

SYDE 556/750  
Simulating Neurobiological Systems  
Lecture 2: Neurons

Andreas Stöckel

Based on lecture notes by  
Chris Eliasmith and Terrence C. Stewart

January 9, 2020



**Accompanying Readings: Chapter 4.1 and Chapter 2.1 of Neural Engineering**

# Contents

<b>1 Overview</b>	<b>1</b>
<b>2 Spiking Neurons</b>	<b>1</b>
2.1 Qualitative neural behaviour . . . . .	2
2.2 The Leaky Integrate and Fire Neuron . . . . .	3
2.3 Characterizing the firing-rate of a LIF Neuron . . . . .	4
2.4 Limitations of the LIF neuron model . . . . .	7
<b>3 Artificial “Rate” Neurons</b>	<b>8</b>

# 1 Overview



**Note:** In the last lecture, we took a deep dive into neuroscience, and we saw how “messy” and complex the nervous systems are. In this lecture we try to back up a little, moving us back to the safe haven of mathematical abstraction.

As we discussed in the last lecture, we consider neurons to be the fundamental computational unit in nervous systems. Neurons compute by receiving action potentials (spikes) from pre-synaptic neurons in their dendrites. Then, under certain conditions, they may themselves emit an action potential that is being propagated along the neuron’s axon and in turn received by post-synaptic neurons (cf. fig. 1).

From an engineering perspective, we could say that individual neurons exchange coded information. An important part of understanding nervous systems is thus to understand the “code” that is being used for neural communication.

Unfortunately, there is no scientific consensus as for what exactly this “neural code” is. Most evidence points at a combination of population coding (i.e., information is encoded in the relative activities of a group of neurons) and time coding (i.e., the timing of individual spikes matters) [1].



**Note:** There isn’t a single neural code; different coding strategies are employed in different parts of the nervous system. Codes differ significantly between the peripheral nervous system (i.e., the sensory and motor neurons distributed throughout the body; especially the latter are clearly using a rate code) and the central nervous system.

What we do have however, are detailed *models* that describe how individual neurons generate spikes, i.e., what the conditions are under which incoming spikes are translated into a corresponding output spike. Thus, we will approach the problem of trying to decipher neural codes in two stages. First, in this lecture, we will have a look at single neurons and try to get an understanding of how neurons generate action potentials. We discuss the so called “Leaky Integrate-and-Fire” neuron, and summarize its behaviour in a simple analytical expression. Second, in the next two lectures, we think about the neural code in terms of *neural representation*, which will lead us to a theory predicting what the neural code may be.

## 2 Spiking Neurons



**Note:** We’re going to have a slightly closer look at biologically detailed spiking neuron models towards the end of the class. For now, we’re skimming over the details. Have a look at [3] (particularly Chapter 7 and 8) if you want to learn more about basic neurobiology.

Neurons are cells that specialise in the integration and transmission of electrical signals. Cells in general are separated from the environment by a thick, impermeable “barrier”, the *cell membrane*, consisting of a bi-layer of lipid molecules. The cell membrane establishes an

“intracellular” space that is isolated from the “extracellular” space. Both spaces are filled with a watery liquid, called the *intracellular fluid* and *extracellular fluid*, respectively (fig. 1).

