Probabilistic spiking neuronal nets - Neuromathematics for the computer era

Antonio Galves, Eva Löcherbach, Christophe Pouzat

Probabilistic spiking neuronal nets – Neuromathematics for the computer era

Antonio Galves (galves@usp.br)
Eva Löcherbach (eva.locherbach@univ-paris1.fr)
Christophe Pouzat (christophe.pouzat@math.unistra.fr)

April 12, 2021

This document is *work in progress*! Additional chapters and sections will be added. Don't hesitate to contact us if you have comments, suggestions, find errors, etc.

Contents

List of Symbols

$\mathbb{N}$ is the set of natural numbers $\{0, 1, 2, \ldots\}$

$\mathbb{Z}$ is the set of integer numbers $\{\ldots, -n, -n+1, \ldots, 0, 1, 2, \ldots\}$

$\mathbb{R}$ is the set of real numbers

Chapter 1

Introduction

This book presents and studies a class of stochastic models for biological neural nets. A biological neural net is a system with a huge number of interacting components, the *neurons*. The activity of each neuron is represented by a point process, namely, the successive times at which the neurons emit an *action potential*, also called a *spike*. It is generally considered that the spiking activity is the way the system encodes and transmits information.

Most of our understanding of the working of the cerebral cortex originates from data that are de facto observations of point processes. Neurophysiologists analyzed these data through averaging or aggregation, building what they call the peristimulus time histogram (46). To the best of our knowledge, (50) is the first to estimate the intensity of a sequence of spike times, even if he does not make use of the mathematical framework, which, by the way, did not exist at this time. (46) clearly represent data mathematically as realization of a point process even if they do not use the terminology. Explicit use of the point process formalism will come shortly after that with the book of (25). We believe that modern statistical research of systems of spike trains starts in the 70ties with Brillinger (see for instance (11) and (13)).

Point processes are stochastic sequences of time points. There are indeed biological evidences that the spiking activity of neurons is intrinsically stochastic. Following Brillinger, in our model the spiking probability of a given neuron is a function of its membrane potential. The membrane potential can be roughly defined as the addition of the overall activity of a set of neighboring neurons called *presynaptic neurons*. When the neuron spikes, its membrane potential is reset to an equilibrium potential. Simultaneously, the set of neurons if influences undergoes a membrane potential

change. These neurons are called *postsynaptic neurons*. In general, for a given neuron, the sets of its pre- and its post-synaptic neurons are not the same. This means that the interaction graph among neurons is oriented.

The reset of the membrane potential following a spike makes the time evolution to be dependent of a variable length of the past. More precisely, it depends on the influence received from its presynaptic neurons since its last spiking time. Therefore our model is a system with a large number of interacting components each one evolving as a stochastic point process with memory of variable length. Our class of models can be considered an extension of both the interacting particle systems, which are Markov, see (90) and the stochastic chains with memory of variable length which have finite state space, see (85) or (41).

Our framework is flexible enough to deal with both discrete and continuous time settings, as well as with various kinds of synapses (chemical, electrical, plastic) and with spontaneous leakage effects. The discrete time version of this class of models was introduced in (43). The continuous time version of the model was first studied in (34).

We believe that apart from the fact that our model is interesting from a pure and an applied mathematical point of view, it is also biologically relevant as we will try to show in the next pages. We close this section with a (non-exhaustive) list of questions to be addressed.

Some questions to be addressed

- Biological data are often considered to be *stationary* in time. Most statistical tools suppose the underlying data to be stationary. Therefore it is an important question to decide whether a stationary version of the process exists, and if so, how many are they.
- Assuming that a stationary version of the process exists, how long does it take for the system to relax back to equilibrium after having been exposed to a stimulation or perturbation? Is it possible to relate this relaxation time to network parameters?
- It has been often conjectured that the brain operates in a metastable regime. Does our model exhibit such metastable properties, and if so, which?
- Is it possible to relate the functional graph, that is, the fact that the activities of different neurons are correlated, with the anatomical graph, that is, the fact that neurons are linked through actual

synapses? More specifically, suppose the system is initially exposed to one local stimulus, how many neurons will be affected by this stimulus at a given time, and is it possible to link the spread of this activity with the interaction structure?

- A central question in contemporary neuroscience is how to explain macroscopic behavior (EEG, fMRI) from a description at a microscopic level. Local mean field limits could be a way to address this question. A main difficulty here is to check if the properties of the limit system are observed experimentally.

- If we wanted to describe the behavior of a region of the cortex, using our model, we must consider that there are different populations of neurons interacting. Taking into account that we have dozens of kinds of neurons, it is easy to understand that obtaining analytic results for such models is a complicated and heavy task for mathematicians. Since an attractive feature of our model is, as will be demonstrated in this book, that we can easily and exactly simulate it on a computer, this opens the possibility of numerically studying these situations through simulations of large systems.

- At this point it is important to discuss wether these simulations are feasible in a reasonable amount of time, using a reasonable amount of memory. It is also important to compare this computational cost with the one of standard procedures in computational neuroscience which are usually based on deterministic time evolution, described by differential equations.

- It is commonly accepted that neurobiology is drowning in data but starving for theory. To face this problem it is crucial to compare mathematical theory with empirical data, by doing statistical model selection. The basic question to the class of models we are introducing is how to identify the interaction graph, especially in situations where only a tiny part of the system is observed.

All these questions are both mathematically interesting and susceptible to be compared to physiological measurements. These questions have been partially addressed in different instances of the class of models introduced in this book. The list of articles devoted to these questions is included in the Bibliography.

In order to help our readers familiar with one field (*e.g.* Neuroscience) and not with the other (*e.g.* Probability) we have tried to include links to

`Wikipedia` articles defining and discussing terms and concepts. These links appear colored in this text.

Roadmap

We have tried to write this book for several types of readers, namely mathematicians interested in stochastic models for neurobiological systems, statisticians interested in the analysis of neurobiological data and neurobiologists interested in mathematical models that could be relevant to better understand the complex phenomena exhibited by actual data and, last but not least, computational neuroscientists interested in stochastic models that run fast on their computers. This means that most chapters can be read at several levels; in particular, we tried to always summarize the biological phenomenon as well as the main mathematical ideas in the beginning of each chapter.

Chapter 2 is a detailed summary of 'basic' neurophysiological results that justifies the models and simplifications used throughout the book. Only the first section of the chapter is required to proceed in the subsequent chapters. Chapter 3 introduces the basic model as well as most of the notations that will be used throughout the book. The reading order of the subsequent chapters is mostly left to the reader's taste.

Acknowledgments

This research is part of FAPESP project Research, Innovation and Dissemination Center for Neuromathematics (Grant 2013/07699-0).

