Spiking Neural Networks

Zahira El mahdaouy   Bence Csabai   Lechlech Latifa   Barta Zoltán

May 10, 2022

Abstract

AI mainly tries to replicate some aspects of human brain function such as body motion or even recognizing people just by the way they walk, and human brain itself is right place to go get hints on how to do it in an efficient manner,so information processing in biology where sparse and asynchronous binary signals are communicated and processed in a massively parallel fashion inspired Spiking Neural Networks.

1 Introduction

Spiking Neural Networks has favorable properties exhibited in real neural circuits like brains, such as analog computation, low power consumption, fast inference, event-driven processing, online learning, and massive parallelism.Information transfer in these neurons mimics the information transfer in biological neurons, i.e., via the precise timing of spikes or a sequence of spikes.

2 how Spiking Neural Networks works

2.1 How biological neurons create and transmit information

There are three basic parts of a neuron: the dendrites, the cell body, and the axon. In order for neurons to communicate, they need to transmit information both within the neuron and from one neuron to the next. This process utilizes both electrical signals as well as chemical messengers.

The dendrites of neurons receive information from sensory receptors or other neurons. This information is then passed down to the cell body and on to the axon. Once the information has arrived at the axon, it travels down the length of the axon in the form of an electrical signal known as an action potential. Once an electrical impulse has reached the end of an axon, the information must be transmitted across the synaptic gap to the dendrites of the adjoining neuron. In some cases, the electrical signal can almost instantaneously bridge the gap between the neurons and continue along its path.

In other cases, neurotransmitters are needed to send the information from one neuron to the next. Neurotransmitters are chemical messengers that are released from the axon terminals to cross the synaptic gap and reach the receptor sites of other neurons. In a process known as reuptake, these neurotransmitters attach to the receptor site and are reabsorbed by the neuron to be reused.

2.2 Artificial spiking neurons

In addition to neuronal and synaptic status, SNNs incorporate time into their working model. The idea is that neurons in the SNN do not transmit information at the end of each propagation cycle (as they do in traditional multi-layer perceptron networks), but only when a membrane potential – a neuron's intrinsic quality related to its membrane electrical charge – reaches a certain value, known as the threshold. The neuron fires when the membrane potential hits the threshold, sending a signal to neighbouring neurons, which increase or decrease their potentials in response to the signal. A spiking neuron model is a neuron model that fires at the moment of threshold crossing.

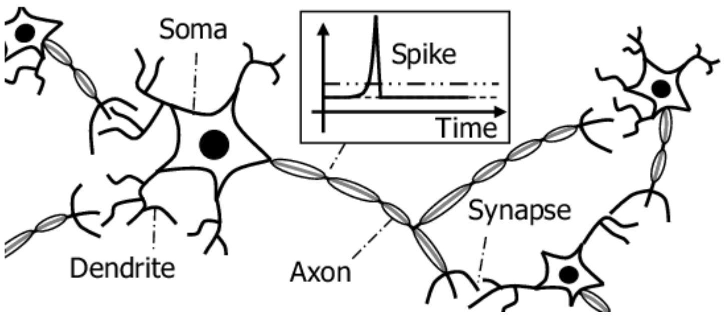

Figure 1: Biological neurons and their interactions

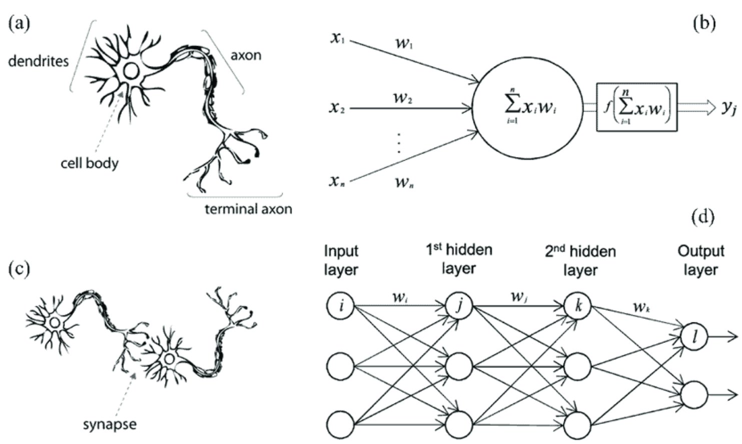

Figure 2: A biological neuron in comparison to an artificial neural network a human neuron

3 How SNN works

3.1 The key concept of SNN operation

The key difference between a traditional ANN and SNN is the information propagation approach.

