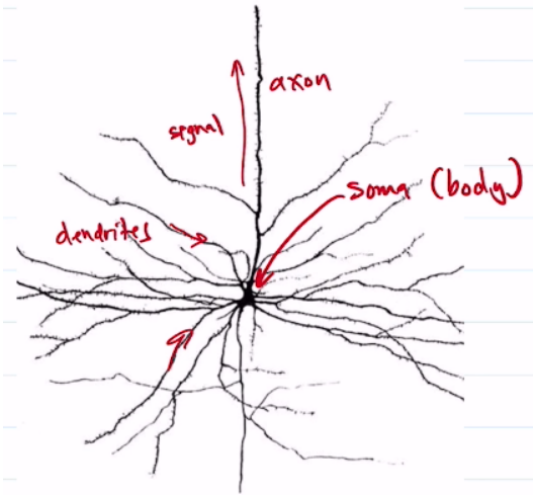
1 Topic 1.

1.1 The Hodgkin-Huxley Neuron Model.

1.1.1 Neurons

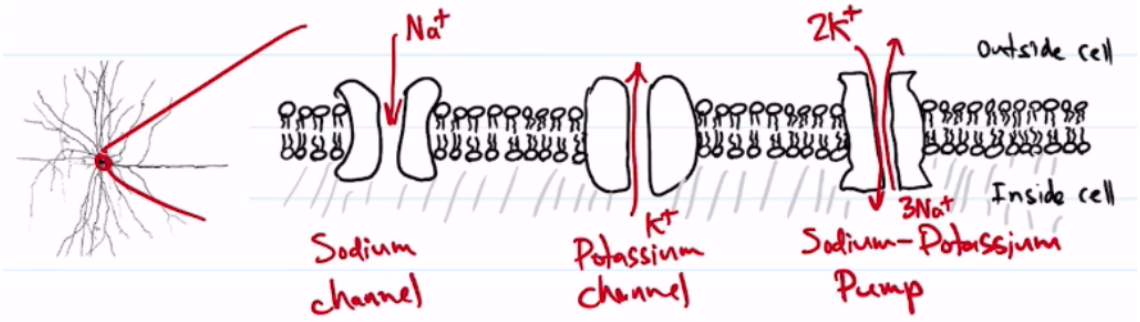
A **neuron** is a special cell that can send and receive signals from other neurons.



- **Soma**: generate electrical signals.
- **Axon**: transmit electrical signals.
- **Dendrites**: receive electrical signals.
- **Synapses**: send electrical signals.

1.1.2 Neuron Membrane Potential

Ions are molecules or atoms in which the number of electrons (-) does not match the number of protons (+), resulting in a net charge. Many ions float around your cells. The cell's **membrane**, a lipid bi-layer, stops most ions from crossing. However, ion channels embedded in the cell membrane allow ions to pass. There exist **sodium** and **potassium channels** which permits Na^+ and K^+ ions to move across the cell membrane, respectively.



The Na^+ channel moves Na^+ ions into the cell while the K^+ channel moves K^+ ions out of the cell. The **sodium-potassium pump** exchanges 3 Na^+ inside the cell for 2 K^+ ions outside the cell. This causes a higher concentration of Na^+ outside the cell and a higher concentration of K^+ inside the cell. It also creates a net positive charge outside and a net negative charge inside the cell. This difference in charge across the membrane induces a voltage difference and is called the **membrane potential**.

1.1.3 Action Potential

Neurons have a peculiar behavior: they can produce a **spike** of electrical activity called an **action potential**. This electrical burst travels along the neuron's **axon** to its **synapses**, where it passes signals to other neurons.

1.1.4 The Hodgkin-Huxley Model

The **Hodgkin-Huxley models** describes how action potentials in neurons are initiated and propagated. Their model is based on the non-linear interaction between membrane potential (aka **voltage**) and the opening/closing of Na^+ and K^+ ion channels. Both Na^+ and K^+ ion channels are voltage-dependent, so their opening and closing changes with the membrane potential.

Let V denote the membrane potential. A neuron usually keeps a membrane potential of around -70mV. We now wish to model the opening/closing of the channels.

