ADVANCING KNOWLEDGE TRANSFER IN **SPIKING NEURAL NETWORKS**

BT5051
TRANSPORT PHENOMENA IN BIOLOGICAL SYSTEMS

CHOOSE-FOCUS-ANALYZE (CFA)

Work submitted by

Anshul Bagaria
BE21B005

1 Motivation

The human brain has always held a special fascination for me. It's a truly remarkable organ, enabling us to perform incredible feats, from recognizing a familiar face in a crowd to creating beautiful music. What's even more astonishing is that it accomplishes all this with remarkable efficiency, using far less computational power than our most advanced artificial neural networks.

In today's world, artificial neural networks strive to replicate the brain's incredible abilities. They help us tackle complex computer-related challenges. However, they often demand substantial computational resources, extensive datasets, and significant time to learn. You could think of them as the voracious eaters of the computational world, quite unlike our brains.

To delve deeper into the intricate workings of the human brain, I embarked on a journey into the realm of Computational Neuroscience. This path introduced me to novel concepts, providing a clearer and more rigorous understanding of the brain's inner workings. This newfound knowledge forms the bedrock of my decision to model Spiking Neural Networks. These artificial neural networks are considered the closest approximations to the human brain, we've achieved so far.

2 Objective

In a multitude of engineering domains, complex systems are governed by a web of simultaneous forces and conservative equations. These equations encompass mass conservation, momentum conservation, thermal energy transfer and charge conservation alongside Maxwell's equations.

Traditionally, the analysis of complex systems adheres to a rigorous approach, where each relevant conservation equation is simultaneously solved. This method offers precise solutions, vital for situations demanding pinpoint accuracy such as high-fidelity simulations. However, when faced with the intricate nature of multiple operational forces, solving these conservation equations becomes a mathematical labyrinth.

An alternative to the rigorous approach emerges through the utilization of transfer coefficients, often known as conductances. Transfer coefficients act as dynamic parameters that delineate how a specific system reacts to diverse forces or inputs. These coefficients encapsulate the intricate interactions among different variables in the system, effectively simplifying the complexity. This approach is particularly valuable when an approximate solution is permissible, or when constraints on computational resources or time impose practical limits.

This analogy has a profound connection to the realm of Spiking Neural Networks (SNNs). SNNs are computational models inspired by the human brain's neural activity, where information is conveyed through discrete, spiking events. In SNNs, information is processed through the transfer of electrical signals, and the deployment of conductances is akin to the transfer coefficients we employ in complex problem-solving. This linkage highlights the practical applicability of transfer coefficients in modeling neural activity and information processing in SNNs.

3 Introduction

3.1 Mobility of ions across a neural membrane

The in-depth comprehension of neural cell operations, specifically in the context of information propagation, stands as an indispensable component within the domains of both biology and artificial intelligence. In the biological systems, the translocation of ions across neural cell membranes plays a pivotal role in the transmission of signals and the generation of action potentials.

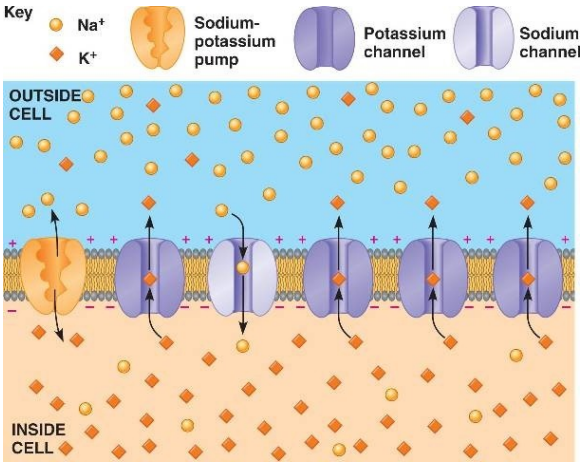
This comprehension is underpinned by the fundamental tenets of transport phenomena, with the passive ion channels or the active transporters serving as the conduits for the migration of charged ions. (Suraishkumar [2014])

$$V_{int} - V_{ext} = -\frac{RT}{z_n F} \ln \frac{c_{(n,int)}}{c_{(n,ext)}} \tag{1}$$

V_{int} = Intracellular electric potential
V_{ext} = Extracellular electric potential
R = Gas constant = 1.98 cal/K-mol
T = Absolute Temperature
F = Faraday's constant = 96480 C/mol
z_n = Valence of ion

The mathematical expression of the Nernst equation plays a pivotal role in elucidating the equilibrium conditions that govern the ion distribution across cellular membranes. This equation quantitatively captures the interplay between the potential difference and concentration gradients within the cell's electrochemical environment. In simpler terms, it provides us with a precise formula to understand how ions, like potassium [**K+**] and sodium [**Na+**], distribute themselves inside and outside the cell.

To conceptualize this phenomenon, we can draw an analogy with an electrical circuit. Imaging the cell as a complex electrical circuit, where ions serve as electrical charges moving within this circuit. Specialized ion channels for potassium and sodium act as electrical switches, controlling the flow of these ions. McCulloch and Pitts' work in 1943, which portrayed neurons as binary logic gates, laid the cornerstone for the field of computational neuroscience, employing electrical circuits to mimic neural processes. (Pospíchal and Kvasnička [2015])

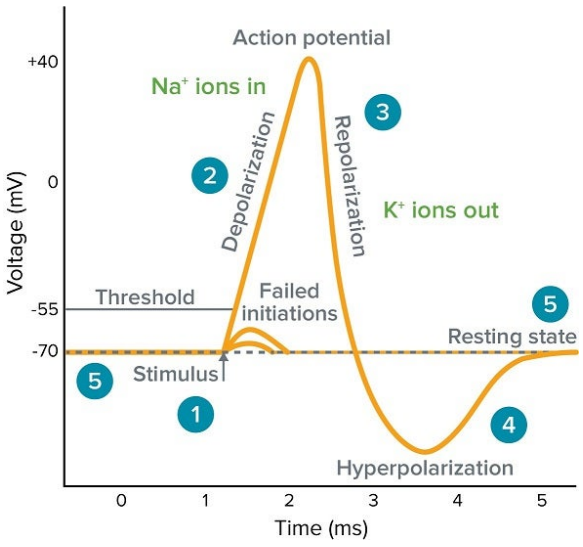


When the potassium channels open, they facilitate the movement of potassium ions (K+) out of the cell, effectively shifting the cell's internal charge in a negative direction. On the other hand, the opening of sodium channels allows sodium ions (Na+) to rush into the cell, inducing a positive change in the cell's internal charge. This controlled opening and closing of the ion channels parallels the operation of electrical components in a circuit, with ions and specialized channels taking the place of traditional wires and switches.

This dynamic interplay, oscillating between positive and negative states, underpins the establishment of the resting state, maintaining a delicate balance that is crucial for transmitting signals within the intricate nervous system. This analogy of the cell to an electrical circuit, albeit one regulated by ions and specialized channels, offers a comprehensible framework for grasping the fundamental biological phenomenon at play. Building upon this understanding, a computational model was developed to further explore and simulate these intricate processes. Details of those have been presented in the subsequent section, offering a practical application of the conceptual insights discussed here.

3.2 Modeling the neuronal membrane behaviour

Upon cellular excitation, a captivating sequence of events transpires, recognized as the **action potential**. This intricate modulation of the cell membrane's membrane potential involves a meticulously arranged sequence of ion channel activities, with a particular focus on those associated with sodium [**Na+**] and potassium [**K+**]. These two ion channels act as gatekeepers, regulating the flow of Na+ and K+ ions in and out of the cell. This delicate balance of ion movement is crucial for the cell to transmit signals effectively. This sequence of behaviour unfolds in discrete phases, akin to a precisely timed and structured sequence:



3.2.1 Depolarization Phase

This phase marks a crucial departure from the cell's resting membrane potential, typically around -70 mV. When an excitatory stimulus triggers an action potential, there is a rapid shift in the cell's electrical charge. The sodium channels swiftly open, leading to a surge of positively charged sodium ions flooding into the cell. This influx of sodium ions serves as the ignition of the action potential, resulting in the membrane potential shooting off to around +40 mV.

3.2.2 Repolarization Phase

Subsequent to the excitement of depolarization, the repolarization phase emerges as a restorative process. It can be likened to the cell resetting itself after the surge of excitement. This phase is marked by the opening of potassium channels, facilitating the efflux of potassium ions from the cell. As potassium ions exit, the cell's charge gradually becomes more negative, effectively restoring the membrane potential to its resting state of approximately -70 mV. Repolarization is essential for readying the cell for subsequent signaling event and ensuring the membrane potential is poised for future excitatory stimuli.

3.2.3 Hyperpolarization

It is a critical yet often overlooked phase, that extends the dynamics of the action potential further. In this phase, the cell's membrane potential becomes even more negative than its resting state, typically reaching values below -70 mV. Hyperpolarization primarily results from the delayed closure of potassium channels. These prolonged openings allow an ongoing efflux of potassium ions, driving the membrane potential to a hyper-polarized state.This phase holds paramount importance as it renders the cell refractory to additional stimulation for a brief interval. This refractory period is indispensable for safeguarding the fidelity of signal transmission and averting premature reversal of the action potential.

Numeric values underpin these transitions, with depolarization elevating the membrane potential to +40 mV, repolarization restoring it to -70 mV, and hyperpolarization dipping below -70 mV, creating a secure temporal framework for neural communication and computation.