## 2.1 Qualitative neural behaviour

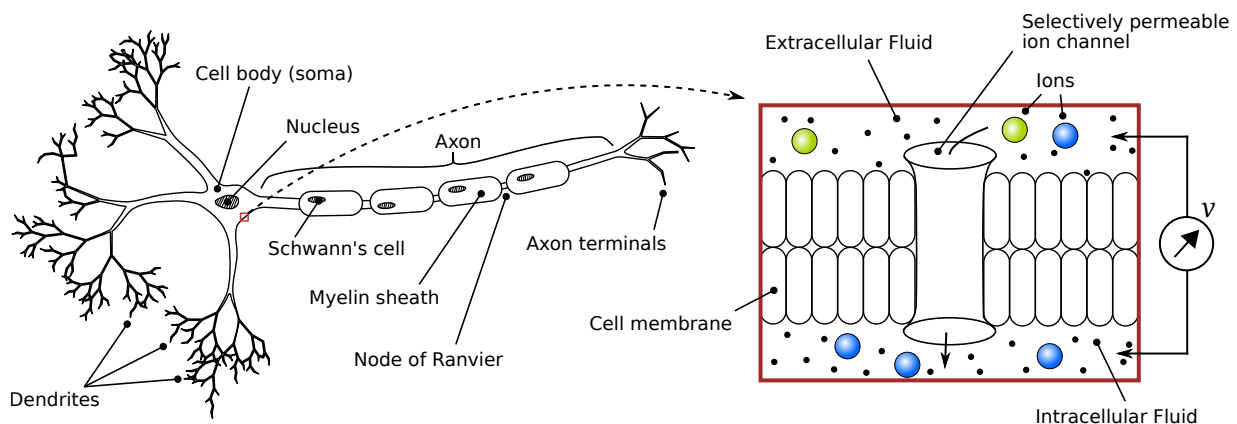
When we insert a sharp electrode into a resting neuron (akin to the “single electrode recording” we saw in the last lecture), we can measure a difference in electrical potential, i.e., a voltage  $v$ , between the intra- and extracellular space (fig. 1). We call this voltage the *resting potential*  $E_L$ .<sup>1</sup>

Instead of just measuring this potential we may also inject an external current into the neuron by hooking it up to a current source (i.e., a precision power supply that controls current instead of voltage). This is similar to what happens whenever a neuron receives a spike from a pre-synaptic neuron: the synapse induces either a positive (“excitatory synapse”), or a negative (“inhibitory synapse”) current that flows into the neuron.

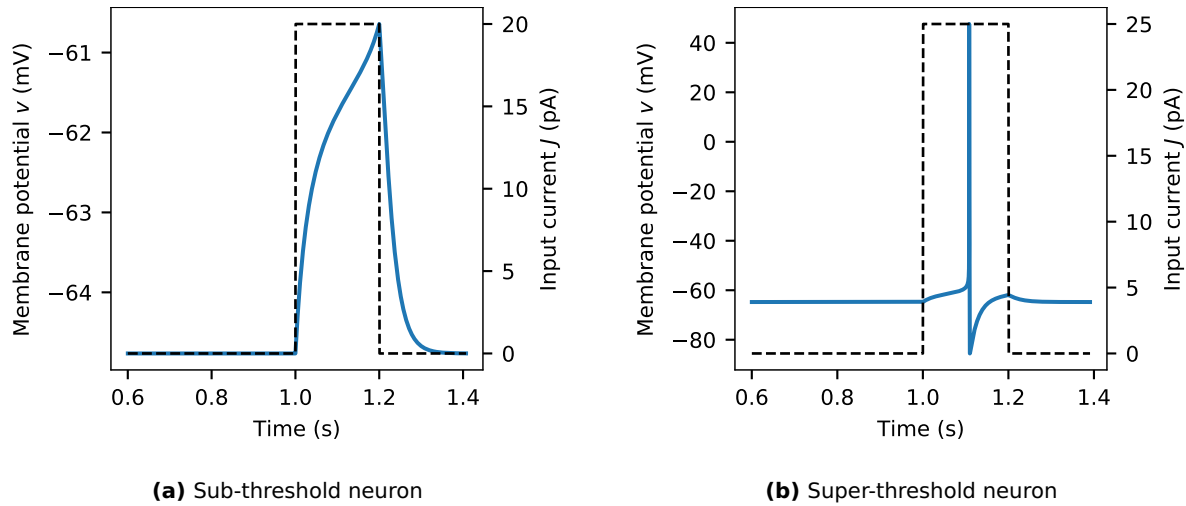
When injecting a positive current into a neuron, we find four things:

1. The cell acts like a *capacitor*, i.e., the voltage increases while we’re injecting a current (fig. 2a).
2. The capacitor is *leaky*. As soon as we stop injecting a current, the voltage collapses back to the resting potential  $E_L$  (fig. 2a).
3. As soon as the voltage surpasses a certain value, the *threshold potential*  $v_{th}$ , the cell will generate a spike (fig. 2b).
4. Shortly after the spike has been produced, the voltage drops below the resting potential. During this period, the *refractory period* of length  $\tau_{ref}$  we cannot get the neuron to spike again, even if we apply large input current  $J$  (fig. 2b).

