque's introduction of the integrate-and-fire model neuron (1907)". In: *Brain Research Bulletin* 50.5-6 (1999), pp. 303–304. DOI: 10.1016/s0361-9230(99)00161-6.

[64] Eugene Izhikevich and Richard Fitzhugh. "FitzHugh-Nagumo model". In: *Scholarpedia* 1.9 (2006), p. 1349. DOI: 10.4249/scholarpedia.1349.

[65] Vladimir Kornijcuk et al. "Leaky Integrate-and-Fire Neuron Circuit Based on Floating-Gate Integrator". In: *Frontiers in Neuroscience* 10 (2016). DOI: 10.3389/fnins.2016.00212.

[66] Marcel Stimberg, Romain Brette, and Dan Fm Goodman. "Brian 2, an intuitive and efficient neural simulator". In: *eLife* 8 (2019). DOI: 10.7554/elife.47314.

[67] Wolfgang Maass. "Networks of spiking neurons: The third generation of neural network models". In: *Neural Networks* 10.9 (1997), pp. 1659–1671. DOI: 10.1016/s0893-6080(97)00011-7.

[68] Gouhei Tanaka et al. "Recent advances in physical reservoir computing: A review". In: *Neural Networks* 115 (July 2019), pp. 100–123. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2019.03.005. URL: http://dx.doi.org/10.1016/j.neunet.2019.03.005.

[69] D. O. Hebb. *The Organization of Behavior*. John Wiley and Sons Inc., 1964.

[70] Flavio Fröhlich. "Synaptic Plasticity". In: *Network Neuroscience* (2016), pp. 47–58. DOI: 10.1016/b978-0-12-801560-5.00004-5.

[71] M. Hu et al. "Memristor Crossbar-Based Neuromorphic Computing System: A Case Study". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.10 (2014), pp. 1864–1878.

[72] Lingyun Shi et al. "Research progress on solutions to the sneak path issue in memristor crossbar arrays". In: *Nanoscale Advances* 2.5 (2020), pp. 1811–1827. DOI: 10.1039/d0na00100g.

[73] Y. Cassuto, S. Kvatinsky, and E. Yaakobi. "Write sneak-path constraints avoiding disturbs in memristor crossbar arrays". In: *2016 IEEE International Symposium on Information Theory (ISIT)*. 2016, pp. 950–954.

[74] Qiangfei Xia and J. Joshua Yang. "Memristive crossbar arrays for brain-inspired computing". In: *Nature Materials* 18.4 (2019), pp. 309–323. DOI: 10.1038/s41563-019-0291-x.

[75] Geoffrey W. Burr et al. "Access devices for 3D crosspoint memory". In: *Journal of Vacuum Science Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena* 32.4 (2014), p. 040802. DOI: 10.1116/1.4889999.

[76] G. Pedretti et al. "Publisher Correction: Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity". In: *Scientific Reports* 8.1 (2018). DOI: 10.1038/s41598-018-26716-7.

[77] *Ardunio*. URL: https://www.arduino.cc/.