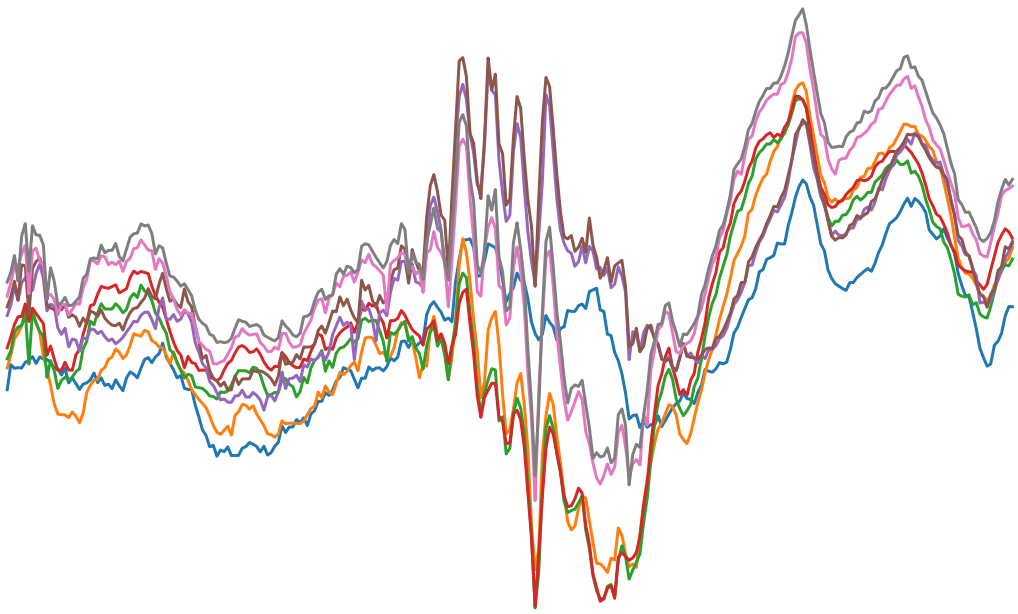
UNIVERSITAT INTERNACIONAL DE CATALUNYA

Leveraging Neuromorphic Computing for Low-Power Detection of High Frequency Oscillations in the Hippocampus

Marcos Oriol Pagonabarraga



Bachelors of Bioengineering

Supervisors: Jose Mateus (i3s), Miguel Ángel Mateos (UIC)

July 2, 2024

Abstract

When dealing with real time signal processing, microprocessors struggle to cope with the slow and high energy consuming artificial neural networks. Inspired by nature, Neuromorphic computation (NC) replicates the neural structures and processing methods of the human brain, achieving faster computing times and lower power requirements, showing great promise for human-implanted devices and brain-machine interfaces, providing a revolutionary approach to interfacing technology with biological systems.

Within Neurodegenerative Diseases (NDs), sharp wave ripples (SWRs), an oscillation pattern originating in the hippocampus, have been recognized as vital for memory consolidation. The hippocampus converts short-term memories into long-term ones, and SWRs are key in this process. Accurately detecting SWRs can play an important role in advancing on the understanding of how our brain processes information. Futhermore, pathological SWRs may be used as a biomarker to assess the onset of memory loss triggered by some NDs.

This bachelor's thesis presents a method to develop a spiking neural network[1] (SNN), which is able to run on Neuromorphic Hardware, for detecting ripple activity in the local field potentials (LFPs) of mice, showcasing the potential of NC in real-time implanted devices for biosignals.

[1] **A Spiking Neural Network** is a type of artificial neural network that mimics the way biological neurons communicate. SNNs use discrete events called "spikes" to transmit information, making them capable of capturing temporal dynamics of neural activity, similar to the brain.

Acknowledgements

I am very grateful for having had the opportunity of doing this work, and all the support, materials, attention, and formation I have been provided with. Throughout these 4 months I have developed myself a lot and learned from an amazing team. Getting involved in the field of neuroengineering was always a dream I had, and what better way than with such a project.

Special thanks to Jose Mateus, who has supervised my work and provided valuable insights of which strategy and procedure should be used, as well as motivation to follow my intuition. Since the start he showed very comprehensive, and informal relationship was built. Also, to Paulo Aguiar, who was always available to give advice and discuss any concerns or doubts. He showed interest in the project and ensured that I would enjoy my stance.

Thank you a lot also to Miguel Ángel, who supervised my work from Spain. He was fully comprehensive and adapted to my timings, letting me give the most of me. He also was available any time I would request.

Lastly, thank you to NCN group, it was a pleasure to be part of the team in this stance. The environment was very friendly, and I have learned a lot from you. You were always willing to help when I had a problem and always made efforts to make my stance the most pleasant possible.

"It always seems impossible until it's done."

Nelson Mandela

Contents

List of Figures

Abbreviations

AD	Altheimer's Disease
AI	Artificial Intelligence
ANN	Artificial Neural Network
CA_x	Cornus Amminois (number "x") Subregion
CNN	Convolutional Neural Network
CNS	Central Nervous System
DBS	Deep Brain Stimulation
DG	Dentate Gyrus
DNN	Deep Nerual Network
EC	Entorinhal Cortex
EC-B	Electronic cell body
E-NEURON	electronic neuron
E-SYNAPSE	electronic neuron
HFO	High Frequency Oscillation
HRS	High Resistance State
LFP	Local Field Potential
LIF	Leaky Integrate-and-Fire
LRS	Low Resistance State
MEA	Micro Electrode Array
NC	Neuromorpic Computation
ND	Neurodegenerative Disease
PD	Parkinson's Disease
SF	Sampling Frequency
SNN	Spiking Neural Network
STDP	Spike Timing Dependent Plasticity
SWR	Sharp Wave Ripple
TLE	Temporal Lobe Epilepsy

Chapter 1

Introduction

The human brain, with its intricate web of neurons and synapses, remains one of the most complex and enigmatic systems in nature. Understanding its fundamental workings holds the key to unraveling mysteries such as cognition and memory formation, which represent one of the main downsides of Neurodegenerative Disorders. Among the brain's many regions, the hippocampus stands out as a focal point for memory encoding and retrieval, spatial navigation, and emotional processing (2; 3).

Within the hippocampus, high-frequency oscillations (HFOs), such as sharp wave ripples (SWRs) have emerged as crucial neural events implicated in various cognitive functions and dysfunctions (3). These oscillations, often ranging from 100 to 250 Hz, play a pivotal role in information processing and communication within neural circuits. Detecting and deciphering these HFOs provide invaluable insights into the underlying mechanisms of learning, memory, and pathologies such as epilepsy and Altheimer's Disease (AD) (1; 4; 5).

However, capturing and analysing HFOs pose significant challenges, particularly concerning power consumption and computational efficiency (6). Traditional computing approaches struggle to cope with the complexity and real-time demands of neural data processing, especially when targeting low-power applications (7). These applications, among others, include closed-loop systems (8), which typically involve the integration of sensing and stimulation components that can monitor neural activity in real-time and deliver therapeutic interventions accordingly, therefore becoming independent from outer machines and more comfortable for the patient. In Parkinson's Disease (PD) closed loop deep brain stimulation (DBS) systems are being developed willing to monitor neural activity and adjust stimulation parameters in response to variations in neural activity (9; 10; 11). Of course, these systems require minimum energy consumption to minimize the number of recharging interventions.

Neuromorphic computing (NC), a groundbreaking approach inspired by the brain's own architecture and functionality, emulate the parallelism, plasticity, and energy efficiency of biological brains, offering promising opportunities for real-time and low-power neural data analysis (12; 13). By harnessing the principles of Spiking Neural Networks (SNNs) and event-driven computation[1],

[1] **Event-driven computation** is a computational paradigm where the execution of tasks or processes is triggered

neuromorphic platforms hold immense promise for revolutionizing the way we study and understand brain dynamics (14; 15).

Artificial intelligence has supposed a technological explosion in the last ten years. Current AI are performing better than humans, and the development possibilities it offers are uncountable. The relentless improvement of algorithms has been accompanied by a notable increase in energy consumption, which at first was overlooked. However, its widespread implementation is starting to raise concerns about sustainability and environmental impact (16). Efforts underway to develop energy-efficient AI algorithms and hardware, are one of the causes pushing the advances of NC. Other pushing force of this research branch includes advancements in brain computer interfaces due to its further resemblance in the way of processing information (14) (which is more similar to neurons). Furthermore, these platforms are facilitating breakthroughs in understanding the complex dynamics of neural systems. By simulating the behaviour of biological neurons and synapses, fundamental principles of brain function, such as learning and memory can be studied (14; 17).

SNNs and neuromorphic computing have their origins many years ago, but it was near 2000 when it got accelerated due to the explosion of semiconductors industry and due to notable contributions from researchers such as Eugene Izhikevich, who introduced a revolutionary concept of neuron model in 2003. Up to date, NC have a solid background and implementation in diverse low-power engineering tasks, such as image classification systems (12). Namely embedded cameras in cars, street traffic, security systems... (18). Nonetheless, very few works about closed loop systems for brain applications have been reported (19). Enabling low power and faster computation on implanted systems in the brain holds immense promise for revolutionizing health care, human-machine interaction, and seamless integration of advanced technologies within the human brain (20; 8).

1.1 Motivation

In light of the benefits offered by NC in real-time processing tasks, this work is commited to implement it in a neuroengineering landscape. In this field, there is a growing interest to interface electronical devices with the brain. After all, being able to understand brain signals, despite the ethical concerns it may raise, holds uncountable applications to improve people lives. Thereby, seeing that NC perfectly bridges the gap in applications where energy consumption is a limitation, such as closed loop systems in the body, learning to implement it offers a promising avenue for innovation.

1.2 Objectives

The main goals of this work are:

---

by events rather than being based on a fixed, predetermined schedule. In this approach, tasks are initiated in response to specific occurrences or stimuli, referred to as events.

- Build a Spiking Neural Network capable of detecting SWR events.
- Deploy the trained SNN to neuromorphic hardware and see the advantages of neuromorphic computation.

Chapter 2

Neuromorphic Computation

2.1 Introduction

Neuromorphic computation (NC) could be described as a "computing paradigm designed to mimic the structure and function of the human brain". It involves the use of hardware and software systems that replicate the neural architecture and operational principles of biological neural networks.

The human brain is posed as the most intelligent and efficient machine. It is built by billions of neurons connected between them on an average of more than ten thousand synapses per neuron (21). Its outstanding intelligence and efficient computation have inspired researchers to develop the working principles of artificial intelligence (AI), namely artificial neural networks (ANNs), where intelligence is achieved by small computing nodes forming a network capable to learn patterns. Following inspiration by nature, NC and SNNs have raised as another approach to recreate intelligence through a deeper emulation of the biological neural system (22). At all ends, ANNs achieve intelligence as a simulation of a neural network allocating values in memory and performing operations between them, which is not very efficient and needs more energy to perform simple tasks. In contrast, human brain has physical nodes with real connections and weights which make it way more efficient than our digital technology in spatial and visual tasks (a 10 million factor approximately (13)). That said, the main advantage of neural networks running in NC against traditional deep learning frameworks is that it aims to build the neural structure physically. That is, designing electronic neurons and synapses at a microscopic scale, which are connected as a circuit recreating a neural network. Essentially, NC aims to reinvent computing machines making them fundamentally different from current digital computers.

2.1.1 Strengths and weaknesses

Digital computers work with algorithms and calculations performed by arithmetic circuits[1] Moreover, information is encoded in binary values by analogue-to-digital converters, working with zeros and ones. These values were first computed serialized (like a queue of values) controlled by

[1] **Arithmetic circuits** are components in computers that perform mathematical operations, like addition or multiplication, using binary logic and electrical circuits. They're the building blocks for calculations in digital systems.

timesteps set by an oscillating crystal inside the processing unit. Each timestep was able to perform a single operation. With the advances in the field, more parallel computation has been pursued and implemented. However, it supposes a hurdle and limits its capabilities. These limitations observed in the working nature of digital computation are overcomed in NC. Physical electronic nodes and synapses allow to process information in a decentralized manner, so the computation is much more parallelized. Moreover, a major part of the process happens by analogous electrical circuit operations which increase notably the speed compared with time step guided operations. Furthermore, as a consequence of parallel operation and improved efficiency in performing operations, much less energy is required to achieve the same computation (100 times less aprox. (23; 24))

Of course, recreating the physical circuits to mimic brain neural network architecture is much more complicated than building a silico microprocessor unit. In addition, NC cannot produce as small processing units as in silico processors. Currently, transistors are in the nanometric scale whereas electronic neurons are in the range of micrometres, limiting their capability to form so complex processing systems as we see in silico computation, because they would need much more space. Considering the abovementioned features of NC, it can be understood why this computation paradigm have not replaced yet the silico industry. Further steps of tailoring and nanoscale electronical neurons would be required for its widespread implementation in our society.

2.1.2 Applications of NC

Its broad potential due to parallel processing and low-latency operations make it suitable for tasks involving real-time analysis of streaming data from various sources, such as sensors, cameras, IoT devices, etc. Its main strength could be assessed as instant decision making. This capability is often demanded in industries such as autonomous vehicles, predictive maintenance systems and healthcare devices. In healthcare industry, there is a growing demand for low power, non-invasive quick treatments for handling biosignal analysis (20). Current implantable devices which attempt to perform real time deep neural network (DNN) applications often end up needing cloud computing due to the required processing power. These constrains limit their applications in such sensitive fields due to risks associated with communication interferences and delays (25). Unlikewise, hardware-based neuromorphic systems address these limitations by offering implantable devices capable to perform DNN computations locally in real time (25).

In signal detection applications, it is very usual to process the input signal before feeding the network for classification (for example filtering). This helps to overexpress the features wanted to be detected, which in consequence improve the accuracy of the model. A hurdle in the implementation of NC for real-time signal classification is this first process of signal filtering and preparation for setting the optimal input to the network. In digital computers is very easy because a complex code can be compiled to perform any modification to the signals. However, in NC an equivalent electronic circuit must be developed and integrated in the system to feed a SNN. Fortunately, there is a wide knowledge and years of research in filtering circuits which are used by the community as preprocessing in hardware (6; 25). These include bandpass filters and analogue-to-digital converters.

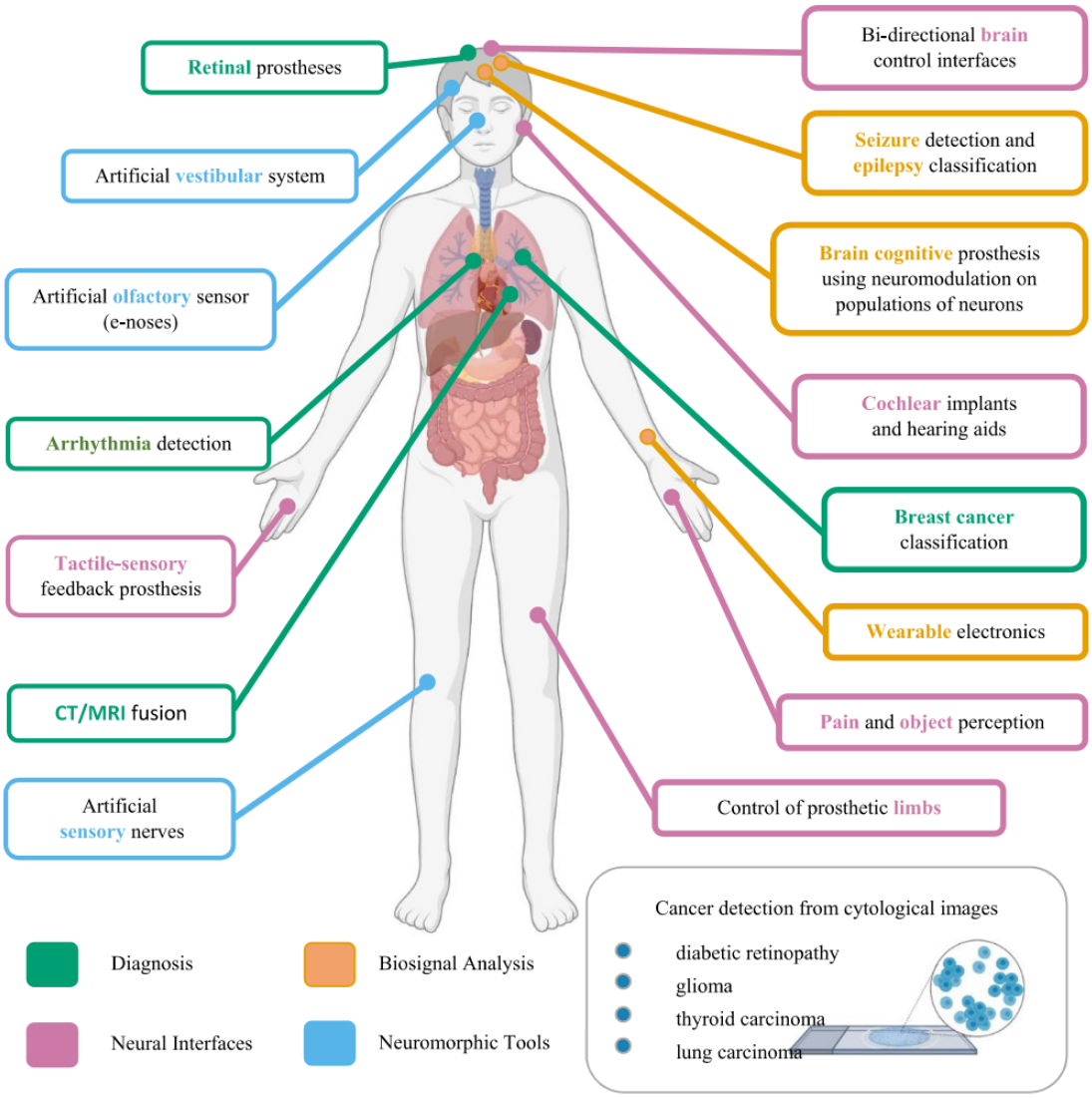


Figure 2.1: Applications of neuromorphic devices in real-time analysis of biological signals. Among the most relevant it can be found applications for: Deep Brain Stimulation closed-loop systems, cancer detection from biomarkers, cardiac anomalies detectors, and high computation cost imaging techniques.(20)

2.2 Hardware Neurons

In order to realize highly efficient neuromorphic computations that can be comparable to biological systems, bioinspired computing frameworks, involving biorealistic artificial synapses and neurons need to be developed. A circuit consisting of at least ten transistors is required to achieve the function of one biological synapse (21). This makes the implementation of neuromorphic computation very difficult for large networks. Fortunately, another realm for electronic circuits have been tailored to mimic the electrical inputs and outputs of neurons. To achieve this, profound research in materials and physics has been done (21).

To recreate the brain environment, two electronic systems have been developed to define an

electronic neuron (e-neuron). First, the cell body (e-cb), which computes the electrical inputs from the presynaptic neuron and regulates the state of its membrane potential (determining when to fire a spike[2]) and second, the neuron synapses (e-synapse), which transmit the electrical pulses generated in the e-cb. The electrical circuits defining the synapses also mimic how biological synapses retain synaptic strength[3], enabling the natural learning method behaviour of biological neurons to happen in hardware, namely Spike-Timing-Dependent Plasticity (STDP).

These two building blocks are sufficient to construct complex networks with high level of fidelity, in the way of processing information, to a biological neural network. E-cbs and e-synapses circuits can be configured together to mimic the architecture of neural networks that, up to date, could only be simulated. The level of resemblance to biological neurons obtained by the development of e-neurons, have made this computing paradigm much more energy efficient and fast than transistor-based computation (14; 12; 15). Furthermore, previous e-neuron trials integrating many transistors suffered from high leakage power dissipation and difficulty in integration due to increased space needed.

2.2.1 Building the blocks

The key to overcome traditional methods mimicking neurons behaviour (transistor based) was the discovery of memristors. They were conceived in 1971 as a theoretical concept, but they were not applied to physical devices until 2008, by Hewlett-Packard (HP) lab (26). The word memristor states for “memory” and “resistor”. Basically, is a resistance with memory. So, the value of the resistance, therefore also of the conductance, depends on the last voltage applied to it. This feature makes them capable of working as non-volatile memory. In other computing frameworks, static random-access memory (SRAM) is used as non-volatile memory framework. With memristors, a resistance random-access memory (RRAM) can be achieved.

Working principle of memristors

Memristors are, indeed, very simple structures. They are composed of a two-terminal metal-insulator-metal (MIM) sandwich. The outer layers are conductive metals, and the inner layer is made of an insulating material (figure 2.2). This insulating layer can change its resistance based on the history of voltage applied across the terminals, effectively "remembering" past electrical activity. When a voltage is applied, ions or charge carriers within the insulator move, altering its resistance (27). Different insulating materials can be used to achieve memristive behaviour. Titanium dioxide (TiO_2) memristors consist of a central insulating layer of TiO_2. The TiO_2 layer contains regions with oxygen vacancies, which can migrate under an electric field, altering the material’s resistance. When a positive voltage is applied, these vacancies form conductive pathways, switching the device to a low resistance state (LRS). Reversing the voltage causes the vacancies to disperse, returning the device to a high resistance state (HRS) (26).

[2] **Spike** refers to an action potential, which is a rapid and temporary electrical signal that travels along the membrane of a neuron.

[3] **Synaptic strength** refers to the efficacy or efficiency of synaptic transmission between two neurons.

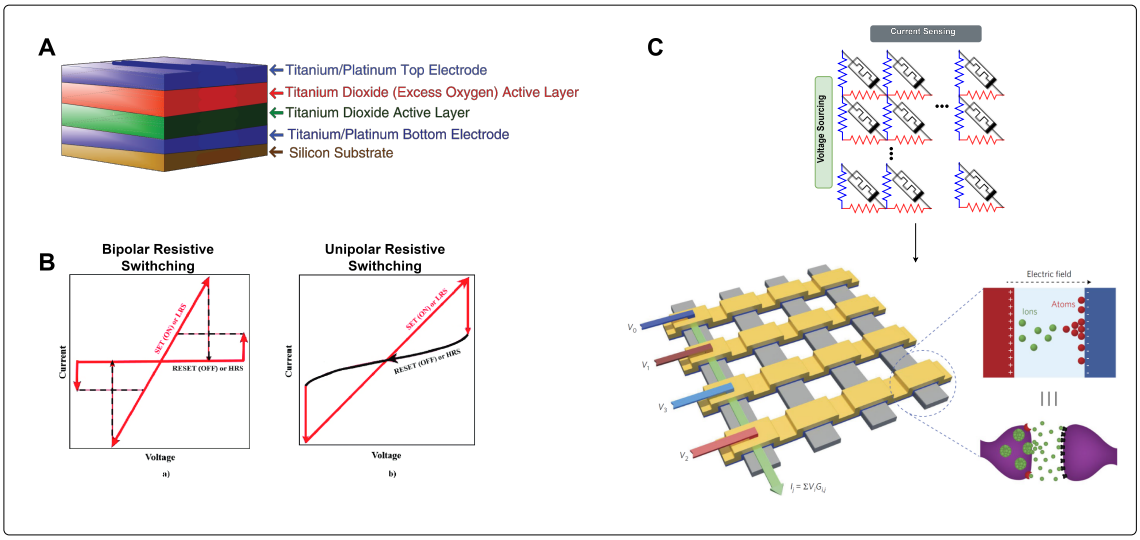


Figure 2.2: **Memristors Overview. How memristors materials are distributed and their behaviours towards electrical stimulus.**
A) Spatial distribution of memristors layers. The choice of silicon substrate can impact factors such as thermal conductivity, mechanical strength, and integration with other semiconductor technologies, top and bottom electrodes establish electrical contact with the memristor and enable flow of current. Finally, Dioxide layers undergo reversible changes in its resistance in response to applied electric fields or currents (Image recreated from (28))
B) Current-Voltage characteristic curves of memristors, showcasing their hysteresis behaviour. Image "a" shows a bipolar state transition from high resistant to low. When the voltage is reduced back to zero and further int the negative range, the memristor remains in the LRS until the negative voltage exceeds a certain threshold, causing it to switch back to HRS (marked as "RESET (OFF)"). This creates a hysteresis loop, indicating the memory effect of the device. Image "b", unlike the bipolar case, the reset process (switching back to HRS) happens at a higher voltage but of the same polarity as the set process. The device remains in the LRS until a higher voltage is applied, causing it to reset to HRS.
C) Memristive cross-bar arrays in SNN and Neuromorphic Computing. The convolution product needed in convolutional neural networks in image classification tasks is the most energy demanding process and it can be naturally performed in a cross-bar array on the physical level based on Ohm and Kirchoff laws (27; 29) (bottom figure from (30))

This ability to change and maintain the resistance in response to the voltage applied is very similar of how synapses work. Synaptic plasticity is the term used for actualizing the "synapse weights" within neurons in response to electrical activity. This behaviour is key for learning, because the output of a network can vary a lot when weights are changed. Modifying weights can imply that a post-synaptic neuron that was firing in response to a stimulus, do not fire anymore, because the inputs are no longer enough to achieve the membrane threshold. Therefore, introducing these components to the electrical circuits defining neurons has supposed a revolution in hardware based SNNs.

