

Memory Technologies for Machine Learning -

EITP25

Hand-in 2 - Spring 2020

Examining the STDP learning mechanism

Hicham Mohamad, hi8826mo-s

27 April 2021

1 Introduction

In this assignment, we will work on writing a simulation for **Spiking Neural Networks** (SNN) in a Python software package called **Brian**. Then, we will practice and get familiar with the dynamics of the **STDP** learning mechanism and comparing it to pure **Hebbian** learning.

The resulting plots and values in this report are obtained by running the implemented Python script `EITP25_STDP_assignment.py`.

2 Background

In order to understand how the **STDP learning mechanism** is performing, we need to have a basic understanding of the biological neuron activity.

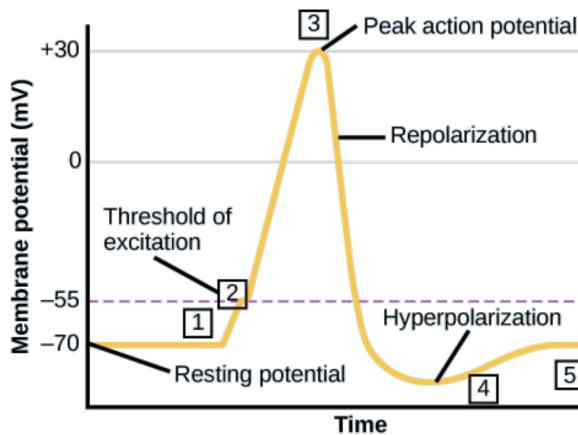


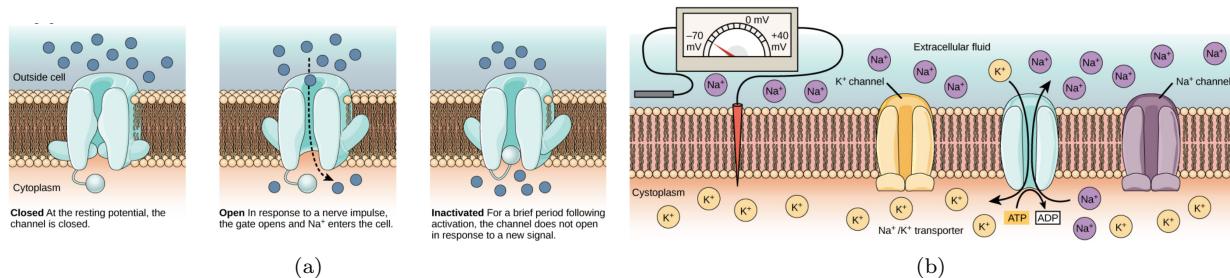
Figure 1: Formation of an action potential can be divided into five steps: (1) A **stimulus** from a sensory cell or another neuron causes the target cell to depolarize toward the threshold potential. (2) If the **threshold of excitation** is reached (-55 mV), all Na^+ channels open and the **membrane depolarizes**, i.e. having a decrease in the difference in voltage between the inside and outside of the neuron. (3) At the **peak action potential** (+40 mV), K^+ channels open and K^+ begins to leave the cell. At the same time, Na^+ channels close. Once depolarization is complete, the cell must now "reset" its membrane voltage back to the resting potential. (4) The membrane becomes **hyperpolarized** as K^+ ions continue to leave the cell (diffusion), in that the membrane potential becomes more negative than the cell's normal resting potential. The hyperpolarized membrane is in a **refractory period** and cannot fire and produce another action potential because its sodium channels will not open. (5) The K^+ channels close and the Na^+/K^+ transporter restores the **resting potential**. At this point, the sodium channels will return to their resting state, meaning they are ready to open again if the membrane potential again exceeds the threshold potential.