3.3 Transition to Spiking Neural Networks

The utilization of action potential and membrane potential dynamics for information encoding lays the foundation for the seamless transition to Spiking Neural Networks (SNNs). (Wulfram Gerstner [2002]) In the biological realm, action potentials precisely encode information through variations in the membrane potential, emphasizing the significance of timing and voltage changes. These principles are seamlessly transposed into the computational domain, where SNNs harness the membrane potential to evoke discrete spikes. This enables the SNNs to efficiently process information, making them a powerful paradigm for cognitive computation tasks. This continuum of encoding principles highlights the striking convergence of biological inspiration and computational innovation within the domain of neural information processing.

In this work, we delve into the realm of SNNs, with the intent of modeling and deciphering the intricacies of information flow within the neural systems. SNNs draw their inspiration from the operational principles of biological neurons, encompassing the conceptualization of action potentials and ion transport. By means of simulating these biological processes within artificial neural networks, our objective is to acquire a profound understanding of the processing and propagation of information, faithfully mirroring biological systems. This endeavour bears immense promise, particularly in applications such as pattern recognition and decision making, where a biologically inspired approach demonstrates the potential to yield superior outcomes.

4 Related Work

In the study of information processing in SNNs, it is essential to draw insights from related fields that explore information transmission and signaling mechanisms. Significant research has been conducted on information transmission and signaling mechanisms related to the action potential. Moreover, extensive studies have explored the implications of ion channel fluctuations in various diseases.

4.1 Action potential signals in organisms

Degli Agosti's work on **"Touch-induced action potentials in Arabidopsis thaliana"**, (Degli Agosti [2014]) highlights the sensitivity of plants to mechanical stimuli, resulting in reversible depolarization resembling APs. Furthermore,

Awan et al. investigated the information transfer facilitated by single AP signals in their paper, **"Communication and Information Theory of Single Action Potential"**. (Awan et al. [2018]) This work quantifies information rates associated with AP signals and underscores their role in enhancing mutual information among neighbouring plant cells.

Kisinieriene et al. investigated the response of *Nitellopsis obtusa* cells to osmotic and saline stress in **"Modeling the Action Potential in Characeae *Nitellopsis obtusa*."**(Kisnieriene et al. [2019]) This study provides insights into the complex dynamics of APs and the underlying mechanisms, contributing to the understanding of plant excitability under different stress scenarios. Awan et al. further explores the properties of an electrochemical signal-based model in **"Impact of Multiple Action Potentials on Communication Properties of Plants"**.(Awan et al. [2019]) This work delves into how multiple AP signals influence mutual information and propagation speed in plant systems, shedding light on their potential applications in plant signaling.

All of these studies emphasize the significance of action potential in plants, particularly in the context of inter-cellular communication and under diverse conditions. Transitioning from this contemporary plant-focused exploration, we delve into the annals of neuroscience where early pioneers laid the foundational groundwork for comprehending action potentials in animal nervous systems.

Julius Bernstein's seminal **"Membrane Theory of Electrical Potentials"** (Seyfarth [2006]) was instrumental in unraveling the intricacies of the resting membrane potential and ion permeability, paving the way for subsequent investigations into the phenomenon of APs. Edgar Adrian's pioneering recordings of APs in **"The discharge of impulses in motor nerve fibres."** (Adrian ED [1929]) provided invaluable insights into the electrical activity of neurons, making a crucial juncture in AP research. John Zachary Young's instrumental studies on the giant axon of the squid in **"You've Got to Work on This Axon"** (Jones [2022]), shed light on the ionic basis of APs in excitable cells.

5 Modelling a Single Spiking Neuron

Inspired by the intricate mechanisms observed in the action potential studies of both plants and animals, our current work delves into the realm of artificial neuron models. These models replicate the remarkable neuronal functions exhibited in these diverse biological systems, allowing us to harness their principles for various applications.

5.1 Hodgkin-Huxley Model

The Hodgkin-Huxley (HH) describes the dynamics of ion movement through voltage-sensitive channels in neurons during an action potential. (Nelson and Rinzel [1998]) It is underpinned by a set of equations that encapsulate the behaviour of various ion currents:

5.1.1 Membrane Voltage Dynamics

The membrane potential experiences dynamic changes due to the interplay of multiple ion currents and external stimuli. The equation governing this is:

$$C_m \frac{dV}{dt} = I_{ext} - I_{Na} - I_K - I_L \tag{2}$$

where

C_m = Membrane Capacitance
$\frac{dV}{dt}$ = Change in membrane potential with respect to time
I_{ext} = External current applied to the neuron
I_{Na} = Sodium current
I_K = Potassium current
I_L = Leakage current

This equation describes how the membrane potential changes over time in response to external current and the currents flowing through sodium, potassium and leakage channels. The ion channels can be integrated with the theory of ion channel activation and inactivation, which explains the intricacies of neuronal signaling. It is essentially the regulatory function of the activation and inactivation sites that result in influx or efflux of ions across the ion channels.

5.1.2 Modelling a voltage-dependent Na+ channel

The voltage dependent Na+ channel in the model is thought to have 4 gates - three of them being activation gates, and the last one being an inactivation gate. The three activation gates are thought to be identical, dented by a common

gating variable, m. The inactivation gate is denoted by the gating variable, h. Thus the open probability of the entire gate is m^3h. If the conductance of the population of Na+ channels in which all Na+ channels are fully open is ${g_{Na}}^{max}$, then the conductance g_{Na} of the population in general conditions is:

$$g_{Na} = {g_{Na}}^{max} m^3 h \tag{3}$$

The gate kinetics of m variables and h variable may be described as,

$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m \tag{4}$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h \tag{5}$$

5.1.3 Modelling a voltage-dependent K+ channel

The voltage dependent K+ channel in the model is thought to have 4 gates - all of them being identical activation gates, denoted by the gating variable, n. Thus the open probability of the entire gate is n^4. If the conductance of the population of K+ channels in which all K+ channels are fully open is ${g_K}^{max}$, then the conductance g_K of the population in general conditions is:

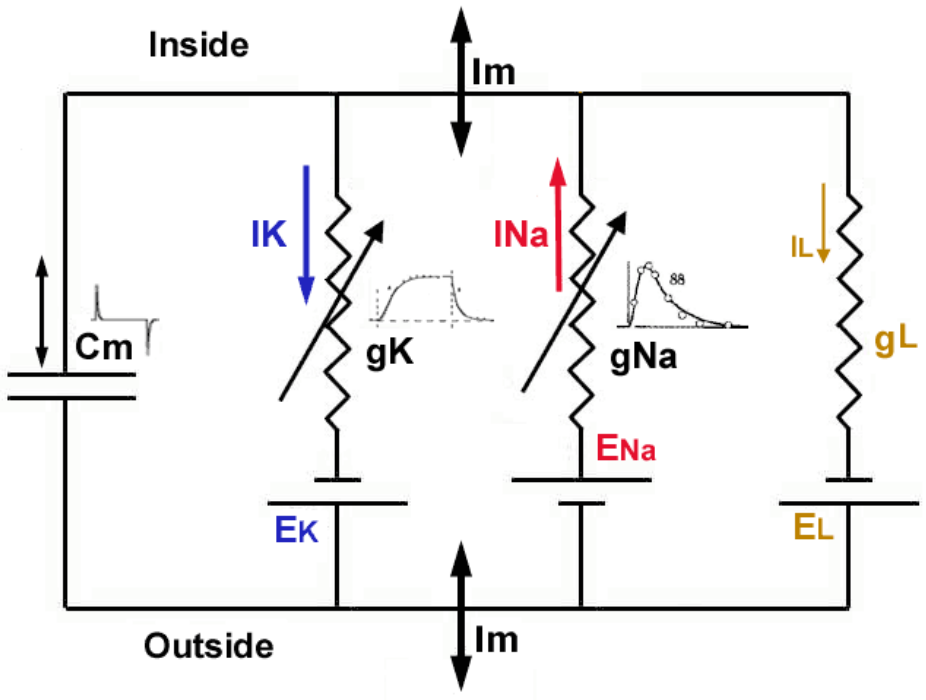
$$g_K = {g_K}^{max} n^4 \tag{6}$$

The gate kinetics of n variables may be described as,

$$\frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n \tag{7}$$

5.1.4 Modelling the voltage-independent leakage channels

Na+ and K+ channels were modelled as voltage dependent channels. All of the remaining channels, which are voltage-independent, can still be compressed to a single branch. This branch was thought to represent an ion channel that is equivalent to the sum total of all the voltage-independent ion channels in the membrane. This ion channel is called a leakage channel, since it is constantly open, allowing the leakage of current from the neuron. We thus have four branches in the circuit, as seen in the figure below.



Building upon the foundational equation, which governs the dynamic changes, the work was extended to derive the following refined expression for membrane voltage dynamics:

$$C_m \frac{dV_m}{dt} = -{g_{Na}}^{max} m^3 h(V_m - E_{Na}) - {g_K}^{max} n^4 (V_m - E_K) - g_l(V_m - E_l) + I_{ext} \tag{8}$$

5.2 Unsteady State Diffusion

Unsteady state diffusion (Fick [1855]) provides a lens through which we observe the dynamic movements of ions in the Hodgkin Huxley model. This theory describes the migration of particles in response to changing conditions over time. This phenomenon is eloquently encapsulated by Fick's second law:

$$\frac{\partial c}{\partial t} = D\nabla^2 c \tag{9}$$

where

D = Diffusion coefficient, governing the speed of particle migration
∇^2 = Laplacian operator, representing the spatial second derivatives
c = The concentration of particles
$\frac{\partial c}{\partial t}$ = represents the change in concentration with respect to time

This equation illustrates how the concentration of particles changes over time due to diffusion, which depends of the diffusion coefficient and the spatial concentration gradients.