Chapter 2

A neurophysiology primer for mathematicians

2.1 A summary for the impatient

The building blocks of the models discussed in this book are the *neurons*. Neurons are "active" cells; they receive inputs via *synapses* from *presynaptic* neurons, "sum" these inputs and when the sum is large enough, an action potential or spike is generated giving rise to inputs to *postsynaptic* neurons. This action potential exhibits a fixed shape and amplitude, neurophysiologists say that it is an "all-or-none" phenomenon. The only way a variable quantity can be represented by sequences of action potentials is therefore through either their precise timing or their local time density. The distinction just made between "presynaptic" neurons (the set of neurons from which a given neuron receives inputs) and "postsynaptic" ones (the set of neurons to which a given neuron gives inputs) makes clear that synapses are *not* symmetrical. Formally neurons can be pictured as nodes/vertices and synapses as edges of a graph. Since synapses are not symmetrical, we are dealing with directed graphs. Neurons come in two main types, *excitatory* neurons: their input to their postsynaptic partners make the latter *more* likely to generate an action potential; and *inhibitory* ones: their input to their postsynaptic partners make the latter *less* likely to generate an action potential. The strength of the inputs that a given neuron gives to its postsynaptic partners is generally not uniform leading to the notion of *synaptic weight*. This synaptic weight will be positive for excitatory neurons and negative for inhibitory ones. When a neuron generates an action potential, it is "reset" and starts its input summation again "from scratch". At

this point, our description of neural networks and the units/neurons making them fits the one of (72) and the only potential source of variability is the absence/presence of a synapse between two given neurons and, if there is a synapse, the strength of the latter. But the last decades have made abundantly clear that many fluctuation sources can be identified in neurons. The input that a neuron, say j receives from its presynaptic partner i is variable from one presynaptic action potential to the next; the synaptic weight must therefore be understood as the *mean* effect of a spike in i upon j. The generation of the action potential/spike depends upon the opening of many ion channels that are membrane proteins going spontaneously back and forth between open and closed states, leading to fluctuation of the time at which the action potential is triggered for a given fixed input. Actual neurons are rather large branching cells along which structure the action potential propagates, but action potential propagation can fail haphazardly at branch points, giving yet another source of variability. It is then tempting to amend the canonical (72) neurons by lumping all the known and yet unknown sources of variability at a single locus, the one of the spike generation. Namely, we will adopt in this book a somewhat crude but handy simplification consisting in making the spike generation a probabilistic function of the summed inputs the neuron received since its last spike.

The "impatient" reader can jump at that stage to the next chapter, where the description just stated is given a proper formal expression. The remaining of this chapter exposes a justification of this description, as well as a discussion of the adopted simplifications.

2.2 Neurons

Neurons are network forming cells whose (biological) function is to receive signals from other neurons, "integrate" these signals and transmit the "integration result" to other neurons or *effector* cells like muscles (71).

2.2.1 Morphological outline

Two key features make most neurons peculiar cells, namely: they do not divide in adults; they are large—eukaryotic cells are typically 10-100 μm large, while neuronal processes can be 0.1 to more than a meter long. This second feature becomes a serious challenge as soon as *quick and reliable* signal transmission from one end of the neuron to the other is required.

Figure 2.1 shows a "typical" cerebral cortex pyramidal cell, the most numerous neuronal type in this brain region, 80% of the neurons are of this

type (see (8), for a comprehensive survey). Typical neurons—we are dealing with Biology here so for "rules" stated as a "typical" feature, there are many exceptions—have three well defined anatomical parts:

the soma or cell body where the nucleus is located, the large structure in the middle of Fig. 2.1, its diameter varies between a few and 20 μm depending on the neuron type,

the dendrites upper part of Fig. 2.1, dentrites can be a few μm to 400 μm long, they are thin, a few μm in diameter to less than $1\mu m$,

the axon the lower part of Fig. 2.1, a thin 10 to 0.1 μm in diameter (see (76)) and potentially very long, 100 to 10^6 μm —axons of giraffes and whales can be several meters long.

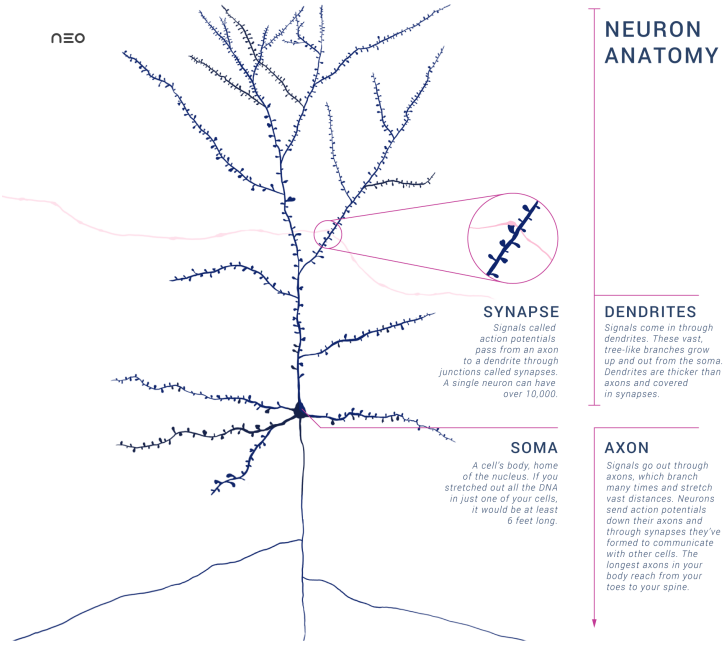


Figure 2.1: This drawing shows a single pyramidal neuron, the dominant excitatory neuron of cerebral cortex, with a synaptic connection from an incoming axon (red circles). It was created by Amy Sterling and Daniela Gamba for Neo Brain Game (license: CC BY-SA 4.0).

Pictures and drawings of neurons are very often misleading since the very long axon is usually "cut" giving the false impression of a balance between

dentritic and axonal length—a brain region where drawing are not misleading in that way is the (vertebrate) retina where neurons are "tiny", except the output one, the ganglion cell. These differences are not only anatomical but also functional in the sense of biological/physiological function, as Liqun Luo (71, pp. 12-13) puts it:

> How do information flow within individual neuron? After systematically observing many types of neurons in different parts of the nervous system, Ramóm y Cajal proposed a **theory of dynamic polarization**: the transmission of neuronal signal takes place from dendrites and cell bodies to the axon. Therefore every neuron has (1) a receptive component, the cell body and dendrites; (2) a transmission component, the axon; and (3) an effector component, the axon terminals.

This "theory of dynamic polarization" formulated by Santiago Ramón y Cajal in the 19th century must of course be nuanced in light of modern knowledge (71), but it remains a very good first approximation.

Neurons come in two categories

Neurons come in two main categories: i) *excitatory* neurons make their *postsynaptic* partners more likely to "be active"; ii) *inhibitory* neurons make their *postsynaptic* partners less likely to be active. There are many subtypes within both of these categories (8; 71), but we are most of the time interested in studying a very simplified model of the *neocortex*—the outmost and most recently evolved part of the vertebrate brain—and we are going to consider just two neuronal types, excitatory and inhibitory with the actual neocortical proportions of 80 and 20% (8).

2.2.2 Membrane potential

Like every cell, neurons are delimited by a plasma membrane—a lipid bilayer—across which the concentrations of some ions are different. The key ionic players here are sodium (Na^+, a monovalent cation), potassium (K^+) and, to a lesser degree, calcium (Ca^{2+}, a divalent cation) and chloride (Cl^-, a monovalent anion). The combination of different ionic concentrations and ion specific permeability gives rise to the membrane potential (an electrical potential difference between the inside and the outside of the neuron). The latter is usually obtained from the Nernst potential—valid when a single ion is permeable; when more than one ionic species is permeable, the Nernst potential equation takes a more general form given by the Goldman equation.

