

# Deep Spiking Neural Network, Deep Liquid State Machine e Deep Echo State Network

André Furlan - [ensismoebius@gmail.com](mailto:ensismoebius@gmail.com)

Universidade Estadual Paulista Júlio de Mesquita Filho

2023

Rascunho

Spiking Neural Networks (SNNs) mimic how the brain works by utilizing action potentials in contrast to continuous values transmitted between neurons.

The term "Spiking" originates from the behavior of biological neurons, which sporadically emit action potentials, creating voltage spikes that are measured; these spikes represent information (KASABOV, 2019). Figure 4 illustrates these spikes.

It is crucial to emphasize that an SNN **is not** a one-to-one simulation of neurons. Instead, it approximates certain computational capabilities of specific biological properties. Some studies explore the nonlinearity of dendrites and other neuron features (JONES; KORDING, 2020) yielding remarkable results in classification of the MNIST database.

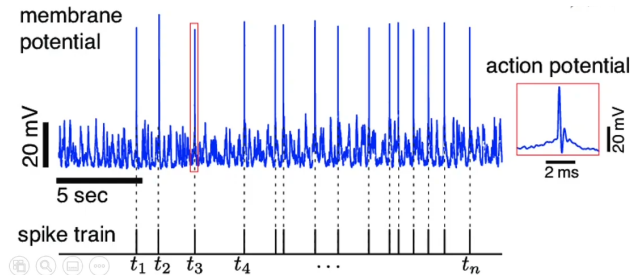


Figura: Spikes from a noisy signal. Source (GOODMAN et al., 2022)

SNNs possess several noteworthy characteristics that distinguish them from traditional machine learning techniques, including classical neural networks. These distinctions encompass (KASABOV, 2019):

- ▶ Proficiency in modeling temporal, spatial-temporal, or spectro-temporal data.
- ▶ Effectiveness in capturing processes involving various time scales.

- ▶ Seamless integration of multiple modalities, such as sound and vision, into a unified system.
- ▶ Aptitude for predictive modeling and event prediction.
- ▶ Swift and highly parallel information processing capabilities.
- ▶ Streamlined information processing.
- ▶ Scalability, accommodating structures ranging from a few tens to billions of spiking neurons.
- ▶ Minimal energy consumption when implemented on neuromorphic platforms.

In order to emulate such behavior, let's begin with a simple model: The "Leaky Integrate and Fire neuron" (LIF). The LIF model describes the evolution of membrane potential as follows.

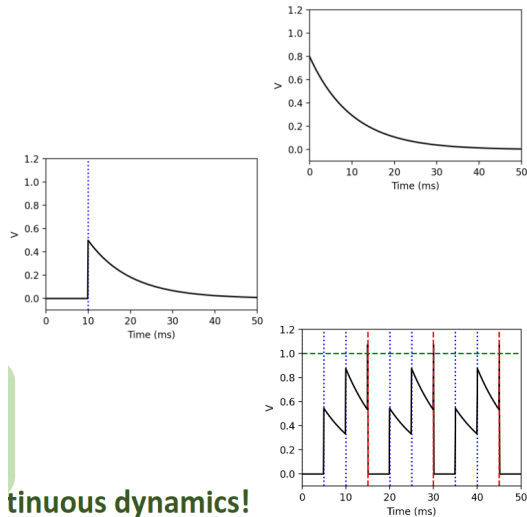
Here we have the "leaking" equation, which models the potential decay over time.

$$\tau \cdot \frac{dV}{dt} = -V \quad (1)$$

When a neuron receives a spike, the membrane potential  $V$  increases according to a synaptic weight  $w$ .

$$V = V + w \quad (2)$$

# Rascunho V



continuous dynamics!

**Figura:** Evolution of a Spike. Source (GOODMAN et al., 2022)

As shown in Figure 5, when a neuron reaches a certain threshold, it resets ( $V = 0$ ) and enters a refractory period.

