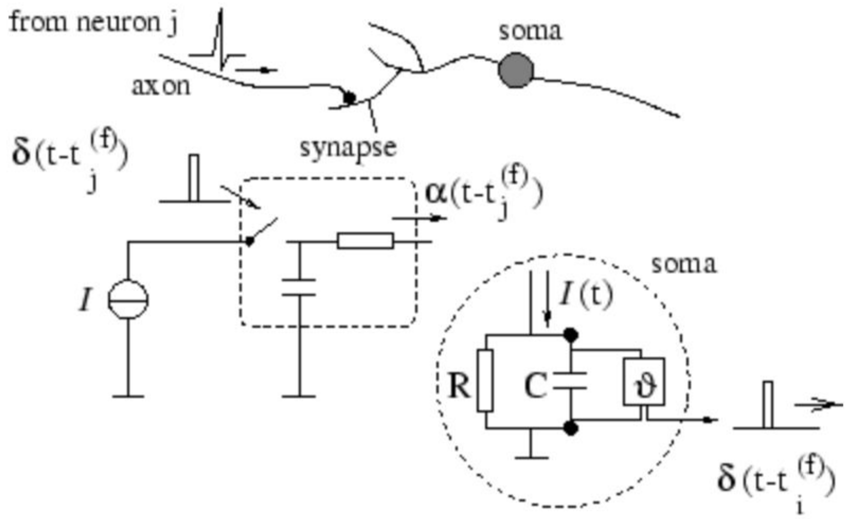
Milestone 0 Report

Neuron Models:

Leaky Integrate and Fire Neuron model:

Neuronal dynamics can be conceived as a summation process (sometimes also called ‘integration’ process) combined with a mechanism that triggers action potentials above some critical voltage.



The basic circuit of an integrate-and-fire model consists of a capacitor *C* in parallel with a resistor *R* driven by a current *I*(*t*); see Fig. 4.1. The driving current can be split into two components, $I(t) = I_R + I_C$. The first component is the resistive current I_R which passes through the linear resistor *R*. It can be calculated from Ohm's law as $I_R = u/R$ where *u* is the voltage across the resistor. The second component I_C charges the capacitor *C*. From the definition of the capacity as $C = q/u$ (where *q* is the charge and *u* the voltage), we find a capacitive current $I_C = C\, \mathrm{d}u/\mathrm{d}t$. Thus

$$I(t) = \frac{u(t)}{R} + C\,\frac{\mathrm{d}u}{\mathrm{d}t}\ . \qquad - (1)$$

We multiply the above equation by *R* and introduce the time constant $\tau_m = R\,C$ of the `leaky integrator'. This yields the standard form

$$\tau_m\ \frac{\mathrm{d}u}{\mathrm{d}t} = -\,u(t) + R\,I(t)\ . \qquad - (2)$$

We refer to *u* as the membrane potential and to τ_m as the membrane time constant of the neuron. In integrate-and-fire models the form of an action potential is not described explicitly. Spikes are formal events characterized by a `firing time' $t^{(f)}$. The firing time $t^{(f)}$ is defined by a threshold criterion

$$t^{(f)}:\quad u(t^{(f)}) = \vartheta\ . \qquad - (3)$$

Immediately after $t^{(f)}$, the potential is reset to a new value $u_r < \vartheta$,

$$\lim_{t \to t^{(f)}; t > t^{(f)}} u(t) = u_r \ . \qquad - (4)$$

For $t > t^{(f)}$ the dynamics is again given by (2) until the next threshold crossing occurs. The combination of leaky integration (2) and reset (4) defines the basic integrate-and-fire model. We note that, since the membrane potential is never above threshold, the threshold condition reduces to the criterion (3), i.e., the condition on the slope du/dt can be dropped.

In its general version, the leaky integrate-and-fire neuron may also incorporate an absolute refractory period, in which case we proceed as follows. If u reaches the threshold at time $t = t^{(f)}$, we interrupt the dynamics (2) during an absolute refractory time Δ^{abc} and restart the integration at time $t^{(f)} + \Delta^{abc}$ with the new initial condition u_r.

SRM neuron model:

Spike response model is a generalization of the leaky integrate and fire model. The main difference being we use time since the last spike in contrast to using voltage as in the lif model. Another difference between integrate-and-fire models and the SRM concerns the formulation of the equations. While integrate-and-fire models are usually defined in terms of differential equations, the SRM expresses the membrane potential at time t as an integral over the past.

$$u(t)= \eta(t-t') + \int \kappa(s) * I^{ext} \ (t-s) ds$$

The explicit dependence of the membrane potential upon the last output spike allows us to model refractoriness as a combination of three components,
(i) a reduced responsiveness after an output spike
(ii) an increase in threshold after firing
(iii) a hyperpolarizing spike after-potential.

SRM is considered to be better over the lif model because of its similarity to the real world neurons.

Coding Method

Learning Method
The chosen neuron and network models have an impact on the algorithm chosen, as certain algorithms are specific to certain network topologies, neuron models, or other network model characteristics. Also, it needs to be decided if the training should be done on chip or off-chip. One last issue is to decide between supervised and unsupervised learning.
Back-propagation:

The most commonly utilized algorithm for programming neuromorphic systems is back-propagation.

STDP:

The most popular unsupervised learning mechanism in neuromorphic systems is spike-timing dependent plasticity which is a form of Hebbian-like learning that has been observed in real biological systems. Synaptic plasticity refers to the ability of synaptic connections to change their strength, which is thought to be the basic mechanism underlying learning and memory in biological neural networks. The rule for STDP is generally that if a presynaptic neuron fires shortly before (after) the postsynaptic neuron, the synapse’s weight will be increased (decreased) and the less time between the fires, the higher the magnitude of the change.

SNN Architecture

The SNN Architecture we are planning to implement is a shallow network with two layers, the input layer with 784 neurons and the output layer with 20% more neurons than the number of classes. SRM and LIF are two commonly used neuron models. LIF model is the most used one. As of now, we are going to try modeling SRM neuron model. Our target for now is to validate the neuron model and also check if the weights are being properly trained. STDP algorithm will be used to train the network. Once we achieve good results with the shallow network, we will try implementing a deep network with hidden layers.