At "rest" the neuronal membrane is mainly permeable to potassium (K^+) whose typical concentrations (in mammalian neurons) are: $[K^+]_{\text{out}} = 4$ mM and $[K^+]_{\text{in}} = 120$ mM leading to a Nernst potential of $\approx$ -90 mv. This value is close to, but slightly more negative than, the measured resting potentials. *It is customary in neurophysiological modeling to give all membrane potential values with respect to the resting value and this convention will be followed in the subsequent chapters of this book.*

2.2.3 The "signal transmission problem" and the action potential

The "large" size of most neurons poses serious problems when one considers the reliable transmission of a signal from one end of a neuron to the other (17). Molecular diffusion is indeed used in bacteria whose size is of the order of 1 μm but is much too slow (the time grows with the square of the size) for larger cells like neurons. Axonal transport—energy consuming transport following elongated proteins, *microtubules*, running all along the axon—exists and is used but is also too slow (50-400 mm/day) to account for the fast reaction times typical of animals. A more effective signal propagation mechanism involves membrane potential deviations from their resting value: the *action potential* or *spike*.

The action potential

The key properties of the action potential are illustrated in Fig. 2.2 which was obtained from a numerical model that provides an excellent approximation to experimental data. This numerical model is tailored to the squid giant axon but the conclusions drawn from it have a general applicability (that's an empirical statement). The black trace on the top panel of Fig. 2.2 shows computed values of $V(x,t)$ (membrane voltage deviation from resting value) at two different locations, 1 cm apart, when a "small" stimulation (current injection) is applied. The curve with the largest amplitude shows $V(0,t)$ ($x = 0$ is the site of current injection) and the other one shows $V(1,t)$. If we increase slightly the stimulation amplitude (grey curves) the voltage deviations increase correspondingly, that is almost linearly, except for the small hump on the decaying phase of $V(0,t)$ (around 0.8 ms). If we keep increasing the stimulation amplitude, Fig. 2.2 bottom panel (notice that the range of the ordinate at the bottom is five times larger than at the top), when the current pulse ends at 0.2 ms at the stimulation location, *the initial potential decay is quickly followed by a rise* (left black curve). This

rise goes up to 90 mv before the potential starts decaying. If we now look at $V(x,t)$ for $x = 1, 2, 3, 4$ cm (successive unimodal black curves from left to right), we see a *standard potential waveform propagating at a constant velocity*. Increasing the stimulation amplitude further (set of grey curves) makes the "standard waveform" appear earlier, but does neither change its shape, nor its velocity. **This standard waveform propagating at a constant velocity is the action potential**.

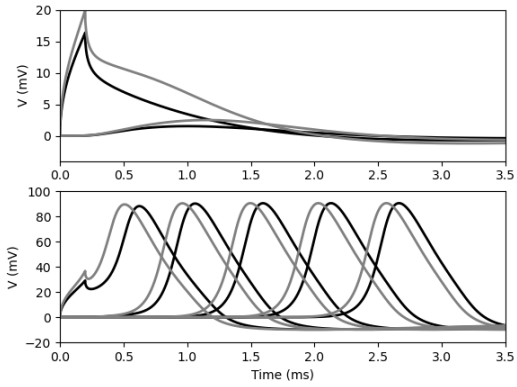


Figure 2.2: A replicate of (23) numerical study of the Hodgkin and Huxley squid giant axon model. Four simulations are illustrated with an applied current pulse lasting 0.2 ms and an increasing amplitude. Top: responses to "subthreshold" pulses (1500, black and 1800 $\mu A/cm^2$ grey) at two locations, the injection location, and 1 cm away. Bottom: responses to "suprathreshold" pulses (2500, black, and 3000 $\mu A/cm^2$, grey) at five different locations: the injection site, 1, 2, 3 and 4 cm away.

The action potential understood as a standard potential waveform propagating at constant velocity clearly provided a way of transmitting a (fast) signal over long distances. The "standard" aspect of the waveform will provide robust signal transmission, in the same way as the TTL pulses of electronic devices do: the exact amplitude (or shape) of the pulse (waveform) is not "interpreted", the pulse (waveform) is there or not. This leads directly to the all-or-none law of the response to a stimulus: there is either an action potential or nothing, the response cannot be graded. This robust feature of the standard waveform comes at a price: if stimulation of different amplitudes have to be represented, these different amplitudes have to be "coded" by different frequencies of action potential sequences or by different

latencies of the first action potential. A time varying continuous signal can clearly not be represented by sequences of action potentials from a single neuron.

We don't have the space to enter into the biophysical details of action potential generation and propagation here (55) and we will only summarize the general feature illustrated by Fig. 2.2. The membrane of (most) neurons exhibits two qualitatively different behaviors: i) when the stimulation (injected current) is small, the membrane response is essentially linear; ii) when a threshold is exceeded, a standard, self propagating potential waveform of brief duration is generated. The models considered in this book will also "forget" about the biophysics of the action potential and schematize the neuronal dynamics by postulating that the neuron spends most of its time "integrating its inputs" (the linear regime above) and, when the integrated inputs are large enough, an action potential is generated and propagated. Neither the action potential nor its propagation are going to be explicitly modeled.

2.2.4 Talking to other neurons: the synapse

A "signal" gets transmitted from one neuron to the next by the activation of a *synapse*: the action potential of the *presynaptic* neuron (in red on Fig. 2.1) reaches the presynaptic terminal, this triggers the release of packets of neurotransmitters (small molecules like glutamate for the excitatory synapses and GABA for the inhibitory ones) that diffuse in the small space between the pre- and post-synpatic neurons and bind to *receptor-channels* (or ligand-gated ion chennels) located in the membrane of the postsynaptic neuron (71)—channels are macromolecules spanning the cell membrane and forming a pore through the latter; the pore can be closed or opened—; after transmitter binding to these receptor-channels, the latter open and let specific ions flow through their pore (mainly sodium for excitatory synapses and chloride for the inhibitory ones). These ion fluxes or *currents* will induce a change of the postsynaptic neuron membrane potential. What we just described are *chemical synapses*, they are by far the most numerous in cortical regions, but we also find, between specific cell types, *electrical synapses* (71) that are likely to play an important role in synchronizing the outputs of groups of neurons.

2.2.5 The integrate and fire model

A simple quantitative neuronal model compatible with the description we just gave was introduced in 1907 by Lapicque (68). It is now known as the *Lapicque* or the *integrate and fire* model and it describes the membrane potential dynamics of 'point-like' neurons (their spatial extension is not explicitely modeled) as follows (93; 16):

$$C_m dV_i/dt = -V_i/R_m + \sum_{j\in\mathcal{S}_{i,E}} I_{j\to i}(t) + \sum_{k\in\mathcal{S}_{i,I}} I_{k\to i}(t) \quad \text{if} \quad V_i(t) < V_{thr}\,, \tag{2.2.1}$$

where i is the neuron index; C_m is the neuron capacitance; R_m is the neuron membrane resistance; $\mathcal{S}_{i,E}$, respectively $\mathcal{S}_{i,I}$, are the indices of excitatory, respectively inhibitory, neurons presynaptic to i; $I_{j\to i}(t) \geq 0$, respectively $I_{k\to i}(t) \leq 0$, are the synaptic currents due to neuron j, respectively k, at time t; V_{thr} is the 'threshold' voltage. Every time $V_i(t) = V_{thr}$ an action potential is emitted and V_i is reset to 0. Very often the $I_{j\to i}(t)$ are set to:

$$I_{j\to i}(t) = w_{j\to i} \sum_l \delta(t - t_{j,l})\,, \tag{2.2.2}$$

where $w_{j\to i}$ is referred to as the *synaptic weight*; δ stands for the Dirac delta distribution/function; the $t_{j,l}$ are the successive spike times of neuron j. In this model, when there are no inputs, the membrane potential relaxes towards 0 with a time constant $\tau = R_m C_m$. A presynaptic spike from an excitatory neuron j generates an instantaneous upward 'kick' of amplitude $w_{j\to i}$, while a presynaptic spike from an inhibitory neuron k generates an instantaneous downward 'kick' of amplitude $w_{k\to i}$. The actual action potential is not explicitly modeled—it was not doable at the time of Lapicque since the biophysics of this phenomenon was not understood—but is replaced by a point event. Notice that with the synaptic input description illustrated by Eq. 2.2.2 the current generated in the postsynaptic neuron by a given synapse does not depend on the membrane voltage of the former. This constitutes a crude approximation of the actual biophysics of synaptic current generation. A much better approximation (66; 94)—but harder to work with analytically—is provided by using:

$$I_{j\to i}(t) = g_{j\to i}(V_{rev} - V_i) \sum_l \delta(t - t_{j,l})\,, \tag{2.2.3}$$

where $g_{j\to i}$ is the *synaptic conductance*, V_{rev} is the synaptic current *reversal potential*—as its name says this is the voltage at which the current changes

sign—, it is negative or null for inhibitory synapses and larger than the threshold voltage for excitatory ones. Taking $V_{rev} = 0$ for the inhibitory inputs we see an important and empirically correct feature appearing: the downward 'kick' generated by the activation of an inhibitory input becomes *proportional* to the membrane voltage V_i; the larger the latter, the larger the kick.

2.3 The "stochastic picture"

The description presented so far is essentially deterministic and is far from giving the whole picture. The biophysical and molecular events leading to the action potential emission and propagation are by now well understood (71) and are known to involve the *stochastic* opening and closing of *voltage-gated ion channels*: the probability of finding a channel closed or opened depends on membrane voltage. This stochastic dynamics of the channels can lead to fluctuations in the action potential emission or propagation time when a given (deterministic) stimulation is applied repetitvely to a given neuron (98; 103). Similarly, the synaptic receptor-channels do fluctuate between open and close states, but that's the binding of the transmitter rather than the membrane potential that influences the probability of finding the channel in the open state. An even (much) larger source of fluctuations at the synapse results from the variable number of transmitter packets that get released upon a presynaptic action potential arrival (71)—even if the same presynaptic neuron is repetitively activated in the same conditions. The result of all these fluctuation sources is a rather "noisy" aspect of the membrane potential of cortical neurons that legitimates the use of "stochastic units" as building blocks of neural network models (94; 103). Historically, the first stochastic units were built by adding a Brownian motion process term to the right hand side of Eq. 2.2.1 (47; 16; 87). This approach leads to Chapman-Kolmogorov / Fokker-Planck equations that are hard to work with analytically and numerically. But identifying a neuron's sequence of action potentials with the realization of a point process suggests another strategy: modeling "directly" the process stochastic intensity (12; 21). This is the approach that will be followed in this book.

To be continued...

Chapter 3

A discrete time stochastic neural network model

3.1 Introduction

The goal of this chapter is to introduce our model in the most accessible hands-on way, avoiding as much as possible mathematical difficulties. In what follows, $\mathbb{N} = \{0, 1, 2, \dots, n, \dots\}$ denotes the set of positive integers including 0, $\mathbb{Z} = \{\dots - n, -n+1, \dots, 0, 1, \dots, n, \dots\}$ denotes the set of all integers, and $\mathbb{R}$ denotes the set of real numbers. The basic ingredients of our model are

1. a family of *synaptic weights* $w_{j\to i} \in \mathbb{R}$, for $j, i \in I$;

2. a family of *spiking probability or rate functions* $\phi_i : \mathbb{R} \to \mathbb{R}_+, i \in I$.

We interpret $w_{j\to i}$ as the *synaptic weight of neuron j on neuron i*. The functions ϕ_i are non-decreasing. The contribution of components j is either excitatory or inhibitory, depending on the sign of $w_{j\to i}$. We shall introduce

$$\mathcal{V}_{\cdot\to i} = \{j \in I : w_{j\to i} \neq 0\},$$

which is the set of *presynaptic neurons of i*, and

$$\mathcal{V}_{i\to\cdot} = \{j \in I : w_{i\to j} \neq 0\},$$

the set of *postsynaptic neurons of i*.

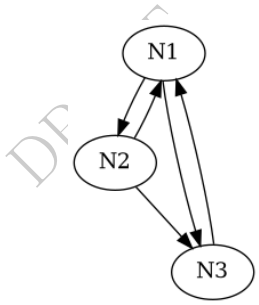
We observe that this defines a graph in which the neurons are the vertices and the synaptic connections are the edges. Since synaptic connections

are generally not symmetrical, we are dealing with *directed graphs*. For the simple illustrations considered in this chapter we are going to consider $w_{i\to j} = 0$ or 1 and we can then see these weights as the elements of the graph adjacency matrix.

For a case with 3 neurons: N1, N2 and N3, where N1 has N2 and N3 as postsynaptic partners, both with weights 1; N2 has N1 and N3 as postsynaptic partners, both with weights 1; and N3 has N1 has postsynaptic partner with weight 1, the adjacency matrix is:

	post		
pre	.	1	1
	1	.	1
	1	.	.

where we write "." instead of "0" following a convention commonly used for sparse matrix representations. The $\mathcal{V}_{\cdot\to i}$ are then the columns of the adjacency matrix, while the $\mathcal{V}_{i\to\cdot}$ are its rows. We will also use graphical representations for these matrices as illustrated here:



We can and will generalize the construction of the adjacency matrix by using elements that are not necessarily 0 (no connection) or 1 (connection) but by plugging-in the actual $w_{i\to j}$ values.

3.2 Basic discrete time model

In the discrete time setting, our model describes the spiking activity of a finite set I of neurons over time, where time is binned into small windows of length around 1 to 5 milliseconds. In this setting, all functions ϕ_i are supposed to take values in $[0, 1]$. For any neuron $i \in I$, $X_t(i) = 1$ indicates the presence of a spike within the time window of index t, and $X_t(i) = 0$ indicates the absence of a spike within the same time window. In what follows, we will simply speak of the value at time t instead of speaking of the time window of index t.

To describe the model, we need to introduce some extra notation. For each neuron $i \in I$ and each time $t \in \mathbb{Z}$, let $L_t(i)$ be *last spike time of neuron i before time t*. Formally,

$$L_t(i) = \max\{s \leq t : X_s(i) = 1\}. \tag{3.2.1}$$

In the following table that can be viewed as simplified *raster plot*, the top row contains the time index. Each subsequent row contains a snapshot of the *realizations* of 3 processes. The notation $x(i)$ $(i = 1, 2, 3)$ should be understood as

$$x(i) \equiv (\ldots, x_{-2}(i), x_{-1}(i), x_0(i), x_1(i), x_2(i), x_3(i), x_4(i), x_5(i), \ldots) \ .$$

The right columns shows the realization $l_5(i)$ of $L_5(i)$ for each of the three processes.

t	...	-2	-1	0	1	2	3	4	5	...	
x(1)	...	.	1	.	.	1	.	.	**1**	...	$l_5(1) = 5$
x(2)	...	1	.	**1**	.	.	.	.	.	...	$l_5(2) = 0$
x(3)	...	1	.	.	.	.	**1**	.	.	...	$l_5(3) = 3$

We also introduce the *membrane potential* of neuron i at time t,

$$V_t(i) = \sum_{j \in I} w_{j \to i} \left(\sum_{s=L_t(i)+1}^{t} X_s(j) \right), \ \text{ if } \ L_t(i) < t, \tag{3.2.2}$$

where we put

$$V_t(i) = 0 \ \text{ if } \ L_t(i) = t.$$

Thus, the membrane potential value is of neuron i obtained by adding up the contributions of all *presynaptic neurons $j \in \mathcal{V}_{\cdot \to i}$ of i* since its last spiking time. Moreover, the membrane potential is reset to 0 at each spiking time of the neuron.