SNN tries to more closely mimic a biological neural network. This is why instead of working with continuously changing in time values used in ANN, SNN operates with discrete events that occur at certain points of time. SNN receives a series of spikes as input and produces a series of spikes as the output (a series of spikes is usually referred to as spike trains).

The general idea is as follows:

At every moment of time each neuron has some value that is analogous to the electrical potential of biological neurons; The value in a neuron can change based on the mathematical model of a neuron, for example, if a neuron receives a spike from the upstream neuron, the value might increase or decrease; If the value in a neuron exceeds some threshold, the neuron will send a single impulse to each downstream neuron connected to the initial one; After this, the value of the neuron will instantly drop below its average. Thus, the neuron will experience the analog of a biological neuron's refractory period. Over time the value of the neuron will smoothly return to its average.

3.2 SNN neuron models

SNN neurons are actually built on the mathematical descriptions of biological neurons. There are two basic groups of methods used to model an SNN neuron. Conductance-based models describe how action potentials in neurons are initiated and propagated Threshold models generate an impulse at a certain threshold Although all these methods try to describe biological neurons, the devil is in detail, so SNN neurons built based on these models might slightly differ.

The most commonly used model for an SNN neuron is the Leaky Integrate-and-fire threshold model. This model suggests setting the value in the neuron to the momentary activation level modeled as a differential equation. Then the neuron receives incoming spikes that affect the value until it either vanishes or reaches a threshold. If the threshold is reached, the neuron sends impulses to the downstream neurons and the value in the neuron drops below its average.

4 Simple Neural Models

In the following paragraphs we will take a look at some basic neuron models. It should be noted that these are just the most simple examples, this list is by no means complete. Also, depending on the source, the definition of these models can slightly differ.

4.1 1D LIF Model

4.1.1 What Is It?

A 1D LIF (Leaky Integrate-and-Fire) neural model is one of the simplest ones. It resembles a biological neuron that receives spikes from other neurons. The membrane potential of our neuron increases each time it gets a spike, however it also decays over time exponentially, hence the name "Leaky". The potentials from the received spikes add up (or "Integrate"), and once a threshold parameter is reached, the neuron Fires a spike of its own. After firing, the membrane potential is reset. This is how a simple 1D LIF neuron works in a few sentences. Now let's take a look more formally!

4.1.2 How It Works?

The evolution of the membrane potential is defined by the following differential equation:

$$\tau \frac{dV}{dt} = -V$$

Where V is the membrane potential, t is the time, and τ is a constant. We can see that this will cause an exponential decay, since if we take the derivative of something, and we result with the same things times some parameter, we know that is the exponential function. An other intuitive way is to take a look at V the closer to 0 it is, the more $-V$ and the derivative will also be closer to 0, and the flatter the function will be. On the other hand if V is large, the function derivative will be a large amplitude negative number causing a steep decrease of the potential over time.

Once the neuron receives a spike the membrane potential will increase by a synaptic weight of w

$$V = V + w$$

And finally, if a threshold is reached, a spike will be emitted and V will reset to 0.

4.1.3 Conclusion

So what does this mean? Well, if a neuron receives signals in quick successions, it will activate, sending out a spike. If synaptic weights our higher, or τ is lower, less incoming signals, or ones that are further apart can activate a neuron.

4.2 2D LIF Model

4.2.1 What Is It?

A 2D LIF Model is very similar to a 1D LIF in its structure and behavior. The added "dimension" is in this case, the threshold, which increases every time the neuron's potential crosses it. Also, the threshold also has an exponential decay here with a baseline of 1.

4.2.2 How It Works?

The evolution of the membrane potential, and other key concepts are the same as in the 1D model. The threshold is defined by a differential equation similar to the one used for the potential:

$$\tau_t \frac{dV_t}{dt} = 1 - V_t$$

With V_t being the threshold potential. Every time the threshold is reached we add a δV_t to it.

$$V_t = V_t + \delta V_t$$

4.2.3 Conclusion

In this case, the later we go, the harder it is for the neuron to fire. WHY IS THIS GOOD?