<sup>1</sup> The weird symbol “ $E_L$ ” stems from the alternative name of this potential, the “**L**eak channel **E**quilibrium potential”.



**Figure 1:** Illustration showing a text-book neuron, as well as a schematic cross-section through the cell membrane. Left part of the illustration from [2], adapted from [3].



**Figure 2:** Computer simulation of a Hodgkin Huxley model neuron [4, 5]. The blue line corresponds to the membrane potential  $v$ , the dashed line to the current  $J$  that is being injected into the neuron. Time axis starts at 0.6 s to make sure that the neuron has settled into its resting state.



**Note:** Importantly, neurons in biology are *dynamical systems*, i.e., they possess a behaviour that evolves over time. *Artificial neurons* (see below) are time-independent. They are mathematical functions that take an input that is “immediately” being mapped onto an output.

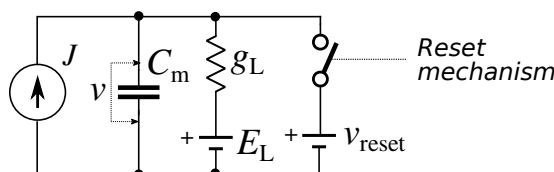
## 2.2 The Leaky Integrate and Fire Neuron

We can qualitatively summarize this behaviour in a very simple model, the so called “Leaky Integrate and Fire” neuron model. This model was first proposed by the French scientist Louis Lapicque in 1907 [6, 7].

**Sub-threshold behaviour** First, we have the *sub-threshold* behaviour, a so called leaky integrator.

$$\frac{d}{dt}v(t) = \frac{1}{C_m}(g_L(E_L - v(t)) + J), \quad \text{if } v(t) < v_{th}. \quad (1)$$

This differential equation corresponds to a capacitor with capacity  $C_m$  that is charged with a current  $J$  and that slowly discharges to a potential  $v_{reset}$  over a resistor with conductance (the inverse of the resistance)  $g_L = \frac{1}{R}$  (the *leak conductance*):



**Super-threshold behaviour** Second, we have the *super-threshold behaviour*, i.e. the spike production and refractory period. Assume  $v(t) = v_{th}$  at  $t = t_{th}$ . Then

$$\begin{aligned} v(t) &\leftarrow \delta(t - t_{th}), & \text{if } t = t_{th}, \\ v(t) &\leftarrow v_{reset}. & \text{if } t > t_{th} \text{ and } t \geq t_{th} + \tau_{ref}, \end{aligned} \quad (2)$$



**Note:**  $\delta(t)$  is the Dirac delta function, i.e., the function defined as

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0, \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(t) dt = 1.$$

**Normalized equations** For our modelling purposes, we don't really care about the exact values of  $v_{th}$ ,  $v_{reset}$  and  $E_L$ . We can just normalise these voltages, i.e., assume that  $v_{th} = 1$ , and  $v_{reset} = E_L = 0$ . We can rewrite eqs. (1) and (2) as

$$\begin{aligned} \frac{d}{dt} v(t) &= -\frac{1}{\tau_{RC}}(v(t) - Rj), & \text{if } v(t) < v_{th}. \\ v(t) &\leftarrow \delta(t - t_{th}), & \text{if } t = t_{th}, \\ v(t) &\leftarrow 0, & \text{if } t > t_{th} \text{ and } t \geq t_{th} + \tau_{ref}, \end{aligned} \quad (3)$$

where,  $\tau_{RC} = C_m R$  and  $R = \frac{1}{g_L}$ .