Using the the previous adjacency matrix, the above realizations lead to:

t	...	-2	-1	0	1	2	3	4	5	...
v(1)	...	?	0	1	1	0	1	1	0	...
v(2)	...	0	1	0	0	1	1	1	2	...
v(3)	...	0	1	2	2	2	0	0	1	...

We see that $v_{-2}(1)$ is not defined ("?") since $l_{-2}(1)$ is missing.

Warning for physiologists: At this point, to simplify the presentation, we did not take into account any leakage effects, they will be introduced in Sec. 3.3. Synaptic delays and synaptic activation times can also easily be added to the present framework as will be illustrated in Sec. **à compléter**.

We now give an informal description of our model. Assuming that at time $t \geq 0$ each neuron has spiked at least once before time t, we do the following steps.

1. We compute $V_t(i)$ for every neuron i.
2. Every neuron i decides to spike at time $t+1$ with probability $\phi_i(V_t(i))$, independently of the others.
3. For every neuron i, we update the values of $X_{t+1}(i)$ according to the previous step and calculate $V_{t+1}(i)$ according to (3.2.2).

The above algorithm can be formally translated as follows. We start at time $t = 0$ from some initial condition $X_t(i) = x_t(i)$ for all $t \leq 0, i \in I$. We suppose that for all $i \in I$, there exist $\ell_i \leq 0$, such that $x_{\ell_i}(i) = 1$. This means that $\ell_i \leq L_0(i) \leq 0$ is well-defined for any $i \in I$, and that we are able to compute $V_0(i)$ for each neuron i.

We consider a family of uniform random variables $U_t(i), i \in I, t \geq 1$, which are i.i.d., uniformly distributed on $[0, 1]$. Then we define in a recursive way for every $t \geq 0$,

$$X_{t+1}(i) = \begin{cases} 1, & \text{if } U_{t+1}(i) \leq \phi_i(V_t(i)) \\ 0, & \text{if } U_{t+1}(i) > \phi_i(V_t(i)) \end{cases}, \tag{3.2.3}$$

where for each $t \geq 1$ and $i \in I$, $V_t(i)$ is the membrane potential of neuron i at the previous time step t, defined according to (3.2.2). By construction of the process, the probability that neuron i spikes at time $t + 1$ is a function of its membrane potential one time step before.

Rule (4.2.1) can be rephrased in the following way. At each step, given the past, neurons decide to spike or not independently the one from the others. This means that for any choice $a(i) \in \{0, 1\}, i \in I$,

$$P(\bigcap_{i \in I} \{X_{t+1}(i) = a(i)\} | X_s(j), s \leq t, j \in I) = \prod_{i \in I} P(X_{t+1}(i) = a(i) | X_s(j), s \leq t, j \in I), \tag{3.2.4}$$

where

$$P(X_{t+1}(i) = 1 | X_s(j), s \leq t, j \in I)$$
$$= P(X_{t+1}(i) = 1 | X_s(j), L_t(i) + 1 \leq s \leq t, j \in \mathcal{V}_{\cdot \to i}) = \phi_i(V_t(i)). \quad (3.2.5)$$

Observe that the process $(X_t)_{t\geq 0}, X_t = (X_t(i), i \in I)$, describing the spiking activity of each neuron, is not a Markov chain since its dependence on the past is not restricted to a fixed finite number of steps.

On the contrary, the process $(V_t)_{t\geq 0}, V_t = (V_t(i), i \in I)$, is a Markov chain and therefore more suitable for simulation issues.

Warning for probabilists The price to pay to obtain a Markovian description is that we replace a process having compact state space $\{0,1\}^I$ by another one having non-compact state space. Therefore simple issues as the existence of invariant probability measures requires more involved arguments.

The transitions of the Markov chain $(V_t)_{t\geq 0}$ can be described as follows:

$$V_{t+1}(i) = \begin{cases} 0, & \text{if } U_{t+1}(i) \leq \phi_i(V_t(i)) \\ V_t(i) + \sum_{j \neq i} w_{j \to i} 1_{\{U_{t+1}(j) \leq \phi_i(V_t(j)\}}, & \text{if } U_{t+1}(j) > \phi_i(V_t(j)). \end{cases} \quad (3.2.6)$$

Pseudocode to simulate the basic discrete time model

The following objects appear in the pseudocode:

N network size (number of neurons)

T simulation duration (number of time steps)

w $N \times N$ adjacency matrix (matrix of synaptic weights)

V vector of N elements with the membrane potential of each neuron

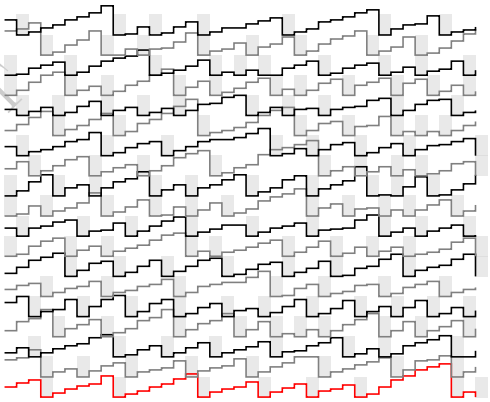
U vector of N independent and uniformly distributed random numbers (the elements of this vector are drawn at each time step)

The pseudocode is:

```
for t ← 1, T do
    for i ← 1, N do
        if U_i ≤ φ(V_i)) then                    ▷ Neuron i spikes
            X_i ← 1
```

```
5:            V_i ← 0
6:        end if
7:    end for
8:    for i ← 1, N do
9:        if X_i = 0 then                    ▷ Neuron i did not spike
10:           for j ← 1, N do
11:               V_i ← V_i + w_{j→i} × X_j
12:           end for
13:       end if
14:   end for
15: end for
```

For a complete `Python` implementation see Sec. 3.4.



The simulated membrane potential path of one neuron of the network (red line) together with the paths of all its presynaptic partners (alternating black and grey lines) are shown for the last 40 time steps. Observe that the paths go downward only when the corresponding neuron spikes (clear grey rectangles), since all the synaptic weights are positive.

This model includes two main features of stochastic integrate and fire models : synaptic integration and the release of action potentials depending on the current value of the membrane potential. One important aspect, present in most neurons, is however missing, this is the effect of leakage that we are now going to describe.

3.3 Introducing leakage

The presence of leakage channels in the membrane of a neuron tends to push the membrane potential of each neuron towards zero. We take this fact into account by adding to the above dynamics

$$\text{a family of } \textit{leak functions } g_i : \mathbb{R}_+ \to [0, \infty[, i \in I.$$

Every function g_i describes how neuron i looses potential due to leakage effects over time. Introducing leakage in the model, the membrane potential of neuron i at time t is now given by

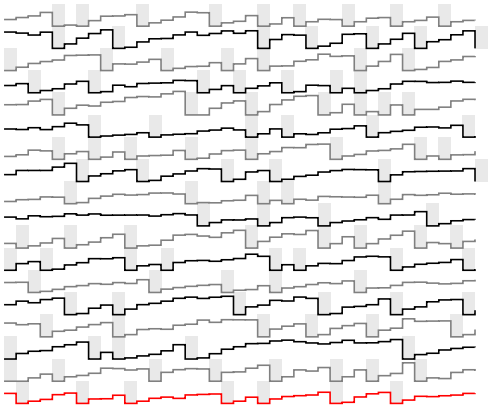
$$V_t(i) = \sum_{j \in I} w_{j \to i} \sum_{s=L_t(i)+1}^{t} g_i(t-s) X_s(j), \quad \text{if} \quad L_t(i) < t, \tag{3.3.7}$$

and $V_t(i) = 0$ if $L_t(i) = t$. The update rules for the process are given by (3.2.4) and (3.2.5) and the construction of the process according to (4.2.1) still applies.