4.3 Others

5 Learning models in SNN-s

To understand spiking neural networks, it is essential to understand not only the working principles of single neurons, but the behaviors of connected neurons.

5.1 STDP

Donald Hebb, a Canadian psychologist stated a famous rule in his 1949 book, *The Organization of Behavior*: "Neurons that fire together, wire together." What this means is that in case of two connected neurons if the postsynaptic cell fires shortly after the presynaptic cell, the synaptic efficacy should increase. This strengthens the connection between these two neurons.

In SNN-s, we achieve this behavior with the spike-time dependent plasticity or STDP model. With this, we update the weigths (corresponding to the synapses) between neurons based on how close they fire to each other.

We take a look at two connected neurons: one of these is the presynaptic and the other is the postsynaptic. If the postsynaptic cell fires shortly after the presynaptic, the weight of the connection between them is greatly increased. However if the post neuron fires right before the pre, this weight is greatly decreased. If the firings are far apart, the weight only changes a small amount. All in all we can say that the magnitude of the weight change is inversely proportional to the time elapsed between the applications of the two neurons.

A formula to calculate the changes to the weights can be the following:

$$\Delta w = \begin{cases} A^- e^{\frac{\Delta t}{\tau^-}}, & \text{if } \Delta t < 0 \\ A^+ e^{\frac{\Delta t}{\tau^+}}, & \text{if } \Delta t > 0 \end{cases}$$

Where A^- and A^+ are constants determining maximum excitation and τ^- and τ^+ are responsible for the steepness of the exponential curves. Also,

$$\Delta t = t_{pred} - t_{post}$$

The Δw function is shown in the following figure: The new weights of the connections can then be calculated by adding Δw to the old weigths. We can also introduce a weight change rate parameter σ to control how fast the weights will adapt.

$$w_{new} = w_{old} + \sigma \Delta w$$

The STDP is firstly an unsupervised training method, but can be modified to serve in a supervised training (e.g. rewards - R-STDP)

5.2 Surrogate Gradients

As we saw when defining the LIF neuron models, the membrane potential of the neurons follow a nonlinear function thanks to the spikes. Therefore, if we want to create a similar neural network as we usually do with ANN-s, we have to be a bit tricky.

Traditional neural networks usually have a loss function, and the gradient of this function is used to tell the model how to change each weight by backpropagating through the network. This cannot be done for the nonlinear LIF based SNN-s... Or can it?

This is where the surrogate gradient technique comes in. The aim of this is to

- Keep the unique spike property of LIF-s
- Be able to differentiate on the loss function

We can reach this by approximating the Heaviside functions in the loss function with smoother, differentiable versions. This is only in effect during the backward step however, the forward step still keeps all of its nonlinear elements.

This surrogate gradient method gives us a good starting point to be able to train supervised learning tasks with SNN-s

5.3 SNN architectures

Despite an SNN being unique at its concept, it is still a neural network, so SNN architectures can be divided into three groups:

- Feedforward Neural Network is a classical NN architecture that is widely used across all industries. In such an architecture the data is transmitted strictly in one direction – from inputs to outputs, there are no cycles, and processing can take place over many hidden layers. The majority of the modern ANN architectures are feedforward;
- Recurrent Neural Network (RNN) is a bit more advanced architecture. In RNNs connections between neurons form a directed graph along a temporal sequence. This allows the net to exhibit temporal dynamic behavior. If an SNN is Recurrent, it will be dynamical and have a high computational power;
- In Hybrid Neural Network some neurons will have feedforward connection whereas others will have recurrent connection. Moreover, connections between these groups might also be either feedforward or recurrent. There are two types of Hybrid Neural Networks that can be used as SNN architecture:

  Synfire chain is a multilayer net that allows impulse activity to propagate in the form of a synchronous wave of transmission of spike trains from one layer to the other and back;

  computing can be used to build a Reservoir SNN that will have a Recurrent Reservoir and output neurons.