The Hodgkin Huxley equations parallel the unsteady state diffusion equation as they involve time-dependent changes (*in voltage V and gating variables*) that impact ion flux. Just as Fick's second law describes how particle concentrations change over time in response to changing conditions, the Hodgkin Huxley model (AL [1952]) equations describe how ion currents change over time due to varying membrane potential. The ions act as diffusing particles, voltage serves as the changing environment condition, and conductance represents the time-dependent permeability. The analogy lies in the dynamic and responsive nature of ion movement in neurons during action potentials, which is in line with the principles of unsteady state diffusion, making it a valuable tool for understanding neural communication and computation.

5.3 Pseudo steady state approximation(PSSA) for unsteady state diffusion

The Pseudo steady state approximation (Suraishkumar [2014]) is a valuable concept that is employed in various scientific disciplines to simplify the analysis of complex systems. It is particularly useful when dealing with systems that consist of multiple interconnected processes occurring at different rates. The essence of PSSA lies in the assumption that certain processes in a system reach a state of approximate equilibrium much faster than others, allowing us to simplify the analysis. To understand the PSSA more comprehensively, let's break down its key elements:

5.3.1 Complex systems

Many natural and artificial systems involve numerous interconnected processes. These processes are often inter-wined in intricate ways, making it challenging to fully comprehend the system's behaviour by examining each component in isolation. These systems exhibit emergent behaviour, where the collective interactions of their components lead to outcomes that cannot be easily predicted by analyzing individual parts.

5.3.2 Fast and slow process

Within a complex system, various processes occur at different rated. Some processes may evolve rapidly, while others change more slowly. The fast processes tend to reach equilibrium or steady state conditions relatively quicker, often within a short time-span. Conversely, slow processes take longer to adopt or change significantly. This differentiation allows for a simplified analysis of complex systems, enabling a focus on aspects that matter most for the system's behaviour.

5.3.3 Approximate equilibrium

The core of the concept of the PSSA lies in the idea that the fast processes can be approximated as being in a state of equilibrium when compared to the slower processes. Equilibrium, in this context, means that the fast processes have stabilized, and their behaviour is relatively constant over the time scales of interest. This approximation is often valid because many complex systems exhibit a clear separation between fast and slow processes.

5.3.4 Focus on slow processes

With the assumption that the fast processes are stable, we can now concentrate on the aspects of the system that change more gradually over time. By doing so, we simplify the system's analysis, as we don't have to consider the fast processes' dynamic behaviour.

For example,

Let's consider a chemical reaction system where a complex reaction pathway leads to the formation of a desired product. Within this system, some intermediate reactions happen very quickly, reaching their equilibrium within milliseconds. However, the final step, which produces the desired product, progresses much more slowly and can take hours to reach equilibrium. To apply PSSA in this context, we assume that the fast intermediate reactions have already stabilized. This assumption simplifies our analysis of the slower, rate-determining step. By focusing on the latter, we can gain insights into the overall reaction without delving into the intricate details of every intermediate step.

5.4 Key features of gates regulating action potential

The knowledge about the working of the gates (activation and inactivation) is essential to appreciate how these gates contribute to the action potential. (Koch [1998])

5.4.1 Activation gate (m) of sodium channels

The activation gates, symbolized as **m**, open rapidly in response to a change in membrane potential, particularly when the membrane undergoes depolarization. As stated earlier, this flinging open of the activation gate permits the surge of sodium ions to rush into the neuron.

5.4.2 Inactivation gate (h) of sodium channels

The inactivation gate, symbolized as **h**, operates at a much slower pace. When the membrane depolarizes, it t oo responds by closing, but the closure is a more sluggish process. This temporal asymmetry between the activation and inactivation gates results in a brief window during which the sodium channels remain open. Yet, a critical characteristic of this inactivation gate emerges that it will not entertain the idea of reopening until the membrane potential reverts to its original resting state.

5.4.3 Activation gate (n) of potassium channels

The activation gates, symbolized as **n**, open in response to depolarization, allowing potassium ions to flow outward. However, there is a slight delay in their opening, such that they tend to open somewhere in between the opening and closing of the sodium channels. This synchronous event is no accident. The interplay between the decrease in sodium influx and the simultaneous increase in the potassium efflux accelerates the repolarization process.

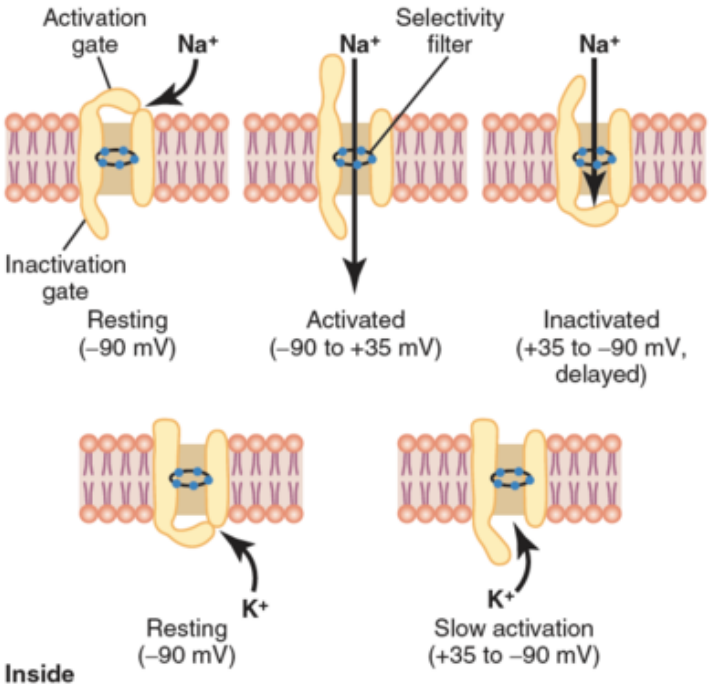


Table 1: Pseudo Steady state approximation

Part		
Name	Description	Relative rates
m	Na+ activation gate	Rapid
h	Na+ inactivation gate	Slow
n	K+ activation gate	Appropriate

5.5 FitzHugh Nagumo Model

While the Hodgkin-Huxley model is more realistic and biophysically sound, among the 4 differential equations, we can clearly see that there are non-linear equations too. Solving equations with 4 ion channels would be very cumbersome mathematically. It could be made simpler by applying suitable approximations to reduce the non-linearity. The approximation should be such that the model's dynamics is preserved. FitzHugh Nagumo neuron model is one of the prime examples to accomplish that. (Sherwood [2013])

5.5.1 PSSA on Hodgkin Huxley model

After having modelled the Hodgkin-Huxley model to an unsteady state diffusion process, we need to study the diffusion process that is giving rise to the action potential. Also, under the section **Key features of gates regulating action potential**, we looked at the activation and inactivation gates of sodium and potassium channels. We shall look at them, while keeping in mind the principles of pseudo steady-state approximation. It is crucial to note that the relative rates for the **m,h** and **n** ion channels are not the same. These variations give us the opportunity to eliminate some gating variables to obtain equations which are mathematically solvable.

The activation gate of sodium channels, **m** react rapidly upon depolarization. The pace at which it opens and responds to the changing conditions, it can be considered to have attained a pseudo steady state. It implies that changes happening within this short span, will not influence the action potential generation greatly. On the other hand, the closure of the inactivation gate of sodium channels, **h** is so slow that, for the interest at hand, this process can be taken as frozen.

Now, we have a two variable equation, against the four variable equation we had in case of Hodgkin Huxley model. We maintain the gating variable of primary interest (**n**) and its associated equations. For the gating variables (**m** and **h**), after the above considerations, we assume their time derivatives are effectively zero. This simplification can be achieved by setting:

$$\frac{dm}{dt} \approx 0, \quad \frac{dh}{dt} \approx 0 \tag{10}$$

$$0 = \alpha_m(V_m)(1-m) - \beta_m(V_m)m \tag{11}$$

$$0 = \alpha_h(V_m)(1-h) - \beta_h(V_m)h \tag{12}$$

$$m = \frac{\alpha_m(V_m)}{\alpha_m + \beta_m(V_m)} \qquad h = \frac{\alpha_h(V_m)}{\alpha_h + \beta_h(V_m)} \tag{13}$$

$$I_{Na} = g_{Na}.m^3.h.(V - E_{Na}) \approx g_{Na}.const.(V - E_{Na}) \tag{14}$$

On substituting the above approximated results back to the membrane voltage dynamics equation (8), we get:

$$C_m \frac{dV_m}{dt} = I_{ext} - {g_{Na}}^{max} const(V_m - E_{Na}) - {g_K}^{max} n^4 (V_m - E_K) - I_l \tag{15}$$

Now, let's transform the variables to dimensionless variables and make some approximations. Introduce the dimensionless voltage (*v*) and the dimensionless time (*t*)

$$v = \frac{V_m - E_{Na}}{E_K - E_{Na}} \tag{16}$$

$$w = \frac{n}{m} \tag{17}$$

This transformation brings v and w into dimensionless form relative to the sodium and potassium equilibrium potentials respectively. With these transformations and approximations, we arrive at the FitzHugh-Nagumo model equations. (Izhikevich [2006])

$$\frac{dv}{dt} = f(v) - w + I_m, \qquad where \qquad f(v) = v(a-v)(v-1) \tag{18}$$

$$\frac{dw}{dt} = bv - rw \tag{19}$$

The FitzHugh Nagumo(FN) model, often employed to describe excitable systems such as neurons, emerges as a simplified counterpart to the intricate Hodgkin-Huxley(HH) model. While the FN model may typically be derived through mathematical approximations, it's worth noting that the same outcome can be achieved by applying the principles of transport phenomena.