Biological neuron activity

Here it is nice to explore some of the basics of the **neuron activity**. The cell membrane of a neuron encloses **cytoplasm** with various ions dissolved in it. The neuron itself is immersed in a salt solution, the extracellular fluid. The ions of the cytoplasm consist mainly of positively charged **potassium ions** K^+ and large negatively charged organic molecules, such as proteins. Outside the cell, the extracellular fluid contains mostly positively charged **sodium ions** Na^+ and negatively charged **chloride ions** Cl^- .

Resting potential

Unstimulated, neurons maintain a constant electrical **difference**, or potential, across their cell membranes. This potential, called **resting potential**, is always negative inside the cell. It ranges from -40 to -90 mV. If a neuron is stimulated, the negative potential inside the neuron can be made either more or less negative, depending on the stimulus.



*Figure 2: a) Voltage-gated ion channels open in response to changes in membrane voltage. After activation, they become inactivated for a brief period and will no longer open in response to a signal. b) At the **resting potential**, all voltage-gated Na^+ channels and most voltage-gated K^+ channels are closed. The Na^+/K^+ pump pumps K^+ ions into the cell and Na^+ ions out.*

Action potential

If potential is made sufficiently less negative, it reaches a level called **threshold** (-55 mV), and **action potential** is triggered. During the action potential, the neuron suddenly becomes 20 to 50 mV positive inside. Action potentials last a few milliseconds before the cell restores its negative resting potential.

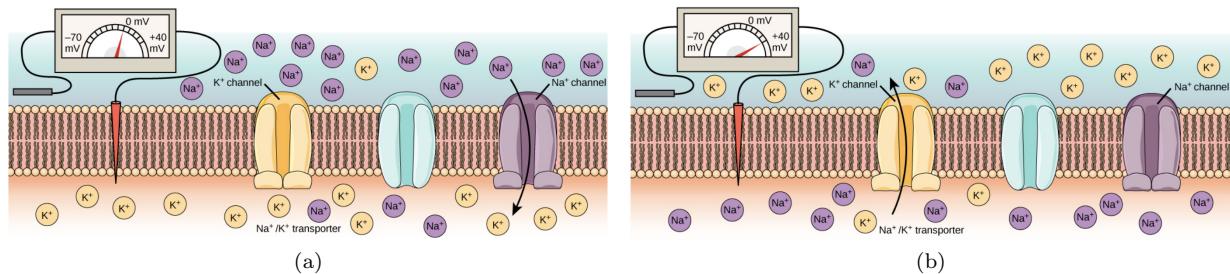


Figure 3: A nerve impulse causes Na^+ to enter the cell, resulting in (a) depolarization. At the peak action potential, K^+ channels open and the cell becomes (b) hyperpolarized.

Hodgkin-Huxley Model

Hodgkin and Huxley performed experiments on the giant **axon of the squid** and found three different types of **ion current**, sodium, potassium, and a leak current that consists mainly of Cl^- ions. This Hodgkin-Huxley model can be understood with the help of Figure 4.

The semipermeable **cell membrane** separates the interior of the cell from the extracellular liquid and acts as a **capacitor**. If an input current $I(t)$ is injected into the cell, it may add further charge on the capacitor, or leak through the **channels** in the cell membrane. Each channel type is represented by a resistor. The unspecific channel has a leak resistance R , the sodium channel a resistance R_{Na} and the potassium channel a resistance R_K .

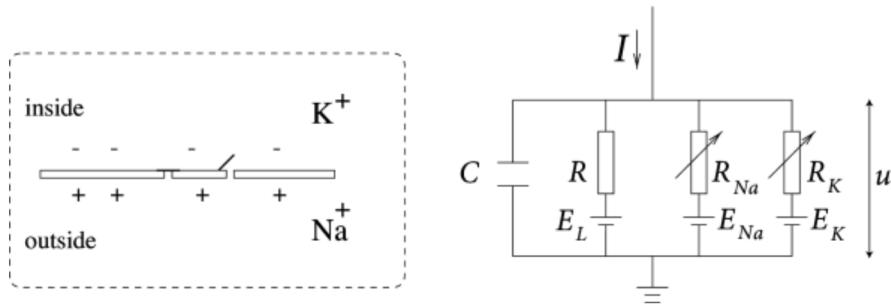


Figure 4: Schematic diagram for the Hodgkin-Huxley model.

