Spiking Neural Networks

The OG Neural Networks

Nature’s tried and tested

A lot of neural networks are made to teach computers tasks that humans can do. If we want to improve them, the obvious site of inspiration is right there, inside our heads: The Human Brain. They’ve been doing their job (almost) perfectly for millions of years, and have “evolved” into superior versions of themselves.

Spiking Neural Networks, or SNNs do exactly that with neural networks designed similarly to neurons in our brains….and as expected, they give fantastic results!

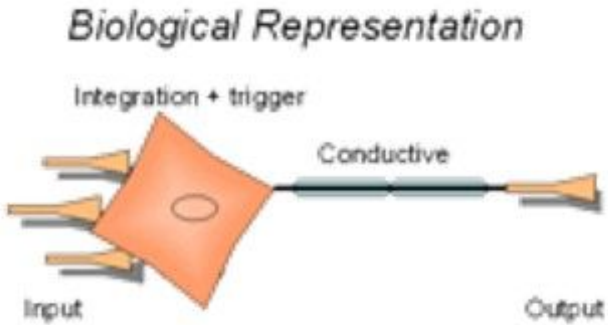
A look into our brains

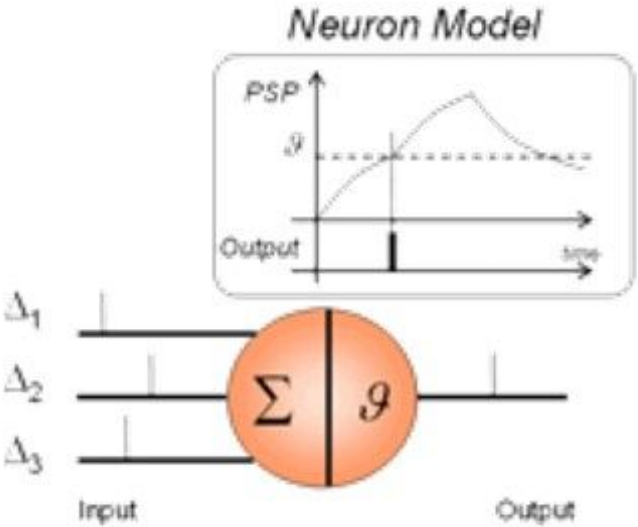
Inside our brains, information is carried by neurons.

A neuron “fires” when its potential reaches a threshold value, sending a signal to neighbouring neurons - which changes their potentials.

This is different from traditional AI NNs in which each neuron transmits information at the end of propagation cycles.

SNNs try to mimic this discrete time behaviour of natural nuerons.

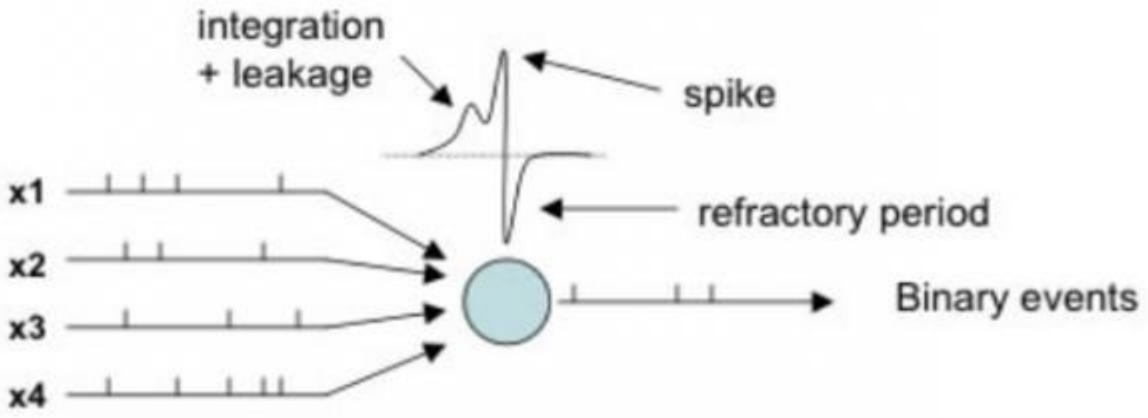

Biological Representation
Integration + trigger
Conductive
Input
Output


Neuron Model
PSP
ϑ
Output
Δ_1
Δ_2
Δ_3
Σ
ϑ
Input
Output

Spikes

In an SNN, each neuron has a value that is equivalent to the electrical potential of biological neurons at any given time. The value of a neuron can change according to its mathematical model; for example, if a neuron gets a spike from an upstream neuron, its value may rise or fall.