Through the pseudo steady-state approximation followed by necessary transformations and approximations, we arrive at a set of equations that capture the essential characteristics of action potentials. Specifically, the voltage variable, **v** is governed by a cubic function **f(v)** and the influence of **w** along with an external current term, **I**. Meanwhile, **w** plays a role in shaping the dynamics of the system, following a linear equation involving the parameters **b** and **r**.

This approach to obtaining the FitzHugh Nagumo model highlights the applicability of transport phenomenon concepts in understanding and simplifying the behaviour of excitable cells, providing an alternate perspective on its derivation.

5.5.2 Modelling Spiking Neural Networks

In the context of modelling spiking neural networks (SNNs), each neuron can be represented by a FitzHugh-Nagumo (FN) unit. The FN model, with its two variables, *v* and *w*, is a simplified but effective way to capture the essential dynamics of neuronal membrane potential and spiking behaviour. In this representation, *v* corresponds to the neuron's membrane's potential, and *w* typically accounts for the recovery variable or gating mechanisms. Hence, neurons in an SNN can be modelled as interconnected FN units. The interactions between these units, such as synaptic connections and coupling between neurons, can be described using appropriate mathematical functions.

6 Modelling Spiking Neural Networks: Bridging the gap

In our journey through the realm of computational neuroscience (Dayan and Abbott [2001]), we have traversed the intricacies of biological transport phenomena, from the Hodgkin-Huxley model's portrayal of unsteady state diffusion to the elegant pseudo-steady state approximations encapsulated within the FitzHugh-Nagumo model. Our exploration leads us to Spiking Neural Networks (SNNs), where we shall model the neural membranes, akin to FitzHugh Nagumo model, as the neurons of the network. This transition is a seamless continuation, as SNNs embody the very essence of biological plausibility.

6.1 Emulating Neurons with FitzHugh-Nagumo

The FitzHugh Nagumo model, serves as a bridge from the deterministic Hodgkin-Huxley model to the more abstract realm of SNNs. By employing the FitzHugh-Nagumo framework, we traverse the path of simplification, while retaining the essential elements of neural dynamics. This model offers a balance between the biological realism and computational tractability, allowing us to fathom the intricate spiking behaviour of neurons with mathematical elegance.

6.2 Spiking Neural Networks: Mimicking nature

While working with spiking neural networks, the action potentials will be referred to as spikes, which encode for the neuronal behaviour. SNNs offer a unique departure from the traditional artificial neural networks by embracing an event-driven paradigm, where spikes, the discrete units of neural activity, hold paramount significance. The pivotal feature of SNNs lies in its event-driven processing. (Eshraghian et al. [2021]) Neurons within SNNs fire spikes, analogous to the biological reality, making SNNs an ideal candidate for modelling neural systems.

6.3 The essence of Integration and firing

In SNNs, the neurons are often represented as integrate and fire units, encapsulating the essence of membrane potential buildup and spike initiation, mirroring the biophysical processes that transpire within biological neurons. (Burkitt [2006]) The thresholding mechanism, similar to Hodgkin-Huxley model, determines when a neuron emits a spike.

6.4 Towards transport phenomena with SNNs

As we delve deeper into the realm of SNNs, the transition from pseudo-steady state approximations to spiking networks offers a unique perspective on modeling transport phenomena. SNNs excel in capturing the temporal precision of information transfer, mirroring the movement of ions and molecules within biological systems. The synchronicity of spikes becomes a focal point, emphasizing the correlation between neural communication and the transfer of physical entities.

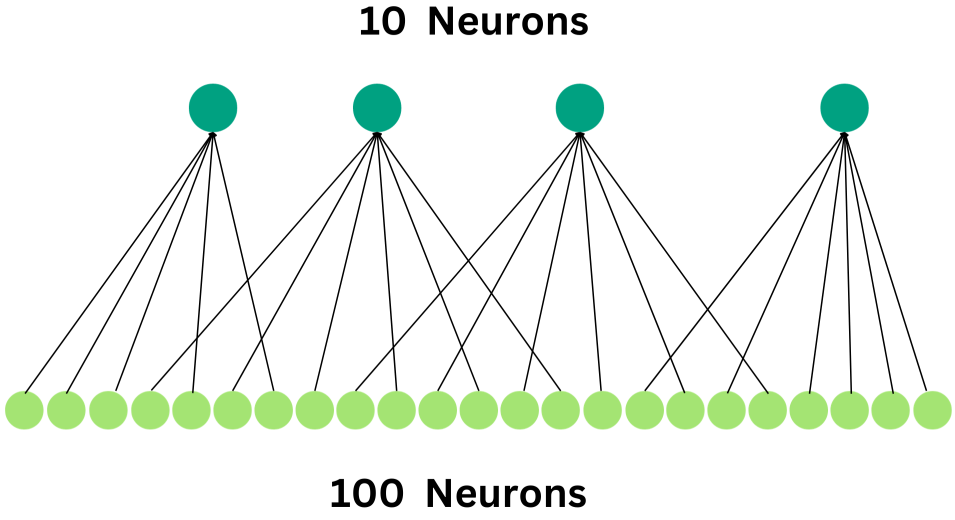
6.5 Applications and Beyond

The universality of SNNs is exhibited in a multifaceted array of applications, transcending cognitive computing, neuromorphic hardware, brain-computer interfaces, and advanced biological and medical research. These applications reinforce the versatile and biologically inspired nature of SNNs, further extending our understanding of transport phenomena and neural systems.

7 Development of Spiking Neural Networks

The development of an SNN begins with defining the model's parameters and simulation conditions. In this particular case, we set the parameters **a**, **b**, τ and the threshold for spike initiation. We also define simulation parameters, including the time step **dt**, the simulation duration **T** and the number of neurons in each layer.

The network architecture consists of two layers. The first layer contains 100 neurons, while the second layer has 10 neurons. We establish the hierarchical connectivity, where each neuron in the first layer is connected to one neuron in the second layer. Input currents are randomly assigned to neurons in the first layer.



The SNN simulations proceeds by numerically solving the FitzHugh-Nagumo equations at each time step. The network captures the firing dynamics of the neurons, with spikes recorded for each neuron. These spike times provide insights into the spiking behaviour and information flow within the network. The results are visualized by means of plotting the spiking timings, offering a graphical representation of how neurons in different layers communicate through spikes. This representation aids in understanding the information flow from one layer to another, guided by the hierarchical connectivity and input currents.

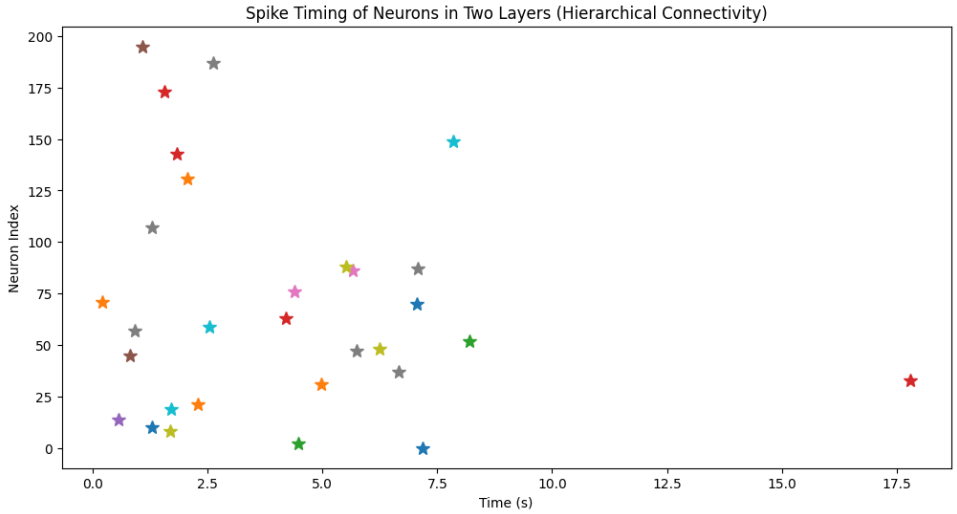
In a spiking neural network, to understand the hierarchical connectivity, we define a synaptic connectivity between the two layers. Each neuron in the first layer is connected to exactly one neuron in the second layer, creating a hierarchical

connectivity structure. The code enters a loop to simulate the spiking neural network for the specified number of time steps. It calculates the changes in the membrane potential, *dv* and recovery variable, *dw* based on the FitzHugh-Nagumo equations and input currents. This provides insights into how neurons in different layers interact and generate patterns over time. The code for the same is below.