Memristor

A memristor (named as a portmanteau of memory and resistor) is a non-volatile electronic memory device that was first theorized by Leon Ong Chua in 1971 as the fourth fundamental two-terminal circuit element following the resistor, the capacitor, and the inductor.

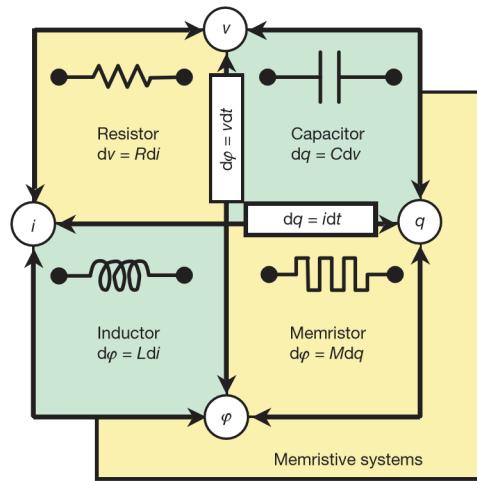


Figure 5: The four fundamental two-terminal circuit elements: resistor, capacitor, inductor and memristor. Resistors and memristors are subset sof a more general class of dynamical devices, **memristive systems**. Note that R , C , L and M can be **functions** of the independent variable in their defining equations, yielding **nonlinear elements**. For example, a charge-controlled memristor is defined by a single-valued function $M(q)$.

Properties

Its special property is that its resistance can be programmed (resistor function) and subsequently remains stored (memory function). Unlike other memories that exist today in modern electronics, memristors are stable and remember their state even if the device loses power.

Today, most computers use random access memory (RAM), which moves very quickly as a user works but does not retain unsaved data if power is lost. Flash drives, on the other hand, store information when they are not powered but work much slower. Memristors could provide a memory that is the best of both worlds: fast and reliable.

Memristors have several attractive features that make them compelling for computer scientists: They require **less energy** to operate and are **faster** than present solid-state storage technologies and they can store at least twice as much data in the same area. Memristors are virtually immune from radiation, which can disrupt transistor-based technologies.

Memristors and biological synapses

The high interest in memristor devices also stems from the fact that these devices emulate the memory and learning properties of biological synapses. i.e. the electrical resistance value of the device is dependent on the history of the current flowing through it.

A memristor is similar to a synapse in the human brain because it exhibits the same switching characteristics, i.e. it is able, with a high level of plasticity, to modify the efficiency of signal transfer between neurons under the influence of the transfer itself. That's why researchers are hopeful to use memristors for the fabrication of electronic synapses for neuromorphic computing that mimics some of the aspects of learning and computation in human brains.

Synapses - STDP equations

For a biological synapse, in Spike Timing Dependent Plasticity (STDP), the difference in time between the firing of pre- and post-synaptic neurons determines whether the "strength" of the connection between those neurons increases or decreases.