## 2.3 Characterizing the firing-rate of a LIF Neuron

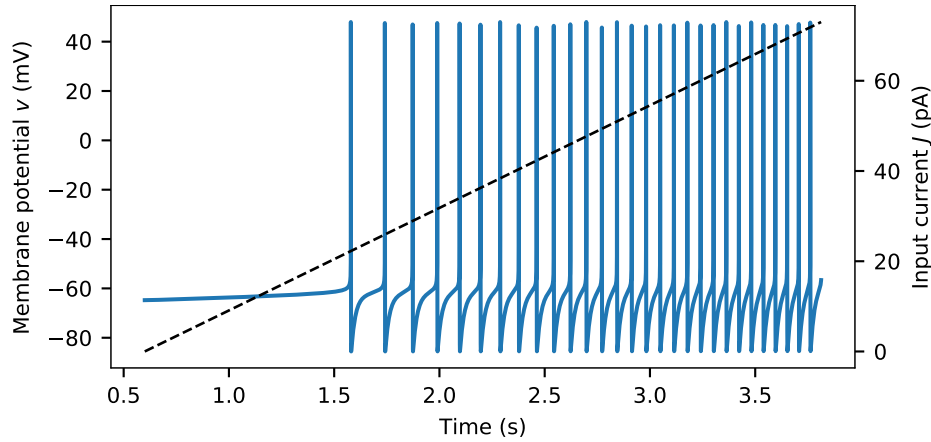
Figure 3 depicts the behaviour of the LIF neuron model eq. (3), as well as the detailed Hodgkin-Huxley type neuron model from before, as we slowly increase the input current  $j$ . Overall, we can see that the behaviour of the LIF neuron matches the behaviour of the more complex model quite well – at least qualitatively speaking.

Furthermore, we observe that the neurons fire quite regularly, yet the rate at which each neuron is emitting spikes (is “firing”) increases as we increase the input current.

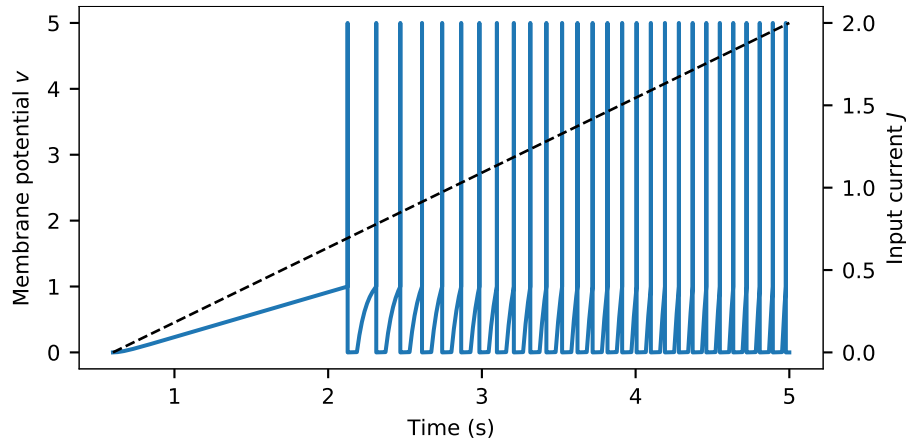
This raises the question: instead of dealing with these complicated differential equations, could we just summarize the behaviour of the neuron, i.e., compute the *firing rate*  $a$  for a given input current  $j$ ? Yes, we can! At least for the LIF neuron we can derive an analytic expression  $G[j]$ , the so called *neural response curve* that maps a current  $j$  onto the average number of spikes per second.



**Note:** Of course, with this *rate approximation*, we loose any information about spike timing. And having access to spike timing is still very useful when we want to bridge to lower levels of organization. We will deal with this in two weeks, when we talk about *temporal representation*.



(a) Hodgkin-Huxley model neuron



(b) Normalized Leaky Integrate-and-Fire neuron

**Figure 3:** Effects of a current ramp on an Hodgkin-Huxley type model neuron and a (normalized) LIF neuron. As above, the blue line is the membrane potential, the dashed line is the input current.

Since the sub-threshold portion of eq. (3) is a simple first-order linear differential equation we can actually fully characterize the behaviour of the LIF neuron under the assumption of a constant input current. Particularly, we can derive a closed form expression that describes the time  $t_{th}$  it takes for the threshold potential to be reached and a spike to be produced, assuming that the neuron is in its reset state at  $t = 0$ ,  $v(0) = 0$ . Writing down such an analytic expression is not possible most – even slightly more complex – neuron models.