It is important to observe that for general leak functions, the process V_t is not a Markov chain any more. This is due to the fact that the erosion rate at every step depends on the time elapsed since the last spike. An exception occurs when $g_i(s) = \rho_i^s$ for all $s \geq 0$, for all $i \in I$, for some $\rho_i \in [0,1]$. Then $(V_t)_{t \geq 0}$ is a Markov chain whose transitions are given by:

$$V_t(i) = \begin{cases} 0 & \text{if } U_t(i) \leq \phi_i(V_{t-1}(i)) \\ \rho_i V_{t-1}(i) + \sum_{j \neq i} w_{j \to i} 1_{\{U_t(j) \leq \phi_j(V_{t-1}(j)\}} & \text{if } U_t(i) > \phi_i(V_{t-1}(i)) \end{cases}. \tag{3.3.8}$$

The previous network simulated in the same conditions *except that a 20 % leakage effect has been added.* As before, the membrane potential path of one neuron of the network (red line) together with the paths of all its presynaptic partners (alternating black and grey lines) are shown for the last 40 time steps. Neurons spike times are indicated by clear grey rectangles.



3.4 Python code

A simple Python implementation of Eq. 3.2.6 and 3.3.8 is presented next. A network made of 100 neurons is considered. The connections are generated with a uniform probability of 0.2 per pairs of neurons (an Erdös-Rényi graph). The membrane potential looses 20 % of its value at each time step (80 % are therefore remaining)

when the neuron does not spike. The synaptic weight is 1 when there is a synapse. The rate function is linear between 0 and 40 (40 is critical in order to have a network that does not die too soon and that does not fire too many spikes), it is 0 below 0 and 1 above 40. The initial value of the membrane potential is drawn uniformly and independently from the discrete set $\{0, 1, \ldots, 40\}$. After each time step, the time index together with the membrane potential of each neuron are written to a text file. 1000 steps are simulated. This a "bare-bone" illustrative implementation, it could be made much more efficient (but less readable).

```
import random  # random number module
random.seed(20200622)  # set RNG seed
N = 100  # Number of neurons in network
connection_prob = 0.2  # connection probability
remain = 0.8 # fraction of potential remaining after one step due
             # to leakage (set to 1 for no leakage)
graph = [[i for i in range(N) if random.random() <= connection_prob and
          i != j]
         for j in range(N)]  # generate Erdos-Renyi graph

thresh = 2*N*connection_prob # voltage threshold
# If V_t(i) >= thresh the spiking probability is going to be 1
# This is what is implemented by the following function
def phi(v):
    return min(v/thresh,1.0)

V = [random.randint(0,thresh) for i in range(N)]  # initialize V

fout = open('Vproc_leak','w') # write simulation to file name 'Vproc'
fout.write(str([0]+V).strip('[]')+'\n')

n_step = 1000 # the number of time steps
for t in range(1,n_step+1): # Do the simulation
    P = [phi(v) for v in V]  # spiking probability
    # Find out next if each neuron spikes or not
    S = [random.random() <= P[i] for i in range(N)]
    for i in range(N):
        if S[i]:  # neuron n spiked
            V[i] = 0  # reset membrane potential
        else:  # neuron n did not spike
            V[i] *= remain # leakage effect
            for k in graph[i]: # look at presynaptic neurons
                if S[k]: # if presynaptic spiked
                    V[i] += 1 # identical weight of 1 for each synapse
    fout.write(str([t]+V).strip('[]')+'\n') # write new V

fout.close()
```

Some comments on the code for `Python` "newbies" The reader new to `Python` is strongly encouraged to read first the official tutorial (a short and really enjoyable read), while the impatient looking for a quick

reminder can consult the very useful Python syntax and semantics article from Wikipedia.

line 1 If Python like C is now everywhere and is here too last, it's partly because of its Standard Library, a collection of modules (in Python parlance) dedicated to specific tasks like manipulating compressed files (*eg* zlib), performing mathematical operations (math) or dealing with random numbers (random). To use the functions and variables of a module/library in Python code or in an interactive session, we must import it, that's what the first line is doing for the random module.

line 2 the default pseudorandom number generator (PRNG) of the random module is a Mersenne Twister; like any PRNG it is in fact a fully *deterministic* generator whose output "looks" random–and passes strong statistical tests of randomness–. It is therefore possible to generate the exact same sequence if one starts from the same generator state. This code line sets the starting state, in PRNG parlance, we are "seeding" the generator. Remark that we are calling a function seed defined in the random module and the way we are "telling" Python that "we want to use seed from the random module" is by prefixing seed with the module name using a dot in between. In a Python session we can get help about any function by calling the help function on the latter, *eg*, help(random.seed)). By using the same seed as we do in line 2, the reader will reproduce *exactly* our "random" simulation.

lines 7-9 An example of nested list comprehensions; lists are the most general way to store many objects in one place in Python. They start with "[" and end with "]", the different objects are separated by ",". Objects within a list need not to be of the same type: [1,2,3,4] is a valid list; so is [1,"2",3,"4"]. List elements can be accessed via a sub-setting syntax; if foo = [1,2,3,4], then the line foo[2] returns 3 (indices in Python start from 0). Concerning types, 2 is not the same as "2", the former has an int type (this implies, among other things, that Python returns 4 when we use the command 2+2), while the latter has a str type (a text sequence, so Python returns "22" when we use "2"+"2" because the + operator is a concatenation operator for text sequences). The idea of the list comprehension is the same as the mathematicians' way of defining a set by specifying a property shared by all set members rather than explicitly listing all of them (a clearly impossible task for infinite sets!). The call to random.random() is a *function call*, that's what the two "()" imply (they must be there,

otherwise `Python` would assume we are referring to a variable named `random`). random is a function of module `random` that takes *no argument* and that returns a random floating point number uniformly distributed in the range $[0, 1)$. The `list` referred to by the variable `graph` is a list of lists. It contains 100 sub-lists. The sub-list `graph[i]` (as in line 32) contains the indices of the neurons presynaptic to neuron `i`. Function range(N) generates sequentially the integers from 0 to `N-1`.

lines 14-15 Definition of function `phi`. This function takes a single formal parameter/argument `x`. To have a value a function *must* return something and what it returns in `Python` is what comes after the keyword `return`, here the minimum of the argument `x` divided by `thresh` (defined outside of the function body at line 9, see Python scoping rules do understand why this is valid) and of `1.0`. This is the first example of block definition by indentation: The function's body starts after the "`:`" and *must be indented*, here by four white spaces (the number of spaces is conventional, we could have used 3, but it must be consistent within a block and tabs are not identical their equivalent number of white spaces!). This indentation rule is a hotly debated issue in the programmers' world, although it was not introduced for historical reasons but readability ones (thereby enforcing in the language syntax something that is normally dealt with by the editor), it reminds me of `fortran 77` syntax...