Before diving into the electrical circuits forming e-cbs and e-synapses, let's see the neuron models proposed in the literature.

2.3 Neuron Models

Neurons are the main variable of SNNs. How the nodes of the network will communicate be-tween them provide notable insights of the net-work behaviour. In the literature there are many neuron models reported and some of them have been widely used for different applications (31; 18).

Leaky Integrate-and-Fire (LIF)

Popular due its simplicity and efficiency. It captures the essence of neuronal behaviour by integrating incoming signals (synaptic inputs) over time. When the accumulated membrane potential reaches a specific threshold, the neuron fires a spike, and the potential resets, beginning the integration process (see Figure 2.4. This model, with its straightforward threshold mechanism and leaky integration, has been foundational in many SNN applications, offering a balance between biological realism and computational manageability (31; 32).

Hodgkin-Huxley

Developed through Nobel Prize-winning research, this model describes how action potentials in neurons are initiated and propagated, based on detailed ionic currents through the neuronal membrane. The Hodgkin-Huxley equations capture the dynamics of voltage-gated ion channels, offering an unparalleled level of detail. Although computationally intensive, this model is useful if electrochemical mechanisms underlying neural activity need to be considered (31; 21).

Izhikevich

Aiming to bridge the gap between the sim-plicity of the LIF model and the complexity of the Hodgkin-Huxley model, the Izhikevich model was introduced. This model combines biological plausibility with computational efficiency, capa-ble of replicating a wide variety of neuronal firing patterns observed in real neurons. By using fewer computational resources, the Izhikevich model

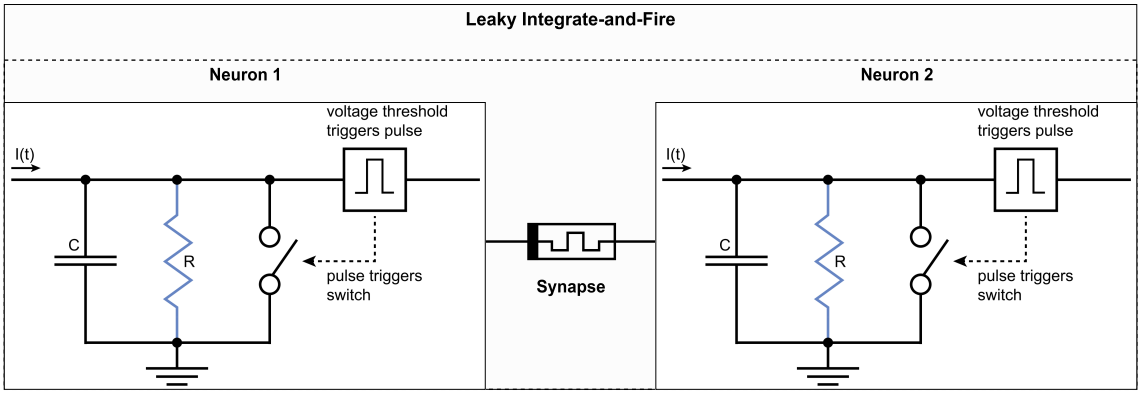


Figure 2.3: **Two LIF e-neurons connected with a memristive synapse.** The capacitor accumulates voltage until it reaches a threshold, where the trigger makes the switch connect and send the cur-rent (spike) out of the circuit. The memristor connects the neurons and adjusts its resistance in response to history of voltage applied. The natural behaviour of the capacitor leaks voltage at a certain rate making the neuron “leaky”. (LIF circuit reconstructed from (32))

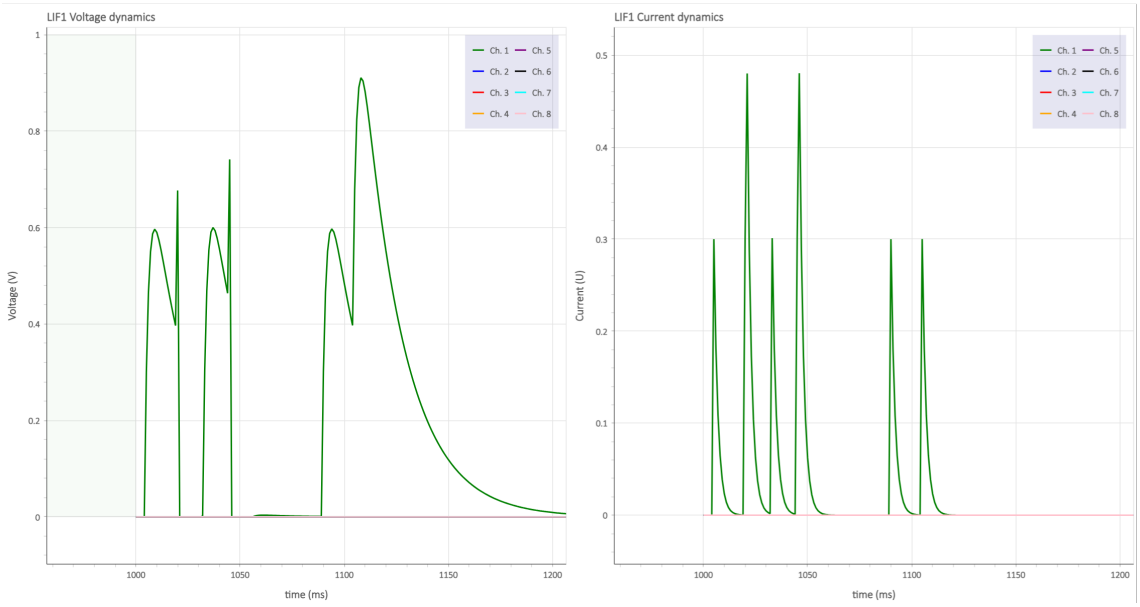


Figure 2.4: **Voltage dynamics of simplified LIF neuron**. Right figure shows input current arriving to a LIF neuron. Each current peak represent an impulse. Left figure shows the voltage of the LIF membrane in response to the input current. The addition of a small and big impulses in a low space of time causes the neuron to cross the membrane threshold (1) and fire a spike. After a spike, there is a refractory period where voltage is set to 0. Two small spikes cannot make it to reach the membrane threshold, and the voltage of the membrane decreases with time until 0 (resting potential).

captures complex spiking behaviors, making it a versatile tool for simulating large-scale neural networks without compromising on realism (31).

Adaptative Exponential Integrate-and-Fire (AdEx)

AdEx enhances the LIF framework by incor porating adaptation mechanisms. This model can replicate the adaptative behaviour of real neurons, such as frequency adaptation and spike-frequency adaptation. The AdEx model's ability to adjust its response based on prior activity provides a more refined simulation of neuronal behaviour, being able to capture the dynamic nature of neuronal adaptation observed in biological systems (31).

2.3.1 Electronic circuitry of e-neurons

E-neurons, as the basic building blocks of SNNs, must be the smaller the better (to build complex networks in small devices), non-heating (to not damage cells if implanted in the body) and easy to produce (to minimize the costs). Of course, the circuit must behave similar to neurons. In the context of the LIF neuron model, as its name suggests (leaky integrate-and-fire), accumulates voltage from inputs, while it is leaking the membrane voltage over time at a certain rate. This makes that the neuron only reaches its firing threshold when inputs arrive within a certain interval

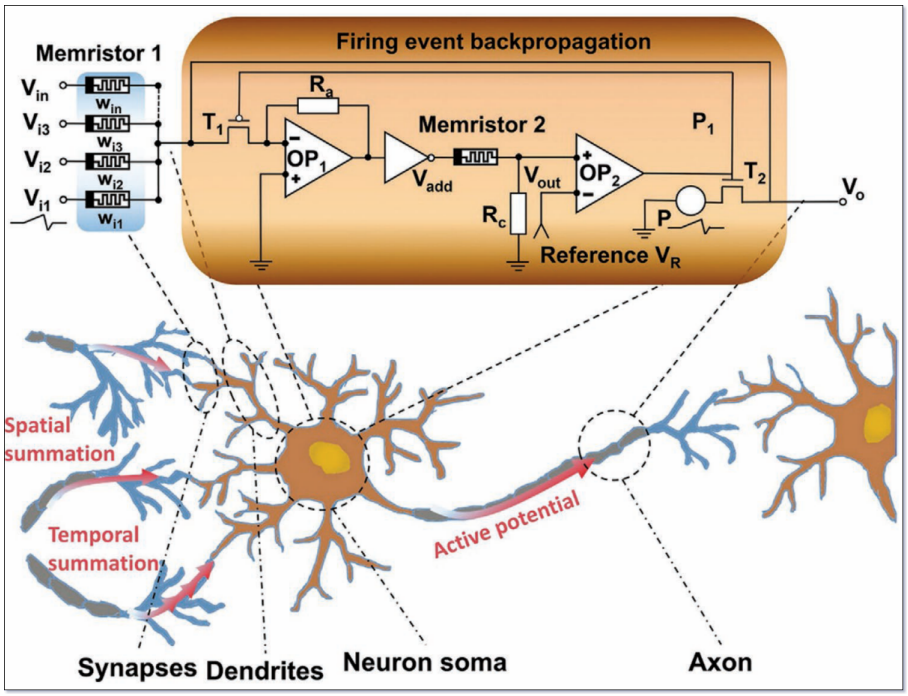


Figure 2.5: **Newer LIF models using memristors for membrane voltage leakage** Memristor 1 are synapses with other neurons. The circuit enables for backpropagation learning. Memristor 2 is the component regulating the accumulation and leakage of voltage. (Reproduced from (21)).

of time. LIF models, traditionally, perform the leakage pattern with the capacitor (33). However, with the arrival of memristors, the use of capacitors will probably get obsolete.

Memristors inherently exhibit both integration and leakage behaviours similar to biological neurons. They can integrate incoming signals by changing their resistance state in response to the input voltage and then gradually return to their baseline state, mimicking the natural decay of graded potentials in neurons (34). This leaky behaviour is naturally achieved in memristors without additional circuitry. In terms of density and integration, memristors can be fabricated at a very small scale and integrated into high-density crossbar arrays, allowing for the creation of compact, large-scale neuromorphic systems. This high integration density is harder to achieve with capacitors, which generally require more space and additional components to emulate complex neuronal dynamics. Consequently, memristors offer a more efficient and scalable solution for advanced neuromorphic circuit design.

2.4 Spiking Neural Networks

Neural communication is well known to be mediated by spikes. The neuroscientific research community have widely debated whether if neural communication primary relies on the shape of spikes or precise timing (rate coding vs temporal coding). Initially, some theories suggested that the precise shape of an action potential might carry important information. However, contemporary research tends to emphasize the importance of the spike rate over the plain shape of individual

spikes for neural encoding (35).

Rate-based coding theories propose that the frequency of spikes over a period is the primary way neurons encode and transmit information. This approach is supported by evidence showing that many neural functions, such as sensory processing and motor control, correlate strongly with changes in spike rate (35; 36). In these theories, the spike rate reflects the amount of information transmitted and is crucial for understanding neural activity.

Additional theories highlight the significance of the precise timing and patterns of spikes, arguing that the exact timing of spikes, in relation to each other, can carry additional layers of information that might be missed if only the rate was considered (37; 36). One notable example is the "chronotron (38)", which is a model of neurons that learn to fire at specific times. This model demonstrates how precise timing of spikes can encode information. Overall, the consensus in modern neuroscience is that both spike rate and timing play important roles in neural communication. In contrast to older theories, shape of the spike seems to be irrelevant in the task.

Discovering that biological neurons process information with spike timing independently of the information of the spike led to the idea of event-driven computation. Spikes are the events and neurons act in response to them. When a stimulus is provided to the biological network through sensors (touch, vision, noise...), the neurons propagate the stimuli through the corresponding areas of the brain in the form of events (spikes). The rate and time patterns of events encode the information, ultimately leading to an output. This process can involve higher or lower amount of neuronal complexes. Similarly, artificial SNNs have been developed to process information in an event domain. An architecture of spiking neurons can be engineered to mimic the processing task that biological neurons would do in the brain. To achieve such task, neuron models have been proposed and tailored to behave as neurons (see chapter 2.2).

2.4.1 SNN Learning Mechanisms

The training process of SNNs is still at the stage of development because of its novelty and lack of deployment in our society. There are several methods which enable the network to learn patterns, but there is a lot of improvement to be done. ANNs are able to achieve better performances than SNNs in many workloads (39).

Learning in neural networks implies changing the weight connections between nodes. In ANNs these nodes are responsible to modify the input signal throughout the layers to enhance "invisible" pattern features so that finally a simple operation can determine the correct label of the input. These modifications rely on "values as inputs". For example, an image as the input of a convolutional network, will experiment kernel operations on its pixel values that will change the image itself through the layers. However, in SNN, the inputs are not values, but spikes. Therefore, the network must be able to modify the strength of neuron synapses to learn time domain patterns within input rates or timings instead of input values (40; 41). The network will have a response for each sample. That is, in the context of detecting an event, composed of 100 samples, each sample will modify the membrane potentials of neurons and the network will produce spikes (or not) at the output layer. The spikes of the event will therefore cause that the network fire at higher

or lower frequency. Thus, setting a rate threshold for detection can be done (36; 42). As it is showcased, the rationale behind learning is notably different within ANNs and SNNs.

Mechanisms to achieve a trained SNN, with correct weights set within their layers, have been engineered in the past decades.

Spike-Timing-Dependent Plasticity (STDP)

This unsupervised learning rule adjusts the synaptic weights based on the precise timing of spikes from pre- and post-synaptic neurons. If a presynaptic spike precedes a postsynaptic spike within a certain time window, the synaptic strength is increased (long-term potentiation, LTP). If the presynaptic spike follows the postsynaptic spike, the synaptic strength is decreased (long-term depression, LTD) (42; 41).

Surrogate Gradient (SG) Methods

Due to the non-differentiability of spike events, traditional gradient-based optimization is challenging for SNNs. Surrogate gradient methods approximate the gradient of the spike function, enabling the use of backpropagation[4]. This approach has been effective in training deep SNNs directly and handling temporal data efficiency (43).

ANN-to-SNN Conversion

Another approach involves converting a pretrained ANN into an SNN. This is done by replacing the activation functions of the ANN with spiking neuron models. While this method can quickly yield SNNs, it often relies on rate coding and might not fully exploit the temporal dynamics of SNNs (44; 43).

The training framework may sound quite straightforward until now. In fact, it is just required the code to recreate the mentioned behaviour, which we, as humans, are perfectly capable of. The problem comes when switching from software to hardware. That is, trespass the equivalent network to a physical network of electrical neurons.

2.5 Loihi2

Loihi 2, Intel's second-generation neuromorphic chip, offers several significant improvements over its predecessor, Loihi. Key advancements include up to 10 times faster processing and up to 15 times greater resource density, which allows for the integration of up to 1 million neurons per chip (23). This enhanced performance is achieved through Intel's latest process technology and asynchronous design methods, including the use of extreme ultraviolet (EUV) lithography. Loihi 2 also supports more complex, programmable neuron models and graded spike events, which improve the efficiency and capabilities of neuromorphic applications. Additionally, Loihi 2 provides better

[4] **Backpropagation**, short for "backward propagation of errors", is an algorithm for supervised learning of artificial neural networks using gradient descent. Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights.

support for new learning algorithms, such as variations of backpropagation, and more flexible input/output interfaces, which facilitate its use in a wider range of real-world applications (23; 45). In a study carried out by accenture, they were testing voice recognition tasks for cars, and compared GPU AI computing againt Loihi2 NC SNN. Results showed that with similar performances Loihi2 comsumed 1000 times less and was 200 ms faster in inferencing (46).

Loihi2 is fully supported by Lava framework, enabling seamless mapping of processes. Furthermore, intel labs have developed an api access to test neuromorphic applications in physical loihi2 chips through the cloud.

2.5.1 Access to loihi2

Reaching a loihi2 neuromorphic chip is not straightforward. As it is still in development phase, number of devices is very low, which makes that intel carefully selects whom to give access to their chips. An application request and a project needs to be presented. Finally, after revision it is determined if allowed or not.

Once validated, there are various ways to acces a loihi2 chip.

- **Through the cloud** access is the easier way. Intel labs have created a system to interface their chips consisting of a machine with access to internet which is intelligently connected to a loihi2. The user can connect to the machine through SSH, and send compiled processes to the neuromorphic hardware. Furthermore, full control of the running timings and energy consumption can be monitored.

- **Physically** access is the way to go if aiming to personalize a neuromorphic system. Optimal for real time closed loop systems, where the response need to be as fast as possible, therefore not wasting time in communication through the cloud.

2.5.2 Loihi2 Architecture

Loihi2 device is composed of 128 neurocores, each of them containing programmable neuron models communicating in an asynchronous mode (see Figure 2.6). Its shape and ports allows them to connect between them in a 3D structure, optimizing the space. Its size is very appropriate for implantable devices as it can be easily embedded in any system (it only covers mm^2 and is thin as a paper).

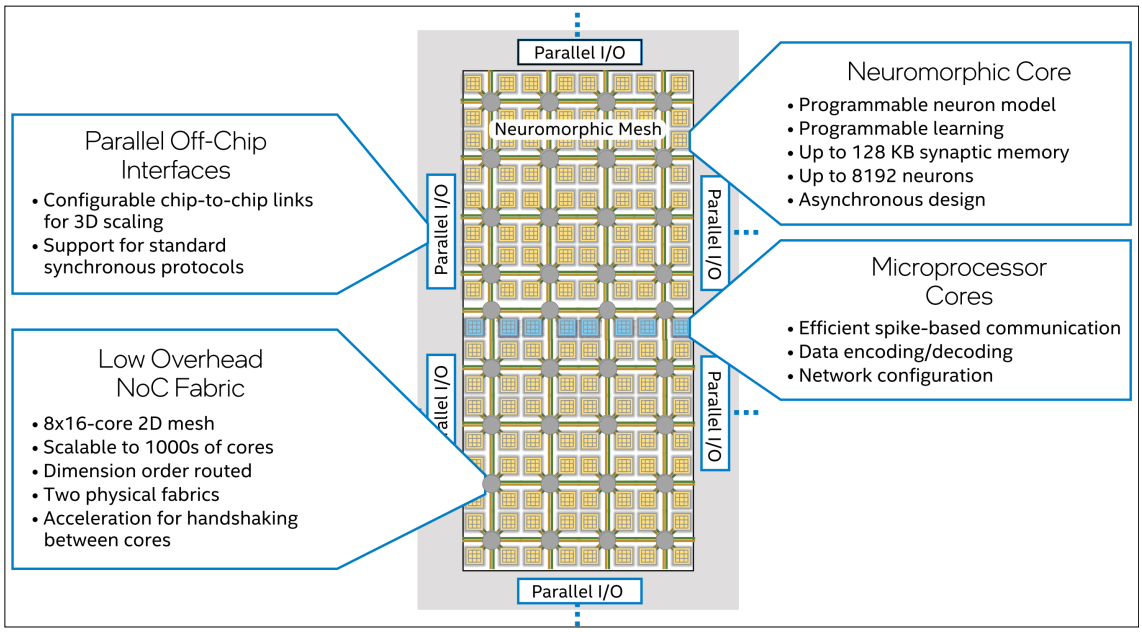


Figure 2.6: **loihi2 chip architechture** (47)

2.6 Neuromorphic Frameworks

Working with such complex systems require a well-organized framework. Through the last decades several neuromorphic environments have been developed to carry out neuromorphic computation (44; 48).

A clear leader in Spiking Neural networks for neuromorphic computing is SpykeTorch. SpykeTorch is an extension of PyTorch oriented to build SNNs. Many further emerged frameworks are supported on SpykeTorch and focused more in deploying the network to hardware (48). Within all the frameworks, LAVA by intel provide a versatile workspace to build custom SNNs with fully integrated neuron models. Furthermore, it is very fined tuned to compile the models and deploy them to their chip (loihi2) and other neuromorphic chips.

2.6.1 Lava Neuromorphic Computing

Lava[5] is an open-source software framework developed by Intel. It is designed for developing neuro-inspired applications and mapping them to neuromorphic hardware. Lava provides developers with tools to harness the principles of neural computation, enabling neuromorphic platforms to process, learn, and respond to real-world data efficiently and quickly compared to other computer architectures.