Potassium Channels

The fraction of K^+ channels that are open is $n^4(t)$,[1] where

$$\boxed{\frac{dn}{dt} = \frac{1}{\tau_n(V)}(n_\infty(V) - n).}$$

Here n is the dynamic variable and $n_\infty(V)$ is the equilibrium solution constant. Both $\tau_n(V)$ and $n_\infty(V)$ depend on voltage. Thus, the dynamics of the K^+ channel depends on the voltage and varies over time. As a remark, the DE converges to level $n_\infty(V)$; the rate of convergence is inversely proportional to τ, i.e., it converges faster if τ is smaller.

Sodium Channels

The fraction of Na^+ ion channels open is $(m(t))^3 h(t)$, [2] where

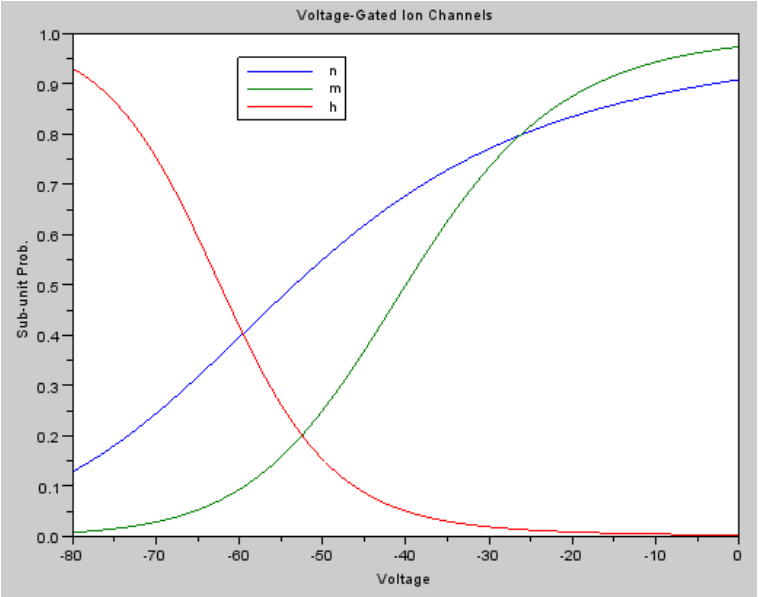
$$\boxed{\begin{aligned}\frac{dm}{dt} &= \frac{1}{\tau_m(V)}(m_\infty(v) - m)\\ \frac{dh}{dt} &= \frac{1}{\tau_h(V)}(h_\infty(v) - h)\end{aligned}}$$

Note all quantities like τ_m, τ_h, τ_n, etc., are measured empirically.

[1] The intuition is that each K^+ channel is controlled by four gates wherein the probability of one gate being open is n, hence the probability of all gates being open is n^4.

[2] Similar to above, we can interpret this as the Na^+ channel is controlled by three gates with probability m being open and one gate with probability h being open.

Below is a graph showing how $h(V), m(V), n(V)$ change as functions of voltage. As we can see, as voltage increases (move rightward) the n-gates and m-gates tend to open while the h-gate tend to close. To see how the DEs work, fix membrane potential at $V = -40$. Then we have $m(-40) \approx 0.5$ and $h(-40) \approx 0.05$. With this, you can compute the number (fraction) of sodium channels that are open as $(m(t))^3 h(t)$.



Channels and Membrane Potential

Now these two types of channels allow ions to flow into and out of the cell, inducing a current, which affects the membrane potential V. We can thus describe the membrane potential as a DE in terms of the fraction of K^+ and Na^+ channels that are open:

$$\boxed{C\frac{dV}{dt} = J_{in} - g_L(V - V_L) - g_{\text{Na}} m^3 h(V - V_{\text{Na}}) - g_{\text{K}} n^4 (V - V_{\text{K}}).}$$

- C: **capacitance**.
- $\frac{dV}{dt}$: time rate of change in voltage, or **current**.
- J_{in}: **input current**, usually from other neurons.
- $V_L, V_{\text{Na}}, V_{\text{K}}$: **zero-current potentials**.
- $g_L, g_{\text{Na}}, g_{\text{K}}$: **maximum conductance**.
- $g_L(V - V_L)$: **leak current**.
- $g_{\text{Na}} m^3 h(V - V_{\text{Na}})$: **sodium current**.
- $g_{\text{Na}} N^4 (V - V_{\text{K}})$: **potassium current**.

This system of four DEs governs the dynamics of the membrane potential.

1.1.5 Python Simulations

Remark. The following two graphs come from my friend Sibelius's notes.

The following two graphs, where x-axis is time, can hopefully give you more intuition.