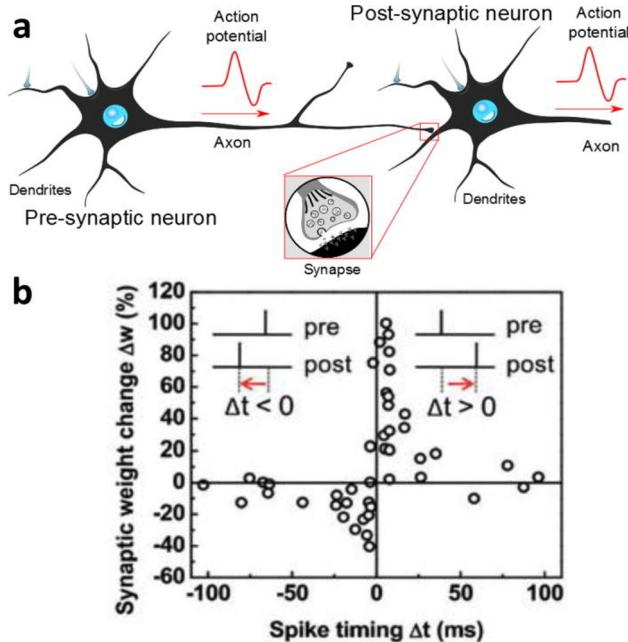


Figure 6: (a) Schematic illustration of a synaptic connection and the corresponding pre- and post-synaptic neurons. The synaptic connection strengthens or weakens based on the spike activity of these neurons; a process referred to as synaptic plasticity. (b) A well-known plasticity mechanism is spike time-dependent plasticity (STDP), leading to weight changes that depend on the relative timing between the pre- and post-synaptic neuronal spike activities.

In other words, when the pre-synaptic spike arrives before the post-synaptic spike ($\Delta t > 0$), the synapse exhibited increased synaptic weight ($\Delta w > 0$) and when the pre-synaptic spike arrives after the post-synaptic spike ($\Delta t < 0$) that the synapse exhibited decreased synaptic weight ($\Delta w < 0$).

STDP implementation

Now we go through an **event-based** way of defining **STDP behaviour**. Here, `stdp_eq` defines the **weight** variable `w` (unit 1) and **two decaying state** differential equations for `a_pre` and `a_post` as:

$$\frac{da_{pre}}{dt} = -\frac{a_{pre}}{\tau_{pre}}, \quad \frac{da_{post}}{dt} = -\frac{a_{post}}{\tau_{post}} \quad (1)$$

These states decay over time and are increased upon **pre-neuron spike** (`a_pre` by an amount `A_pre`) and **post-neuron spike** (`a_post` by an amount `A_post`) as is defined in `on_pre_eq` and `on_post_eq`, respectively.

On a **pre-spike event** the post-neuron potential is increased by `w`, and `w` in turn is decreased by `a_post` (clipped between 0 and 1). Thus if a post-event just occurred then `a_post` is large and the weight is strongly decreased, otherwise the impact is minimal.

On a **post-spike event** `w` is increased by `a_pre`, which if a pre-event just happened has not decayed much and will strongly increase the weight.

Listing 1: Equations for the synapses - STDP equations

```

1 # STDP equations
2 Apre = Apost = 0.03
3 wmax = 1.0
4 tau_pre = 20*ms
5 tau_post = 35*ms #depression decays slower
6 stdp_eq = '''
7     w : 1
8     dapre/dt = -apre/tau_pre : 1 (event-driven)
9     dapost/dt = -apost/tau_post : 1 (event-driven)
10    '',
11 on_pre_eq = '''
12     v_post += w
13     apre += Apre
14     w = clip(w-apost, 0, wmax)
15    '',
16 on_post_eq = '''
17     apost += Apost
18     w = clip(w+apre, 0, wmax)
19    '',

```

Neuron model - Differential equations

Here we define the membrane potential v differential equation in `eqs`:

$$\frac{dv}{dt} = \frac{-v}{\tau_v} \quad (2)$$

Which describes the "leakiness" of the neuron.

Neuron model implementation

The membrane potential v differential equation is implemented in `eqs`.

`thres` defines what the condition for spiking should be, set to $v > T = 1.0$, but this value you may play around with.

Listing 2: Neuron model - Neuron differential equations

```

1 #Neuron differential equations
2
3 tau_v = 20*ms #20ms
4 T = 1.0
5
6 # neuron model
7 eqs = '''
8 dv/dt = -v/tau_v : 1
9 '',
10 thres = 'v > T'
11 res = '',
12 v=0
13 ''

```

Building the network

Here we define the **pre-neurons** as a number, `num_inputs`, of Neurons firing randomly with an average frequency of `input_rate` (50 Hz). These are represented by the **PoissonGroup** object `P`.

The single **post-neuron** is defined as a **NeuronGroup** object `G` defined by the equations `eqs`, `thres` and `res`.