Lava has a versatile structure, allowing re-searchers to integrate a wide range of algorithms and build complex neuromorphic applications. It supports the definition of processes, such as individual neurons, neural networks, and interfaces to external devices, which can be encapsulated

[5]Documentation: https://lava-nc.org/ – Github: https://github.com/lava-nc

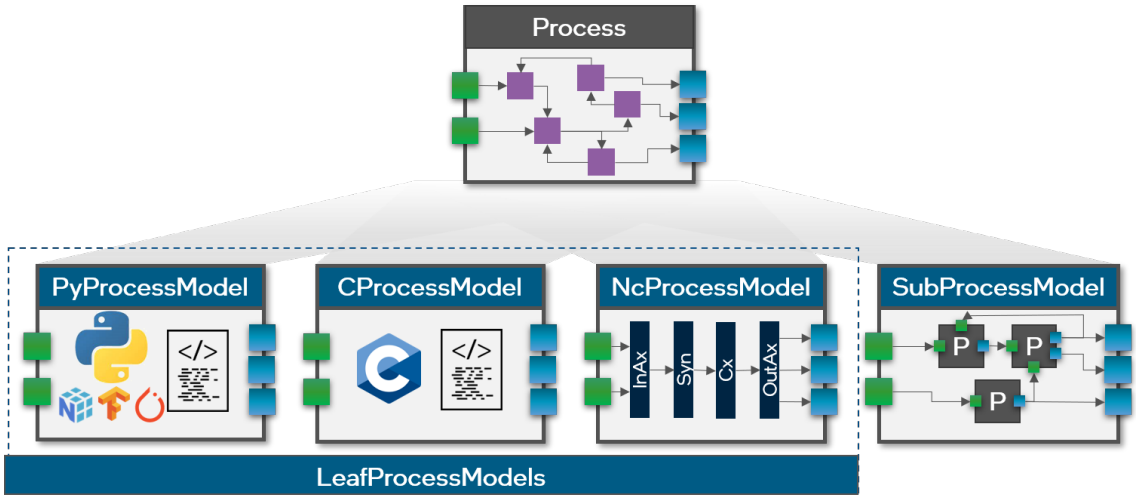


Figure 2.7: **Versatility of Lava Neuromorphic Framework.** Lava supports the creation of custom processes which can connect between them setting "IN" and "OUT" ports. (green and blue) These processes are compiled and work by event driven computation. The running of these processes can be on neuromorphic hardware or in a simulated neuromorphic environment (23))

into modules and aggregated to form sophisticated systems. Communication between processes in Lava uses event-based message passing, facilitating the simulation of neural interactions.

Moreover, Lava allows applications to be run on conventional CPUs/GPUs and deployed to various neuromorphic chips, such as Intel's Loihi. This flexibility is enhanced by a low-level interface called Magma, which aids in compiling and executing processes across different backends. Lava aims to unite the neuromorphic computing community, providing a common framework to facilitate collaboration and innovation in the field.

2.6.2 Lava Workflow

Lava framework is divided in two branches:

Lava-nc is the main branch and it is used to compile and run lava processes, as well as migrate them to physical neuromorphic chips.

Lava-dl is an independent module used to create neuron models and networks. It also provides SNN training as well as functions to convert network to lava process, which can be compiled and mapped to neuromorphic hardware.

2.6.2.1 Slayer

Slayer, standing for Spike Layer Error Reassignment in Time, is the tool used by lava-dl for training SNNs. It's an enhanced version of slayerPytorch. It trains the network on pytorch and then exports the trained model with HDF5 network exchange format. Slayer learning tool makes possible a more effective learning by improving the network's ability to recognize patterns and

make decisions, mimicking the way the brain processes information. By implementing a temporal credit assignment policy, slayer enables errors to be backpropagated through the network layers (23), addressing the challenge of non-differentiability in spike generation[6]. This allows for the adjustment of synaptic weights and axonal delays, ultimately leading to better performance in tasks like pattern recognition and decision making, similar to biological neural networks.

Slayer Training - As slayer used in lava is not independent from Pytorch, the training needs to be done in a pytorch SNN, and slayer is involved as an assistant. The assistant takes into account the loss function, learning rate and the classifier object and does the training using the pytorch net and the custom trainig functions.

```
    # Custom SNN
    net = RipplesNetwork(y_input_size=y_size, layers='256_128')
    # Optimizer and learning rate
    optimizer = torch.optim.Adam(net.parameters(), lr=0.001)
    # Error Function
    error = slayer.loss.SpikeRate(
            true_rate=0.2,
            false_rate=0.03,
            reduction='sum'
        )
    # Manager for accuracy and loss
    stats = slayer.utils.LearningStats()
    # Slayer Training assistant
    assistant = slayer.utils.Assistant(
            net,
            error,
            optimizer,
            stats,
            classifier=slayer.classifier.Rate.predict
        )

    # Training loop
    for epoch in range(epochs):
        for input, label in train_data: # training loop
            output = assistant.train(input, label)
        # Save model with hdf5, or the weights as a "state_dict"
        if stats.testing.best_accuracy:
            net.export_hdf5(trained_folder + '/network.net')
            torch.save(net.state_dict(), trained_folder + '/network.pt'
                )
        # Update the training stats
```

[6]The challenge of **non-differentiability in spike generation** refers to the difficulty in calculating the precise changes needed to improve the network's performance because the spiking function is not smooth and doesn't have a straightforward derivative.

```
31          stats.update()
```

Listing 2.1: Training with lava slayer. Lava's assistant does a custom training by using the pytorch SNN and training functions.

2.6.2.2 Lava Processes

Processes are the foundational building blocks of Lava. Each process is a self-contained unit with its own private memory, communicating with other processes through event based communication. These processes can represent simple elements like single neurons, complex structures like entire neural networks, regular computer programs, or even physical devices like sensors and actuators. Lava is very versatile due to its ability to convert different systems into processes, which can communicate between them through ports and be mapped into various neuromorphic platforms. Figure 2.9 shows how a SNN is trained on Lava with PyTorch, and then converted to a process.

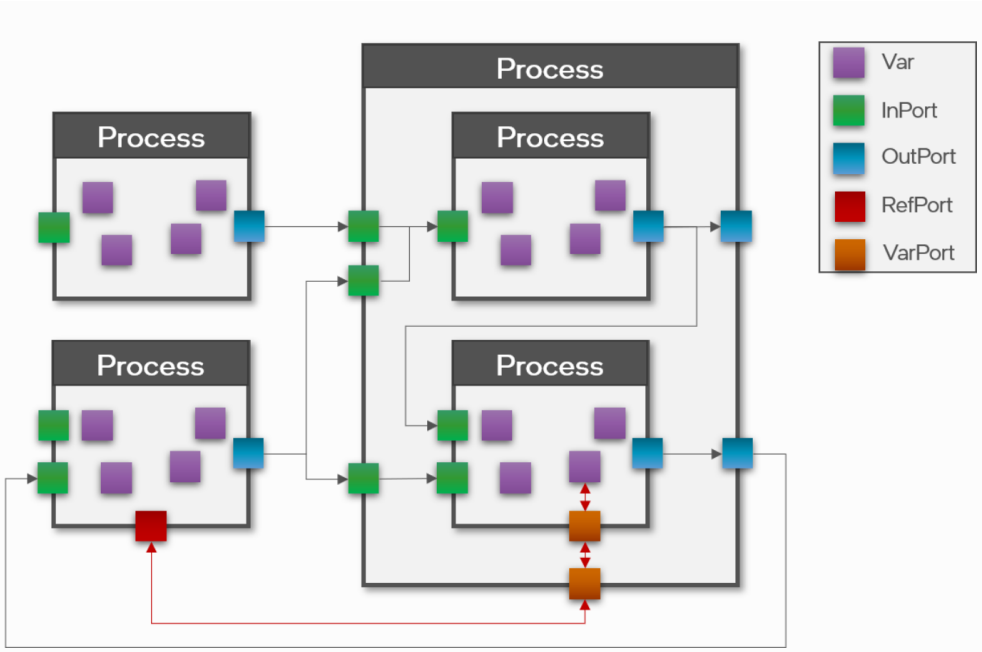


Figure 2.8: **lava Processes Overview.**

2.6.2.3 Running the SNN

To run the network with lava, it has to be converted into a process. Furthermore, a system of processes defining the flow for inferencing has to be built.

- **Loihi2 Simulation** is a virtual simulation of the hardware chip. It is supported by lava python library. To build a simulation, first, the processes need to be assembled together. Connecting lava processes is as simple as:

```
from lava.utils.system import Loihi2
...
# Connect the processes
source.s_out.connect(inp_adapter.inp)
inp_adapter.out.connect(net.inp)
net.out.connect(out_adapter.inp)
out_adapter.out.connect(sink.a_in)
...
# Run the network
net.run(condition=run_condition, run_cfg=run_config)
```

Listing 2.2: Run with loihi simulator.

- Running in **Loihi2** needs a different lava module, with different implementations of the neuron models. With that, the processes can be mapped to hardware. The documentation of this library for running on hardware is very incomplete. Some constrains are explained inside the source code comments, which make it challenging to run it appropriately. However, running "on chip" offers many useful tools, such as the possibility to control everything happening inside the chip.

AxonIn	NeuronGr	Neurons	Synapses	AxonMap	AxonMem	Total	Cores
0.80%	12.50%	0.05%	0.80%	0.01%	0.00%	1.30%	1
1.60%	12.50%	3.13%	72.00%	0.80%	0.00%	60.16%	1
0.31%	12.50%	3.13%	14.06%	0.80%	0.00%	12.79%	2
Total							4

Table 2.1: Loihi2 runing feedback. By running in Loihi2 physically, it can be tracked the number of cores used, the power consumption, number of neurons, axons, etc.

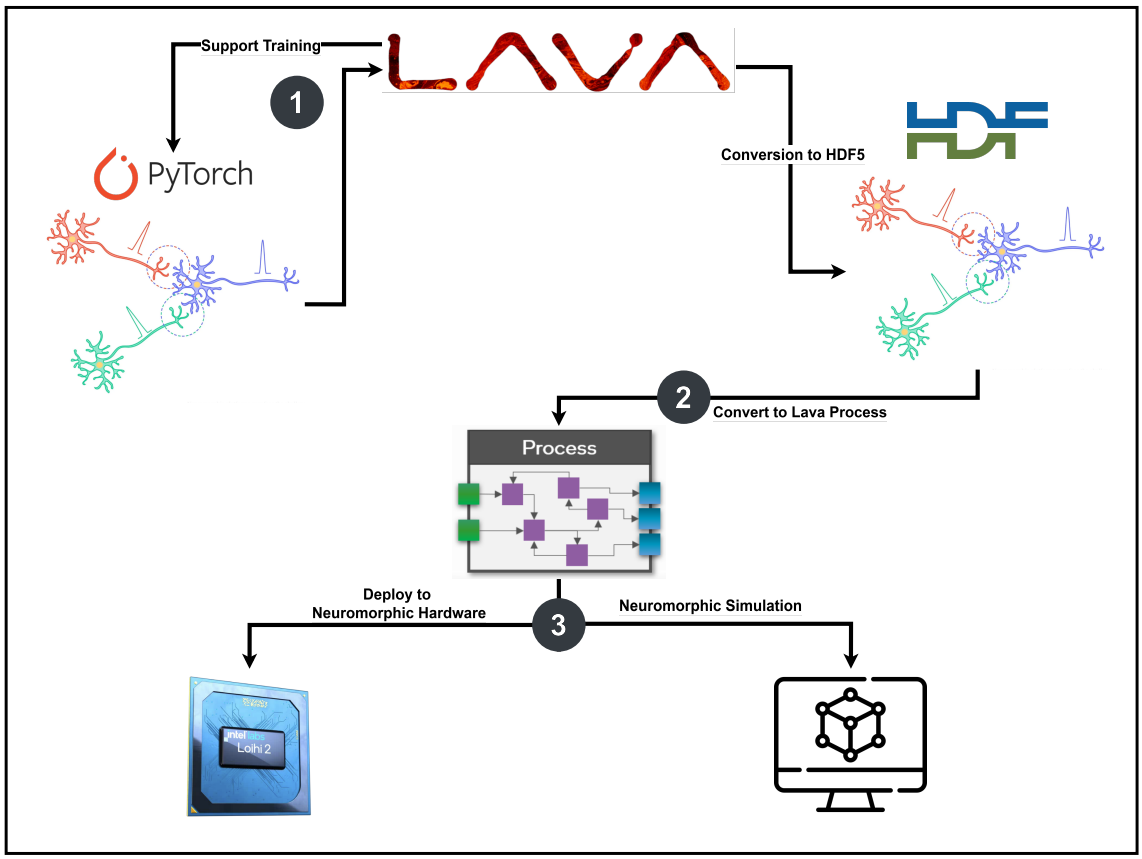


Figure 2.9: **How lava handles a SNN. 1)** The network is defined in PyTorch and trained with lava as an assistant. **2)** The trained model can be exported as a hdf5 structure and converted easily to a lava process. **3)** The process can be mapped to neuromorphic hardware and run *"on chip"*, or can be run as a neuromorphic simulation.

Chapter 3

Current stage of research in NDs

At the current stage of scientific research, Alzheimer's Disease (AD), the most common form of ND, representing the 60% to 80% of all cases (49), has no efficient treatment. This is mainly due to the late manifestation of its symptoms. When they first start appearing, many parts of the brain are already damaged without the possibility to recover (50). Similarly, Temporal Lobe Epilepsy (TLE) show the first symptoms years after the onset of the disease.

NDs are becoming a major health and economic issue due to the aging and lifestyle of the society. Currently, over 50 million people world-wide suffer from a ND. This number is expected to triple in 2050 if no effective preventive measures are found (49).

Throughout the years, the research community has tried to approach and counteract the downsides of NDs by many different techniques, leading to greatest advances and improvements in the lifestyle of patients.

3.1 Drug Treatments

Current treatments for AD are symptomatic-based rather than curative, trying to limit the progression of cognitive, behavioural and psychological symptoms of dementia. There are two main families of drugs approved on the market: Anti-cholinesterase inhibitors and anti-glutaminergics. Both are drugs that when enter the central nervous system (CNS) provoke a desired effect. Anticholinesterase inhibitors are molecules designed to increase acetylcholine levels in the brain and antiglutaminergics regulate the high levels of glutamate observed in persons with AD, which impair learning and memorization (51).

Although these treatments are not curative, they improve the quality of life of the patient by enabling them to maintain independence. Unfortunately, these drugs have a modest effect compared to their potential because of the difficulty in reaching the CNS. The drug present in the bloodstream targeting neurons must trespass many hurdles. The endothelial cells in the walls of the vessels of the CNS are joined tighter together, sealing possible gaps where drugs may go through. Furthermore, their efflux transporters that pump out undesired molecules also minimizes the effect of the drug.

To overcome all these obstacles from the blood to the CNS, known as blood brain barrier (BBB), researchers have developed encapsulation methods such as lipids and exosomes (52; 53). Moreover, other delivery routes were explored to overcome traditional ones. Intranasal delivery (IN) administration provides an alternative to intravenous administration (54). It is non-invasive, painless, and easy to ad-minister without a medical specialist. Furthermore, the IN route bypasses the BBB, enhancing drug bioavailability by avoiding first-pass metabolism and intestinal degradation (55).

With all the proposed techniques significant advances with improving results were obtained. However, any of them was able to revert the progression of the disease nor neutralize its primary cause. Furthermore, they were not able to restore the damage. As a result, dealing with NDs implied long-term treatments without foreseeable total cure. Drug therapies are thought to be more effective in the early asymptomatic stage before the process of neurodegeneration occurs (51). (56) claim the need for better diagnosis in the early stages of AD using additional biomarkers to improve their prospects.

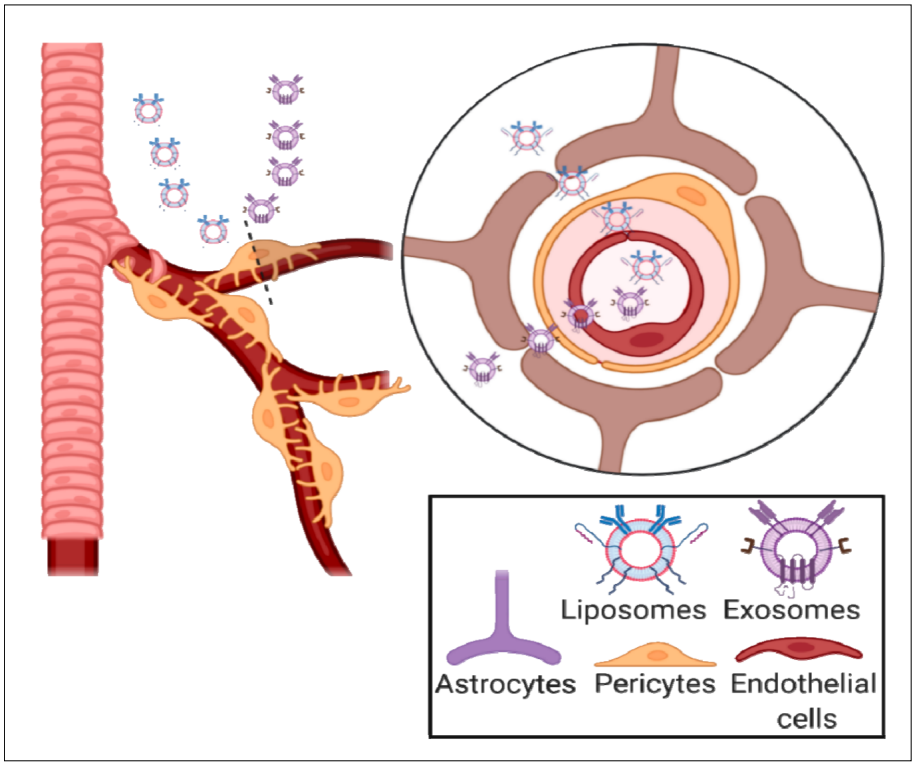


Figure 3.1: Schematic representation of BBB complex interac-tion with inorganic and polymeric nanoparticles (51). Liposomes or exosomes must cross endothelial cells, as well as pericytes to finally reach the CNS.

3.2 Electrical Treatments

Since achieving fair effectiveness of drug therapy in the CNS has been a major challenge, other therapy methods have been explored. Patients with Parkinson (PD), AD and TLE have shown good response to determined electrical stimulus in specific regions of the brain (57).

There are several types of electrical stimuli applied to brain tissue (58). Their main goal is to induce some controlled neuronal activity that will improve the condition of the disease. It was first applied in the mid 20s. (59) observed positive reinforcement in rats by electrical stimulation on septal area and (58) reported changes in rats behaviour as a result of daily brain electrical simulation. This opened a new realm in the context of therapies for neurological diseases. Since then, many electrical stimulations methods have been explored.

3.2.1 Brain Stimulation Systems

Nowadays, there are some methods that have gained higher relevance and are more often used.

Deep brain stimulation (DBS) has been approved for the treatment of Parkinson's disease, essential tremor, and dystonia. Research is ongoing to explore its potential in other neurodegenerative disorders such as AD and Huntington's disease (49; 57; 60).

Transcranial Magnetic Stimulation (TMS) is non-invasive and uses magnetic fields to induce electrical currents in targeted brain regions. It is primarily used as a treatment for depression but has also shown promise in conditions such as Parkinson's disease and Alzheimer's disease (61).

Transcranial Direct Current Stimulation (tDCS) delivers a low-intensity direct current through electrodes placed on the scalp, modulating the excitability of cortical neurons. It is a non-invasive technique that has been investigated for various neurodegenerative diseases, including Parkinson's disease, Alzheimer's dis-ease, and multiple sclerosis (62).

DBS consists of electrodes implanted in deep areas of the brain. These electrodes emit regular electrical impulses to modulate neuronal activity. It is primarily used as a treatment for chronic neurological disorders such as PD, AD, obsessive-compulsive disorder (OCD), dystonia, essential tremor, epilepsy, and depression (63).

The working principle of DBS is not fully understood. However, there are some neuronal behaviours strongly linked to DBS. The main hypothesis is that high-frequency stimulation of DBS helps to normalize dysfunctional neuronal firing patterns in the brain (64). It may disrupt pathological oscillations, synchronize neuronal activity, and induce plastic changes in neural circuits, ultimately leading to therapeutic effects. (9) proposed a closed-loop system to detect abnormal high frequency oscillations in the hippocampus and perform local electrical stimulation to re-store normal firing activity.

Dysfunctional neuronal firing patterns are one target for DBS regarding PD and epilepsy (57). The high frequency oscillations of the electrodes end up exhausting the synapse neurotransmitters

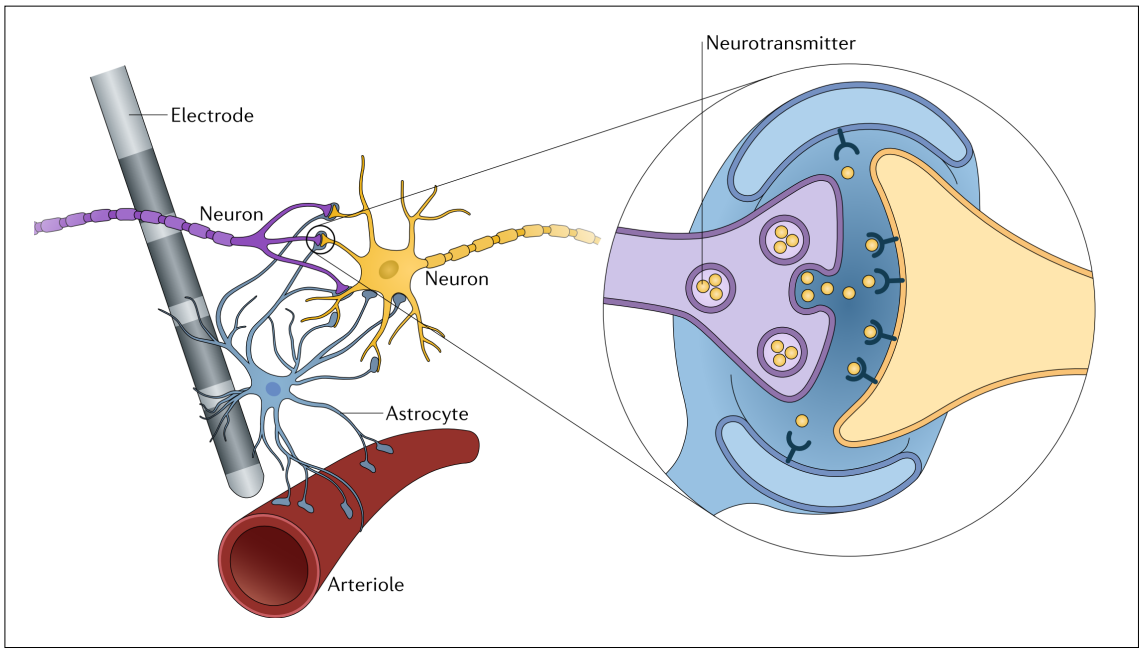


Figure 3.2: **Overview of DBS on neuroplasticity.** Neurotransmitters (inset) are released in response to stimulation, leading to calcium waves and subsequent release of gliotransmitters. This release influences synaptic plasticity, leading to arteriole dilation and increased regional blood flow ultimately leading to tissue regeneration (63).

of nearby neurons and block the pathway to subsequent action potentials of the pathological network. This disrupts excessive oscillatory synchronization leading to normal brain function (9; 65).

Depending on the frequency of stimulation and other parameters, different outcomes can be achieved on the target tissue. It has also been re-ported increased neuroplasticity resulting from DBS (63; 60; 66). This neuroplasticity enhanced a notable recovery of damaged tissue on neurodegenerative diseases, where most of the times the affected areas cannot achieve tissue recovery.

Overall, DBS and other types of electrical stimulation devices have opened another realm for approaching NDs. This realm probably boosted the field of neuronal signal processing. The bound line between stimulating and measuring is very thin. If a cable can be introduced precisely to deliver controlled current same can be achieved with an electrode to measure. So, with the development of DBS, local invasive electrodes have also gone through a process of tailoring (67). The combination of measuring and stimulating provides another degree of freedom and control in the context of electrical stimulation as a therapy for NDs, where stimulations can be intelligently delivered in response to known biomarkers (9; 10; 11). Some of these biomarkers are localized in specific parts of the brain, namely the hippocampus (1). The hippocampus has shown to play an important role in ND. Oscillation pat-terns within its networks show to mediate important tasks related to memory (3). Studies in the literature involve numerous recordings and analysis of signals within its neural populations (68).