If a neuron's value surpasses a certain threshold, the neuron will "**Spike**" and send a single impulse to each downstream neuron connected to the first one, and the neuron's value will immediately drop below its average.

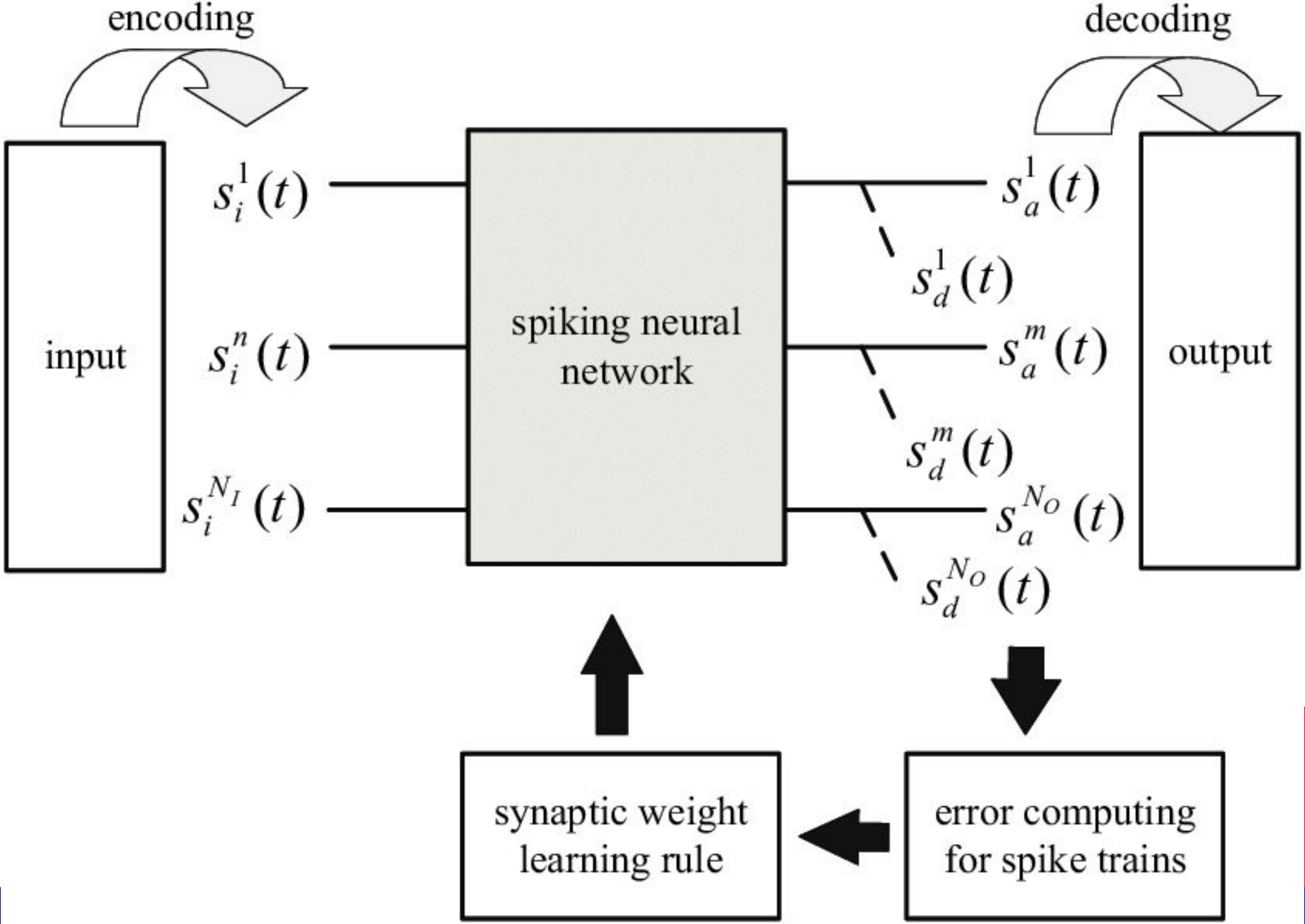

integration
+ leakage
spike
refractory period
x1
x2
x3
x4
Binary events

Spike based neural codes

In an SNN, neurons communicate through “**spikes**”. This means that the only available input and output are in the form of spikes (whether the neuron is “firing” or not).

In order to make our network learn and do useful stuff, we need it to process things like audio, images and words. For that to happen, we need to encode data points(pixels, n-grams and audio signals) into spikes.