The **Synapses** object `S` connects the two neuron groups `P` and `G`, and uses the synapse models we defined in equations `stdp_eq`, `on_pre_eq` and `on_post_eq`. We connect all inputs to the output with `S.connect()`, and then sets the weights to a random value in the range $[0, w_{\max}]$.

Finally, we define **two Monitors** that record things during the simulation. This is a convenient way to collect data to use to analyse the network afterwards. For large simulations they slow down the simulation too much, however.

The `PopulationRateMonitor()` `M` measures the spiking rate of `G` as a function of simulation time. The `StateMonitor()` `WM` records the synapse weights, and how they change over time.

Synapse with excitatory and inhibitory plasticity

In neurobiology, **lateral inhibition** is the capacity of an excited neuron to reduce the activity of its neighbors. Lateral inhibition disables the spreading of action potentials from excited neurons to neighboring neurons in the lateral direction.

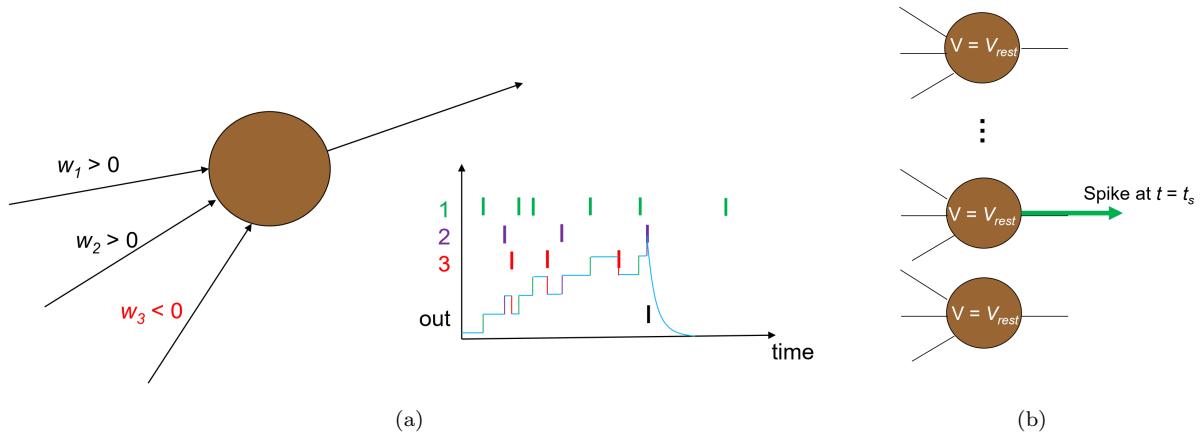


Figure 7: a) Excitatory/Inhibitory synapses: Not all synapses shift the neuron potential towards threshold (excitation). Some reduce the neuron potential (inhibition). Inhibition can be a crucial feature for learning in SNNs. Inhibition can be modeled by synapse with negative weight. b) Winner-takes-all (WTA): If all neurons in a layer can completely inhibit all others in the layer, i.e. the first to fire will be the only one to fire. Typically implemented as inhibition for a certain time, matching refractive period $V = V_{rest}$ on all until $t = t_s + t_{refr}$.

The **winner-takes-all** (WTA) mechanism, that is often assumed to take place in biological nervous systems, allows only the most strongly-activated unit in the network to output something that distinguishes it as the winner (e.g., as **delta function**) over other units. When a unit fires, we also require it to inhibit all units in the network after a delay of time.

In simulation, **Excitatory cells** receive input and the neuron receiving the most input activation is the first to reach its spiking threshold. Spiking excites the **inhibitory cell**, which in turn prevents other cells from responding.