Hierarchical Connectivity for 2 layer Spiking Neural Network

```
import numpy as np
import matplotlib.pyplot as plt

# Define FitzHugh-Nagumo neuron parameters
a = 0.7
b = 0.8
tau = 12.5
threshold = 1.0

# Simulation parameters
dt = 0.01
T = 100
steps = int(T / dt)
neurons_in_first_layer = 100  # Number of neurons in the first layer
neurons_in_second_layer = 10  # Number of neurons in the second layer

# Neuron variables
v = np.random.rand(2, neurons_in_first_layer) * 2 - 1  # Two layers
w = np.random.rand(2, neurons_in_first_layer) * 2 - 1
spike_times = [[] for _ in range(2 * neurons_in_first_layer)]

# Define hierarchical connectivity
synaptic_weights = np.zeros((neurons_in_first_layer, neurons_in_second_layer))

# Connect each neuron in the first layer to one neuron in the second layer
for i in range(neurons_in_first_layer):
    synaptic_weights[i, i % neurons_in_second_layer] = 1.0

# Input to Neurons
# For simplicity, random input currents to neurons
input_currents=np.random.uniform(0,0.5,neurons_in_first_layer)

# Simulate the SNN and record spike times
for step in range(steps):
    # Create a 2D array for input currents
    I_ext = np.zeros((2, neurons_in_first_layer))
    # Neurons in the first layer receive input currents
    I_ext[0, :] = input_currents

    dv = (v - (v ** 3) / 3 - w + I_ext) / tau
    dw = (v + a - b * w) / tau
    v += dv * dt
    w += dw * dt

    for layer in range(2):
        for i in range(neurons_in_first_layer):
            if v[layer, i] >= threshold:
                spike_times[layer*neurons_in_first_layer+i].append(step*dt)
                v[layer, i] = -1.0
                w[layer, i] += 0.5

# Visualize spike times
plt.figure(figsize=(12, 6))
```

```
for i in range(2 * neurons_in_first_layer):
    plt.plot(spike_times[i], [i] * len(spike_times[i]), '*', markersize=10)
plt.xlabel('Time (s)')
plt.ylabel('Neuron Index')
plt.title('Spike Timing of Neurons in Two Layers (Hierarchical Connectivity)')
plt.show()
```



7.1 Mass Flux analysis: Information flow

The mass flux matrix provides insights into how information or spikes are transferred between the neurons, particularly between different layers of the network. Each element in the matrix signifies the intensity of information flow between the neurons. Concentration profiles are derived from solving the diffusion equations and serve as proxies for spike distribution within a network layer. Steeper concentration profiles indicate rapid changes in spike concentration.

7.1.1 Diffusion through mass flux equations

In the context of neural networks and mass flux analysis, the diffusion equations can be adapted to describe the flow of information or spiking events from one layer of neurons to another.

Assumptions

- Consider two layers of neurons (Layer 1 and Layer 2)
- The neurons in Layer 1 send information (spikes) to the neurons in Layer 2
- We want to describe the steady state flow of spikes from Layer 1 to Layer 2

Variables

- $C_1(x,t)$: Concentration of spikes in Layer 1 at position x and time t
- $C_2(x,t)$: Concentration of spikes in Layer 2 at position x and time t
- J(x) : Mass flux of spikes from Layer 1 to Layer 2 at position x

The rate of change of spike concentration in Layer 1 is balanced by the mass flux of spikes to Layer 2

$$\frac{\partial C_1}{\partial t} = -\frac{\partial J}{\partial x} \tag{20}$$

Under steady state conditions, $\frac{\partial C_1}{\partial t} = 0$, and there is no change in concentration over time.

Fick's First Law

The mass flux of spikes from Layer 1 to Layer 2 is proportional to the concentration gradient between the layers.

$$J(x) = -D.\frac{\partial C_1}{\partial x} \tag{21}$$

D is the diffusion coefficient that characterizes the spread of spikes between the layers. Equate the terms in Fick's law and the mass conservation equation under steady-state conditions:

$$0 = -D.\frac{\partial {C_1}^2}{\partial x^2} \tag{22}$$

The solution obtained on double integrating the above relation is of the form,

$$C_1(x) = A * x + B \tag{23}$$

However, to satisfy the boundary conditions, we often consider the diffusion problems that are symmetric around a midpoint. In the context of spiking neural networks, we can consider:

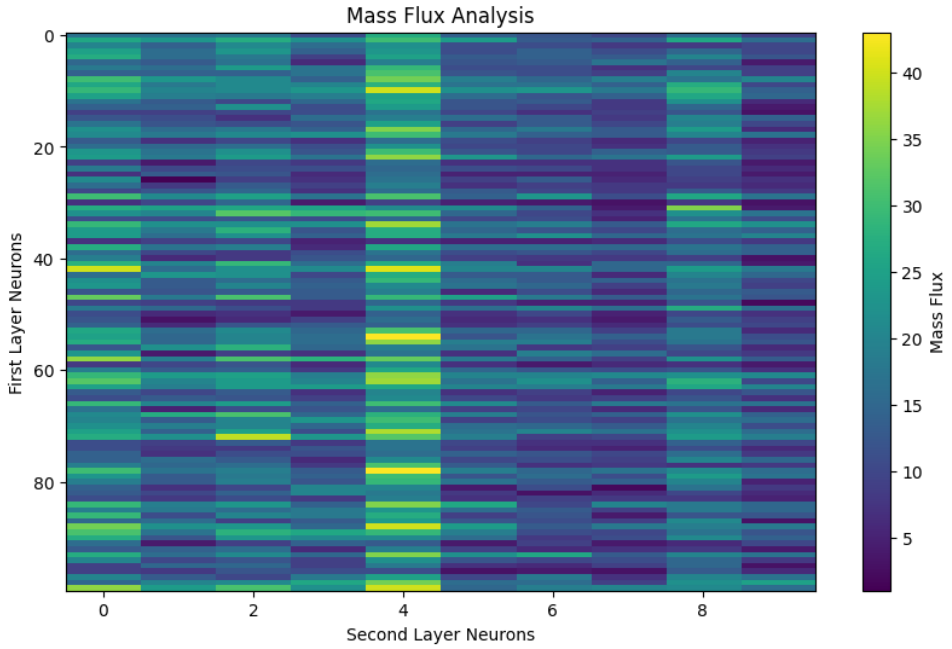
- **Initial Concentration** : The concentration at time $t = 0$, $C_1 = 0$, is often specified or can be taken as boundary condition.
- **Zero flux at boundary** : At the boundaries of the system, we can impose zero mass flux, $\frac{\partial C_1}{\partial x} = 0$. This condition ensures that spikes do not enter nor leave the system at the boundaries.

In this work, a python script has been coded to compute the mass flux matrix in a spiking neural network. Before diving into the mass flux analysis, the simulated spike times are recorded. In a real-world scenario, this section would be replaced by the actual network simulation and spike recording code. For demonstration purposes, the code generates random spike times for neurons in the first layer. The python code for the same is below.

Generate the mass flux matrix

```
import numpy as np
import matplotlib.pyplot as plt

# Define the network structure
neurons_in_first_layer = 100
neurons_in_second_layer = 10
synaptic_weights = np.zeros((neurons_in_first_layer,
neurons_in_second_layer))

# Simulate the network and record spike times
# Replace this section with your actual network
# simulation and spike recording code
spike_times = [[] for _ in range(neurons_in_first_layer)]

# Simulated spike times (for demonstration purposes)
for i in range(neurons_in_first_layer):
    spike_times[i] = np.sort(np.random.uniform(0, 100,
    size=np.random.randint(10, 30)))

# Mass flux analysis
mass_flux = np.zeros((neurons_in_first_layer, neurons_in_second_layer))

for i in range(neurons_in_first_layer):
    for j in range(neurons_in_second_layer):
        source_spike_times = spike_times[i]
        target_spike_times = spike_times[j]

        for source_time in source_spike_times:
            for target_time in target_spike_times:
                if source_time < target_time:
                    mass_flux[i, j] += 1

# Visualize the mass flux matrix
plt.figure(figsize=(10, 6))
plt.imshow(mass_flux, cmap='viridis', interpolation='none', aspect='auto')
plt.colorbar(label='Mass␣Flux')
plt.title('Mass␣Flux␣Analysis')
```

```
plt.xlabel('Second Layer Neurons')
plt.ylabel('First Layer Neurons')
plt.show()

# Print the mass flux matrix (for demonstration)
print('Mass Flux Matrix:')
print(mass_flux)
```



The code is used to calculate the mass flow analysis of a spiking neural network, which has been initialized with random spikes. It iterates through all pairs of neurons in the first and second layers, calculating the mass flux of information from the first layer to the second layer. If the spike time of the source neuron is earlier than the spike time of the target neuron, it signifies that information has been transmitted from the source to the target. For each such event, the mass flux value is incremented by 1. In this manner the information being passed is recorded in the mass flux matrix.