3.3 Neural Electrode Systems

To effectively interface with the brain and record neural activity, a good and reliable recording system is required. It is as important to have good quality signals as to being able to detect them. Since we know that the brain has electrical activity happening within its tissue, many trials have been done aiming to decipher their meaning. Currently, many recording systems are enabling neuroscientific community to learn many insights from neuronal signals (69), thus boosting the appearance of new technologies and advances. Electrodes implanted in the brain have created a renaissance in the study of normal and pathological brain function. These devices are being developed to treat a growing number of medical conditions, including Parkinson's disease, paralysis, Alzheimer's disease and depression (70). These systems range from non-invasive to invasive techniques, each offering differ-ent levels of spatial resolution and specificity.

3.3.1 Novel Technologies

Common and traditional methods such as electroencephalography (EEG) and magnetoencephalography (MEG) are widely used for their ability to monitor brain activity without the need for surgical intervention. EEG measures electrical activity through electrodes placed on the scalp, while MEG detects magnetic fields produced by neuronal activity. Both methods provide interesting insights into brain function but are limited by their relatively low spatial resolution.

Other newer more invasive techniques include Electrocorticography (ECoG), which involves placing electrodes on the surface of the brain, typically during neurosurgery. This approach offers better spatial resolution than EEG and MEG, allowing for more precise localization of neural activity. Some works have relied on this technique to assess epileptogenic tissue areas during brain interventions (6). The more precise recorded signals provide with a higher signal-to-noise ration and the detection of biomarkers results to be more accurate. In epileptic patients, high frequency oscillations originating from epileptogenic tissue can be detected by filtering the signal in its frequency band and training a simple network to detect them (71).

More invasive methods involve implanting electrodes directly into brain tissue, providing the highest spatial resolution and the ability to record from individual neurons or small groups of neurons. These include:

Single-Unit Recording uses fine microelectrodes to record the activity of single neurons. It offers unparalleled detail but can only sample a limited number of neurons at a time. This type of electrodes can be found in patch clamp method, a powerful technique widely used in electrophysiology to study the ionic currents flowing through individual ion channels on the membranes of neurons.

Multi-Electrode Arrays (MEAs) consist of multiple electrodes arranged in a grid, allowing simultaneous recording from many neurons. This approach is valuable for studying neural networks and their dynamics (72). MEAs can be valuable to study 2D distributed cell populations. Being

able to place electrodes uniformly as a "matrix" through the cellular space can assess signals flowing through the network and population oscillations. Nevertheless, they may not be suitable for deep tissue recordings due to their limited capabilities of being embedded in a 3D cell complex (73). MEAs are primarily applied in vitro. Where neurons are delivered in top of the electrodes to grow in top of them. Recordings are directly in contact with neuron membranes, leading to much less interferences and pure raw signals (72). Although, they have also been used in vivo (72; 74).

Shank Invasive Electrodes [1] , such as silicon probes, are a specific type of invasive electrode designed to penetrate brain tissue and record from multiple sites along their length. These electrodes can sample from different layers of the brain, providing detailed information about neural activity in various depths. Shank electrodes have been instrumental in advancing our understanding of complex neural circuits and their role in behaviour and cognition. Their appearance was driven by the need of recording deep areas of the brain with high resolution (75).

In applications where the goal is to study deep brain complexes, invasive technologies show better outcomes compared to superficial ones (70). In neurodegenerative diseases (NDs) where structures such as the hippocampus and temporal lobe are object of study, shank electrodes, despite their invasive damage, are worth the resolution they provide at the local site (76).

3.3.2 Key advances in newer systems

In top of improvements in recording methodologies, advances in electrode materials and physics and logics have also been done. Softer, less invasive, and more broadly distributed approaches outperform older technologies (77; 75; 62). It has been well reported that traditional metal and silicon probes often cause an undesirable immune response, being reflected in local neural loss and glial reactivity (78; 79). Improved electrodes influenced by underlying concerns reveal the well known "Michigan" style arrays. These devices consist mainly on silicon with iridium or platinum recording sites embedded in the silicon shank (80; 81). Their main advantage is the possible different configurations easily achieved with them (70).

Another key point in neural recording is the electronical advancements. These made possible to overcome a limitation which was the low capability of multiplexing[2]. Previously, this low capability limited the number of simultaneous recordings that could be obtained from neural electrode systems (69). This has allowed researchers to record neural activity from many more sites simultaneously, greatly enhancing the ability to study complex neural net-works and brain functions in greater detail (67; 75; 62).

[1] A **shank** is typically a slender, elongated structure that houses multiple electrode sites along its length, allowing for the recording of electrical activity from different depths within the brain or other neural tissues.

[2] **Multiplexing** refers to combining multiple signals from various electrodes into a single data stream for transmission or processing.

3.3.3 Prospects for learning and memory in NDs

Recording oscillatory behaviours, such as ripples in the hippocampus, in local parts of the brain is a thoroughly studied realm and has let to the understanding of cognitive processes (see chapters 4.2, 4.3). However, there are still many gaps and unclear processes that limit the development of more efficient treatments for NDs. NeuroGrid electrodes were able to determine anatomical locations of ripple-generating cortical areas (82). Furthermore, Neuropixel electrodes, a kind of deep implanted shank electrodes, achieved to record simultaneous activity of multiple cortical areas (with the outstanding number of >2000 neurons per site) and deeper brain structures (83; 84). Further investigation works to deeper understand the transfer of memory traces and consolidation of memory could be achieved by combining these two recording systems simultaneously, because information of subcortical and cortical areas would be obtained at the same time, thus being able to relate patterns within deeper and surface observing points (67).

Chapter 4

The Hippocampus and SWRs

The hippocampus is one of the most thoroughly investigated parts of the brain. As a complex brain structure, it is found embedded into the temporal lobe and is involved in many processes of learning and memory. Since the famous report of the case study H.M.[1] who lost the ability to acquire new memories after the removal of the hippocampus in a desperate approach to suppress invalidating epileptic seizures, it has been posed at the centre of research in memory consolidation (85). The intense research raised to the discovery of several subregions, with a complex interplay between them (firstly defined as trysinaptic loop), where input information from sensory systems was processed following a specific path (see 4.1), thus providing the brain with a spatiotemporal framework where memories could be stored and consolidated.

4.1 Hippocampal subregions

The hippocampus is divided into two main complexes: Entorhinal Cortex (EC) and Dentate Gyrus (DG). EC provides the major cortical input source to the hippocampus. As information flows through it, different subregions (CA1, CA2, CA3, CA4) are distinguished (2; 77; 86) (see figure 4.1). It is said to mediate neural communication between neurons from hippocampus and neocortex. It primarily targets Dentate Girus (DG), which then targets Cornu Ammonis 3 (CA3), finally leading to CA1, which will project back to neocortical neurons, through EC, to complete the loop (86; 77). Apparently, the hippocampus seems predictable, each part doing a defined job. However, it has been found that the complex is not so stratified into separate regions. Simultaneous activity, parallel processing and widespread connectivity has been revealed in further studies (2; 86; 3; 77).

[1] **H.M.**, whose full name was Henry Molaison, was a patient who became one of the most famous cases in the history of neuroscience. In 1953, when he was 27 years old, Molaison underwent experimental brain surgery to alleviate severe epileptic seizures. The surgery, performed by Dr. William Scoville, involved the removal of large portions of his medial temporal lobes, including most of his hippocampus (85).

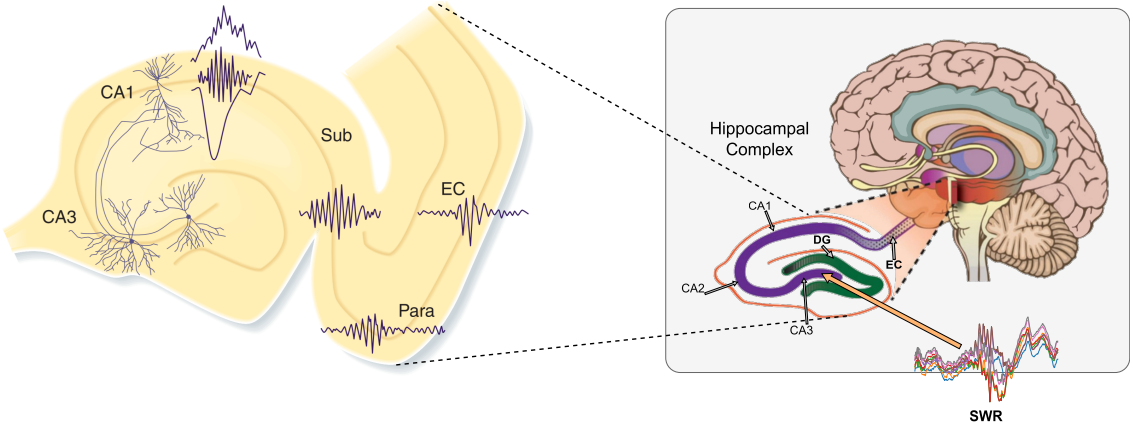


Figure 4.1: Transversal axis sliced view of the hippocampus showcases its subregions and their spatial distribution. Entorinal Cortex (EC) and Dentrite Gyrus (DG) are sandwiched between them creating the collectively known "hippocampal formation". SWRs originate within the CA1 and CA3 regions and travel throughout entorhinal cortex to reach other brain regions. (Figure reconstructed from (2).)

4.2 Memory Consolidation

These processes happening within different subregions of the hippocampus are believed to somehow encode the lived experiences in a way that later the sense of it can be recreated in our brain with conscious awareness (known as declarative memory). When exposed to an experience, the learned material remains vulnerable to interference for a period of time before consolidating in other cortical areas or getting replaced by new ones (3). In the literature it has been reported two types of declarative memory (3):

- **Episodic memories** are those related to events that happened in a specific place and time throughout the day. They are easily forgotten. For example “this morning I closed the door when leaving”.
- **Semantic memories** are stored as general knowledge about the world. For example “if you touch the fire you get burnt”. Also referred to as “long-term memory”.

Many researchers believe that the hippocampus is specifically important for forming new episodic memories, whereas other parts of the temporal lobe and neocortex are more critical for semantic memories (77). Therefore, the process of memory consolidation, where an experience is interiorized in our brain to contribute to our knowledge, relies on the ability of the hippocampus (as the first step of the process) to retain episodic memories. Episodic memories will follow further steps of consolidation if they are relevant or not (3) (repetition of an event usually makes it more relevant), ultimately becoming more independent of the hippocampus (semantic memory). In support to these, it has been observed that patients with hippocampal damage could remember events that happened years before but could not remember what they ate for breakfast (77).

The process of how memories gradually get independent from the hippocampus is not fully comprised. It is widely believed that the memory storage framework used by the mammalian brain is based on the synaptic weights and connections that neurons have between them (87; 88). Connections and weights encoding for episodic memories are built during events (89). Some oscillation and firing patterns originating from neuronal ensembles[2] (among them SWRs) are thought to be involved in the process of transferring those connections and weights to the neocortex (3; 90), where they will be available for a long period of time. Then, the hippocampus will leave those connections free to new incoming events.

4.3 Sharp Wave Ripples (SWR)

SWRs are distinctive patterns of neural activity observed in the hippocampus. These patterns consist of high-frequency oscillations (ripples) superimposed on sharp-wave complexes, which are characterized by brief, high-amplitude deflections in the local field potential (LFP)[3]. They represent the most synchronous population pattern in the mammalian brain (1). In the ~100 ms time window of a hippocampal SWR, 10-20% of the total neural population in the rat hippocampus discharge simultaneously in the CA3-CA1 subregions (2).

They arise from neuronal ensembles in the hippocampus. Their excitatory output stimulates a wide area of the cortex and several subcortical nuclei. SWRs occur during "off-line" states of the brain, associated with consummatory behaviours[4] and non-REM sleep, and are influenced by numerous neurotransmitters and neuromodulators. Numerous studies unveil their relevance in the

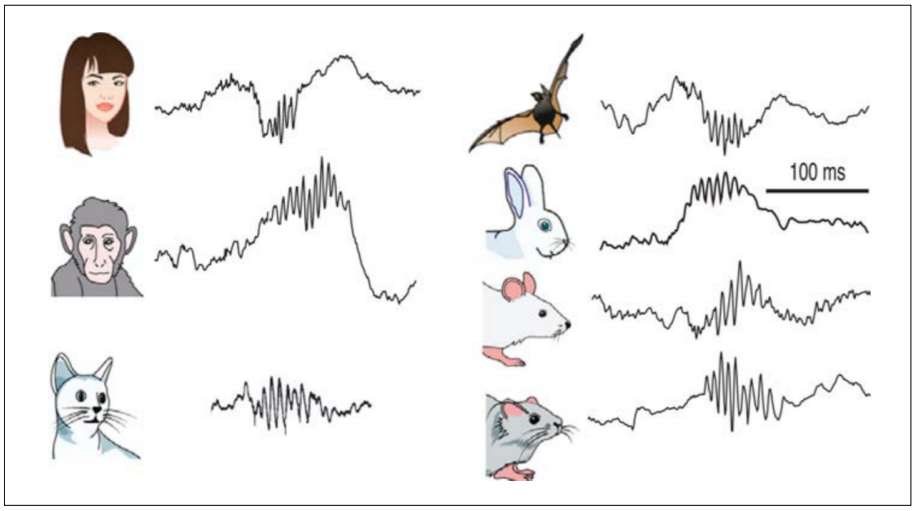


Figure 4.2: **SWRs: the most synchronous oscillating pattern in the mammalian brain.** SWR recorded from different mammal hippocampus. They all have a similar pattern: 3-9 high amplitude waves in a frequency ranging from 100-250 Hz (1)

[2] A neuronal **ensemble** refers to a group of neurons that function collectively to perform specific tasks or processes.

[3] **LFPs** reflects the synchronized activity of a group of neurons in the vicinity of the recording electrode.

[4] **Consummatory behaviours** are actions or activities that fulfil a biological or psychological need, typically associated with the satisfaction of a primary drive or motivation. (i.e. Eating, resting . . .)

process of episodic memory encoding (1; 4; 5). The memory traces are encoded via weak synaptic potentiation[5] in the CA3 network induced by theta oscillations during consummatory behaviours. Then, synapses strengthened during the process contribute to the generation of SWRs (77), which target, through entorhinal cortex, neocortical regions. Thereby, SWRs stablish a bidirectional communication between hippocampus and neocortex, being able to gradually transfer memories outside the hippocampus and playing a pivotal role in the process of knowledge and memory consolidation.

4.3.1 SWRs related terms and definitions (1)

Sharp waves are characterized by brief, high-amplitude deflections in the LFP recorded in the hippocampus. They typically have a frequency range of around 0.1 to 4 Hz. These sharp waves represent synchronous depolarization of populations of neurons, often associated with the reactivation of neuronal ensembles involved in memory consolidation.

Ripples refer to high-frequency oscillations superimposed on the sharp waves. They are fast oscillations in the frequency range of approximately 100 to 250 Hz. Ripples are thought to reflect synchronized activity within local neuronal circuits, particularly involving the coordinated firing of interneuron[6] populations.

Fast Ripples are ripples occurring at higher frequencies, typically above 250 Hz. They are often observed in pathological conditions such as epilepsy and are believed to be related to abnormal neural firing patterns associated with seizure generation.

4.3.2 Pathological SWRs

In rat brains, SWRs duration range from 30 to 150ms, and its amplitude should never exceed 3mV (1). Alteration of the physiological mechanisms supporting SWRs leads to a pathological signal

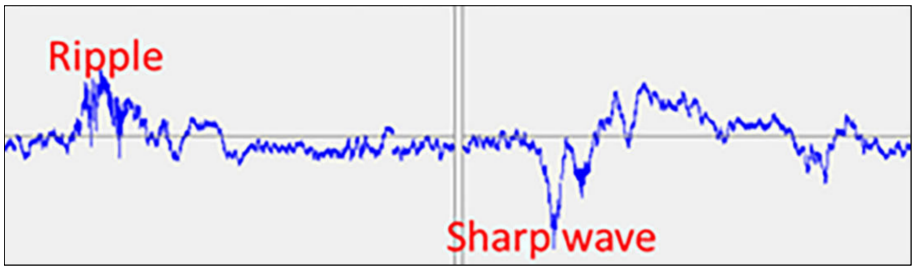


Figure 4.3: Ripples refer to high amplitude oscillatory activity in frequencies ranging from 100-250 Hz. Sharp Waves show a deeper low frequency wave sharply decreasing at the beginning (5).

[5] **Weak synaptic potentiation** typically involves a moderate increase in the efficiency of neurotransmission at the synapse, leading to a relatively modest enhancement in the postsynaptic neuron's excitability or responsiveness.

[6] **Interneurons** also known as association neurons, are a type of neuron that serves as a mediator or connector within the nervous system.

morphology, which is a marker of epileptogenic tissue and can be observed in Schizophrenia and AD. In addition, dysfunctional SWRs could be an important cue for early stage detection of AD, as these signals show different morphological features in the affected host and start appearing near the start of the disease, when there are no symptoms and no damage is yet done to brain tissue. It is true that genetic and environmental risk factors can be used to predict future AD, however, the confidence of these predictions and temporal prediction for the appearance of symptoms remains poor. Alternatively, these factors could be combined with an analysis of SWRs to detect the disease and treat it at an early stage.

Being these signals a promising biomarker for future neurodegenerative diseases affecting memory and cognition, it is of valuable interest to learn to correctly assess and detect them.

Chapter 5

Methodology

The study was able to train a SNN that could detect ripple oscillations with good recall[1]. The network was trained with lava neuromorphic framework. Within lava, lava deep learning (lava-dl) was used to define the neuron parameters and network architecture as well as the learning method. A n-dataset was conceived from the original provided dataset. The n-dataset contained the positive and negative events equally distributed (same number of events) in a format that the SNN could work with (spikes). Once trained, it was validated and performance metrics were obtained. The SNN was compared with a state-of-the-art CNN[2].

5.1 Dataset

The dataset was recorded by Neural Circuits Lab from CSIC, Instituto Cajal, Madrid guided by Liset M. de la prida[3]. *μ-LED optoelectrodes* (32 channels, 4 shanks of 8-channels) were used in head-fixed awake transgenic Thy1-GCaMP7 mice[4] to record local field potentials (LFPs) at 30 KHz in the dorsal hippocampus (91). Recordings were done several days after the implantation to let them habituate.

The dataset used for this study consisted on two recording sessions in mice: "Amigo2" and "Som_2". Each session contained ripples and SWR events tagged altogether as ground truth ripples. The times when ripples occurred were manually annotated by an expert in a postprocess phase. The expert would tag the start and the end of the event, setting a time window where the ripple occurred.

Each session had a binary raw file with the recorded LFPs, and an info file containing all the tagged events.

[1] Ratio of true positive predictions to the total number of actual positive instances

[2] **C**onvolutional **N**eural **N**etwork

[3] https://cajal.csic.es/laboratorios/circuitos-neuronales/

[4] These mice express GCaMP7 in their neurons, allowing researchers to monitor neuronal activity in real-time by measuring changes in fluorescence. This provides insights into the dynamics of neural circuits during different behaviors or conditions.

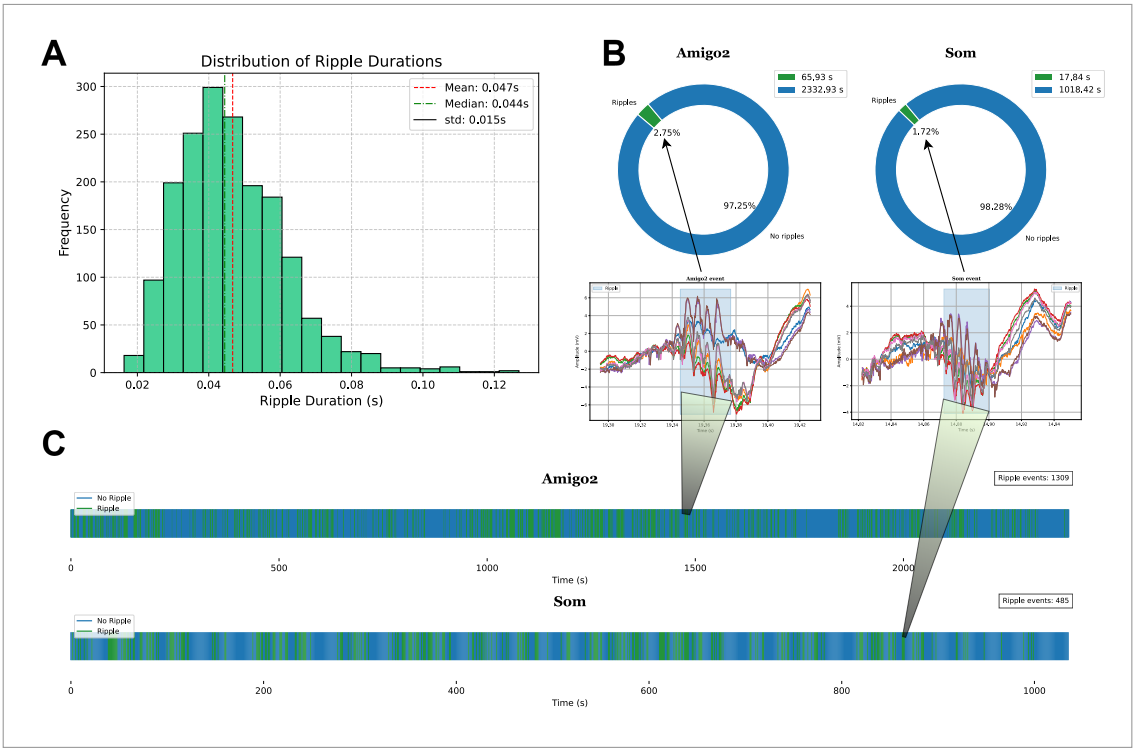


Figure 5.1: **Overview of the two recording sessions: Ripple events, durations, and features.** Both datasets have very similar events. Amigo2 represents a major part of the data. Both sessions contain altogether 1794 ripple events. **A)** Normal distribution of the duration of the handtagged ripples. Most of them are in the range of 30-60 ms, however there is a non small part of the population with longer durations ranging from 80-120 ms. **B)** Ripple events represent a 2% of the recording and are **C)** uniformly distributed through it. Each vertical green line represent a second of the recording which contained at least one event.

5.2 Data Processing

The annotated events of the recording were extracted and processed to make it easier for the network to detect differences between "ripple" and "non-ripple" events. The data processing consisted of two main operations: Filtering and conversion to spikes. Figure 5.7 shows the processing pipeline from raw LFP until training data. 1794 positive events (ripples) were extracted from the recording and processed to spikes. Equivalent negative events were extracted to balance the dataset. The data was normalized with z-score algorithm and downsampled from 30000 Hz to 4000 Hz. Then the signals were filtered with a butterworth banpass filter in the ripple band range (100-250Hz).