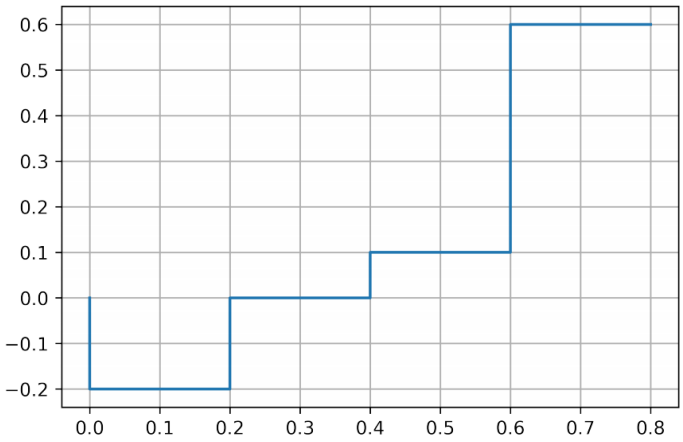


Figure 1.1: Amount of current.

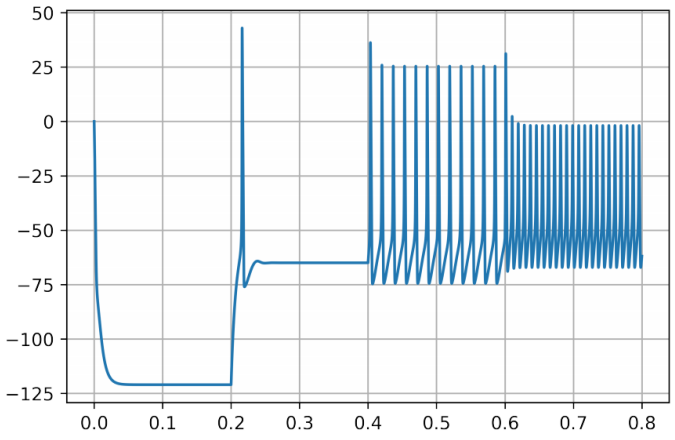


Figure 1.2: How neuron behaves.

The initial membrane potential is around -120 (right) and goes up as we increase the input current. At $J = 0.1$, it's high enough that causes regular action potentials. As we increase input current even more (to $J = 0.6$), the neuron spikes faster.

1.1.6 Summary and Motivation for LIF

The HH model is already greatly simplified:

- A neuron is treated as a point in space.
- Conductances are approximated with formulas.
- Only considers K^+, Na^+, and generic leak currents.

But to model a single action potential (spike) takes many time steps of this 4-D system. However, spikes are fairly generic, and it is thought that the *presence* of a spike is more important than its specific shape.

1.2 *Leaky Integrate-and-Fire Model.*

1.2.1 The Leaky Integrate-and-Fire Model

The **leak integrate-and-fire** (LIF) model only considers the sub-threshold membrane potential (voltage), but does NOT model the spike itself. Instead, it simply records when a spike occurs (i.e., when the voltage reached the threshold). We express it as

$$C\frac{dV}{dt} = J_{in} - g_L(V - V_L)$$

- C: capacitance.
- g_L: conductance.
- J_{in}: input current.

Note that $g_L = 1/R$ where R is the resistance. Multiply both sides by R, we get

$$RC\frac{dV}{dt} = RJ_{in} - (V - V_L)$$

Let $\tau_m := RC$ be the time constant and define $V_{in} = RJ_{in}$ (Ohm's Law). Then the voltage can be modelled as

$$\tau_m\frac{dV}{dt} = V_{in} - (V - V_L).$$

Note this model is valid only when $V < V_{th}$ (later). Change of variables:

$$v := \frac{V - V_L}{V_{th} - V_L}.$$

Then $v \to 0$ if $V_{in} = 0$ and $v = 1$ is the threshold. With this change of variable, we get

$$\boxed{\tau_m\frac{dv}{dt} = v_{in} - v.}$$

We integrate the DE for a given input current (or voltage) until v reaches the threshold of 1. Then we record a spike at that time. After it spikes, it remains dormant during its refractory period (denoted by τ_{ref}, often just a few ms). Then we start integrating again from zero (more on this below).

1.2.2 LIF Firing Rate

Suppose we hold the input v_{in} constant. We can solve the DE analytically between spikes.