**Computing the time-to-threshold  $t_{th}$**  To compute this time-to-threshold  $t_{th}$ , we first need to solve the subthreshold differential equation under the assumption that  $J$  is constant and the initial condition  $v(0) = 0$ . In general, the solution to this differential equation according to the

“variation of constants” formula is

$$v(t) = - \int_0^t \frac{1}{\tau_{RC}} (v(t') - RJ) dt' = RJ \left( 1 - e^{-\frac{t}{\tau_{RC}}} \right).$$

For the given assumptions, this equation can be used to compute the membrane potential at any time  $t$ . Of course, this does not take the super-threshold behaviour into account.

Second, we can compute the point in time  $t_{th}$  at which the membrane potential will reach  $v_{th}$ :

$$\begin{aligned} v_{th} &= RJ \left( 1 - e^{-\frac{t_{th}}{\tau_{RC}}} \right), \\ \Leftrightarrow 1 - \frac{v_{th}}{RJ} &= e^{-\frac{t_{th}}{\tau_{RC}}}, & \left| RJ \neq 0 \right. \\ \Leftrightarrow -\tau_{RC} \log \left( 1 - \frac{v_{th}}{RJ} \right) &= t_{th}. & \left| 1 - \frac{v_{th}}{RJ} > 0, RJ \neq 0 \right. \end{aligned}$$



**Note:** The terms right of the “|” specify the conditions for which equivalence holds. It is always useful to remember the conditions under which an equation remains valid.

Note that for  $1 - \frac{v_{th}}{RJ} < 0$  the input current is so small that the membrane potential will converge to an equilibrium below  $v_{th}$ , and hence  $t_{th} \rightarrow \infty$  – in other words, the neuron does not spike at all in this case.

**Computing the firing rate** Note that the LIF neuron model assumes that neurons have no “memory”. Whenever the neuron spikes, it is reset to its initial state. This means that the LIF neuron response is perfectly periodic as long as the input remains constant.



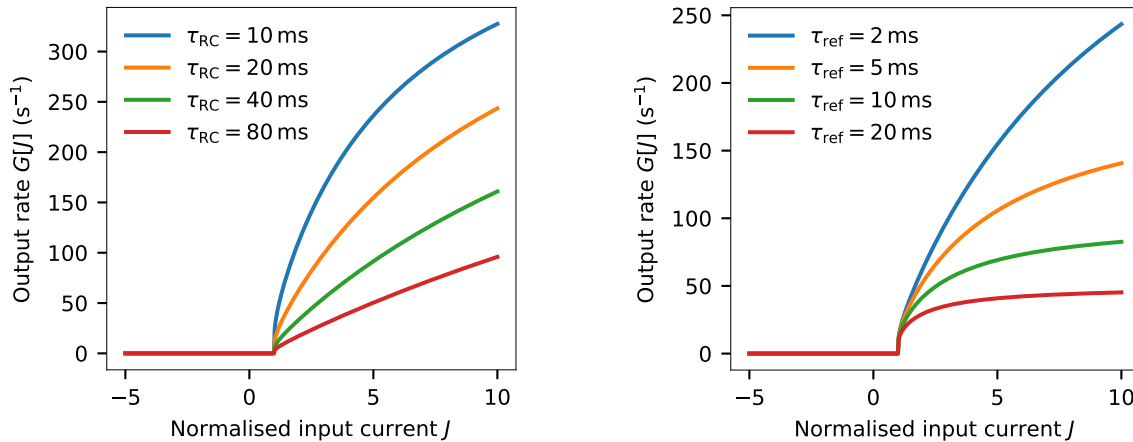
**Note:** The “no memory” assumption – and thus the perfect periodicity of the LIF neuron model – does not hold for other neuron models. However, for any neuron model, we could approximate  $G[J]$  by applying the current  $J$ , simulating the neuron over a time  $T$  and then measuring the number of spikes  $n$  over  $T$ , i.e.,  $G[J] = n(J)/T$ .