5.2.1 Reading the data

The data provided, as mentioned in earlier, was structured containing the signal data in a binary raw file, and the event timings in a separate info file. The binary file consisted of a one dimensional array containing the merged values of all channels from the whole recording time. The recording

contained 43 channels, from which only 32 were recording data. The remaining 32 channels were divided in 4 sections. Each section was a shank containing 8 electrodes in different depths. Only one shank was placed at the region of interest (hippocampus) for each recording. Hence, the corresponding electrode values of the shank (specified in info file) needed to be extracted from the one-dimensional array. To do so, the concatenated channel values from each recording sample were assessed to isolate them from the other values. In order to simplify the process, a python module toolkit (https://github.com/MarcosOriolPago/LAVA_SNN_ripples/tree/main/liset_tk) was programmed to easily load and visualize the data. Helpful cues for understanding how to load this kind of data were found in (91).

The data was normalized and downsampled. For the normalization it was used a custom z-score normalization. Once the data was normalized, it was downsampled to 4000 Hz. Original sampling frequency (SF) was 30000 Hz, but since the model does not need that much detail of the signal, and having higher SF means more input signals for the same signal, it makes no sense to use the original SF. In the machine learning toolbox paper from liset, they downsampled the signals to 1250 Hz. However, the quality of the signals with that SF was not optimal for the SNN.

The downsampling method consisted on taking the middle elements within the recording so that they would have the downsampled number of values each second.

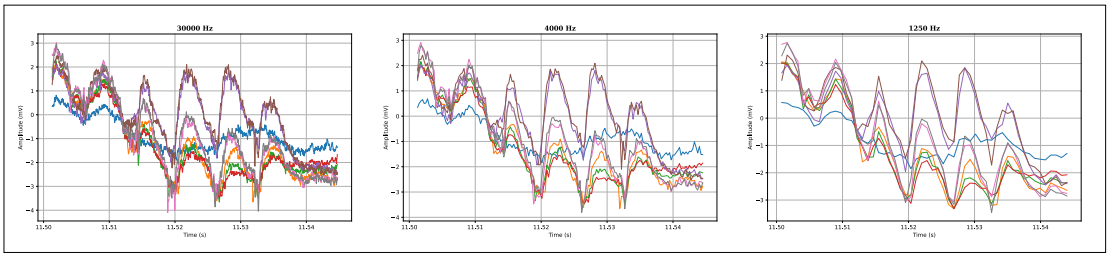


Figure 5.2: Sampling frequency choice. The image shows the same signal for three different sampling frequencies (ripple event before filtering). Using 30000 Hz embraces too much unnecessary noise. 1250 Hz may work, because the shape, at its main, is conserved. However, in order to make sure it gets all the oscillations with a decent resolution, the fs can be increased to 4000 Hz, which seems more reasonable.

5.2.1.1 Defining the window

As the network needed a fixed window as an input for spike rate predictions and training, all the training data must have the same shape. As it can be observed in Figure 5.1 *"a"* the tagged ripples from the recordings had very different durations. Determining a fixed window for all of them was an important decision that would significantly affect the performance of the model. If the window is too short, large ripples may not be detected and otherwise, small ripples may not be detected. In order to set a window that would get the most of all the ripples, the following equation was used.

$$\text{t\_w} = \mu(\text{ripples}_D) + \sigma(\text{ripples}_D) \tag{5.1}$$

Where t_w is the fixed time window for creating the custom n-dataset and "ripples_D" is a one-dimensional array containing the durations of all the ripples tagged from both recordings. That is, the mean summed by the standard deviation of the ripple's durations. Once the time window value is set, the number of samples to take from the recording for extracting ripple and non-ripple events, would depend on the sampling frequency of the data.

5.2.2 Defining the filtering band

Ripples propagating through the hippocampus can show mild variations in its oscillation frequencies (1). This explains the variability in bandpass filtering among the literature (92; 93; 90; 82). Where they go from 80Hz to 150 Hz in the low pass, and from 200 Hz to 300 Hz in the high pass. In this work the filtering bandpass was set to 100-250 Hz as in (93). In other works involving high frequency oscillations to detect epileptogenic brain areas, they filtered the LFPs in the 250-500 Hz band-pass. Figure 5.9 shows how different filtering values modify the LFPs of this study.

The filtering bandpass 100-250 Hz effectively made a game changing difference between the ripple and non-ripple events. Thereby, the resulting datasets to use in the training were sufficient to train the model with hihg accuracy. In Figure **??** it can be seen the difference between labels. The spiking signals from each dataset will provoke the firing specific neurons from the first layer of the SNN.

5.2.3 Conversion to Spikes

There are many possible ways to create a representation of the data with spikes (6; 94). In the context of detecting high frequency oscillations, which differ in shape from other neural patterns,

[a]

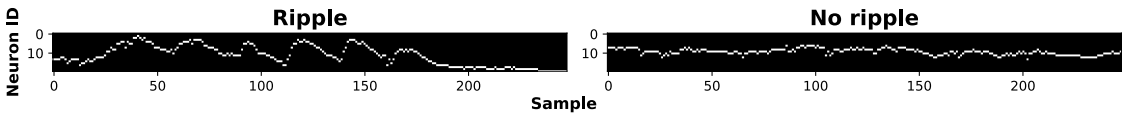


Figure 5.3: Unfiltered.

[b]

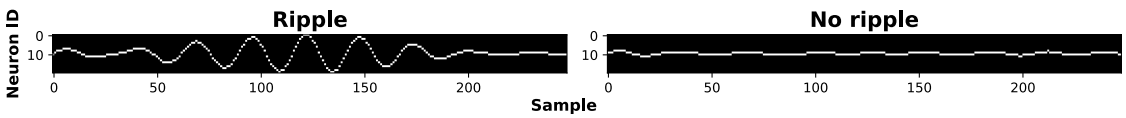


Figure 5.4: Filtered.

Figure 5.5: How filtering clears and improves the data.

the spikes may provide information about the shape. (6) applied an interesting conversion method which was based on UP and DN spikes. That is, if the processed signal crossed a threshold, a spike would be assigned to UP (positive) or DN (negative). That would provide information of the oscillatory frequency and the amplitude. Inspired by his work, this approach tried to add another degree of information to the spike signal. Instead of binarizing to UP and DN, the signal was discretized into a number "n" of values in the y axis. The number "n" could be fine tuned to see how many levels would be necessary to assess the shape of the signal (Figure 5.6).

The conversion method was based on discretizing the y-axis. That is, to have a number of steps for the heigh of the signal. Therefore, for each sample of the signal, a "1" was put in the corresponding height to match the shape of the signal. With that, the first layer of the SNN would have same number of neurons as number of steps. So, each neuron would receive an input (0 or 1) for each sample. The rationale of the approach was to force neurons to fire depending on the shape of the signal (see SNN training from Figure 5.10).

The equation for determining the position of the y-step was defined as follows:

$$y_{\text{val}} = (y\_s - 1) - \left\lfloor \frac{x}{cutoff} \cdot \frac{y\_s}{2} + \frac{y\_s}{2} \right\rfloor \tag{5.2}$$

Where *y_s* is the number of "*y values*", x is the value of the 1D signal, and cutoff is the cutoff amplitude[5], defined with the function:

$$cutoff = \mu(ripples_A) + \sigma(ripples_A) \tag{5.3}$$

Where *ripples_A* is a one-dimensional array containing the maximum amplitudes of all the ripples tagged from both recordings.

For achieving a ripples detector, the network should be able to distinguish between a ripple event and a non-ripple event. If the model would only be trained with ripple data but would never see non-ripple events it may not be able to differentiate between them. In this approach, a n-dataset[6] with the tagged ripples and another one with random non-ripple events was created from the recording sessions. Available code for the extraction of events and creation of n-dataset can be found at https://github.com/MarcosOriolPago/LAVA_SNN_ripples/tree/main/extract_Nripples.

SNNs work with spikes, which means, that the data must be reconverted into a spikes format to fit the network. Spikes, as pulses sent by neurons, are theoretically equal within them. The communication is based on the rate and specific times they are sent. To mimic a spike, the default value is set to 1, that is: 0 not spike, 1 spike. Following that rationale, signals were reconverted to a binary format 5.7.

[5] **Cutoff Amplitude** is the maximum value allowed for the y-axis discretization. Any value greater that the cutoff will be set to max neuron id (if positive) or min neuron id (if negative).

[6] N-dataset refers to neuromorphic dataset. That is a binary dataset for spike supported training.

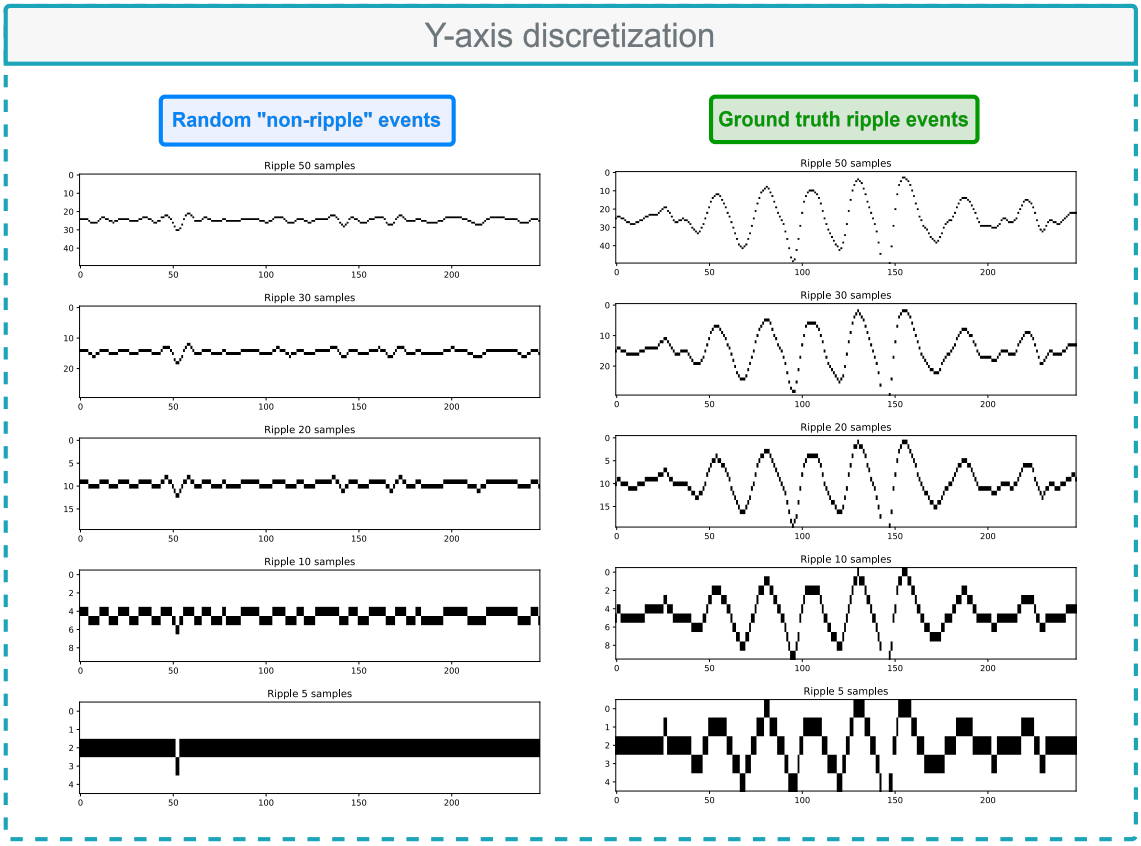


Figure 5.6: Same signal discretized with different y values. Ripple and non-ripple events need to be discretized to train the model. Each dot represents a spike that will make the corresponding input neuron from the network fire. The less values used, the more the neurons from each value will fire, and may learn faster. However, the cost of low level of discretization is the loss of resolution. The shape changes notably in lower values.

5.2.4 Training dataset

There are many possible ways to create a representation of the data with spikes (6; 94). In the context of detecting high frequency oscillations, which differ in shape from other neural patterns, the spikes may provide information about the shape. (6) applied an interesting conversion method which was based on UP and DN spikes. That is, if the processed signal crossed a threshold, a spike would be assigned to UP (positive) or DN (negative). That would provide information of the oscillatory frequency and the amplitude. Inspired by his work, this approach tried to add another degree of information to the spike signal. Instead of binarizing to UP and DN, the signal was discretized into a number "n" of values in the y axis. The number "n" could be fine tuned to see how many levels would be necessary to assess the shape of the signal (Figure 5.6).

5.2.4.1 Extracting the events

The training dataset needed to be a dataset with two labels: "Ripple", "Non-ripple". Each label would contain events (of a fixed time window) representing the label. The tagged events only

included ripple events, but not times with non-ripple activity. Therefore, they were set as random timings where for sure there was not a ripple nearby. On those timings, the half window was added to the right and the other half to the left, leaving a non-ripple event matching the restricted time window. For the ripple events, same method was applied from the middle of it. Next, the extracted events from the recording were bandpass filtered in the ripple band (100-250 Hz) and then converted to spikes (Figure 5.7).

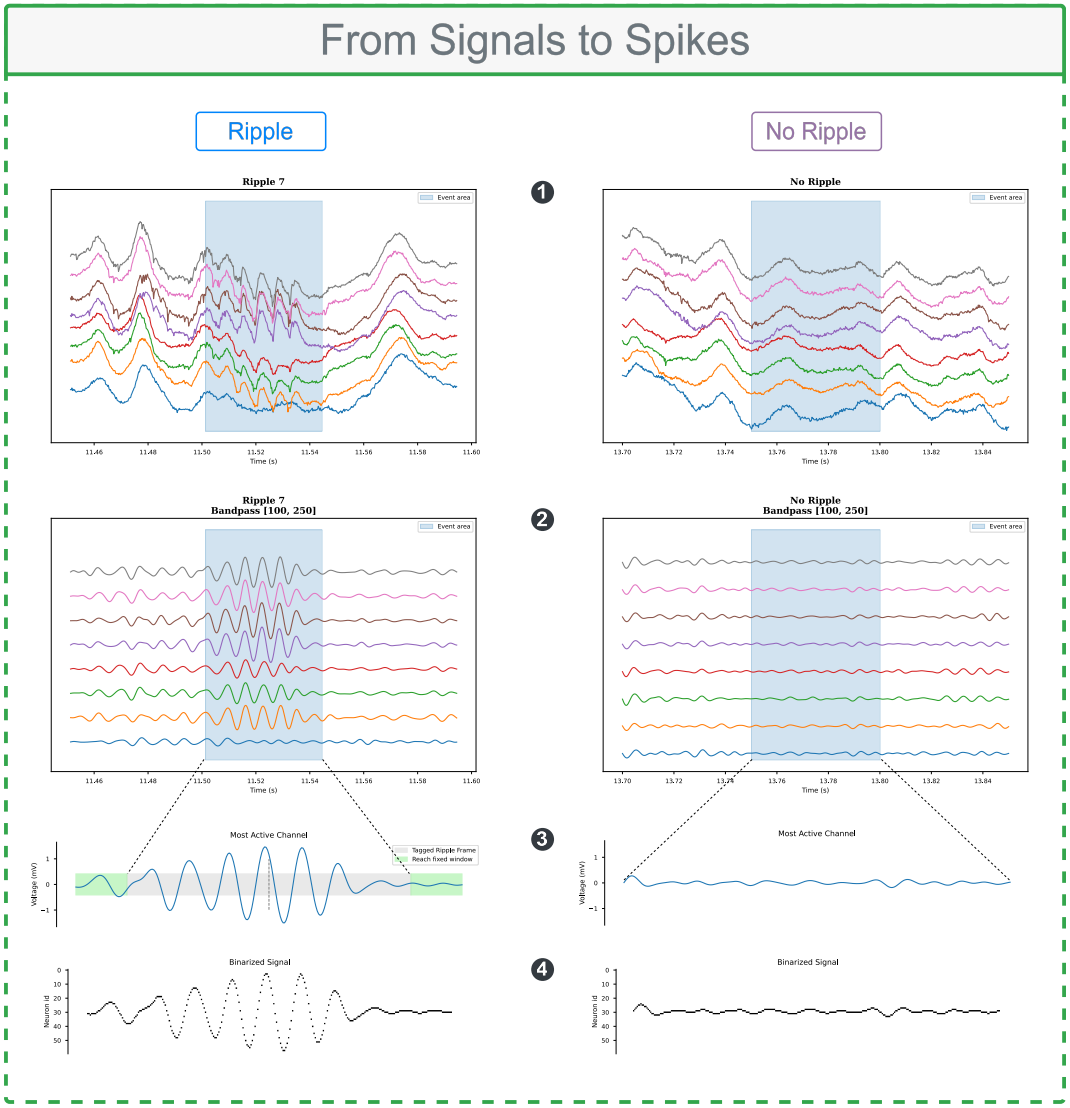


Figure 5.7: Workflow to build the n-dataset. 1) Events are located in the recording, 2) A Butterworth bandpass filter is applied in the range of 100 to 250 Hz, which is the oscillation speed of ripples, 3) highest amplitude channel is selected from the shank and the signal is adjusted in length to fit the previously set window size (in the case of no ripple, as they are not previously tagged, the window length is directly set), 4) the extracted signal is converted to spikes.

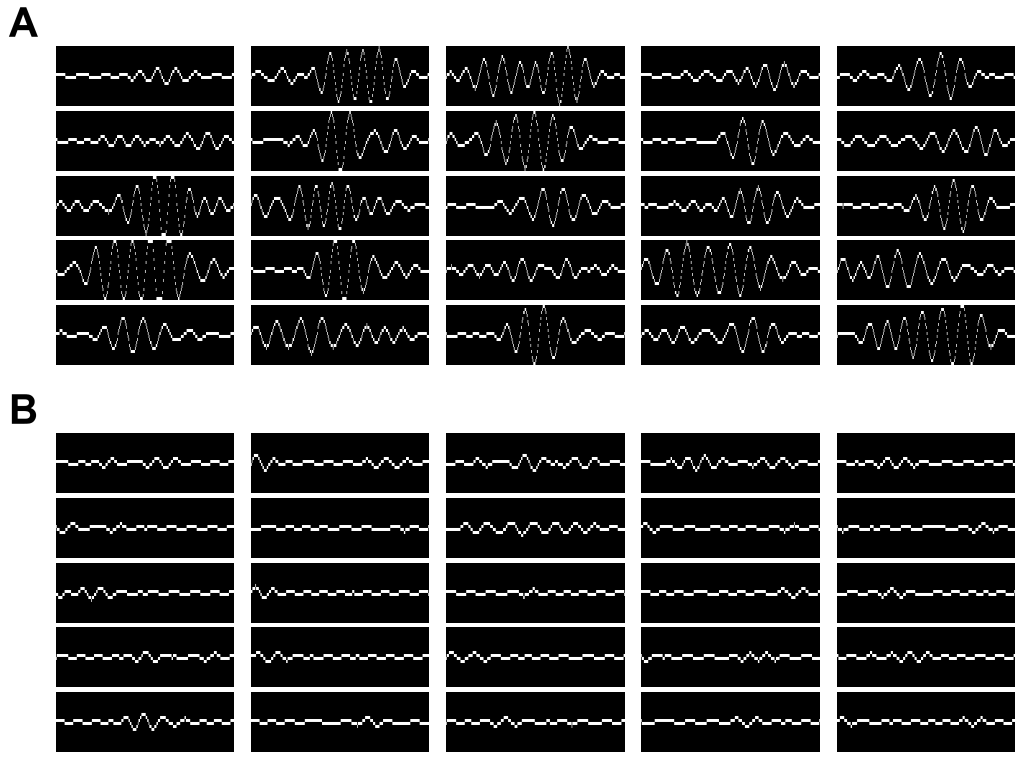


Figure 5.8: **Overview of the training n-dataset. A)** Ground truth ripple activity manually tagged by an expert. **B)** Non ground truth ripple activity selected randomly from the recording.

5.3 SNN Building and Training

Implementing SNNs in biosignals is very recent and has few online documentation. Developing a SNN from scratch can be a challenging task (95).

5.3.1 Options for building the network

At first, the recent publication of a machine learning toolbox for detecting ripples from liset M. de la prida on a similar dataset (96) pushed me to emulate their approach. In their work, they were detecting the ripples with different CNNs with different window sizes. Seeing a good performance achieved, the most logical way to proceed was to build an equivalent SNN. Reading the literature, it can be seen that SNN can be built from a previous ANN by copying the layer shapes and trespassing the weights (44). However, since the nature of SNNs are fundamentally different from ANNs, the networks obtained from the conversion are not as good as the original ANN. Furthermore, a SNN may need less nodes to learn and better performance can be obtained if trained from scratch. Overall, some trials were done to reconvert the models from the liset publication with two different frameworks: Nengo (https://github.com/nengo) and SNN Toolbox (https://github.com/NeuromorphicProcessorProject/snn_toolbox). As the trials were unsuccessful, the approach was discarded.

Figure 5.9: **Finding the optimal filtering bandpass**. Clear Ripple, middle clear ripple and non-ripple events are filtered within three bandpass filters. Middle clear ripples are some events found in the tagged dataset which do not show clear ripple activity. The three ranges make a detectable difference between ripples and non-ripples, but not between middle-clear ripples and non-ripples. The most differential and consistent bandpass is the one seen in the literature (100 – 250 Hz).

Then it was decided to build and train the network from scratch with lava-nc. Lava, as a neuromorphic framework developed by intel, is mainly developed for widespread engineering applications. Furthermore, their documentation is not very complete. The most similar example found consisted of a n-mnist detector (see on https://github.com/lava-nc/lava-dl tutorials). There, they would go through the whole process of building and training the network. N-mnist dataset consists of binary arrays of similar shape than the ripples. Therefore, the approach was supported by the workflow observed in that example.

The learning module used was based on SpikeRate. Learning from spike rate means using the rate of the output neurons to make predictions. Given a window of spikes (representing an

input), the number of times the output neuron fired in response to that input is used to assess the prediction.

The SNN was defined and trained with lava. Lava builds SNNs supporting in SpykeTorch python library, so the models are trained as pytorch models. Then lava converts them to HDF5, which is a file format that later can be mapped to neuromorphic hardware (Figure 2.9).