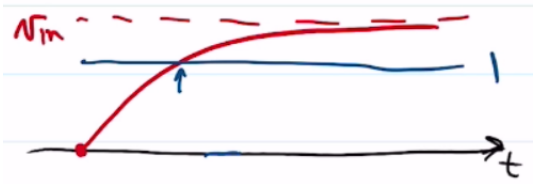
Claim. *$v(t) := v_{in}(1 - e^{-t/\tau})$ is a solution for*

$$\tau \frac{dv}{dt} = v_{in} - v$$

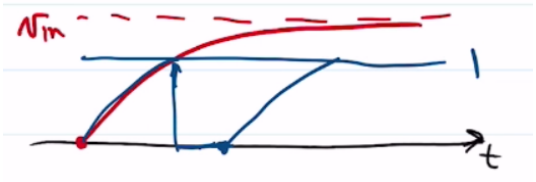
with initial value $v(0) = 0$.

In words, starting at a voltage of 0, follow the dynamics described by the DE, you get $v(t)$ as specified. To prove the claim, plug in the solution to the DE and show that LHS = RHS.

The graph of $v(t)$ looks like this:

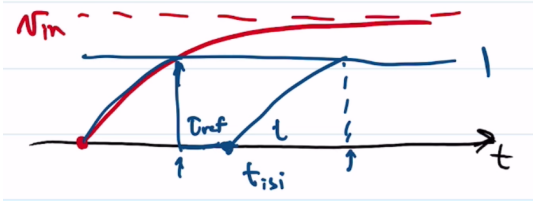


Note v_{in} must be above 1 or the neuron never spikes. The blue arrow points to the intersection of $v(t)$ and 1; at this time the neuron spikes. It then enters the refractory period and reinitiate the curve:



Solving for the Firing Rate

To solve for the firing rate, we need to solve for the time the spike occurs (as a function of v_{in}). Let t_{isi} denote the **inter-spike interval**. Note this value is the reciprocal of the fire rate. Now t_{isi} has two components, the refractory time constant τ_{ref}, plus the time it takes to go from $v = 0$ to $v = 1$, call it t^*.



We need to find t^* where $v(t^*) = 1$. From our above solution,

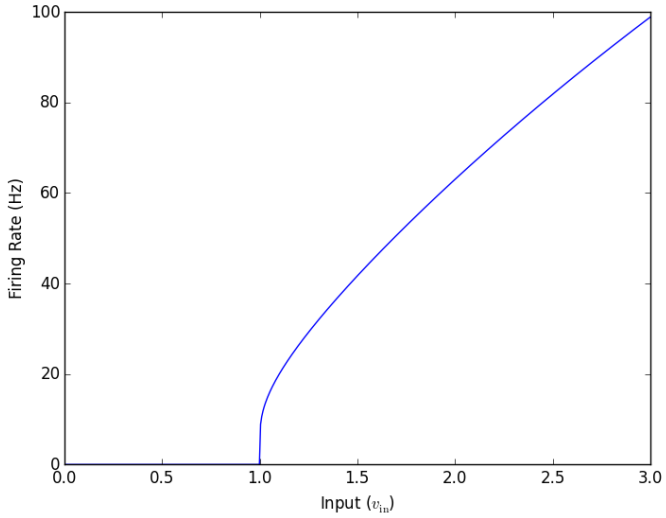
$$v(t^*) = 1 = v_{in}(1 - e^{-t/\tau}) \implies t^* = -\tau \ln\left(1 - \frac{1}{v_{in}}\right), \qquad v_{in} > 1.$$

Therefore, the firing rate is given by

$$\boxed{G(v_{in}) = \begin{cases} \dfrac{1}{\tau_{\text{ref}} - \tau_n \ln(1 - 1/v_{in})} & v_{in} > 1 \\ 0 & v_{in} \leq 1. \end{cases}}$$

Cortical Neurons

Typical values for *cortical neurons* are $\tau_{ref} = 0.002$s or 2ms and $\tau_m = 0.02$s or 20ms. Below is the graph of the firing rate as a function of v_{in}.



Let us now look at even simpler neurons.