This is done through different kinds of neural codes like binary, rate, latency and full temporal codes.


encoding
decoding
input
$s_i^1(t)$
$s_i^n(t)$
$s_i^{N_I}(t)$
spiking neural network
$s_a^1(t)$
$s_d^1(t)$
$s_a^m(t)$
$s_d^m(t)$
$s_a^{N_O}(t)$
$s_d^{N_O}(t)$
output
synaptic weight learning rule
error computing for spike trains

Binary Codes

Binary coding is an all-or-nothing encoding in which a neuron is either active or inactive within a specific time interval, firing one or more spikes throughout that time frame. The finding that physiological neurons tend to activate when they receive input (a sensory stimulus such as light or external electrical inputs) encouraged this encoding.

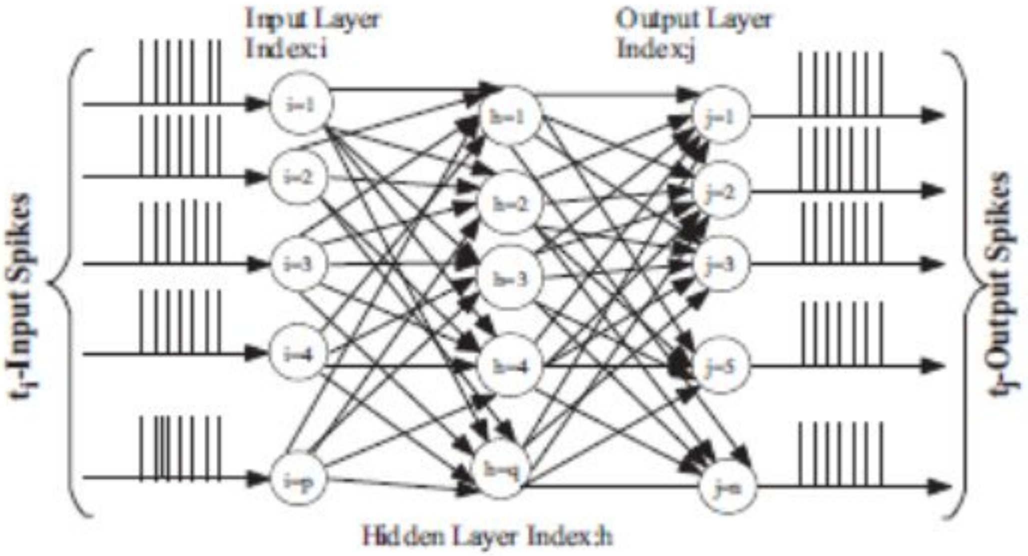
Individual neurons can benefit from this binary abstraction because they are portrayed as binary units that can only accept two on/off values.

Rate Coding

Only the rate of spikes in an interval is employed as a metric for the information communicated in rate coding, which is an abstraction from the timed nature of spikes. The fact that physiological neurons fire more frequently for stronger (sensory or artificial) stimuli motivates rate encoding.

It can be used at the single-neuron level or in the interpretation of spike trains once more. In the first scenario, neurons are directly described as rate neurons, which convert real-valued input numbers “rates” into an output “rate” at each time step.

Spiking neurons and linking synapses are described by configurable scalar weights in an SNN architecture. The analogue input data is encoded into the spike trains using either a rate-based technique, some sort of temporal coding or population coding as the initial stage in building an SNN



Architecture of a multilayer spiking neural network.

Learning in SNNs

Learning, or training happens by altering synaptic weights. Synaptic weights are basically variables that determine the connection between the two neurons being connected by the synapse. That is, they control how much the spiking of one neuron impacts the potential of the next.

Spiking allows for the replication of a form of bio-plausible learning rule that is not possible in non-spiking networks. Many variations of this learning rule have been uncovered by neuroscientists under the umbrella term spike-timing-dependent plasticity (STDP).Its main feature is that the weight (synaptic efficacy) is altered based on their relative spike times within tens of millisecond time intervals. The weight adjustment is based on information that is both local to the synapse and local in time.

Applications of SNNs

SNNs are resource efficient alternatives to artificial neural networks: able to function with fewer neurons and usually being much faster. They are also dynamic and can train while working, providing an edge in dynamic tasks like speech identification.

SNNs can be used for usual deep learning tasks like speech and text processing, image captioning and identification etc. They are also widely used in research to simulate animal brains, because of their proximity to natural neural networks.

Assignment

- We will be using brian2; a python library for simulating spiking neural networks
- The assignment will cover the basics of brian2, and how we can simulate spiking neuron groups with it.
- First you will simulate a single spiking neuron, and then you will slowly add complexity to it. By the end of the assignment, you will be able to simulate multiple spiking neurons connected by synapses.