5.3.2 Choosing the architecture

The inner architecture is foundational for the output of the network. The number of nodes and layers to use can vary with many factors. A heavier network with more nodes will probably require more training data to being able to optimize all the weights, thus learning complex patterns. Otherwise, a lighter network will train easier but with the limitation of simple pattern learning. The architecture of ripple-detector CNNs developed by liset labs were quite simple (low neuron

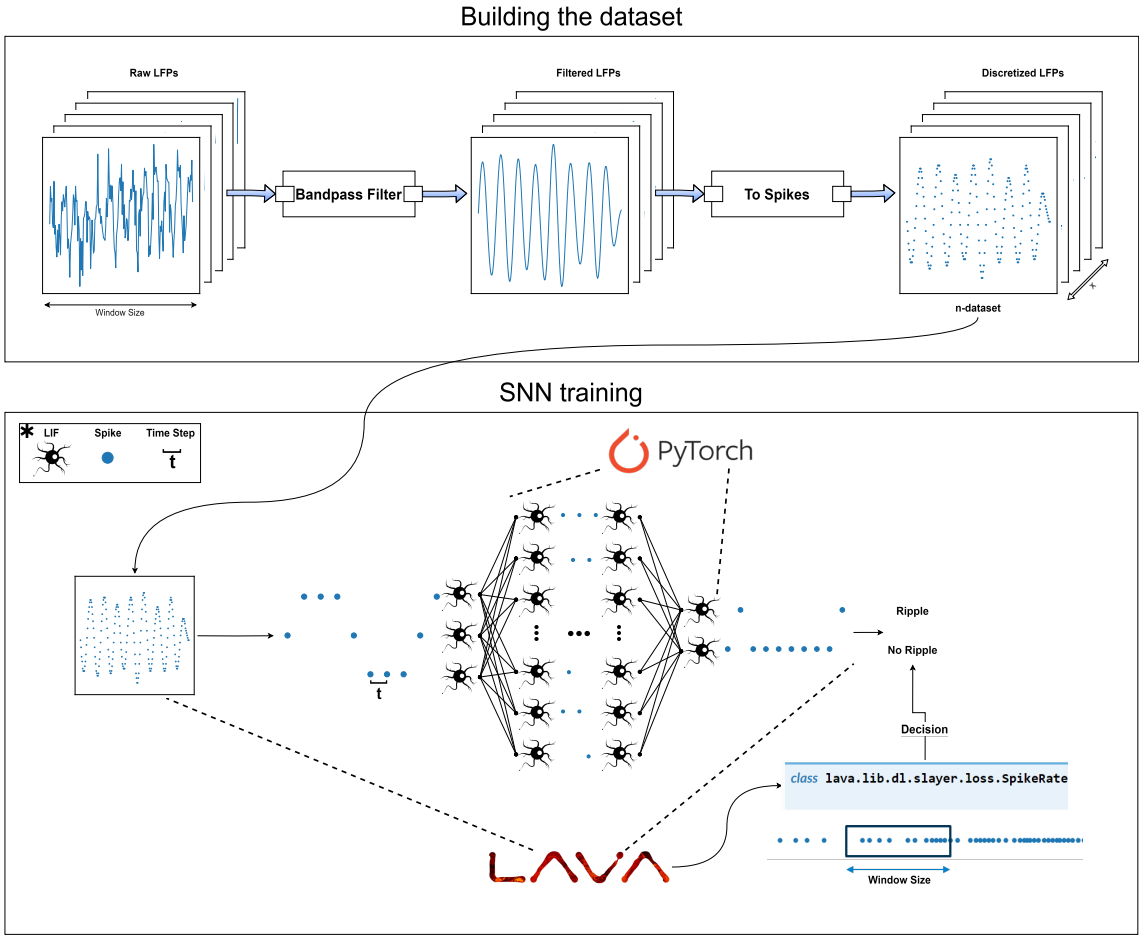


Figure 5.10: Data pocessing and model training pipeline. The n-dataset consisted on spiking representation of the signals conserving the shape of the filtered LFPs. The network must have same number of input neurons as number of y-steps from n-signals. For each time step, only one neuron from the input layer would receive a spike and the network would compute an output on the output neurons.

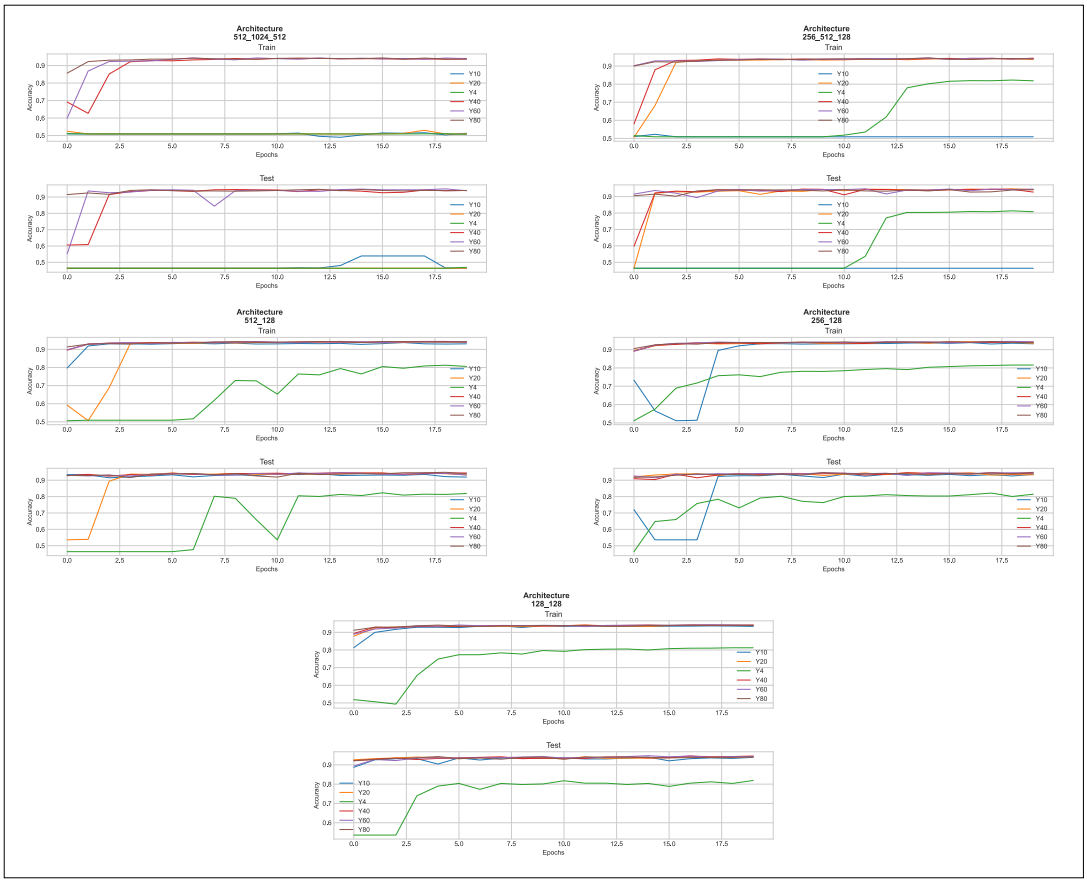


Figure 5.11: **In search of the optimal architecture.** Five trials were tested on n-datasets with different "y values" (4, 10, 20, 30, 40, 60, 80). Based on the results obtained, models with 3 dense layers had more difficulty for learning the patterns, and probably needed more data to reach high accuracy values. Lighter models, with 2 layers seemed to learn the patterns faster and needed less resolution for the y-discretization, that is, needed less neurons as input to detect the ripples.

params), so some trials with simple architectures were done to see performances 5.11. Taking into account the small size of the available data for training (1794 samples per label),it is reasonable that the model should require few nodes to learn. The test was to see which order had the capacity to distinguish the patterns.

Surprisingly, all of the architectures succeded in differentiating the two labels with high confidence. It can be observed however, that heavier models struggled to learn with low Y-resolution signals, whereas lighter ones didn't. The n-dataset with 4 levels (Y4) could only reach 80% accuracy. Probably because in the training dataset there was a part of the population that when using low resolution it resembled the other label.

Seeing that smaller models were sufficient to learn with good confidence, the chosen architecture was 256_128 as an balance between computing time and performance throughout the resolutions. That is, in addition to the input and output, two dense layers with a decreasing number of neurons. Layers had 256, 128 neurons respectively (Figure 5.12). The neurons employed were

current based (CUBA) LIF neurons from the built-in neurons of lava-dl[7]. The neuron parameters for the cuba LIF neurons were defined as follows:

Parameter	***Description***	***Value***
Membrane Threshold	*The voltage level required to trigger a neuron spike.*	***1.25 mV***
Current Decay	*The rate at which synaptic current decreases.*	***0.25 ms*$^{-1}$**
Voltage Decay	*The rate at which membrane potential returns to resting state.*	***0.03 ms*$^{-1}$**
Tau Grad	*The time constant for the gradient of membrane potential or synaptic current.*	***0.03 ms***
Scale Grad	*A dimensionless factor that scales the gradient.*	***3***

Table 5.1: Neural parameters used for the LIF neurons

5.3.3 Building the network

Once familiarized with the data, some approaches were explored to know which one could fit better. As first part of the approach, "Principles of computational modelling in neuroscience" (31) gave good insights of which kind of neurons should be used for the detector. LIF neurons were the most appropriate due to their balance between biorealism and light computation. For applied NC not involving neural simulation they are totally fine. Other neuron models such as izhikevich neurons have similar simplicity in computation but show less behaviour mimicry.

Next, the network had to be defined and built. Following the workflow from n-mnist example in lava-dl github repository, the network was built and trained with ripples n-dataset. However, very bad performance was achieved with the architecture of n-mnist classifier. Next, other architectures were tested to get a glance of which architecture could perform better. The architectures ranged from 128 – 1024, and from 2 – 3 dense layers. Each architecture was run for n-datasets with different "y_samples" (4, 10, 20, 40, 60). With the obtained results, it was determined that the most consistent learning outcomes were given by 256_128 architecture. That is, the input layer, then a dense 256 LIF neuron layer, followed by a 128 LIF dense layer and finally, the output layer, which is 2 neurons referring to ripple and no-ripple. Trained models and code for the training can be found at https://github.com/MarcosOriolPago/LAVA_SNN_ripples/.

The workflow for training a network used by lava consists in training the Spiking Neural Network with pytorch, and then converting it to a format compatible with neuromorphic hardware. Lava has builtin functions to convert the pytorch SNN network into a format that can be compiled into a lava process and deployed to neuromorphic hardware, namely, hdf5 (see Figure 2.9).

[7] https://lava-nc.org/lava-lib-dl/slayer/neuron/neuron.html#module-lava.lib.dl.slayer.neuron.cuba

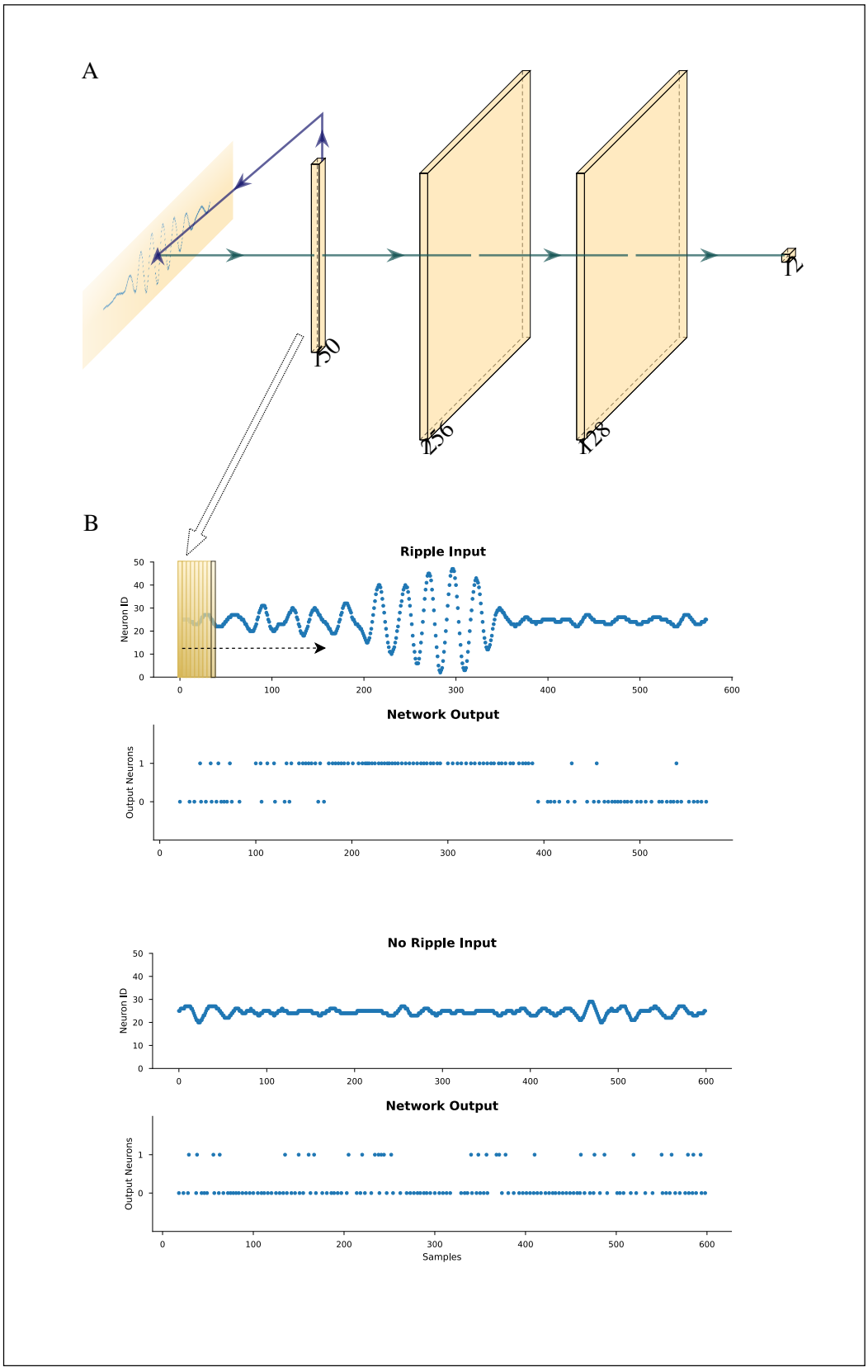


Figure 5.12: **Network Architecture Overview. A)** The network has 2 dense layers, an input, and an output. The window is set as an input and for each sample, the network computes an output. **B)** Difference in the network output between ripple and no-ripple.

5.3.4 Training

Training was done with SLAYER learning method, with an error function based on the spike rate (ref) of the output neurons. The training loop was set to 50 epochs. The assessment of the training performance was set to the best epoch.

5.4 Network validation

Once the model was trained, it was convenient to validate it on real data, and not just on training data. To do so, the network inferenced throught the whole recordings. Performance metrics were obtained by comparing the predictions where the network thought there was a ripple, with the ground truth ripples tagged by the expert.

5.4.1 Performance metrics

- ***Precision***

$$precision = \frac{correct\_predictions}{total\_predictions} \tag{5.4}$$

- ***Recall***

$$recall = \frac{correct\_predictions}{total\_ground\_truth} \tag{5.5}$$

- ***F1 Score***

$$F1\_score = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall} \right) \tag{5.6}$$

5.4.2 Comparison with CNN

Both networks were validated with network validation 5.4 rationale, and the performance metrics were discussed. A python notebook inferencing on the data and providing performance visualization can be found at https://github.com/MarcosOriolPago/LAVA_SNN_ripples/. Both SNN and CNN were optimized in the inference parameters to achieve the maximum F1 score. For the CNN, it turned out to be 0.3 threshold confidence for the detection, and the SNN 30 spikes for a 130 samples moving window (as explained in 5.4.3).

5.4.3 How to determine a correct prediction

As mentioned earlier, ground truth data from the recordings consisted of an array of (start, stop) times when events occurred. Bearing this in mind, for the validation test, the predictions of the network were forced to be in the same format.

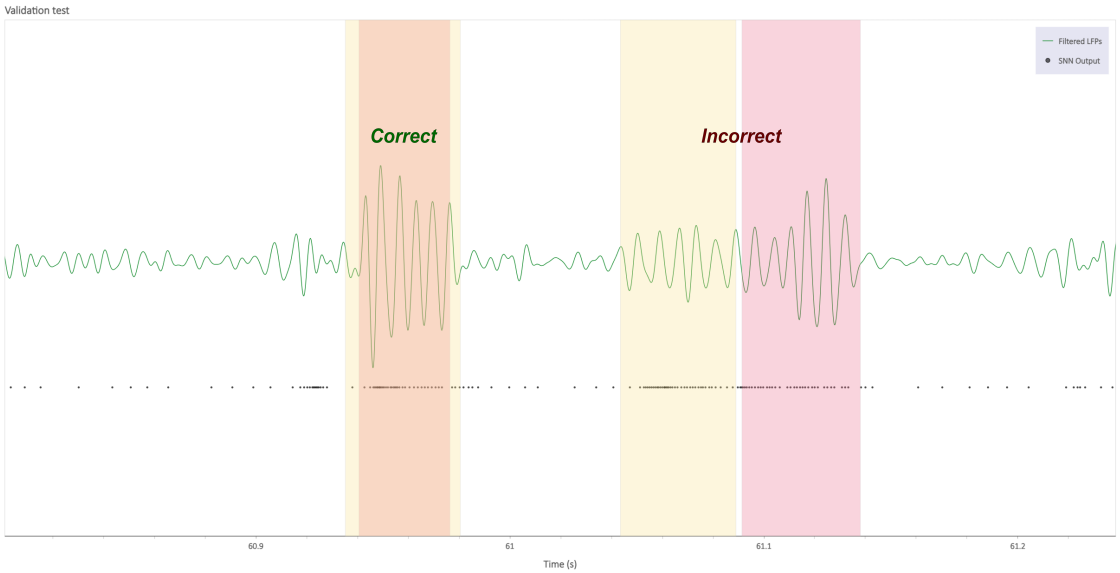


Figure 5.13: **Correct vs incorrect prediction.** Red fill correspond to ground truth, and yellow fill to SNN prediction. If the intersection of union is greater than 20%, it is considered correct.

The trained SNN had as an output spikes from the two output neurons. A postprocess phase was done to assess time intervals where the network had predicted a ripple based in the spike rate. Thereby setting a kind of spike rate threshold. An overlapping moving window would be counting the spikes and assessing the rate, and if the rate was above a threshold, the window would set the "start" of the prediction. Then, when the rate would go below the threshold, the "stop" of the prediction would be set. A moving window of 130 samples and a threshold of 30 spikes per window was determined to be the better performing.

Once with the SNN prediction timings, the performance metrics could be calculated. To match a prediction as correct, it should match with one of the ground truth ripples. Consequently, to determine a match, it was considered to have a 20% timing coincidence.

5.5 Running on loihi2 Simulator

For running in the simulation, the input data, network and output were defined as processes as in Figure 5.14. The ports were connected following the flow logics. Notebook with the code for running in the simulator is available at https://github.com/MarcosOriolPago/LAVA_SNN_ripples.

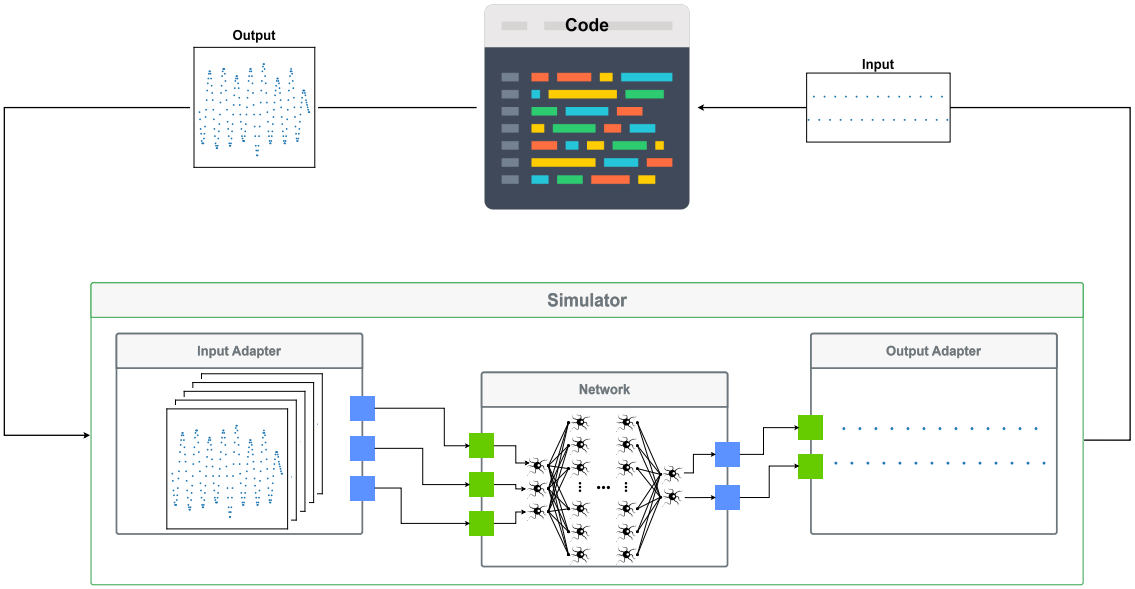


Figure 5.14: **Running Network with Lava.** Processes need to be built and connected between them to enable running on simulator or hardware. These processes are compiled to perform the same action but communicating with events, much more similar to brain structures.

Chapter 6

Results & Discussion

6.1 SNN Training

The network showed an unexpected high learning rate. Since the first 2 epochs the accuracy was raising up to >90% for most of the resolutions and network architectures. This may be explained by the low number of nodes, and the differentiability of the inputs between labels. As shown in 5.5, the difference between ripple and non-ripple after the filtering is very easy to spot, and the data is clean, avoiding confusion. After all, if different neurons from the input layer are being stimulated in each label, the model will easily learn to classify them.

6.1.1 Minimum resolution

Seeing that the model had no difficulty in learning the pattern, training for different signal resolution in the y-axis was done. Interestingly, in figure 6.4 it can be observed that 2 values are too low to learn a pattern, but by raising it to 4, the SNN was able to achieve an 80% train and test accuracy. Then, up to 10 it stabilized near 94% accuracy. In figure 5.6 it can be seen how the shape of the signal changes depending on the level of discretization.

[a]

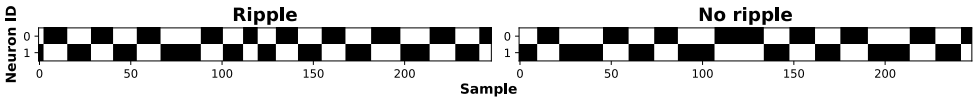


Figure 6.1: 2 level discretized signal

[b]

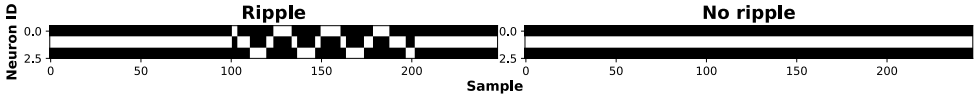


Figure 6.2: 3 level discretized signal

Figure 6.3: Y-axis Discretization Limit. Same signal in Y2 and Y3 resolution.

The reason why 2 values were too low is due to the fact that the algorithm to set the value for each sample makes values indistinguishable for only 2 levels (Figure 6.1).

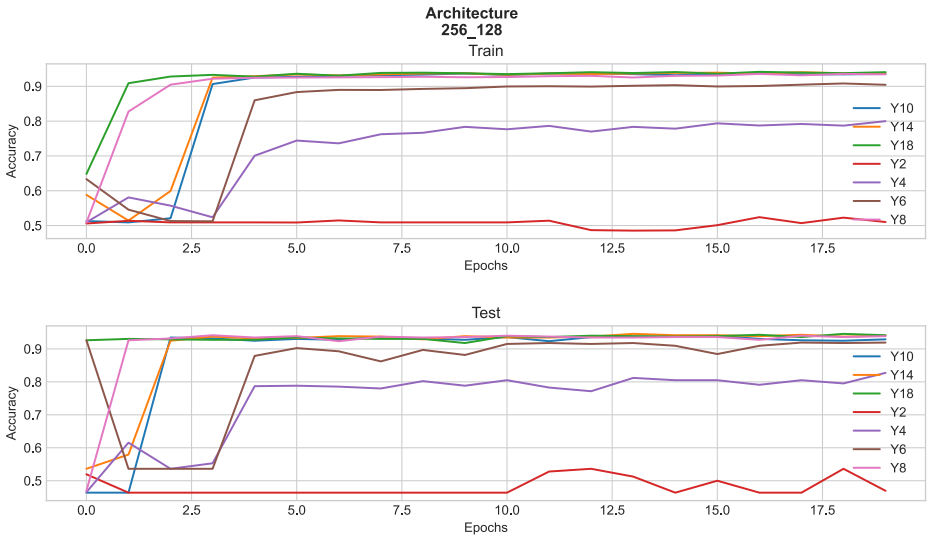


Figure 6.4: **Minimum Y-axis resolution for learning.** Train and test accuracy values throughout training epochs for different levels of discretization in the y axis of the spiking signals.