Solving the diffusion equation, we obtain concentration profiles in the network. For the given network, these profiles reveal significant patterns:

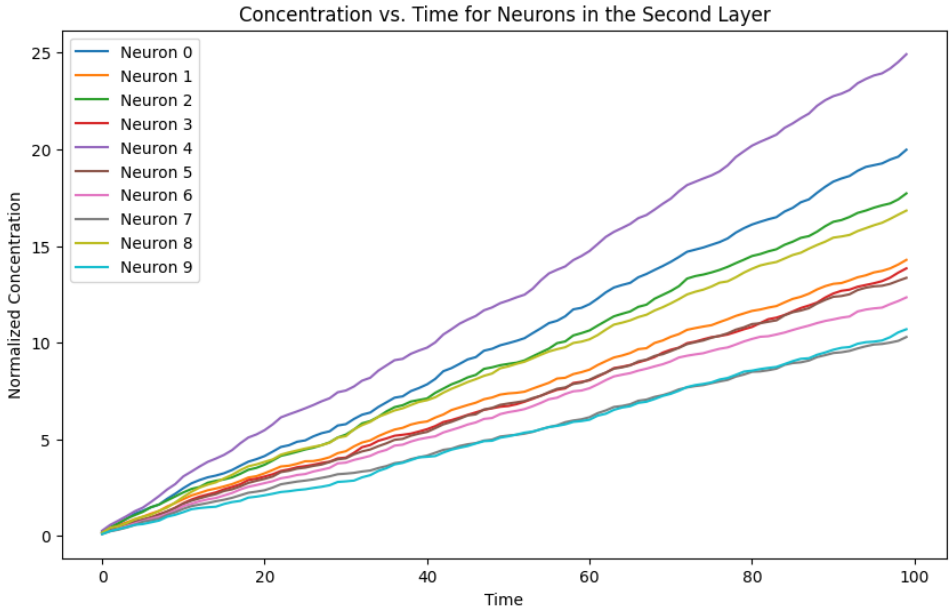
- In scenarios where Layer1 exhibits a high concentration of spike activity and low concentration gradients, this signifies the regions within the neural network where the spike accumulation is prevalent near the source. Think of this as analogous to a crowded area where people tend to gather around a center point. In our neural network context, this means that certain neurons in Layer1 are more active, and their accumulated spikes are centered around specific locations, creating localized activity hubs of information exchange.
- Conversely, when high concentration gradients are observed within Layer1, it indicates the regions where spikes rapidly propagate to Layer2. Imagine this as a steep slope where water flows quickly downstream. In our neural networks, this means that there are regions in Layer1 where spike activity has a significant influence of layer2, and these spikes swiftly transmit information or signals to the neurons in the subsequent layer.

High concentration near the source and subsequent decay implies that certain neurons in Layer 1 act as potent sources of spikes. Concentration profiles allow us to identify these sources and understand their impact on information flow. By interpreting the concentration profiles, we can formulate hypotheses about the network behaviour. The code for getting the concentration profile is below.

Obtaining the concentration profile

```
import numpy as np
import matplotlib.pyplot as plt

# Load your mass flux matrix (mass_flux) here
```

```
# Calculate the cumulative mass flux
cumulative_mass_flux = np.cumsum(mass_flux, axis=0)

# Normalize the cumulative mass flux
normalized_concentration = cumulative_mass_flux / mass_flux.shape[0]

# Plot concentration vs. time for each neuron in the second layer
plt.figure(figsize=(10, 6))
for neuron_index in range(normalized_concentration.shape[1]):
    plt.plot(np.arange(0, normalized_concentration.shape[0]),
    normalized_concentration[:, neuron_index],
    label=f'Neuron {neuron_index}')

plt.xlabel('Time')
plt.ylabel('Normalized Concentration')
plt.title('Concentration vs. Time for Neurons in the Second Layer')
plt.legend()
plt.show()
```



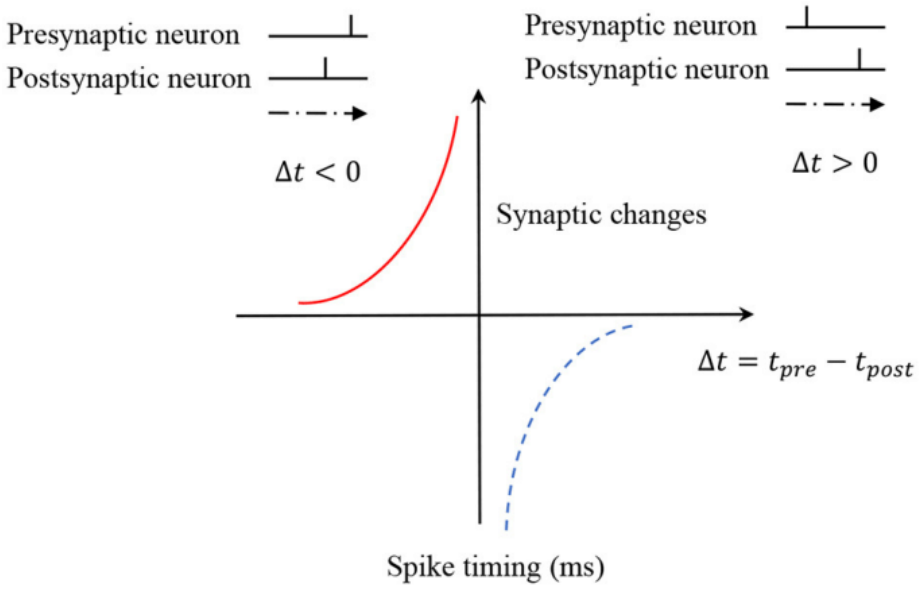
- Within Layer1, distinct clusters of neurons function as pivot points which regulate the flow of information into Layer2. When the information accumulates in certain neurons in Layer1, on propagation to Layer2, we observe the concentration profile to have a greater gradient profile for some neurons, while lesser gradient profile for some other neurons.
- Comparison between the mass flux analysis graph and the concentration profile, signifies the existence of specialized routes for transmitting signals between neurons.
- Analyzing the variation in concentration across individual neurons in Layer2 reveals that certain neurons are more active in processing and transmitting information compared to others. The parameters that regulate this, vary from problem to problem.
- The concentration profile within the second layer indirectly provides insights into the neurons in the first layer, that establish connections with those in the second layer. This interplay unveils the intricate network of synaptic connections and its role in shaping information flow throughout the Spiking Neural Network.

Matching mass flux analysis to concentration profiles in SNNs is a critical step in understanding the information flow within the network. Concentration profiles serve as a practical tool for interpreting complex mass flux patterns.

7.2 Validating Learning Rules in Spiking Neural Networks

Learning rules in Spiking Neural Networks are mathematical mechanisms or algorithms that govern how synaptic connections, *also known as synapses*, between the neurons are modified during the learning process. (Yi et al. [2023]) These rules determine how the strength of connections changes based on the timing and relative activity of pre-synaptic and post-synaptic neurons. Learning rules are crucial for adapting the network's behaviour, enabling it to perform specific tasks and learn from experience, similar to how synaptic plasticity operates in biological neural networks.

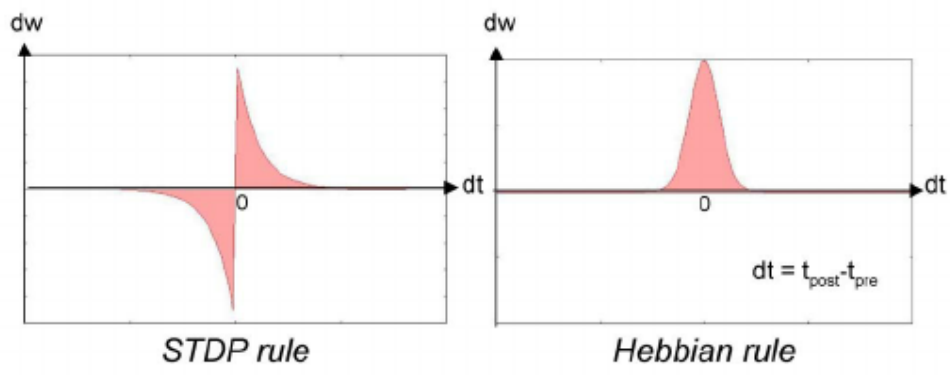
Spiking Timing Dependent Plasticity (STDP) is one of the most well known learning rules in SNNs. (Shouval et al. [2010]) It considers the timing of spikes in pre-synaptic and post-synaptic neurons to determine how the synaptic strength should be adjusted. When a pre-synaptic spike precedes a post-synaptic spike, the synapse is strengthened, and if the post-synaptic spike precedes the pre-synaptic spike, the synapse is weakened.



There are many other governing learning rules too. Some of them are **rate based learning**, **Hebbian learning** (Chakraverty et al. [2019]) and so on. These learning rules play a vital role in enabling the SNNs to learn from experience, respond to environmental changes, and improve their functionality over time. The choice of a learning rule depends on the specific task and the biological plausibility desired for the neural network. Below is a comparison of how the STDP learning rule and Hebbian learning rule would react to certain input.

The Hebbian learning rule is a fundamental concept in the realm of synaptic plasticity. It postulates that **neurons that fire together, wire together**. This rule suggests that when a pre-synaptic neuron consistently activates in conjunction with a post-synaptic neuron, the synaptic connection between them strengthens. This manifests as an increase in synaptic weight, ultimately enhancing the efficacy of signal transmission across the spiking neural network.

These learning rules are foundational concepts that underscore how the neural systems learn and adapt through the experience of correlated neural activity, contributing to our understanding of memory formation, pattern recognition and neural plasticity.



We have followed a novel modelling of Spiking Neural Networks using FitzHugh Nagumo neurons, where each neuron mimics a neural membrane, following the constraints of FitzHugh Nagumo model. We investigate the integration of these neuron units into an SNN and explore the extent to which they adhere to the principles of learning rules, be it STDP or Hebbian.

The SNN architecture is build upon the FitzHugh Nagumo neuron model. We consider a network of multiple neurons, each characterized by the FitzHugh Nagumo equations. The neurons are interconnected by synaptic weights that facilitate information transfer. A key feature of the network is the adaption of a hierarchical connectivity pattern, which mirrors certain structural aspects of biological neural networks.