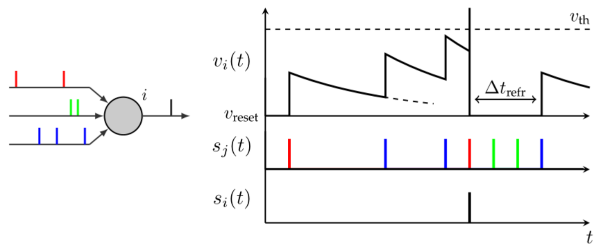

Figure 3: STDP Weight change function

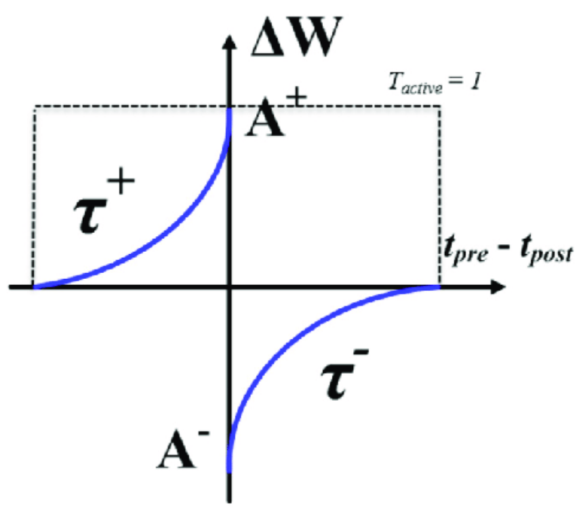

Figure 4: STDP Weight change function

6 Traditional Neural Network Vs SNN

A spiking neural network is a two-layered feed-forward network with lateral connections in the second hidden layer that is heterogeneous in nature. To transfer information, biological neurons use brief, sharp voltage increases. Action potentials, spikes, and pulses are all terms used to describe these signals. Spiking neuron networks are more potent than non-spiking counterparts because they can encode temporal information in their signals, but they also require different and biologically more realistic synaptic plasticity rules.

Spikes can't just hop from one neuron to the next. They must be handled by the neuron's most complex component: the synapse, which is made up of the axon's end, a synaptic gap, and the first piece of the dendrite. The synapse was formerly thought to merely transport a signal from the axon to the dendrite; however, it has now been discovered to be a highly complex signal pre-processor that is critical in learning and adaptation. When a spike reaches the synapse's axonal (presynaptic) side, some vesicles fuse with the cell membrane and release their neurotransmitter content into the extracellular fluid that fills the synaptic gap.

Artificial neural networks are a rather old computer science technique; the original ideas and models date back more than fifty years. McCulloch-Pitts threshold neurons were the first generation of artificial neural networks, a conceptually simple model in which a neuron sends a binary 'high' signal if the sum of its weighted incoming inputs exceeds a threshold value.

Despite the fact that these neurons can only produce digital output, they have been used in sophisticated artificial neural networks such as multi-layer perceptrons and Hopfield nets. A multilayer perceptron with a single hidden layer, for example, can compute any function with a Boolean output; these networks are known as universal for digital computations.

Second-generation neurons compute their output signals using a continuous activation function rather than a step- or threshold function, making them appropriate for analogue in- and output. The sigmoid and hyperbolic tangent are two examples of activation functions that are commonly utilized.
The illustration of forward propagation (blue arrow) and backward propagation (red arrow) of an

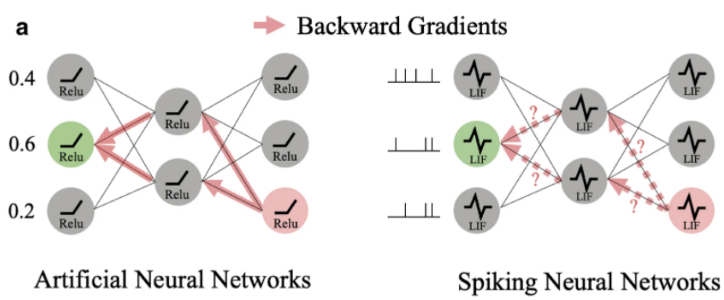

Figure 5: from ANNs, backward gradients of SNNs are difficult to be calculated.