1.2.3 Activation Functions

As we've seen, the activity of a neuron is very low, or zero, when the input is low, and the activity goes up and approaches some maximum as the input increases. This general behavior can be represented by a number of different **activation functions**.[3]

Single Neuron Activation Functions

- Logistic

$$\sigma(z) = \frac{1}{1+e^{-z}}.$$

- Arctan

$$\sigma(z) = \arctan(z).$$

- Hyperbolic tangent

$$\sigma(z) = \tanh(z).$$

- Threshold

$$\sigma(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases}$$

- Rectified Linear Unit (ReLU)

$$\text{ReLU}(z) = \max(0, z)$$

Multi-Neuron Activation Functions

Some activation functions depend on multiple neurons. Here are two examples.

SoftMax is like a probability distribution (or probability vector) as its elements add to 1. Given input $\vec{z} = (z_1, z_2, \ldots, z_N)$,

$$\text{SoftMax}(\vec{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

One-Hot is the extreme of the SoftMax, where only the largest element remains non-zero while the others are set to 0.

$$\text{One-Hot}(\vec{z})_i = \begin{cases} 1 & z_i = \max(\vec{z}) \\ 0 & \text{otherwise} \end{cases}$$

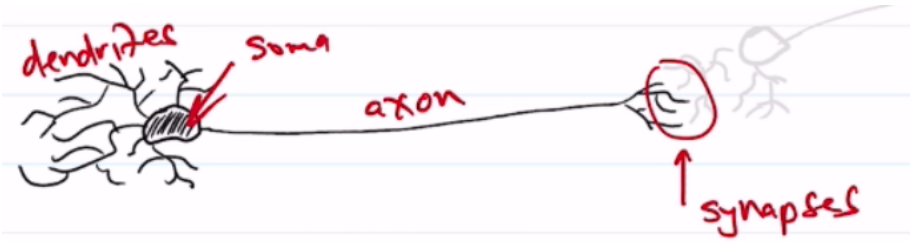
[3] Page 10 of Sibelius's notes has nice graphs for these functions.

1.3 Synapses.

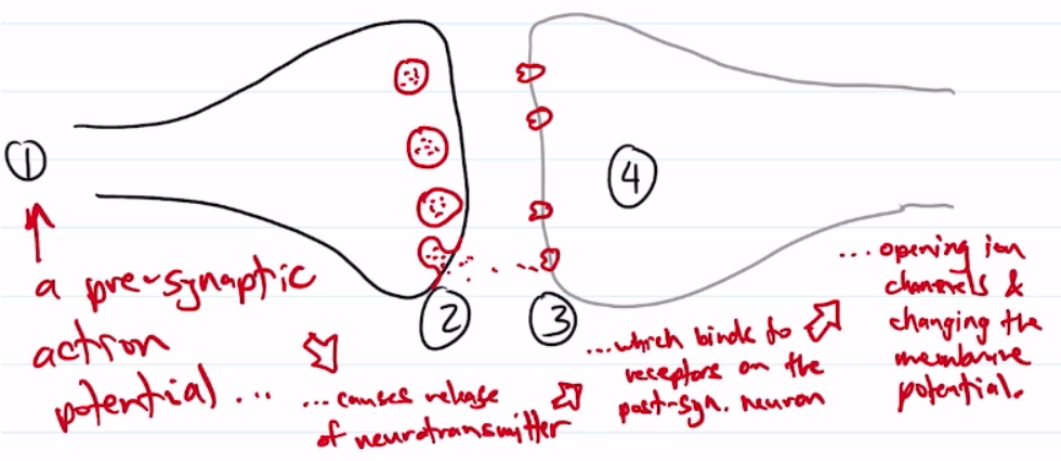
So far, we've just looked at individual neurons and how they react to their input. But that input usually comes from other neurons. In this lecture, we look at how neurons pass information between them and how we can model these communication channels.

1.3.1 Synapses

The action potential (wave of electrical activity) fired by a neuron travels along its **axon**. The junction where one neuron communicates with the next neuron is called a **synapse**.



A **pre-synaptic** action potential causes the release of **neurotransmitters** into adjacent synapses which bind to receptors on the **post-synaptic** neuron. This in turn opens or closes ion channels in the post-synaptic neuron, thereby changing membrane potential and causing the action potential to propagate.