line 17 Membrane potential initialization; a simpler example of list comprehension usage. Potentials are initialized to integer values and `random.randint` is called instead of `random.random`.

line 19 A file called `Vproc_leak` is opened in "writing" mode (that's what the second parameter, `w`, means).

line 20 The initial value of the membrane potential of each neuron is written to file `Vproc_leak` on a single line. The line starts with the time index (here 0). The special meaning of the `+` operator for lists is put to use: `[0]+V` concatenates the list `[0]` with the list `V`. The resulting list is then converted to a string representation (call to `str`) and the square brackets surrounding the string representation of a list are removed by calling the method `strip` with parameter `'[]'` on the string generated by: `str([0]+V)`. The "new line" character `\n` is then added to line end using again the special "concatenation" meaning of

the `+` operator for strings this time. The actual writing to the file is performed by calling the `write` method.

line 23-35 Main simulation loop.

line 24 Spike probability of each neuron is obtained by calling our function `phi` with the membrane potential of each neuron as parameter in a list comprehension.

line 26 A sequence of independent draws is used to "decide" if each neuron spikes or not in a list comprehension; the resulting list `S` is a list of Boolean values.

line 27-34 The individual neurons are processed successively (at a given time step `t`).

line 28-29 If the neuron `i` spikes, `S[i]` is `True`, its membrane potential is reset to zero.

line 30-34 The membrane potential of neuron `i` that doesn't spike, `S[i]` is `False`, is updated.

line 31 The leakage effect; this line could be removed in order to speed up the code for a model without leakage (`remain` set to 1).

line 32-34 The spiking behavior of each presynaptic neuron to neuron `i` is checked.

line 33-34 If neuron `k` presynaptic to neuron `i` spikes, then the membrane potential of neuron `i` is increased by 1.

line 35 The new value of the membrane potential of each neuron is written to our file with the time index starting the line.

line 37 Once the simulation is over, the file is closed.

3.5 Exercises and complements

The first three exercises require some probabilist training.

Exercice 1 : Is the membrane potential process Markov or not ?

i) Put

$$\tilde{V}_t(i) = \sum_{j \neq i} \sum_{s=L_{t-1}(i)+1}^{t-1} g_i(t-s) X_s(j)$$

and show that

$$V_t(i) = 1_{\{\ U_t(i) > \phi_i(V_{t-1}(i))\}} \left(\tilde{V}_t(i) + \sum_{j \neq i} w_{j \to i} 1_{\{U_t(j) \leq \phi_j(V_{t-1}(j)\}} \right).$$

ii) Compare $\tilde{V}_t(i)$ to $V_{t-1}(i)$ and show that $\tilde{V}_t(i)$ is a (deterministic) function of $V_{t-1}(i)$ if and only if $\frac{g_i(t-s)}{g_i(t-1-s)}$ is constant for all $L_{t-1}(i) < s \leq t-1$. Deduce from this that $(V_t)_{t \geq 0}$ is a Markov chain if and only if $g_i(t) = \rho_i^t$ for some $\rho_i \in [0,1]$, for all i.

Exercice 2 : And what about the spike trains ?

Let $k \geq 1$. We say that $X_t = (X_t(i), i \in I)$ is a Markov chain of order k if for any i and t, in (3.2.5),

$$\begin{aligned} &P(X_{t+1}(i) = 1 | X_s(j), s \leq t, j \in I) \\ &\quad = P(X_{t+1}(i) = 1 | X_s(j), (L_t(i)+1) \wedge t - k \leq s \leq t, j \in \mathcal{V}_{\cdot \to i}). \end{aligned} \quad (3.5.9)$$

i) Show that if for all $i \in I$, $g_i = c_i 1_{[0,T_i]}, T_i \leq k$, then (3.5.9) holds.

ii) Show that (3.5.9) implies that for each i, $g_i(t) = 0$ for all $t \geq k+1$, that is, the leakage functions are all of compact support.

Exercice 3: An embedded Markov chain

Notation and extra definitions. Given two integers $s \leq t$, the sequence $(x_r : s \leq r \leq t))$ of symbols belonging to the alphabet A will be denoted x_s^t. Given a sequence x_s^t, its length $t-s+1$ will be denoted $\updownarrow(x_s^t)$. Given a sequence x_s^t and an element $a \in A$, the sequence obtained by concatenating the sequence to the element will be denoted $x_s^t a$. This a sequence of length $t-s+2$, starting with x_s and ending with symbol a. The sequences x_r^t, where $s \leq r \leq t$, are called its *suffixes* of the sequence x_s^t.

i) Let the process $(X_t(i) : i \in I\ ,\ t \geq 0)$ be defined as in Section 3.2 and 3.3. For each $t \geq 0$, define

$$L_t = \inf\{L_t(i) : i \in I\}$$

and

$$C_t = (X^t_{L_t}(i); i \in I)\ .$$

For each $t \geq 0$, show that C_{t+1} is a suffix of $C_t X_{t+1}$.

ii) Show that $(C_t\ :\ t \geq 0)$ is a Markov chain of order 1. Notice that this Markov property is not restricted to settings where the leakage function decays exponentially with time (as in Exercice 1). The price to pay is to work with a much more complicated state space.

iii) Assuming that $\phi_i(0) = 0$ for every $i \in I$, show that the null configuration $n_t(i) = 0$, for any $i \in I$ and any $t \geq 0$, is trap for the chain a $(C_t : t \geq 0)$.

iv) Assuming that $\phi_i(u) \geq \delta > 0$, for any $i \in I$ and any $u \in R$, prove that the Markov chain $(C_t : t \geq 0)$ has at least one invariant measure and never stops spiking, i.e.

$$\mathbb{P}\left(\cap_{i\in I} \cap_{t\geq 0} \cup_{s\geq t}\{X_s(i) = 1\}\right) = 1\ .$$

v) Assuming that $\phi_i(u) \geq \delta > 0$, for any $i \in I$ and any $u \in R$, under which conditions on the set of synaptic weigths $(w_{j\to i} : i \in I, j \in I$, the Markov chain $(C_t : t \geq 0)$ has a unique invariant probability measure.

vi) Let now the process $(X_t(i) : i \in I\ ,\ t \geq 0)$ be defined as in Section 3.3, with leakage. Assuming that $\phi_i(0) = 0$ and $\rho_i < 1$, for every $i \in I$. Prove that this processes spikes only a finite number of times, i.e.

$$\mathbb{P}\left(\cup_{i\in I} \cap_{t\geq 0} \cup_{s\geq t}\{X_s(i) = 1\}\right) = 0\ .$$

Exercice 4: Cross-correlation of simulated trains

Use the Python simulation code of the Section 3.4 to simulate a simple network made of 3 neurons. Neuron 1 has an input with a large synaptic weight on neuron 2; it has a weak input on neuron 3; neurons 2 and 3 do not form synapses on any of the other two neurons. Make sure that neuron 1 never stops spiking by having a positive value of the rate function when the membrane potential is null.

i) Simulate such a network for long enough to observe of the order of 1000 spikes from every neuron.

ii) Write a Python code computing the *cross-correlogram* between a pair of neurons. Given two observed trains, $\{x_1(i), x_2(i), \ldots, x_T(i)\}$ and $\{x_1(j), x_2(j), \ldots, x_T(j)\}$, the *cross-correlogram* at lag τ, $\hat{c}_\tau(i,j)$, between the *reference* i and the *test* j train estimates the probaility that a spike from neuron j follows a spike from neuron i by exactly τ time steps. It is formally defined by:

$$\hat{c}_\tau(i,j) = \frac{\sum_{\max(-\tau,1)}^{\min(T,T-\tau)} x_t(i)\, x_{t+\tau}(j)}{\sum_{\max(-\tau,1)}^{\min(T,T-\tau)} x_t(i)}\,,$$

for $-\tau_{max} \leq \tau \leq \tau_{max}$ and $\tau_{max} > 0$.