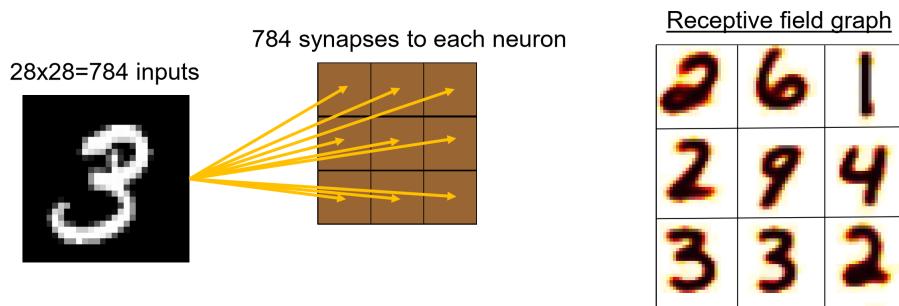


Figure 8: Lateral inhibition promotes neuronal learning of specific prototypes in the data, since only a few neurons can spike and adjust their receptive field towards a specific prototype. Lateral inhibition also prevents overfitting as the competition between neurons forces them to learn different features/prototypes.

3 Tasks

Task 1

After running the provided code as it is, we get the resulting output as shown in Figure 9.

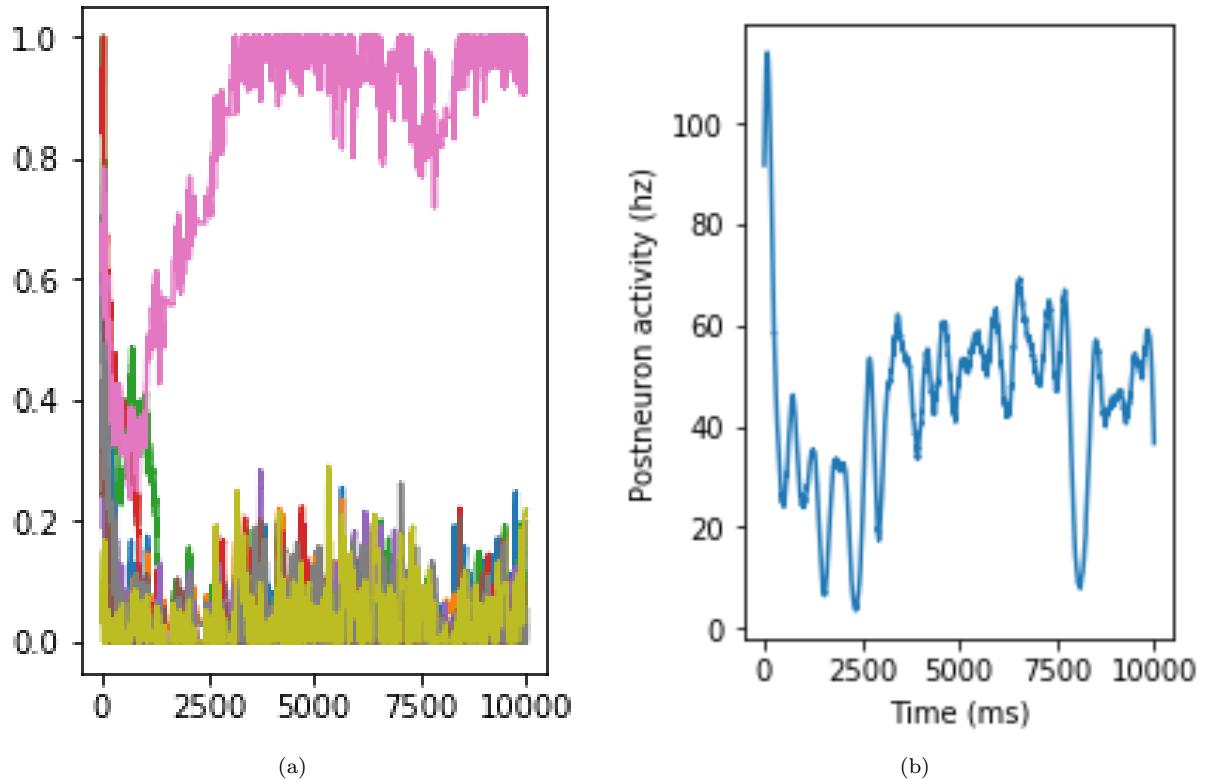


Figure 9: a) Plot of synapse weight where each color indicates the weight of one synapse. b) Graph that indicates change in the postneuron spiking rate over time.