1.3.2 Post-Synaptic Potential Filter

Even though an action potential is very fast, the synaptic processes by which it affects the next neuron takes time. Some synapses are fast (~10ms) while some are slow (~300ms). If we represent that time constant using τ_s, then the current entering the post-synaptic neuron can be written as a function of t:

$$h(t) = \begin{cases} kt^n e^{-t/\tau_s} & t \geq 0 \text{ for some } n \in \mathbb{Z}_{\geq 0} \\ 0 & \text{otherwise} \end{cases}$$

where k is chosen so that

$$\int_0^\infty h(t)\, dt = 1.$$

Solving for k, we get

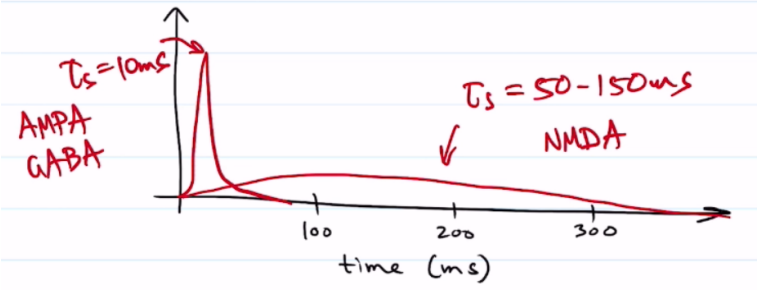
$$k = \frac{1}{n!\tau_s^{n+1}}.$$

The function $h(t)$ is called the **Post-Synaptic Current** (PSC) filter or (in keeping with the ambiguity between current and voltage) **Post-Synaptic Potential** (PSP) filter.

Note we have a split at zero because the spike arrives at the synapse at time $t = 0$ and then we are looking at what's happening after that.

Effect of τ_s

Some neurotransmitters are fast (e.g., AMPA) while others are slow (e.g., NMDA). The area under these curves are 1 (by construction).



1.3.3 Spike Train

The Dirac delta function is defined as

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & \text{otherwise} \end{cases}$$

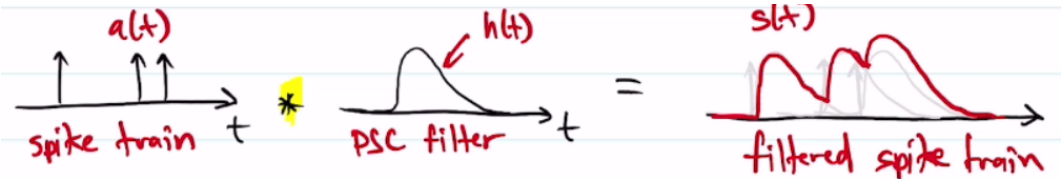
with the following two properties:

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1 \quad \text{and} \quad \int_{-\infty}^{\infty} f(t)\delta(T-t)\, dt = f(T).$$

Multiple spikes form what we call a **spike train** and can be modelled as a sum of Dirac delta functions. Suppose we have a set of spikes indexed by p occurring at time t_p, then we can describe the activity of a neuron based on its spikes:

$$a(t) = \sum_p \delta(t - t_p).$$

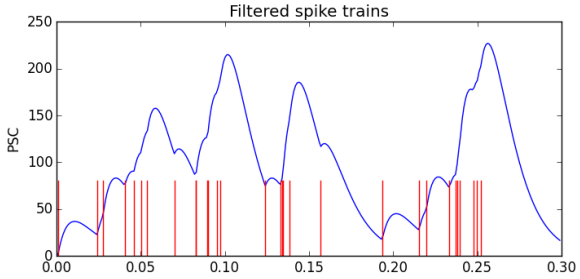
How does a spike train influence the post-synaptic neuron? You simply add together all the PSC filters, one for each spike. This is actually equivalent to **convolving** ($*$ operator) the spike train with the PSC filter.



That is,

$$s(t) = (a * h)(t) = \sum_p h(t - t_p) = \text{sum of PSC filters, one for each spike.}$$

Here's a real demo picture (with horizontal axis = time), where red lines are spikes and blue curve is the PSC induced by this spike train. Observe that (toward the right end) multiple spikes together cause a large increase in PSC.

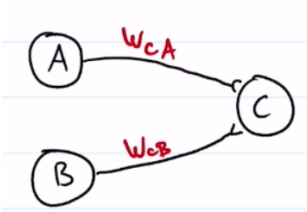


1.3.4 Connection Weight

How much does a spike at the pre-synaptic neuron influence the input current for the post-synaptic neuron? The total current induced by an action potential onto a particular post-synaptic neuron can vary widely, depending on:

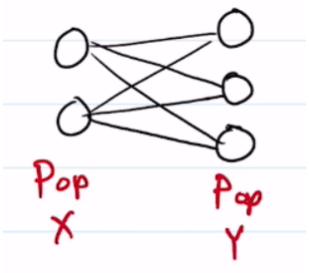
- the number and sizes of the synapses;
- the amount and type of neurotransmitter;
- the number and type of receptors;
- etc.