Given  $t_{th}$  we can compute the firing rate  $G[J]$  of the neuron, i.e., the number of output spikes during one second for a given input current  $J$ . The firing rate is the inverse of the inter-spike-interval (ISI), which is the sum of  $t_{th}$  (time-to-fire) and the refractory period  $\tau_{ref}$ :

$$\begin{aligned} G[J] &= \begin{cases} \frac{1}{\tau_{ref} + t_{th}(J)} & \text{if } 1 - \frac{v_{th}}{RJ} > 0, \\ 0 & \text{otherwise,} \end{cases} \\ &= \begin{cases} \frac{1}{\tau_{ref} - \tau_{RC} \log \left( 1 - \frac{v_{th}}{RJ} \right)} & \text{if } 1 - \frac{v_{th}}{RJ} > 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Remember that we defined  $v_{th} = 1$ . Furthermore, if we only want to characterize the LIF firing rate qualitatively, we can assume  $R = 1$ , since this resistance just rescales our input current (note that all variables are now unit-less). This gives us our final equation that fully summarizes the behaviour of the LIF neuron (fig. 4):





**Figure 4:** LIF rate approximation  $G[J]$  for different values of  $\tau_{RC}$  and  $\tau_{ref}$ .

(LIF neuron rate approximation)

$$G[J] = \begin{cases} \frac{1}{\tau_{ref} - \tau_{RC} \log(1 - \frac{1}{J})} & \text{if } J > 1, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

## 2.4 Limitations of the LIF neuron model

From looking at fig. 3, it may seem as if the LIF neuron model and the biophysically more plausible Hodgkin-Huxley type model exhibit the same qualitative behaviour. This is not true in general, but only for certain sets of neuron parameters (i.e., types of neurons in biology). While the LIF neuron model certainly is a good first-order approximation of neural behaviour, it does not account for some neurophysiological phenomena observed in biology (fig. 5).

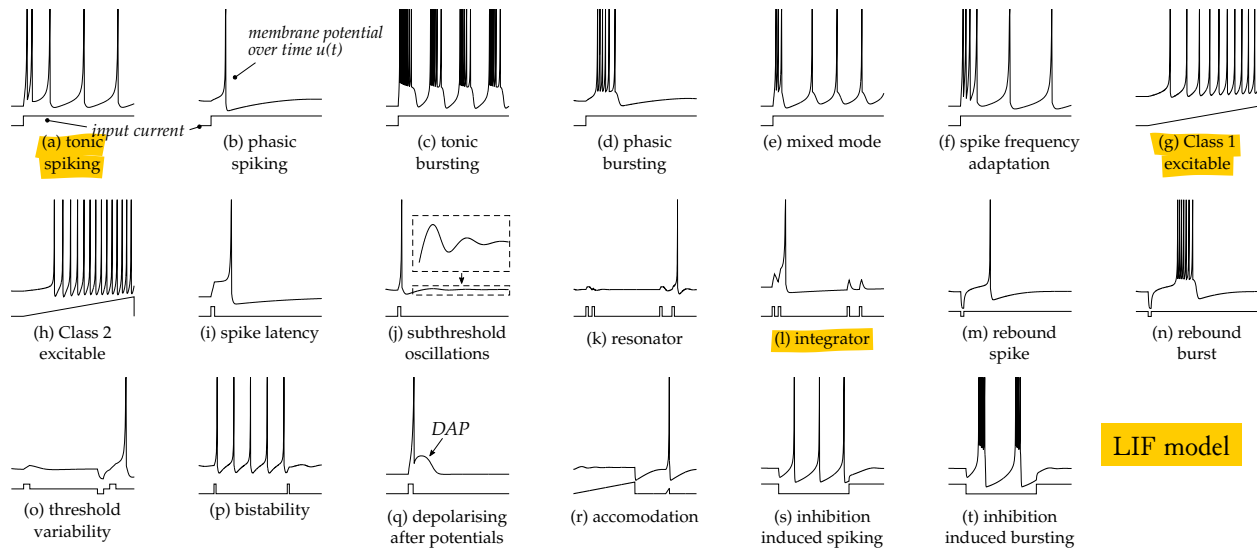
For various reasons, we are not concerned with these limitations for now. First, as we will see when we talk about temporal representation, the methods we are using are not limited to LIF neurons – as long as we have a dynamical system description of the neuron we are using, we can compute the connection weights that implement a certain behaviour.