Energy efficiency:

Spiking Neural Networks (SNNs) are often considered power-efficient for several reasons:

1. **Event-Driven Processing:** SNNs are inherently event-driven. Instead of constantly updating neuron activations and synapse weights as in traditional artificial neural networks (ANNs), SNNs only transmit spikes (action potentials) when a neuron's membrane potential reaches a certain threshold. This event-driven approach reduces the amount of computation required and can lead to significant energy savings.



2. Sparse Activity: SNNs tend to exhibit sparse activity, meaning that only a small percentage of neurons are active at any given time. This sparsity reduces the number of computations that need to be performed, which is especially beneficial for hardware implementations where most of the energy consumption comes from active components.
3. Low Precision: SNNs can often work with lower precision than ANNs. While ANNs typically use high-precision floating-point numbers for neuron activations and synaptic weights, SNNs can use lower precision fixed-point or binary representations. Lower precision computations require less energy to perform.
4. Neuromorphic Hardware: SNNs can be efficiently implemented on specialized neuromorphic hardware, which is designed to mimic the energy-efficient behavior of biological neural systems. These hardware platforms are optimized for the event-driven nature of SNNs, further reducing power consumption.

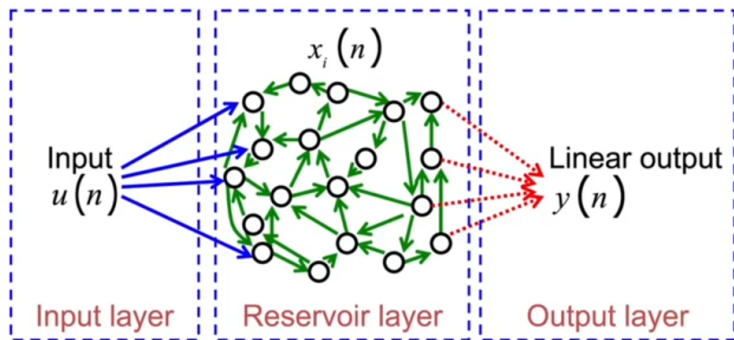
5. Energy-Aware Learning Rules: SNNs can employ learning rules that take into account energy efficiency. For example, some learning rules prioritize strengthening or weakening synapses based on their contribution to network activity, which can lead to more energy-efficient learning.
6. Spike Encoding: SNNs can encode information in the timing and frequency of spikes, which can be a highly efficient way to represent and process data, particularly for event-based sensors like vision sensors or auditory sensors.

How do SNNs get trained? Well, this is still an open question. An SNN neuron has an activation-function behavior that is more relatable to a **step-function**. Therefore, in principle, we can't use gradient descent-based solutions because this kind of function **is not** differentiable (KASABOV, 2019).

But there is some insights out there that may put some light on this subject: While some *in vivo/ in vitro* observations shows that brains in general learns by strengthen/weaken and add/remove synapses or even by creating new neurons or other

cumbersome methods like RNA packets. There are some more acceptable ones like (KASABOV, 2019):

- ▶ Spike Timing Dependent Plasticity (STDP): The idea is that if there is a pre-synaptic neuron and it fires **before** the post-synaptic one there is a strengthening in connection but, if the post-synaptic fire before then we are going to have a weakening .
- ▶ Surrogate gradient descent: The technique **approximates** the step-function by using another mathematical function, which is differentiable (like sigmoid), in order train the network. These approximations are used only **in the backwards pass** keeping the steps function in forward pass (KASABOV, 2019).
- ▶ Evolving algorithms: Uses the selection of the fittest throughout many generations of networks.
- ▶ Reservoir computing: Echo state networks and Liquid state machines. Which will be discussed further in this presentation.



**Figura:** Reservoir computing: The reservoir layer is not trained. Instead just the weights between reservoir and output layer are adjusted. Source: (KASABOV, 2019)

# Introdução

## Estrutura da apresentação

# Deep Spiking Neural Network

Spiking Neural Networks (SNNs) mimic how the brain works by utilizing action potentials in contrast to continuous values transmitted between neurons.

The term "Spiking" originates from the behavior of biological neurons, which sporadically emit action potentials, creating voltage spikes that are measured; these spikes represent information (KASABOV, 2019). Figure 4 illustrates these spikes.