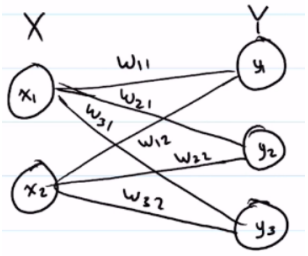
We can combine all those factors into a single number, the **connection weight**. Thus, the total input to a neuron is a *weighted sum* of filter spike trains. In the following graph with three neurons A, B, C, we use w_{CA} and w_{CB} to denote the connection weight from A to C and from B to C, respectively.



Weight Matrices

When we have many pre-synaptic neurons, it is more convenient to use *matrix-vector* notation to represent the weights and activities. Suppose we have 2 populations X and Y, each with N and M nodes. If every node in X sends its output to every node in Y, then we will have a total of NM connections, each with its own weight.





The corresponding weight matrix is given by

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \in \mathbb{R}^{M \times N}$$

Computing Input Current

Storing the neuron activities in vectors, i.e.,

$$\boxed{\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \text{and} \qquad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix},}$$

we can compute the input to the nodes in Y using

$$\boxed{\vec{z} = W\vec{x} + \vec{b}}$$

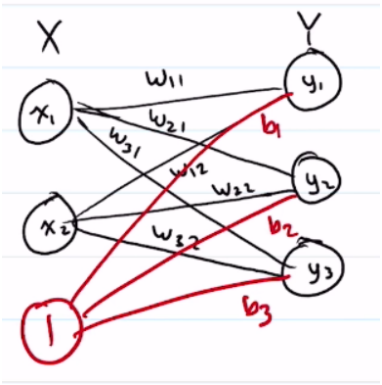
where $\vec{b}$ holds the biases for nodes (neurons) in Y. Thus,

$$\boxed{\vec{y} = \sigma(\vec{z})}$$

where $\sigma(\cdot)$ is an activation function.

Alternating Representation for Bias

Another way to represent the biases $\vec{b}$ is to add an auxiliary node in the input layer X with constant value 1.



With this modification, we can write the same thing but with an augmented matrix:

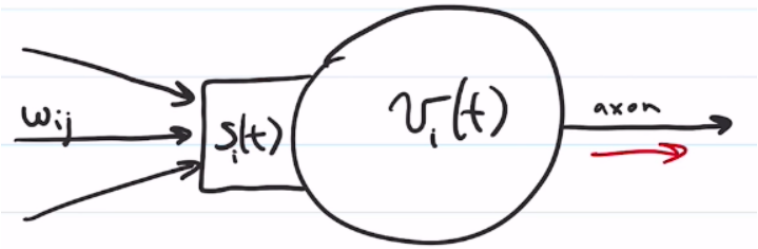
$$W\vec{x} + \vec{b} = [W \mid \vec{b}] \begin{bmatrix} \vec{x} \\ \hline 1 \end{bmatrix} =: \hat{W} \begin{bmatrix} \vec{x} \\ \hline 1 \end{bmatrix}$$

Implementing Connections Between Spiking Neurons

For simplicity, let $n = 0$ so $h(t) = \dfrac{1}{\tau_s} e^{-t/\tau_s}$. This happens to be the solution of the IVP

$$\tau_s \frac{ds}{dt} = -s, \quad s(0) = \frac{1}{\tau_s}.$$

1.3.5 Full LIF Neuron Model



Relevant differential equations:

$$\begin{cases} \tau_m \dfrac{dv_i}{dt} = s_i - v_i & \text{if not refracting, i.e., integrating input current right now} \\ \tau_s \dfrac{ds_i}{dt} = -s_i \end{cases}$$

If v_i reaches 1 (the threshold v_{th}),

1. start refractory period,
2. send spike along axon,
3. reset v to 0.

If a spike arrives from neuron j, increase s_i with

$$s_i \leftarrow s_j + \frac{W_{ij}}{\tau_s}.$$

In words, the amount of current that it injects into the post-synaptic neuron is proportional to the weight, and we divide it by τ_s, the normalizing factor. Thus, the total amount of current that eventually gets injected is its weight.