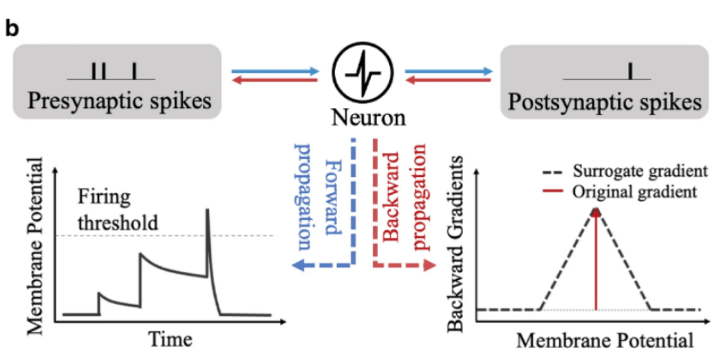

Figure 6: the membrane potential increases according to the pre-synaptic spike inputl

LIF neuron. During forward propagation,If the membrane potential exceeds the firing threshold, the

LIF neuron generates the post-synaptic spike and resets the membrane potential. This leak-integrate-and-fire behavior induces the non-differentiability of the membrane potential. Therefore, surrogate gradient functions are used to implement the backward gradient.

7 How to train a SNN?

Virtually all ANNs, spiking or non-spiking, learning is realized by adjusting scalar-valued synaptic weights. Spiking enables a type of bio-plausible learning rule that cannot be directly replicated in non-spiking networks. Neuroscientists have identified many variants of this learning rule that falls under the umbrella term spike-timing-dependent plasticity (STDP). Its key feature is that the weight (synaptic efficacy) connecting a pre- and post-synaptic neuron is adjusted according to their relative spike times within an interval of roughly tens of milliseconds in length (Caporale Dan, 2008). The information used to perform the weight adjustment is both local to the synapse and local in time. The following subsections describe common learning mechanisms in SNNs, both unsupervised and supervised.

8 Deep Learning in SNNs

As you might know, the more layers you have in your neural network, the deeper it is considered to be. So, in theory, you can stack multiple hidden layers in your SNN and consider it a deep Spiking Neural Network. However, as of today, the performance of directly trained Spiking Deep Neural Networks are not as good as traditional Deep Neural Networks represented in the literature. So, developing a Deep SNN with good performance comparable with traditional deep learning methods is a challenging task that is yet to be solved.

9 Advantages and Disadvantages of SNNs

9.1 Advantages

9.1.1 Energy consumption

One of the main advantages and prospects of spiking neural networks is, that if they run on the right hardware, their energy consumption is a fraction compared to classic ANN trainings. Multiple studies have compared the energy efficiency of SNN-s and ANN-s for the same tasks, and almost always, the SNN was victorious in this category.

9.1.2 Temporal attribute

Simply the nature of SNN-s makes them very practical to utilize on temporal data. They always wait for a new input/spike in every time step, therefore they are well suited for handling this data type. Also, a great advantage of these networks is that they learn on the fly as they receive the data.

9.1.3 Biological viability

One other advantage is that spiking neural networks are more closely modeled after biological neural networks than ANN-s. This leads to two main things:

- We can create SNN models inspired by biological systems
- By creating SNN-s, we might get insights into how some biological networks truly work

9.2 Disadvantages

9.2.1 Performance

Although SNN-s come close to their ANN counterparts, in most tasks they cannot beat them to become new state-of-the-art technologies yet. This is partly due to the fact that less research/money has been put into this field, and also, the next point, hardware.

9.2.2 Hardware

To run most efficiently, SNN-s need neuromorphic chips. There are some examples of these on the market already, like IBM's TrueNorth or Intel's Loihi, they haven't had the same amount of time and resources funneled into them as GPU-s, therefore it is understandable, that they are far from perfect and easily attainable.

10 SNN Applications

While SNN-s are not the state-of-the-art models - as mentioned earlier - and so they don't often appear in industry, there are some cases they can be used in. First of all, there are of course research projecs going on for brain-inspired audio and visual processing, simulating a simple organism's brain etc. Also, an SNN based method, NeuCube has used EEG and MRI signals to detect sleep states and help with prosthetic control.

And let's not forget that this field is still in development - most of all considering the hardware, so future applications could include even more. For example one possibility is to create edge solutions using SNN-s since in those cases it is essential to have low power consumption.