It is crucial to emphasize that an SNN **is not** a one-to-one simulation of neurons. Instead, it approximates certain computational capabilities of specific biological properties. Some studies explore the nonlinearity of dendrites and other neuron features (JONES; KORDING, 2020) yielding remarkable results in classification of the MNIST database.



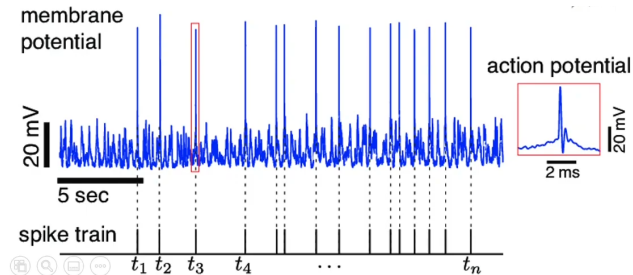


Figura: Spikes from a noisy signal. Source (GOODMAN et al., 2022)

SNNs possess several noteworthy characteristics that distinguish them from traditional machine learning techniques, including classical neural networks. These distinctions encompass (KASABOV, 2019):

- Proficiency in modeling temporal, spatial-temporal, or spectro-temporal data.
- Effectiveness in capturing processes involving various time scales.

- ▶ Seamless integration of multiple modalities, such as sound and vision, into a unified system.
- ▶ Aptitude for predictive modeling and event prediction.
- ▶ Swift and highly parallel information processing capabilities.
- ▶ Streamlined information processing.
- ▶ Scalability, accommodating structures ranging from a few tens to billions of spiking neurons.
- ▶ Minimal energy consumption when implemented on neuromorphic platforms.

In order to emulate such behavior, let's begin with a simple model: The "Leaky Integrate and Fire neuron" (LIF). The LIF model describes the evolution of membrane potential as follows.

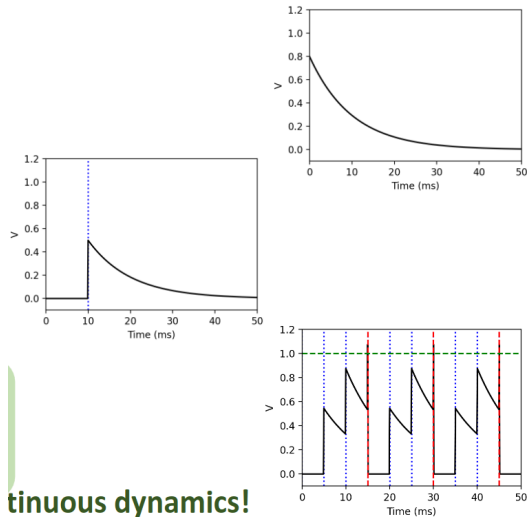
Here we have the "leaking" equation, which models the potential decay over time.

$$\tau \cdot \frac{dV}{dt} = -V \quad (3)$$

When a neuron receives a spike, the membrane potential  $V$  increases according to a synaptic weight  $w$ .

$$V = V + w \quad (4)$$

# Rascunho V



continuous dynamics!

**Figura:** Evolution of a Spike. Source (GOODMAN et al., 2022)

As shown in Figure 5, when a neuron reaches a certain threshold, it resets ( $V = 0$ ) and enters a refractory period.

Energy efficiency:

Spiking Neural Networks (SNNs) are often considered power-efficient for several reasons:

1. **Event-Driven Processing:** SNNs are inherently event-driven. Instead of constantly updating neuron activations and synapse weights as in traditional artificial neural networks (ANNs), SNNs only transmit spikes (action potentials) when a neuron's membrane potential reaches a certain threshold. This event-driven approach reduces the amount of computation required and can lead to significant energy savings.

2. Sparse Activity: SNNs tend to exhibit sparse activity, meaning that only a small percentage of neurons are active at any given time. This sparsity reduces the number of computations that need to be performed, which is especially beneficial for hardware implementations where most of the energy consumption comes from active components.
3. Low Precision: SNNs can often work with lower precision than ANNs. While ANNs typically use high-precision floating-point numbers for neuron activations and synaptic weights, SNNs can use lower precision fixed-point or binary representations. Lower precision computations require less energy to perform.
4. Neuromorphic Hardware: SNNs can be efficiently implemented on specialized neuromorphic hardware, which is designed to mimic the energy-efficient behavior of biological neural systems. These hardware platforms are optimized for the event-driven nature of SNNs, further reducing power consumption.