Question: If we had only 1 pre-neuron in the network, for a 50 Hz average input rate and threshold of 1, what is the rough **firing rate** that we could expect?

Task 2

Vary the value of the variable `tau_post`.

Question: What is the impact on the weights and activity? Why?

Task 3

Vary the magnitude of `Apre` and `Apost` (while keeping them equal).

Question: What is the impact on the weights and activity? Why?

Task 4

Vary the threshold `T`.

Question: What is the impact on the weights and activity? Why?

Task 5

Vary the input rate and the number of input neurons

Question: What is the impact on the weights and activity? Why?

Task 6

Alter the synaptic learning rules to conform with pure Hebbian learning instead:

Question: What is the impact on the weights and activity? Why?

Task 7

Now alter this Hebbian rule to use Oja's rule instead.

```
1 on-pre_eq = ''
2     v-post += w
3     apre += Apre
4     w = clip(w-apost, 0, wmax)
5     '',
6 on-post_eq = ''
7     apost += Apost
8     ''
```

Question: What is the impact on the weights and activity? Why?

4 Appendix

References

- [1] Mattias Borg, Memory Technologies For Machine Learning - EITP25, Lectures notes: <https://canvas.education.lu.se/courses/2101>
- [2] Chen, Hutchby, Zhirnov and Bourianoff, Emerging Nanoelectronic Devices, Wiley 2015.
- [3] Manan Suri: Advances in Neuromorphic Hardware Exploiting Emerging Nanoscale Devices. Springer, 2017, ISBN: 978-81-322-3701-3.
- [4] Dmitri Strukov, G. Snider, D. Stewart and R. Williams, The missing memristor found. 2008 Nature vol 453.
- [5] Action Potential, Lumen Module 14: The Nervous System. <https://courses.lumenlearning.com/wm-biology2/chapter/action-potential/>
- [6] VanderPlas, A Whirlwind Tour of Python. O'Reilly 2016. Online reading: <https://jakevdp.github.io/WhirlwindTourOfPython/>
- [7] Geron, Jupyter notebook on linear algebra from Machine Learning and Deep Learning in Python using Scikit-Learn, Keras and TensorFlow: <https://github.com/ageron/handson-ml2>
- [8] Brian2 Documentation: <https://brian2.readthedocs.io/en/stable/index.html>.
- [9] Lumen Leanrning, Resting Membrane Potential: <https://courses.lumenlearning.com/wm-biology2/chapter/resting-membrane-potential/>.
- [10] Neuronal Dynamics, Online book. The Hodgkin-Huxley Model: <https://neuronaldynamics.epfl.ch/online/Ch2.S2.html>.
- [11] P. Diehl, M. Cook, Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience, Institute of Neuroinformatics, ETH Zurich and University Zurich, Zurich, Switzerland: <https://www.frontiersin.org/articles/10.3389/fncom.2015.00099/full>.
- [12] Harvard Extention School, Action Potential in the Neuron: <https://youtu.be/oa6rvUJlg7o>.
- [13] Ren Hartung, Neuron Action Potential for Beginners. Lecture on Youtube: https://www.youtube.com/watch?v=8g1RBj03Syo&feature=emb_rel_end.
- [14] Brian Hays. The Memristor: <https://www.americanscientist.org/article/the-memristor>, American Scientist.

- [15] L. F. Abbott and Sacha B. Nelson, Synaptic plasticity: taming the beast. 2000 Nature America Inc.
- [16] D. Querlioz, et al, Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches, IEEE/ACM International Symposium on Nanoscale Architectures, 203–210 (2012).
- [17] Spiking neural networks: <https://towardsdatascience.com/spiking-neural-networks-558dc4479903>.