The SNN simulation involves introducing external input currents to neurons, mimicking the neural network's exposure to sensory information. The simulation captures the evolution of membrane potentials and the application of learning rules. The code for the same is below.

Modelling SNN to portray STDP learning rule

```
import numpy as np
import matplotlib.pyplot as plt

# FitzHugh-Nagumo parameters
a = 0.7  # Excitability parameter
b = 0.8  # Inhibition parameter
tau = 12.5  # Time constant

# FitzHugh-Nagumo model
def fitzhugh_nagumo(v, w, I):
    dv = v - (v**3) / 3 - w + I
    dw = (v + a - b * w) / tau
    return dv, dw

# Simulation parameters
dt = 0.01  # Time step
T = 200  # Total simulation time
steps = int(T / dt)

# Neuron parameters
neurons = 10

# Neuron variables
v = np.random.rand(neurons) * 2 - 1
w = np.random.rand(neurons) * 2 - 1

# Synaptic weights and plasticity parameters
synaptic_weights = np.random.rand(neurons, neurons)
synaptic_strength = 0.2
learning_rate = 0.005

# Lists to store data
v_values = []
time_values = []

plt.figure(figsize=(12, 6))
plt.imshow(synaptic_weights, cmap='plasma', aspect='auto', origin='lower')
plt.colorbar()
plt.xlabel('Post-synaptic Neuron')
plt.ylabel('Pre-synaptic Neuron')
plt.title('Synaptic Weights')
plt.show()

# Simulate the SNN with STDP
for step in range(steps):
```

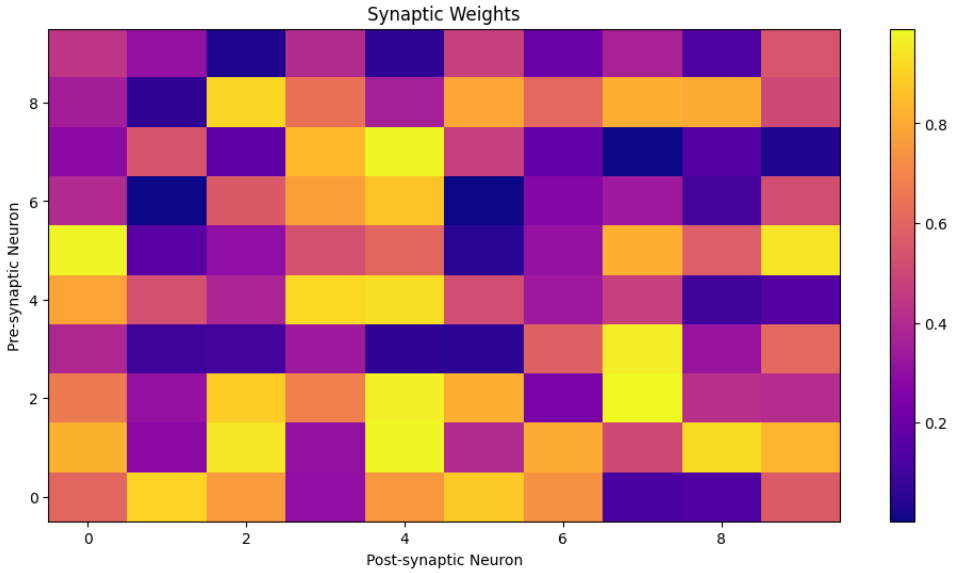
```
        time = step * dt
        time_values.append(time)
        v_values.append(v.copy())

        # Generate random spike events for input
        input_spikes = np.random.rand(neurons) < 0.01  # Random spikes

        # Apply external input to neurons based on spikes
        I_ext = np.dot(synaptic_weights, input_spikes) * synaptic_strength
        dv, dw = fitzhugh_nagumo(v, w, I_ext)
        v += dv * dt
        w += dw * dt

        # Update synaptic weights using STDP
        for pre_neuron in range(neurons):
            for post_neuron in range(neurons):
                if input_spikes[pre_neuron] and post_neuron != pre_neuron:
                    if synaptic_weights[pre_neuron, post_neuron] > 0:
                        synaptic_weights[pre_neuron, post_neuron]
                        -= learning_rate
                    elif synaptic_weights[pre_neuron, post_neuron] < 0:
                        synaptic_weights[pre_neuron, post_neuron]
                        += learning_rate

    # Plot the membrane potentials of neurons and synaptic weights
    v_values = np.array(v_values).T
    plt.figure(figsize=(12, 6))
    for i in range(neurons):
        plt.plot(time_values, v_values[i], label=f'Neuron {i}')
    plt.xlabel('Time')
    plt.ylabel('Membrane Potential')
    plt.title('SNN with STDP Simulation')
    plt.legend(loc='upper right')

    plt.figure(figsize=(12, 6))
    plt.imshow(synaptic_weights, cmap='plasma', aspect='auto', origin='lower')
    plt.colorbar()
    plt.xlabel('Post-synaptic Neuron')
    plt.ylabel('Pre-synaptic Neuron')
    plt.title('Synaptic Weights')
    plt.show()
```

The code encloses the following details.

A two layer spiking neural network was defined, where we have 10 neurons in the first layer, followed by 10 neurons in the second layer. The two layers have Hierarchical connectivity. Each pair of neurons connected have a synaptic weight associated with it. Initially all the neuronal pair connections have been assigned a random value. In real time scenarios, either a random value is assigned or a pre-trained value obtained from some real-time dataset is assigned to the synaptic connections.

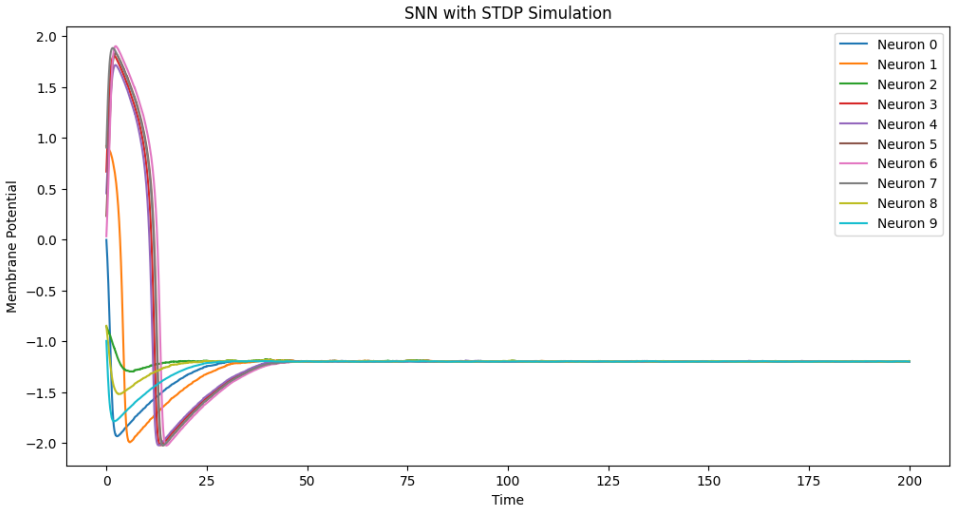
Then we simulate the spiking neural network, which is a network of neurons, each following the FitzHugh-Nagumo model. Inside the loop, we keep a track of the membrane potential of neurons at each time step. At the same time, we are generating input spike events for each neuron. The external input is computed for each neuron based on the synaptic weights and the input spikes. This mimics the effect of spikes from connected neurons on the membrane potential.

In order to verify the learning rules, we first create an array of initial state of synaptic weights in the spiking neural network before any simulation. Initially, since all the synaptic weights are random, no specific pattern or connection between the neurons is observed. This plot serves as the starting point for simulation.



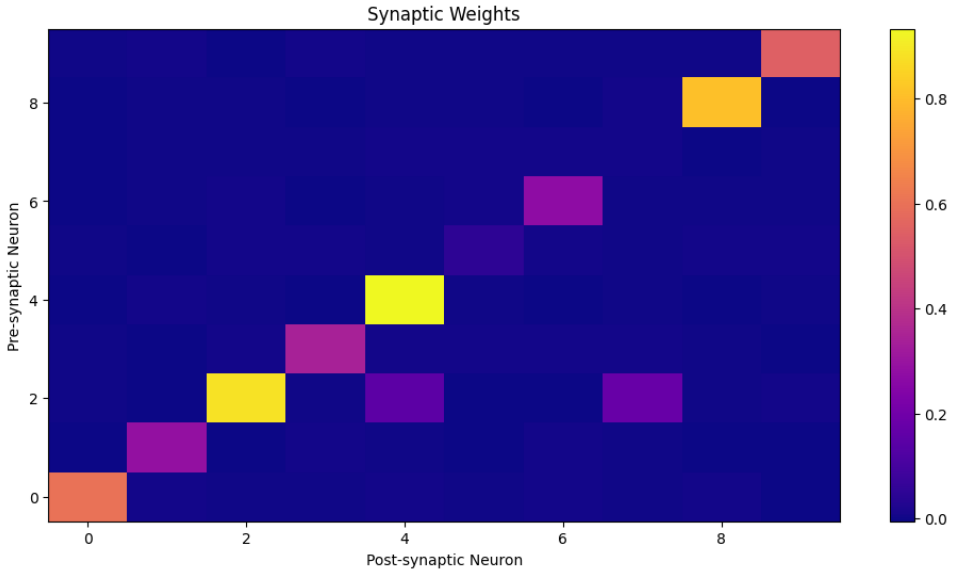
For the purpose of verifying the learning rule, I will be passing randomly generated inputs and tuning the model as per the FitzHugh Nagumo neural membrane directions. The training part involves updating the membrane potential against time. Whenever the membrane potential crosses the threshold value, and action potential is generated and we need to reset the potential.