5. Energy-Aware Learning Rules: SNNs can employ learning rules that take into account energy efficiency. For example, some learning rules prioritize strengthening or weakening synapses based on their contribution to network activity, which can lead to more energy-efficient learning.
6. Spike Encoding: SNNs can encode information in the timing and frequency of spikes, which can be a highly efficient way to represent and process data, particularly for event-based sensors like vision sensors or auditory sensors.

How do SNNs get trained? Well, this is still an open question. An SNN neuron has an activation-function behavior that is more relatable to a **step-function**. Therefore, in principle, we can't use gradient descent-based solutions because this kind of function **is not** differentiable (KASABOV, 2019).

But there is some insights out there that may put some light on this subject: While some *in vivo*/ *in vitro* observations shows that brains in general learns by strengthen/weaken and add/remove synapses or even by creating new neurons or other

cumbersome methods like RNA packets. There are some more acceptable ones like (KASABOV, 2019):

- ▶ Spike Timing Dependent Plasticity (STDP): The idea is that if there is a pre-synaptic neuron and it fires **before** the post-synaptic one there is a strengthening in connection but, if the post-synaptic fire before then we are going to have a weakening .
- ▶ Surrogate gradient descent: The technique **approximates** the step-function by using another mathematical function, which is differentiable (like sigmoid), in order train the network. These approximations are used only **in the backwards pass** keeping the steps function in forward pass (KASABOV, 2019).
- ▶ Evolving algorithms: Uses the selection of the fittest throughout many generations of networks.
- ▶ Reservoir computing: Echo state networks and Liquid state machines. Which will be discussed further in this presentation.



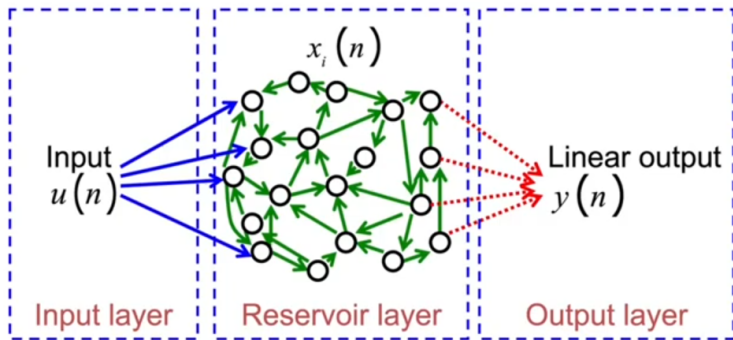


Figura: Reservoir computing: The reservoir layer is not trained. Instead just the weights between reservoir and output layer are adjusted. Source: (KASABOV, 2019)

# Deep Liquid State Machine

The liquid state machine (LSM) has emerged as a computational model that is more suitable than Turing machines for describing neural networks. It is used to process continuous streams of data, typically in the form of spike trains, and it maps streams of inputs into streams of outputs. These outputs may depend on the previous states created by the streaming data (MAASS, ).

LSM is model for adaptive computing systems. Considering that the training stage is the most expensive and sensible one, these kind of networks are trained only at the readouts. These readouts, usually, consists of only a single neuron which is called "*projection neuron*" that extracts information from a micro-circuit somewhere in the LSM. These structures can be used as inputs to another area of the neural network and can be modeled as a perceptron, linear gate, sigmoidal gate, spike neuron and others (MAASS, ). The networks structure consists of several neurons randomly and recurrently connected forming the "*liquid part*" which is interpreted by the readouts. The word "*liquid*" can sometimes be take literally like in this work (FERNANDO; SOJAKKA, 2003) where a bucket of water has been taken the role of the liquid part of the network as can be seen in Figure 7.