As it can be seen in Figure 6.2, signals are already very differential among populations. In fact, a simpler network architecture can easily learn to differentiate them. The problem is that the loss of resolution not always fits well with lower amplitude ripples, therefore loosing accuracy in not so clear ripple activity. A 50_50 network, with 3 levels of resolution can learn up to 70% accuracy (Figure 6.5).

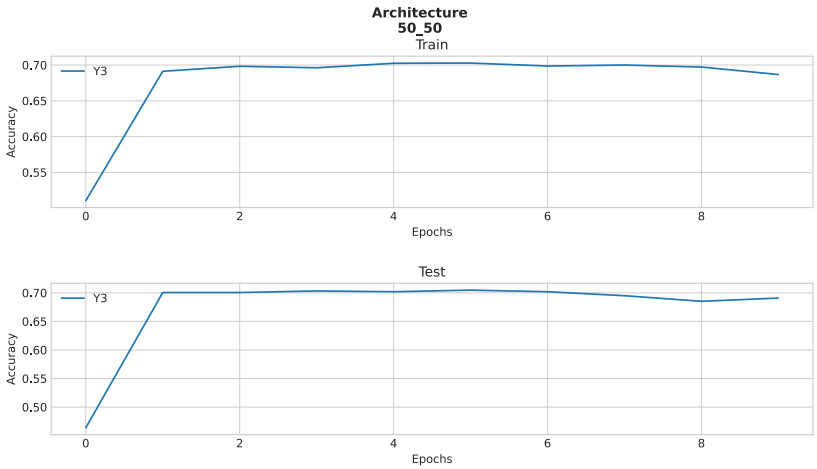


Figure 6.5: 3 levels are able to reach 70% accuracy.

6.1.2 Higher resolutions

For Higher resolution datasets, the accuracies were stuck at ~94%. The reason of why higher resolution datasets got higher accuracies could be due to the network learning not so clear patterns from ripples that with lower resolutions are lost.

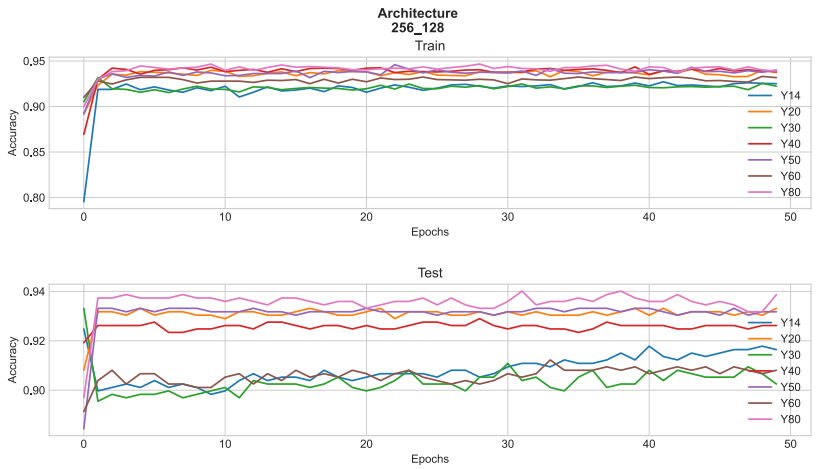


Figure 6.6: **Accuracies on higher resolution datasets**. Apparently, the higher the resolution, the higher the accuracy. However, there is the limit near 95%, where the network seems unable to learn more from the data.

6.2 Optimized Model

The selected resolution of the spiking signals was 50 (50 levels to assess the y value for each x sample). This number of samples seemed a reasonable between simplicity and performance. The accuracy values obtained with slayer training (as in listing 2.1) were the following.

- Train: 94.59%
- Test: 94.56%

As it can be seen, the model has reached >90% accuracy in the second epoch. The epoch 0 appears to be at 90% because epoch 0 should be 50%, so directly shows after training on epoch 0.

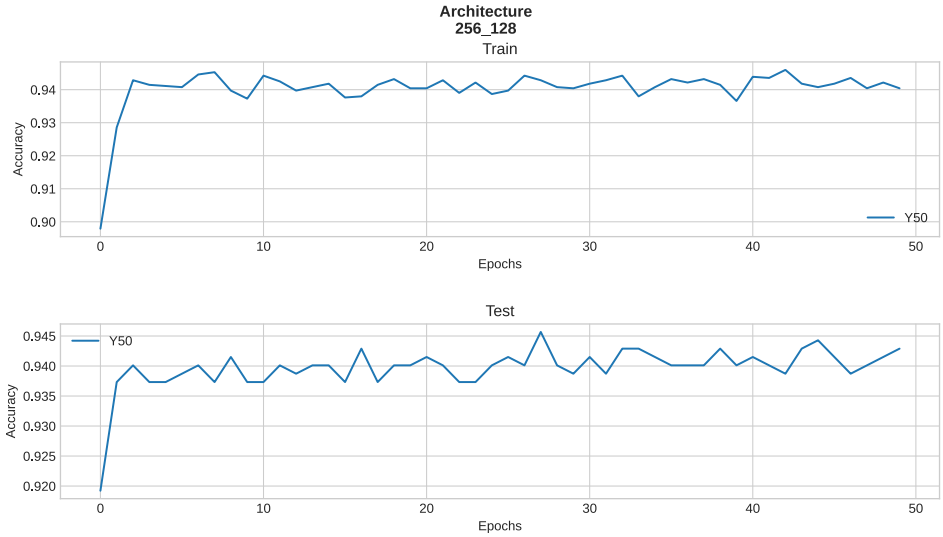


Figure 6.7: Performance of the chosen network for 50 epochs

.

6.3 Validation on loihi simulator

The validation gives insights of the sensitivity of the networks to the signals and helps to assess the robustness across the whole recording. Following the validation test defined in 5.4 for both CNN network previously reported in (96) and the SNN, performance metrics showed in Figure 6.9 were obtained. The SNN successfully run in the loihi simulator (see notebook). The output of the network obtained from the simulation was processed to translate it into (start, stop) predictions.

Observing Figure 6.9, it can be noticed that the SNN has similar outcomes in Som_2 dataset, and better ones in Amigo2. Though, F1 score values do not reflect the accuracy obtained from the training.

6.3.1 SNN vs CNN

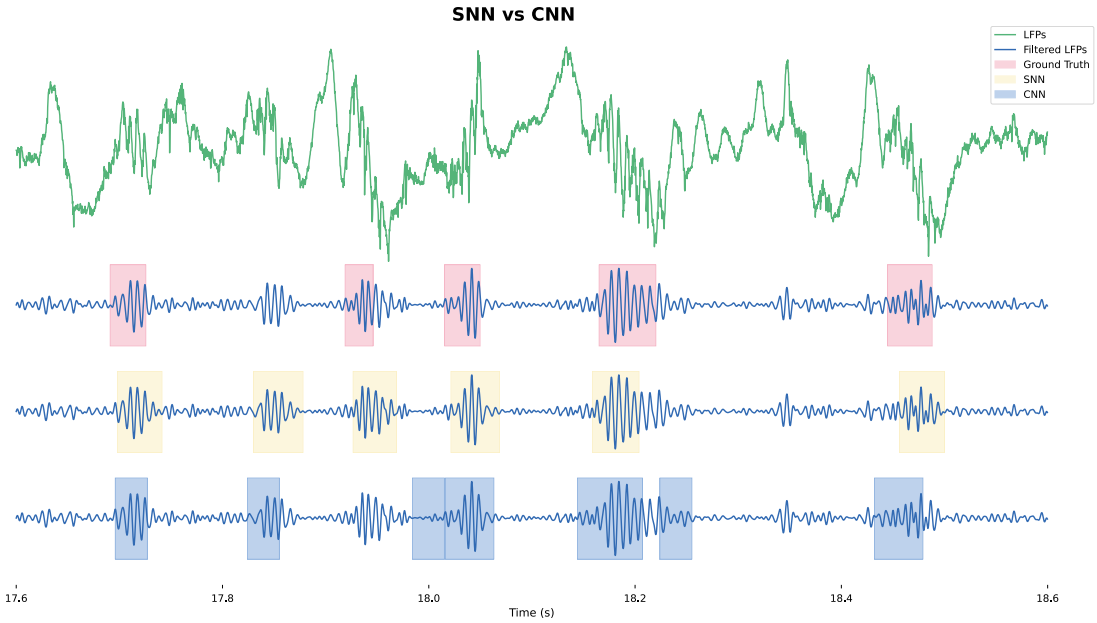


Figure 6.8: Comparison of the predictions of both models and the ground truth. Visually, the SNN has a more consistent pattern of detection, showing more sensibility, by reacting at all the high frequency oscillating events. The CNN also shows good response, but misses one of the events.

6.3.2 Evaluation of ground truth data

Performance of the network in the validation decreases because, training dataset represent a 2% of the dataset where only ripple activity happens. However, through the whole recording there are other events which may be similar oscillation frequencies and may confuse the model. As in Figure 5.8, ripple activity does not always show that great increase in amplitude compared to negative events. These differences among the dataset are probably causing that the training accuracy cannot improve 94% accuracy. This does not mean that those events are not ripples, but that the network is not capable to distinguish them with this filtering approach. Other considerations should probably be needed to take into account when setting the approach. Improving this limitation will need to be addressed in further studies.

6.3.3 Analyzing validation False positives

The parameter affecting the F1 score of both models is clearly the precision, which is the number of correct predictions over the total predictions. This means that the models are predicting much more ripple events than should be. If we take a look to some examples we see that effectively, some "False positives" are very similar to the ground truth data from the dataset (Figure 6.10).

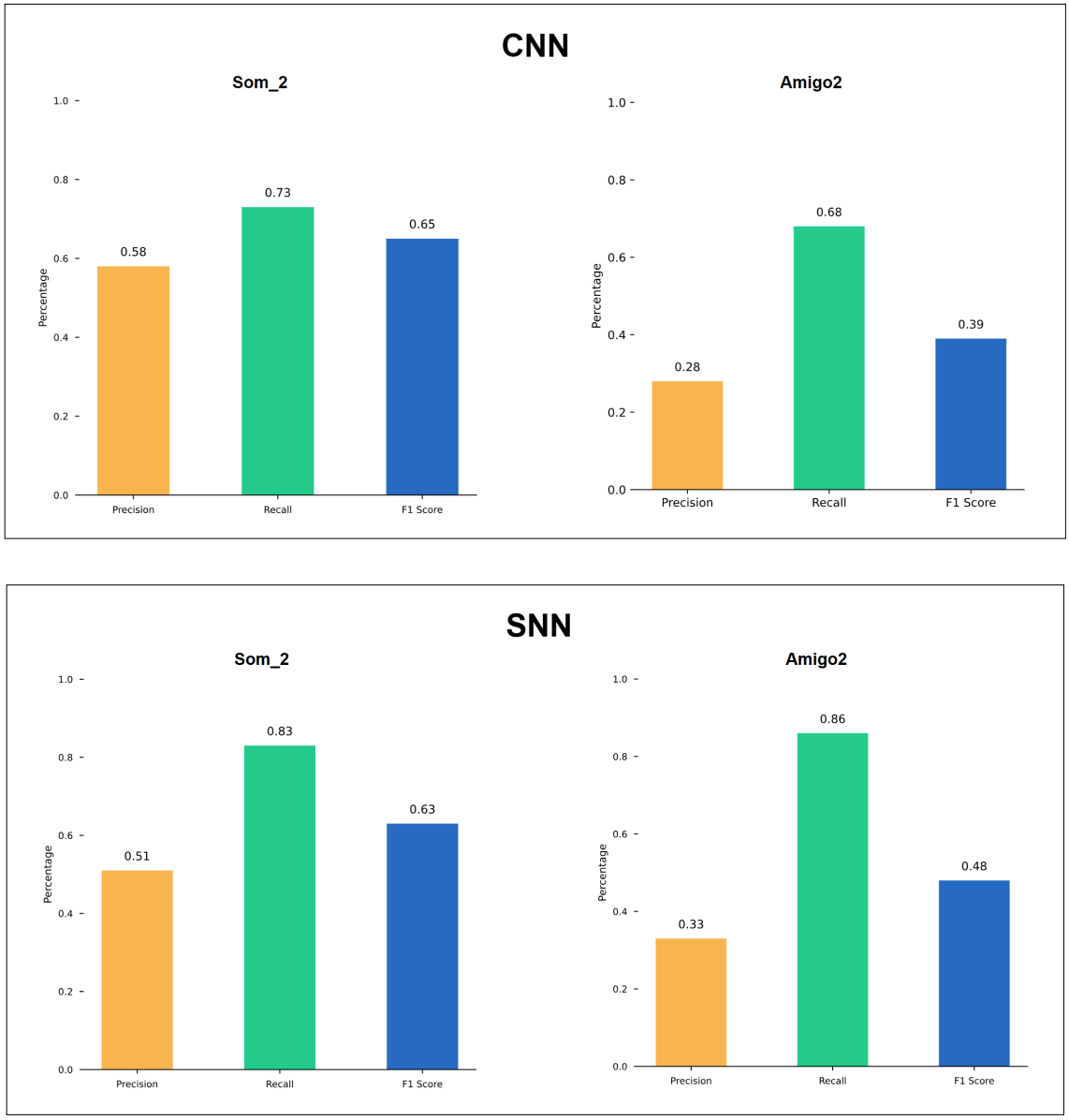


Figure 6.9: Validation of CNN and SNN on datasets used for training.

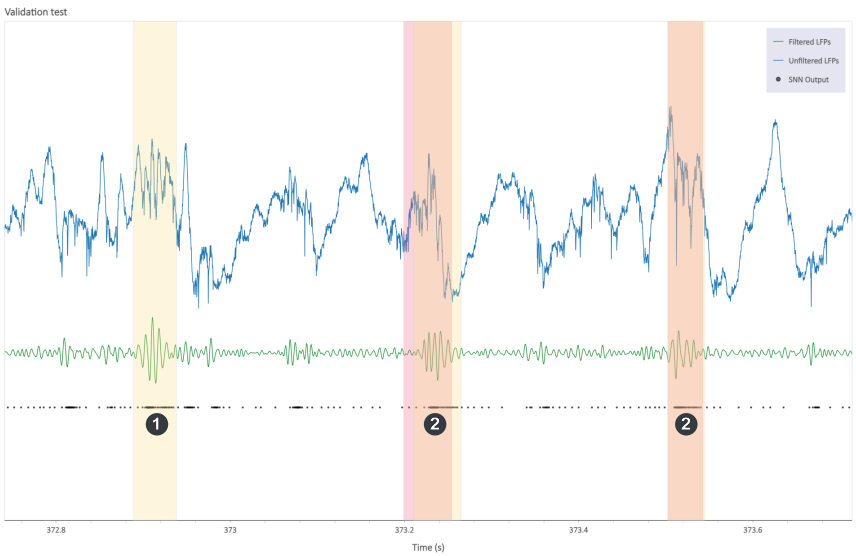


Figure 6.10: Yellow fills correspond to SNN prediction and red fills to ground truth. **1)** Event classified as false positive, but showing clear ripple activity (3-9 high amplitude oscillation in the ripple band). **2)** True positive events matching the ground truth.

Seeing that the false positives affecting the metrics on the validation are similar to real ground truth, two hypothesis can be done.

- **Not all the ripple events were tagged in the recording.** Thus not being appropiate for the validation.
- **Ripples have more complexity not so easy to spot with our vision.** However, the definition of Ripples from (1) states that a ripple is any high amplitude oscillation pattern in the ripple band, including 3-9 ripples (4.3.1).

6.4 Final Remarks

The SNN showed good sensibility to the data it was trained with, meaning that spiking neural networks are as competent as other ANNs. To build a better performing model, a more precise definition of SWRs should be done, and the data should match with the definition. Otherwise, it is very hard not to raise false positives through the validation process.

Regarding the behaviour of the network, it can be seen (Figure 6.12) that the output is clearly sensitive to changes in amplitude withing a certain frequency in the filtered LFPs. It makes much sense because the training data was prepared distinguishing the labels primarily on the amplitude and frequency of the signal. Thereby, seeing that behaviour is good feedback that the network learned the differences. Consequently, for future works, being able to include other feature that overexpress the amplitude of the LFPs in ripple activity would increase significantly the performance of the model.

Chapter 7

Conclusion and future directions

SWRs are an interesting type of high frequency oscillation involved in vital processes of cognition, learning and memory. Thereby being able to correctly detect them can help to understand these processes and advance in innovation for NDs treatments. Neuromorphic computation overcomes several limitations present in current computing frameworks, holding a path for innovation in countless applications. In this project, a specific implementation of AI and NC has been successfully achieved. A SNN was trained to detect SWR events with significant performance, and a simulation of the model running on state-of-the-art neuromorphic hardware was done to assess the validation.

7.1 Perspective

The value provided by NC in real-time bio-signal applications remains on making the system compatible and comfortable for the patient. It would make no sense to use neuromorphic chip for running the SNN and a digital processor for filtering and converting the signal to spikes. Because all the energy saved by the neuromorphic chip would be used by the digital processor, in addition to reducing the speed. Thereby loosing the attractiveness of low-power, non-heating, compact, and real-time system.

This work does not intend to provide a final solution for a device ready for the patient. Rather, it is an approach using cutting edge technology which may help to the development of future projects. The work provides tools to easily use and implement SNNs on a neuromorphic framework. The next step would be deploying it and demonstrating low-power and faster computing times. Time constrains reduced the scope of the project, and it could not be done in neuromorphic hardware. However, results proof that SNN are capable to achieve efficient detection of SWRs in the LFPs of mice. Which can be extrapolated to other biological signals, making NC interesting for human-machine interactions.

7.2 Next steps

The methodology proposed in this thesis was thought to be compatible with a future fully implementation in a closed loop system for sensing and actuating in neuronal population. This implementation, in order to leverage the properties of neuromorphic chip loihi2, must be fully analogue (including the preprocessing phase). That is building a electronic system with measuring electrodes, the neuromorphic chip, and final stimulating electrodes.

7.2.1 Analogue Filtering

In order to implement this approach, a bandpass filter must be used to filter the measured signal in real time. For that an analogue integrated circuit of the filter can be used (Figure 7.1).

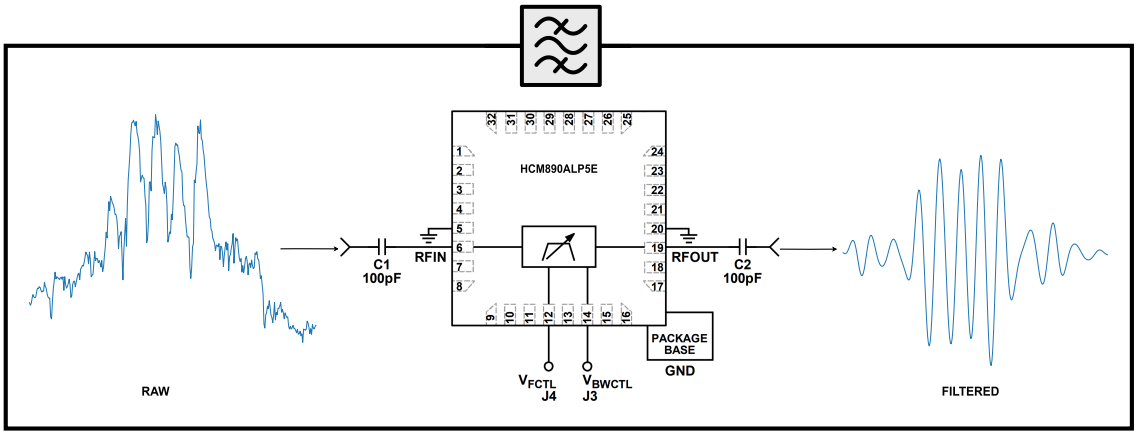


Figure 7.1: The integrated circuit can naturally filter the signal in real time (filter ref.: https://www.mouser.es/datasheet/2/609/HMC890ALP5E-1487476.pdf).

7.2.2 Conversion to spikes

The conversion to spikes process may seem to only be possible perform with digital computation. However, the foundation of the process is to discretize in the y-axis. So basically it is the same as doing analogue to digital conversion. It is therfore, only needed a digital conversion with the appropriate number of bits to covert the input signal into the desired number of steps. The approach proposed in this work was using 50 steps for discretizing the signal. However, performances are very similar since >14 steps. A 4-5 bits could be used to achieve 16-32 resolution (Figure 7.2).

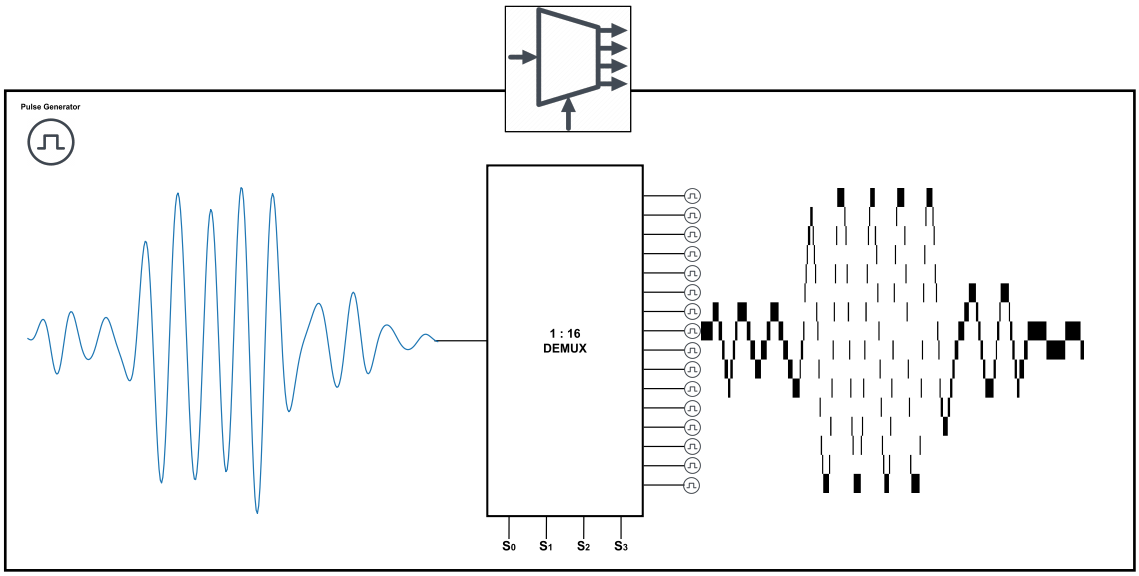


Figure 7.2: The demux stratifies the input analogue values setting them to a value. If a pulse generator is connected to each of these outputs, they can be used to feed the first layer of the SNN.

7.2.3 Overview

Finally, once achieved signal filtering and spike conversion, the spikes can feed the SNN, and the whole system would be nearly analogue (loihi2 has a small microprocessor for orchestrating the cores).