The so obtained plot shows the evolution of the membrane potentials of individual neurons in the spiking neural network over time during the simulation. Each line in the plot represents the membrane potential of a specific neuron. As the simulation progresses, the lines depicting the membrane potentials of neurons may fluctuate. The fluctuations indicate the dynamic behaviour of individual neurons in response to the applied input and synaptic plasticity.



Now, we have run the simulation of the Spiking Neural network for several inputs and updated the synaptic weights as desired. We now plot the synaptic weights after updating the weights. The color intensity of each cell in the matrix represents the strength of the synaptic connection between the pre-synaptic and post-synaptic neurons.

In the plot, we observe that the correlation between the pre-synaptic and post-synaptic spikes have linearly strengthened. This reflects that the neural network has adapted over time, thereby proving the concept of spike-time-dependent-plasticity. In the given example we took the simple example of Hierarchical connectivity, hence we observed a linear trend in learning. However, when it comes to real time scenarios, this linearity need not be true. Rather, it is very rare to observe a linear relation because, it is very difficult to have a linear map between the input characteristics and the output map. Also, the connectivity between the different layers of spiking neural networks is seldom hierarchical. The above developed model is just a simple case. The same can be generalized to any multi-layer network too.



8 Results and Conclusion

The results of this study provide compelling evidence that our FitzHugh-Nagumo based spiking neural network successfully emulates the essential characteristics of spike time dependent plasticity (STDP) based learning. As observed in our simulations, the synaptic weights within the network dynamically adjust in direct response to the relative timing of spikes between pre- and post- synaptic neurons. This phenomenon unequivocally underscores the network's capacity for information processing and adaptive learning, mirroring the principles of synaptic plasticity observed in biological neural networks.

The results also demonstrate the robustness of our biophysical interpretation of the Hodgkin-Huxley model as an unsteady-state approximation. Intriguingly, this biophysical interpretation yielded results consistent with our mathematically derived models. The alignment between our mathematical and biophysical approaches suggests that our SNN encapsulates both the mathematical principles and the biological underpinnings of synaptic plasticity. This observation provides a cohesive and comprehensive understanding of learning and information transfer in neural systems and sets the stage for development of advanced neuromorphic systems.

9 Future Directions

The present study marks a significant stride in the exploration and modelling of neural information processing. While our FitzHugh-Nagumo based spiking neural network has successfully emulated the STDP, there remain many avenues for further research and advancement.

9.1 Biophysical Realism

An interesting prospect for future work is to enhance the modelling, by incorporating more complex neuron models into the network while retaining the STDP learning rule. This approach can lead to a deeper understanding of neural processing and might help uncover additional mechanisms that play pivotal roles in learning and memory.

9.2 Multi-modal Integration

The integration of multiple sensory modalities in SNNs represents a fascinating path for future research. Emulating how the brain integrates and processes information from diverse sensory inputs can pave the way for building advanced neuromorphic systems with real-world applicability, such as in robotics or biomedical technologies.

9.3 Neuromorphic Hardware

In the near future, there is a potential to translate the insights gained from our modelling efforts into practical, energy-efficient neuromorphic hardware systems. These systems could revolutionize fields ranging from artificial intelligence to cognitive neuroscience and offer solutions to complex problems.

10 Acknowledgement

I would like to express my gratitude to Prof. G K Suraishkumar for providing me with this invaluable opportunity to undertake this project as an integral component of his esteemed, **Transport Phenomena on Biological Systems** course. This exercise has not only deepened my understanding of fundamental biological processes but has also opened new horizons in comprehending Spiking Neural Networks (SNNs) from the perspective of biophysical modelling. Prof. Suraishkumar's guidance and mentor-ship has been instrumental in shaping the success of this project, for which I am sincerely thankful.

References

G K Suraishkumar. *Continuum Analysis of Biological Systems*, volume 5. 01 2014. ISBN 978-3-642-54467-5. doi:10.1007/978-3-642-54468-2.

Jiří Pospíchal and Vladimír Kvasnička. 70th anniversary of publication: Warren mcculloch & walter pitts - a logical calculus of the ideas immanent in nervous activity. In Peter Sinčák, Pitoyo Hartono, Mária Virčíková, Ján Vaščák, and Rudolf Jakša, editors, *Emergent Trends in Robotics and Intelligent Systems*, pages 1–10, Cham, 2015. Springer International Publishing. ISBN 978-3-319-10783-7.

Werner M. Kistler Wulfram Gerstner. Spiking neuron models : single neurons, populations, plasticity. page 480. Cambridge University Press, Cambridge, U.K., 2002, 2002.

Robert Degli Agosti. Touch-induced action potentials in arabidopsis thaliana. *Archives Des Sciences Journal*, 67: 125–138, 12 2014.

Hamdan Awan, Raviraj Adve, Nigel Wallbridge, Carrol Plummer, and Andrew Eckford. Communication and information theory of single action potential signals in plants. *IEEE Transactions on NanoBioscience*, PP:1–1, 11 2018. doi:10.1109/TNB.2018.2880924.

Vilma Kisnieriene, Indre Lapeikaite, Vilmantas Pupkis, and Mary Jane Beilby. Modeling the action potential in characeae nitellopsis obtusa: Effect of saline stress. *Frontiers in Plant Science*, 10, 2019. ISSN 1664-462X. doi:10.3389/fpls.2019.00082. URL `https://www.frontiersin.org/articles/10.3389/fpls.2019.00082`.

Hamdan Awan, Raviraj Adve, Nigel Wallbridge, Carrol Plummer, and Andrew Eckford. Impact of multiple action potentials on communication properties of plants. pages 1–6, 05 2019. doi:10.1109/ICC.2019.8761237.

Ernst-August Seyfarth. Julius bernstein (1839-1917): Pioneer neurobiologist and biophysicist. *Biological cybernetics*, 94:2–8, 02 2006. doi:10.1007/s00422-005-0031-y.

Bronk DW Adrian ED. The discharge of impulses in motor nerve fibres. part ii. the frequency of discharge in reflex and voluntary contractions. j physiol. *The Physiological Society*, 1929.

Kathryn Maxson Jones. ?you've got to work on this axon?: J. z. young and squid giant axon preparations in 20th-century neurobiology. *Berichte Zur Wissenschaftsgeschichte*, 45(3):317–331, 2022. doi:10.1002/bewi.202200021.

Mark Nelson and John Rinzel. *The Hodgkin—Huxley Model*, pages 29–49. Springer New York, New York, NY, 1998. ISBN 978-1-4612-1634-6. doi:10.1007/978-1-4612-1634-6_4. URL `https://doi.org/10.1007/978-1-4612-1634-6_4`.

Adolf Fick. Ueber Diffusion. *Annalen der Physik*, 170(1):59–86, January 1855. doi:10.1002/andp.18551700105.

Hodgkin AL. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, August 1952. doi:10.1113/jphysiol.1952.sp004764.

Christof Koch. *Biophysics of Computation: Information Processing in Single Neurons* . Oxford University Press, 11 1998. ISBN 9780195104912. doi:10.1093/oso/9780195104912.001.0001. URL `https://doi.org/10.1093/oso/9780195104912.001.0001`.

William Erik Sherwood. *FitzHugh–Nagumo Model*, pages 1–11. Springer New York, New York, NY, 2013. ISBN 978-1-4614-7320-6. doi:10.1007/978-1-4614-7320-6_147-1. URL `https://doi.org/10.1007/978-1-4614-7320-6_147-1`.

Eugene M. Izhikevich. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press, 07 2006. ISBN 9780262276078. doi:10.7551/mitpress/2526.001.0001. URL https://doi.org/10.7551/mitpress/2526.001.0001.

P. Dayan and L.F. Abbott. *Theoretical Neuroscience*. MIT Press, Cambridge, MA, 2001.

Jason Kamran Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. Training spiking neural networks using lessons from deep learning. *CoRR*, abs/2109.12894, 2021. URL https://arxiv.org/abs/2109.12894.

Anthony Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95:1–19, 08 2006. doi:10.1007/s00422-006-0068-6.

Zexiang Yi, Jing Lian, Qidong Liu, Hegui Zhu, Dong Liang, and Jizhao Liu. Learning rules in spiking neural networks: A survey. *Neurocomputing*, 531:163–179, 2023. ISSN 0925-2312. doi:https://doi.org/10.1016/j.neucom.2023.02.026. URL https://www.sciencedirect.com/science/article/pii/S0925231223001662.

Harel Shouval, Samuel Wang, and Gayle Wittenberg. Spike timing dependent plasticity: A consequence of more fundamental learning rules. *Frontiers in Computational Neuroscience*, 2010. ISSN 1662-5188. doi:10.3389/fncom.2010.00019. URL https://www.frontiersin.org/articles/10.3389/fncom.2010.00019.

Snehashish Chakraverty, Deepti Moyi Sahoo, and Nisha Rani Mahato. *Hebbian Learning Rule*, pages 175–182. Springer Singapore, Singapore, 2019. ISBN 978-981-13-7430-2. doi:10.1007/978-981-13-7430-2_12. URL https://doi.org/10.1007/978-981-13-7430-2_12.