Second, before we add more complexity to our models – such as the ability to produce the complex neural responses from fig. 5 – we should ask ourselves first what the implications for producing high-level behaviour are. In our opinion, it only makes sense to add more detail if this somehow constrains the kind of behaviour our models can produce.

Lastly, many of these phenomena can also be produced by LIF neurons when considering network effects, and not just individual neurons.



**Note:** As hinted at above, we will talk about more complex neuron models in the NEF at the end of the course, if time permits.



**Figure 5:** Single-neuron behaviours observed in nature. The highlighted behaviours correspond to those that are explained by the LIF neuron model. Upper plot of each subfigure corresponds to the membrane potential, lower plot to the input current. Figure from [2], adapted from [8]. All these behaviours can be explained with the slightly more complex (two instead of one state variables) Izhikevich neuron model [9].

### 3 Artificial “Rate” Neurons

The idea to use a rate model instead of spiking dynamical systems is exactly one idea behind the artificial neural networks used in machine learning – neurons are reduced to a function that maps an input (in biology: a current) onto a firing rate. In machine learning, this function is called the “non-linearity”. Being non-linear is an important aspect of computation in neural systems, since purely linear elements cannot perform any computation.



**Note:** As a reminder, here is the definition of what it means for a function to be *linear*.

A function  $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$  is linear *exactly if* the following equality holds

$$f(a \cdot \mathbf{x} + b \cdot \mathbf{y}) = a \cdot f(\mathbf{x}) + b \cdot f(\mathbf{y}),$$

where  $a, b \in \mathbb{R}$  are scalars and  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$  are vectors.

Another definition that makes use of matrix multiplication is: a function  $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$  is linear *exactly if* there exists a matrix  $\mathbf{F} \in \mathbb{R}^n \times \mathbb{R}^m$  such that

$$f(\mathbf{x}) = \mathbf{F}\mathbf{x}.$$

This implies – reading the definition in the backwards direction – that every matrix  $\mathbf{A}$  defines a corresponding linear function.

Next, we will take a look at some examples of neural non-linearities (or, in biological terms, “neural response functions”).

## References

- [1] Fred Rieke et al. *Spikes: Exploring the Neural Code*. Reprinted ed. July 1999. ISBN: 978-0-262-68108-7.
- [2] Andreas Stöckel. "Design Space Exploration of Associative Memories Using Spiking Neurons with Respect to Neuromorphic Hardware Implementations". MA thesis. Germany: Bielefeld University, 2015.
- [3] E. Kandel et al. *Principles of Neural Science*. 5th ed. McGraw-Hill Education, 2012.
- [4] Alan L. Hodgkin and Andrew F. Huxley. "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve". In: *The Journal of Physiology* 117.4 (1952), pp. 500–544.
- [5] Roger D. Traub and Richard Miles. *Neuronal Networks of the Hippocampus*. Vol. 777. Cambridge University Press, 1991.
- [6] Louis Lapicque. "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarization". In: *Journal de Physiologie et de Pathologie Generale* 9 (1907), pp. 620–635.
- [7] L.F. Abbott. "Lapicque's Introduction of the Integrate-and-Fire Model Neuron (1907)". In: *Brain Research Bulletin* 50.5 (1999), pp. 303–304. ISSN: 0361-9230. DOI: [https://doi.org/10.1016/S0361-9230\(99\)00161-6](https://doi.org/10.1016/S0361-9230(99)00161-6). URL: <http://www.sciencedirect.com/science/article/pii/S0361923099001616>.
- [8] Eugene M. Izhikevich. "Which Model to Use for Cortical Spiking Neurons?" In: *IEEE transactions on neural networks* 15.5 (2004), pp. 1063–1070.
- [9] Eugene M. Izhikevich. "Simple Model of Spiking Neurons". In: *IEEE Transactions on neural networks* 14.6 (2003), pp. 1569–1572.