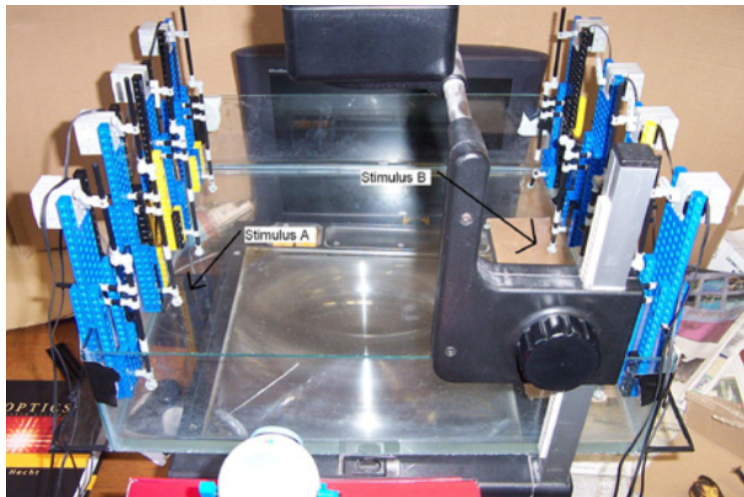
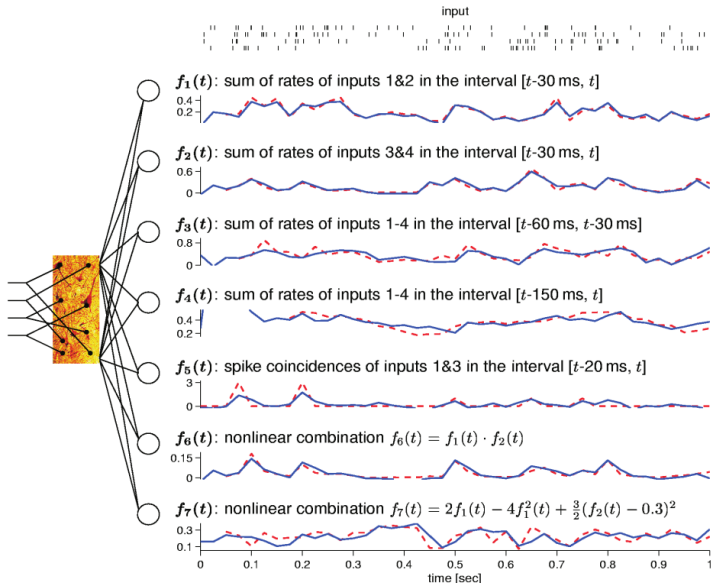


Figura: A liquid network. Source: (FERNANDO; SOJAKKA, 2003)

LSM can be adapted to multiple computation as the projection neurons can extract as many different characteristics one desires as can be seen in Figure 8.





Figura





# Deep Echo State Network




 FERNANDO, C.; SOJAKKA, S. Pattern recognition in a bucket. In: BANTHAF, W. et al. (Ed.). *Advances in Artificial Life*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 588–597. ISBN 978-3-540-39432-7.

 GOODMAN, D. et al. *Spiking Neural Network Models in Neuroscience - Cosyne Tutorial 2022*. Zenodo, 2022. Disponível em: [⟨https://doi.org/10.5281/zenodo.7044500⟩](https://doi.org/10.5281/zenodo.7044500).

 JONES, I. S.; KORDING, K. P. *Can Single Neurons Solve MNIST? The Computational Power of Biological Dendritic Trees*. 2020. Disponível em: [⟨https://github.com/ilennaj/ktree⟩](https://github.com/ilennaj/ktree).

 KASABOV, N. K. *Time-space, spiking neural networks and brain-inspired artificial intelligence*. [S.l.]: Springer, 2019.

 MAASS, W. Liquid state machines: Motivation, theory, and applications. In: \_\_\_\_\_. *Computability in Context*. [s.n.]. p. 275–296. Disponível em: [〈https://www.worldscientific.com/doi/abs/10.1142/9781848162778\\_0008〉](https://www.worldscientific.com/doi/abs/10.1142/9781848162778_0008).