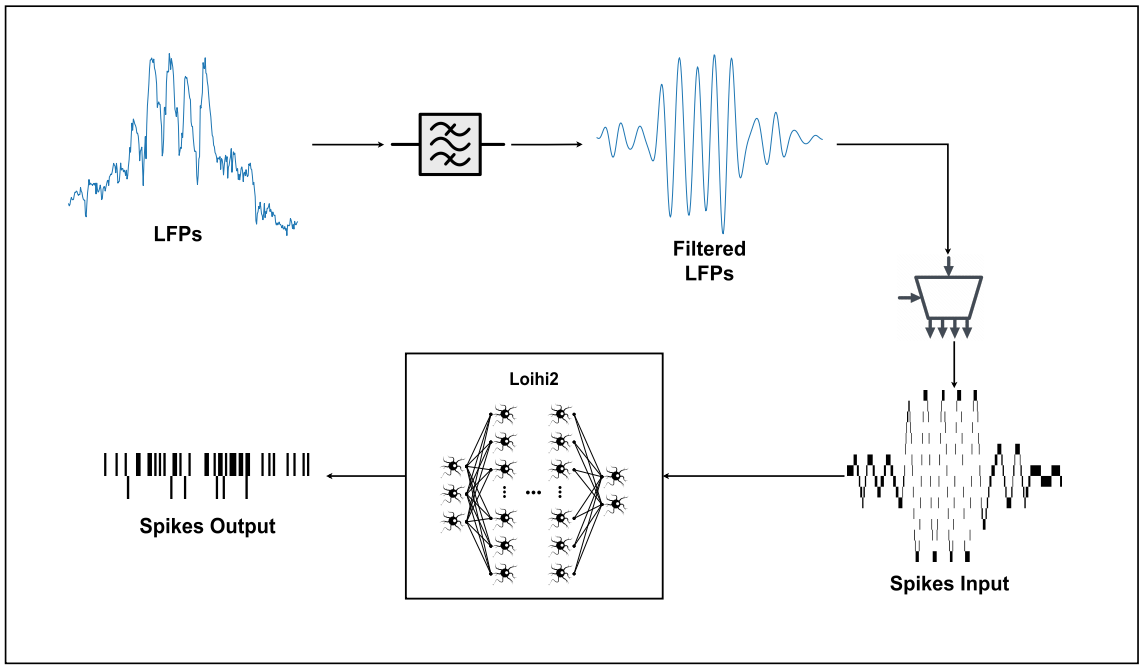


Figure 7.3: Fully analogue preprocessing before feeding the neuromorphic chip, where the SNN will run.).

These kinds of systems offer highly efficient computation and significantly faster performance compared to digital systems. Although modern microprocessors can handle complex tasks autonomously, they often require cooling systems and consume more energy. Additionally, they may not always be suitable for real-time applications.

Given these considerations, this further system based on the approach proposed in this work, could interact more naturally with biological systems, such as for in vitro cell monitoring, head implantable deep brain stimulation (DBS) devices, pacemakers, and blood glucose monitoring for diabetics. Adapting this system to specific data requirements in each application is necessary, but its potential to enhance healthcare outcomes is substantial and motivational.

References

[1] G. Buzsáki, "Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning," *Hippocampus*, vol. 25, pp. 1073–1188, 10 2015.

[2] G. B. . J. J. Chrobak, "Synaptic plasticity and self-organization in the hippocampus," 2005. [Online]. Available: http://www.nature.com/natureneuroscience

[3] L. R. Squire, L. Genzel, J. T. Wixted, and R. G. Morris, "Memory consolidation," *Cold Spring Harbor Perspectives in Biology*, vol. 7, 8 2015.

[4] E. A. Jones, A. K. Gillespie, S. Y. Yoon, L. M. Frank, and Y. Huang, "Early hippocampal sharp-wave ripple deficits predict later learning and memory impairments in an alzheimer's disease mouse model," *Cell Reports*, vol. 29, pp. 2123–2133.e4, 11 2019.

[5] Z. H. Zhen, M. R. Guo, H. M. Li, O. Y. Guo, J. L. Zhen, J. Fu, and G. J. Tan, "Normal and abnormal sharp wave ripples in the hippocampal-entorhinal cortex system: Implications for memory consolidation, alzheimer's disease, and temporal lobe epilepsy," 6 2021.

[6] K. Burelo, M. Sharifshazileh, N. Krayenbühl, G. Ramantani, G. Indiveri, and J. Sarnthein, "A spiking neural network (snn) for detecting high frequency oscillations (hfos) in the intraoperative ecog," *Scientific Reports*, vol. 11, 12 2021.

[7] A. Sanaullah, C. Yang, Y. Alexeev, K. Yoshii, and M. C. Herbordt, "Real-time data analysis for medical diagnosis using fpga-accelerated neural networks," *BMC Bioinformatics*, vol. 19, 12 2018.

[8] T. C. Stewart, T. DeWolf, A. Kleinhans, and C. Eliasmith, "Closed-loop neuromorphic benchmarks," *Frontiers in Neuroscience*, vol. 9, 2015.

[9] D. L. Castro, M. Aroso, A. P. Aguiar, D. B. Grayden, and P. Aguiar, "Disrupting abnormal neuronal oscillations with adaptive delayed feedback control," *eLife*, vol. 13, 3 2024.

[10] J. E. Fleming, E. Dunn, and M. M. Lowery, "Simulation of closed-loop deep brain stimulation control schemes for suppression of pathological beta oscillations in parkinson's disease," *Frontiers in Neuroscience*, vol. 14, 3 2020.

[11] Z. Wang, J. Wang, X. Shi, Z. Zhu, P. Chen, and H. Peng, "Electronic neurons for a new learning paradigm," *Advanced Healthcare Materials*, vol. 12, 7 2023.

[12] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, pp. 607–617, 11 2019.

[13] C. MEAD, "Neuromorphic electronic systems," *IEE Invited paper*, vol. 78, 1990.

[14] F. D. Broccard, S. Joshi, J. Wang, and G. Cauwenberghs, "Neuromorphic neural interfaces: From neurophysiological inspiration to biohybrid coupling with nervous systems," 6 2017.

[15] M. Ziegler, C. Wenger, E. Chicca, and H. Kohlstedt, "Tutorial: Concepts for closely mimicking biological learning with memristive devices: Principles to emulate cellular forms of learning," *Journal of Applied Physics*, vol. 124, 10 2018.

[16] S. E. Bibri, A. Alexandre, A. Sharifi, and J. Krogstie, "Environmentally sustainable smart cities and their converging ai, iot, and big data technologies and solutions: an integrated approach to an extensive literature review," 12 2023.

[17] N. Winter-Hjelm, Åste Brune Tomren, P. Sikorski, A. Sandvig, and I. Sandvig, "Structure-function dynamics of engineered, modular neuronal networks with controllable afferent-efferent connectivity," *Journal of Neural Engineering*, vol. 20, 8 2023.

[18] K. Yamazaki, V. K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," 7 2022.

[19] M. B. Milde, S. Afshar, Y. Xu, A. Marcireau, D. Joubert, B. Ramesh, Y. Bethi, N. O. Ralph, S. E. Arja, N. Dennler, A. van Schaik, and G. Cohen, "Neuromorphic engineering needs closed-loop benchmarks," 2 2022.

[20] K. Aboumerhi, A. Güemes, H. Liu, F. Tenore, and R. Etienne-Cummings, "Neuromorphic applications in medicine," 8 2023.

[21] R. Yang, H. M. Huang, and X. Guo, "Memristive synapses and neurons for bioinspired computing," 9 2019.

[22] N. Zins, Y. Zhang, C. Yu, and H. An, *Neuromorphic Computing: A Path to Artificial Intelligence Through Emulating Human Brains*. Springer International Publishing, 1 2023, pp. 259–296.

[23] I. Corporation, "Taking neuromorphic computing with loihi 2 to the next level technology brief," 2021.

[24] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, pp. 911–934, 5 2021.

[25] P. S. Zarrin, R. Zimmer, C. Wenger, and T. Masquelier, "Epileptic seizure detection using a neuromorphic-compatible deep spiking neural network," vol. 12108 LNBI. Springer, 2020, pp. 389–394.

[26] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 5 2008.

[27] H. Li, S. Wang, X. Zhang, W. Wang, R. Yang, Z. Sun, W. Feng, P. Lin, Z. Wang, L. Sun, and Y. Yao, "Memristive crossbar arrays for storage and computing applications," *Advanced Intelligent Systems*, vol. 3, 9 2021.

[28] S. P. Mohanty, "Memristor: From basics to deployment," *IEEE Potentials*, vol. 32, pp. 34–39, 2013.

[29] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," pp. 309–323, 4 2019.

[30] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature Electronics*, vol. 1, pp. 22–29, 1 2018.

[31] D. Sterratt, *Principles of computational modelling in neuroscience*. Cambridge University Press, 2011.

[32] A. Thomas, "Memristor-based neural networks," 3 2013.

[33] T. Guo, K. Pan, B. Sun, L. Wei, Y. Yan, Y. N. Zhou, and Y. A. Wu, "Adjustable leaky-integrate-and-fire neurons based on memristor-coupled capacitors," *Materials Today Advances*, vol. 12, 12 2021.

[34] N. M. Samardzic, J. S. Bajic, D. L. Sekulic, and S. Dautovic, "Volatile memristor in leaky integrate-and-fire neurons: Circuit simulation and experimental study," *Electronics (Switzerland)*, vol. 11, 3 2022.

[35] R. Brette, "Philosophy of the spike: Rate-based vs. spike-based theories of the brain," 11 2015.

[36] F. Zeldenrust, W. J. Wadman, and B. Englitz, "Neural coding with bursts—current state and future perspectives," 7 2018.

[37] B. Gardner and A. Grüning, "Supervised learning in spiking neural networks for precise temporal encoding," *PLoS ONE*, vol. 11, 8 2016.

[38] R. V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns," *PLoS ONE*, vol. 7, 8 2012.

[39] L. Deng, Y. Wu, X. Hu, L. Liang, Y. Ding, G. Li, G. Zhao, P. Li, and Y. Xie, "Rethinking the performance comparison between snns and anns," *Neural Networks*, vol. 121, pp. 294–307, 1 2020.

[40] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," 9 2018. [Online]. Available: http://arxiv.org/abs/1810.08646

[41] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 3 2019.

[42] T. Iakymchuk, A. Rosado-Muñoz, J. F. Guerrero-Martínez, M. Bataller-Mompeán, and J. V. Francés-Víllora, "Simplified spiking neural network architecture and stdp learning algorithm applied to image classification," *Eurasip Journal on Image and Video Processing*, vol. 2015, 2015.

[43] Y. Guo, X. Huang, and Z. Ma, "Direct learning-based deep spiking neural networks: a review," 2023.

[44] J. Ding, Z. Yu, Y. Tian, and T. Huang, "Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks," 5 2021. [Online]. Available: http://arxiv.org/abs/2105.11654

[45] R. B. Uludağ, S. Çağdaş, Y. S. İşler, N. S. Şengör, and I. Akturk, "Bio-realistic neural network implementation on loihi 2 with izhikevich neurons," 7 2023. [Online]. Available: http://arxiv.org/abs/2307.11844

[46] Accenture, "Neuromorphic computing: Energy-efficient smart cars with advanced voice control | accenture," 2021. [Online]. Available: https://www.accenture.com/content/dam/accenture/final/a-com-migration/r3-3/pdf/pdf-147/Accenture-Neuromorphic-Computing-Energy-efficient-Smart-Cars-with-Advanced-Voice-Control.pdf

[47] I. Corporation, "Taking neuromorphic computing with loihi 2 to the next level technology brief." [Online]. Available: https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf

[48] D. L. Manna, A. Vicente-Sola, P. Kirkland, T. J. Bihl, and G. D. Caterina, "Frameworks for snns: a review of data science-oriented software and an expansion of spyketorch," 2 2023. [Online]. Available: http://arxiv.org/abs/2302.07624

[49] G. Livingston, J. Huntley, A. Sommerlad, D. Ames, C. Ballard, S. Banerjee, C. Brayne, A. Burns, J. Cohen-Mansfield, C. Cooper, S. G. Costafreda, A. Dias, N. Fox, L. N. Gitlin, R. Howard, H. C. Kales, M. Kivimäki, E. B. Larson, A. Ogunniyi, V. Orgeta, K. Ritchie, K. Rockwood, E. L. Sampson, Q. Samus, L. S. Schneider, G. Selbæk, L. Teri, and N. Mukadam, "Dementia prevention, intervention, and care: 2020 report of the lancet commission," pp. 413–446, 8 2020.

[50] A. Chudzik, A. Śledzianowski, and A. W. Przybyszewski, "Machine learning and digital biomarkers can detect early stages of neurodegenerative diseases," 3 2024.

[51] E. Passeri, K. Elkhoury, M. Morsink, K. Broersen, M. Linder, A. Tamayol, C. Malaplate, F. T. Yen, and E. Arab-Tehrany, "Alzheimer's disease: Treatment strategies and their limitations," 11 2022.

[52] H. Choi, K. Choi, D. H. Kim, B. K. Oh, H. Yim, S. Jo, and C. Choi, "Strategies for targeted delivery of exosomes to the brain: Advantages and challenges," 3 2022.

[53] A. M. Cardoso, J. R. Guedes, A. L. Cardoso, C. Morais, P. Cunha, A. T. Viegas, R. Costa, A. Jurado, and M. C. P. de Lima, "Recent trends in nanotechnology toward cns diseases: Lipid-based nanoparticles and exosomes for targeted therapeutic delivery," *International Review of Neurobiology*, vol. 130, pp. 1–40, 1 2016.

[54] S. Talegaonkar and P. R. Mishra, "Intranasal delivery: An approach to bypass the blood brain barrier," 2004. [Online]. Available: http://journals.lww.com/iphr

[55] L. R. Hanson and W. H. Frey, "Intranasal delivery bypasses the blood-brain barrier to target therapeutic agents to the central nervous system and treat neurodegenerative disease," *BMC Neuroscience*, vol. 9, 12 2008.

[56] J. Cummings, A. Ritter, and K. Zhong, "Clinical trials for disease-modifying therapies in alzheimer's disease: A primer, lessons learned, and a blueprint for the future," pp. S3–S22, 2018.

[57] A. Majdi, Z. Deng, S. Sadigh-Eteghad, P. D. Vloo, B. Nuttin, and M. M. Laughlin, "Deep brain stimulation for the treatment of alzheimer's disease: A systematic review and meta-analysis," 2023.

[58] D. C. Mcintyre and C. K. Leech, "A permanent change in brain function resulting from daily electrical stimulation," pp. 295–330, 1969.

[59] J. Olds and P. Milnkr, "Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain'," 1954.

[60] T. F. Yuan, W. G. Li, C. Zhang, H. Wei, S. Sun, N. J. Xu, J. Liu, and T. L. Xu, "Targeting neuroplasticity in patients with neurodegenerative diseases using brain stimulation techniques," 12 2020.

[61] J. P. Lefaucheur, "Transcranial magnetic stimulation," *Handbook of Clinical Neurology*, vol. 160, pp. 559–580, 1 2019.

[62] S. M. Won, E. Song, J. Zhao, J. Li, J. Rivnay, and J. A. Rogers, "Recent advances in materials, devices, and systems for neural interfaces," 7 2018.

[63] A. M. Lozano, N. Lipsman, H. Bergman, P. Brown, S. Chabardes, J. W. Chang, K. Matthews, C. C. McIntyre, T. E. Schlaepfer, M. Schulder, Y. Temel, J. Volkmann, and J. K. Krauss, "Deep brain stimulation: current challenges and future directions," pp. 148–160, 3 2019.

[64] R. S. Fisher and A. L. Velasco, "Electrical brain stimulation for epilepsy," pp. 261–270, 2014.

[65] C. Dias, D. Castro, M. Aroso, J. Ventura, and P. Aguiar, "Memristor-based neuromodulation device for real-time monitoring and adaptive control of neuronal populations," *ACS Applied Electronic Materials*, vol. 4, pp. 2380–2387, 5 2022.

[66] J. Kricheldorff, K. Göke, M. Kiebs, F. H. Kasten, C. S. Herrmann, K. Witt, and R. Hurlemann, "Evidence of neuroplastic changes after transcranial magnetic, electric, and deep brain stimulation," 7 2022.

[67] G. Hong and C. M. Lieber, "Novel electrode technologies for neural recordings," pp. 330–345, 6 2019.

[68] A. Ecker, B. Bagi, E. Vértes, O. Steinbach-Németh, M. R. Karlócai, O. I. Papp, I. Miklós, N. Hájos, T. F. Freund, A. I. Gulyás, and S. Káli, "Hippocampal sharp wave-ripples and the associated sequence replay emerge from structured synaptic interactions in a network model of area ca3," *eLife*, vol. 11, 1 2022.

[69] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," *Nature Neuroscience*, vol. 14, pp. 139–142, 2 2011.

[70] C. H. Thompson, T. E. Riggins, P. R. Patel, C. A. Chestek, W. Li, and E. Purcell, "Toward guiding principles for the design of biologically-integrated electrodes for the central nervous system," 2020.

[71] M. Sharifshazileh, K. Burelo, J. Sarnthein, and G. Indiveri, "An electronic neuromorphic system for real-time detection of high frequency oscillations (hfo) in intracranial eeg," *Nature Communications*, vol. 12, 12 2021.

[72] M. E. J. Obien, K. Deligkaris, T. Bullmann, D. J. Bakkum, and U. Frey, "Revealing neuronal function through microelectrode array recordings," p. 423, 2015.

[73] N. Geramifard, J. Lawson, S. F. Cogan, and B. J. Black, "A novel 3d helical microelectrode array for in vitro extracellular action potential recording," *Micromachines*, vol. 13, 10 2022.

[74] Y. Dai, Y. Song, J. Xie, S. Xu, X. Li, E. He, H. Yin, and X. Cai, "In vivo microelectrode arrays for detecting multi-region epileptic activities in the hippocampus in the latent period of rat model of temporal lobe epilepsy," *Micromachines*, vol. 12, 6 2021.

[75] K. Woeppel, Q. Yang, and X. T. Cui, "Recent advances in neural electrode–tissue interfaces," *Current Opinion in Biomedical Engineering*, vol. 4, pp. 21–31, 12 2017.

[76] A. T. Connolly, R. J. Vetter, J. F. Hetke, B. A. Teplitzky, D. R. Kipke, D. S. Pellinen, D. J. Anderson, K. B. Baker, J. L. Vitek, and M. D. Johnson, "A novel lead design for modulation and sensing of deep brain structures," *IEEE Transactions on Biomedical Engineering*, vol. 63, pp. 148–157, 1 2016.

[77] C. Wang, M. A. Grohme, B. Mali, R. O. Schil, and M. Frohme, "Towards decrypting cryptobiosis - analyzing anhydrobiosis in the tardigrade milnesium tardigradum using transcriptome sequencing," *PLoS ONE*, vol. 9, 3 2014.

[78] J. W. Salatino, K. A. Ludwig, T. D. Kozai, and E. K. Purcell, "Erratum: Publisher correction: Glial responses to implanted electrodes in the brain (nature biomedical engineering (2017) 1 11 (862-877))," p. 52, 1 2018.

[79] R. Biran, D. C. Martin, and P. A. Tresco, "Neuronal cell loss accompanies the brain tissue response to chronically implanted silicon microelectrode arrays," *Experimental Neurology*, vol. 195, pp. 115–126, 9 2005.

[80] K. L. Drake, K. D. Wise, J. Farraye, D. J. Anderson, and S. L. Bement, "Performance of planar multisite microprobes in recording extracellular single-unit intracortical activity," 1988.

[81] R. J. Vetter, J. C. Williams, J. F. Hetke, E. A. Nunamaker, and D. R. Kipke, "Chronic neural recording using silicon-substrate microelectrode arrays implanted in cerebral cortex," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 896–904, 6 2004.

[82] D. Khodagholy, J. N. Gelinas, and G. Buzsáki, "Learning-enhanced coupling between ripple oscillations in association cortices and hippocampus." [Online]. Available: https://www.science.org

[83] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Çağatay Aydin, M. Barbic, T. J. Blanche, V. Bonin, J. Couto, B. Dutta, S. L. Gratiy, D. A. Gutnisky, M. Häusser, B. Karsh, P. Ledochowitsch, C. M. Lopez, C. Mitelut, S. Musa, M. Okun, M. Pachitariu, J. Putzeys, P. D. Rich, C. Rossant, W. L. Sun, K. Svoboda, M. Carandini, K. D. Harris, C. Koch, J. O'Keefe, and T. D. Harris, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, pp. 232–236, 11 2017.

[84] C. Stringer, M. Pachitariu, N. Steinmetz, C. B. Reddy, M. Carandini, and K. D. Harris, "Spontaneous behaviors drive multidimensional, brainwide activity," p. eaav7893, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aav7893

[85] L. R. Squire, "The legacy of patient h.m. for neuroscience," pp. 6–9, 1 2009.

[86] A. Oliva, A. Fernández-Ruiz, G. Buzsáki, and A. Berényi, "Spatial coding and physiological properties of hippocampal neurons in the cornu ammonis subregions," *Hippocampus*, vol. 26, pp. 1593–1607, 12 2016.

[87] A. Holtmaat and K. Svoboda, "Experience-dependent structural synaptic plasticity in the mammalian brain," pp. 647–658, 9 2009.

[88] P. Rao-Ruiz, E. Visser, M. Mitrić, A. B. Smit, and M. C. van den Oever, "A synaptic framework for the persistence of memory engrams," 3 2021.

[89] A. Stuchlik, "Dynamic learning and memory, synaptic plasticity and neurogenesis: An update," 4 2014.

[90] A. P. Vaz, S. K. Inati, N. Brunel, and K. A. Zaghloul, "Coupled ripple oscillations between the medial temporal lobe and neocortex retrieve human memory." [Online]. Available: https://www.science.org

[91] A. Navas-Olive, R. Amaducci, M.-T. Jurado-Parras, E. R. Sebastian, and L. M. de la Prida, "Deep learning-based feature extraction for prediction and interpretation of sharp-wave ripples in the rodent hippocampus," *eLife*, vol. 11, p. 77772, 2022.

[92] D. F. English, A. Peyrache, E. Stark, L. Roux, D. Vallentin, M. A. Long, and G. Buzsáki, "Excitation and inhibition compete to control spiking during hippocampal ripples: Intracellular study in behaving mice," *Journal of Neuroscience*, vol. 34, pp. 16 509–16 517, 12 2014.

[93] J. Patel, E. W. Schomburg, A. Berényi, S. Fujisawa, and G. Buzsáki, "Local generation and propagation of ripples along the septotemporal axis of the hippocampus," *Journal of Neuroscience*, vol. 33, pp. 17 029–17 041, 2013.

[94] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, vol. 9, 2015.

[95] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," 10 2018.

[96] A. Navas-Olive, A. Rubio, S. Abbaspoor, K. L. Hoffman, and L. M. de la Prida, "A machine learning toolbox for the analysis of sharp-wave ripples reveals common waveform features across species," *Communications Biology*, vol. 7, p. 211, 3 2024. [Online]. Available: https://www.nature.com/articles/s42003-024-05871-w