The *cross-correlogram* is called the *cross-intensity* by **(author?)** (13).

iii) Use this code to compute the cross-correlogram between every pair of simulated neurons.

iv) Change the synaptic weights and observe the effect on the cross-correlograms.

Chapter 4

A case study: correlations between successive inter spike intervals

4.1 Are inter spike intervals correlated?

An often measured parameter in neurobiologial experiments is the correlation between successive inter-spike intervals (ISI) of a given neuron (77). Neurophysiologists use this parameter to decide what type of model is appropriate for their spike trains. The simplest spike train model, a Poisson point process, is 'never' adequate because of the refractory period. The next simplest model is the renewal process that is fully specified by the ISI distribution (and the distribution of the first spike). A requirement for a renewal process is the absence of correlation between successive ISI. Starting with (50; 48) experimental studies in several species and brain regions reported both significant and un-significant correlations. The question has been repetitively addressed (*e.g.* (73)) and examples of both have been described.

It is therefore mandatory to check whether a proposed mathematical model aiming at describing actual data exhibits significant ISI correlation or not. This is the purpose of the present chapter. We will show that we can account for the apparently contradictory facts of observing both correlated and uncorrelated ISI, within the framework introduced in Chapter 3.

4.2 Spiking neurons interacting through an Erdös-Rényi random graph

We work in a setup in which the synaptic weights define a directed random graph with a large but finite number of components. The simplest random graph one can think of is one where the probabilty of observing a connection between an ordered pair of arbitrary neurons (the first pair member is the presynaptic neuron, while the second member is the postsynaptic neuron) is the same for all pairs. Moreover the decision concerning a specific ordered pair does not influence in any way the decision relative to any other pair of neurons in the network. In probabilistic terms we toss a biased coin for each ordered pair of neurons to assign or not a synaptic connection. This is done sequentially and independently with the same coin for each ordered pair of neurons. This simple random graph is called a "directed Erdös-Rényi random graph". We will provide more mathematical insight in our Appendix Section **??**. For a general reference on random graphs we refer the reader to (6).

Despite of its apparent simplicity a slightly refined version of this model is commonly used in neural modeling where several coins are used instead of a single one; each coin corresponding to a specific combination of neuronal type and neuronal location. For instance the cortical column model considered by (79) uses this procedure to generate a network with $80\text{x}10^6$ neurons where each neuron belongs to one of 8 possible types making a total of 64 different ordered pairs and therefore using 64 different coins. This is illustrated on Fig. 4.1 and by the following connectivity matrix giving the

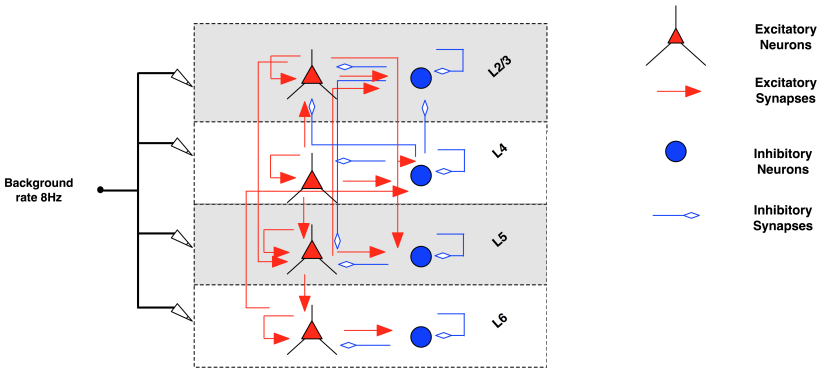


Figure 4.1: Example of network with 6 neuronal classes (81).

coin biais value

Table 4.1: Connectivity matrix between the different populations of the model (extracted from (81)). The connectivity matrix describes the probabilities of the target-specific connections between populations of neurons.

		from								
		L23e	L23i	L4e	L4i	L5e	L5i	L6e	L6i	Th
to	L23e	0.101	0.169	0.044	0.082	0.032	0.0	0.008	0.0	0.0
	L23i	0.135	0.137	0.032	0.052	0.075	0.0	0.004	0.0	0.0
	L4e	0.008	0.006	0.050	0.135	0.007	0.0003	0.045	0.0	0.0983
	L4i	0.069	0.003	0.079	0.160	0.003	0.0	0.106	0.0	0.0619
	L5e	0.100	0.062	0.051	0.006	0.083	0.373	0.020	0.0	0.0
	L5i	0.055	0.027	0.026	0.002	0.060	0.316	0.009	0.0	0.0
	L6e	0.016	0.007	0.021	0.017	0.057	0.020	0.040	0.225	0.0512
	L6i	0.036	0.001	0.003	0.001	0.028	0.008	0.066	0.144	0.0196

In the sequel we will work with the simple directed Erdös-Rényi random graph. This choice is justified if we restrict ourselves locally to a specific cell type in the neocortical region. As illustrated by the first figure in (8), neocortical tissues display a quite chaotic structure, where at first sight, any neuron can locally connect with any other neuron. This impression is confirmed by quantitative and analytical studies focusing on the main neocortical cell type, the pyramidal cell, and adopted, for instance, in a very influential paper by (5).

Let us start by defining the directed Erdös-Rényi random graph in a precise mathematical way. We consider a finite system consisting of a large number N of neurons with random synaptic weights $W_{i\to j}, i \neq j$. Here we use capital letters to distinguish the random variables from their deterministic counterparts $w_{i\to j}$ that we have used up to now. In our model there is no self-interaction, that is, all $W_{i\to i} = 0$. Moreover, the random variables $W_{i\to j}, i \neq j$, are independent and identically distributed random variables taking the values 0 or 1. In the mathematical literature such random variables are called *Bernoulli random variables*. We denote p_N the probability that $W_{i\to j}$ equals 1. Starting from the family $W_{i\to j}, i \neq j$, we now define the associated directed graph such that the directed link $i \to j$ is present if and only if $W_{i\to j} = 1$.

Notice that the synaptic weights $W_{i\to j}$ and $W_{j\to i}$ are distinct and independent random variables. In the sequel, we first choose a random graph, that is, a particular realization of synaptic weights $W = (W_{i\to j}, i \neq j)$. Let us call $\tilde{P}$ the probability measure we used to make this choice. This procedure seems rather complicated, but it actually only requires to simulate a sequence of independent uniformly distributed random variables to choose the values of $W_{i\to j}$ for all $i \neq j$.

Once the graph W is chosen, we define the time evolution of the chain. To do so, we consider another family of uniform random variables $U_t(i), i \in I, t \geq 1$, which are i.i.d., uniformly distributed on $[0,1]$, independent of the one used in the first stage. Then we define in a recursive way for every $t \geq 0$,

$$X_{t+1}^W(i) = \begin{cases} 1, & \text{if } U_{t+1}(i) \leq \phi_i(V_t^W(i)) \\ 0, & \text{if } U_{t+1}(i) > \phi_i(V_t^W(i)) \end{cases}, \tag{4.2.1}$$

where for each $t \geq 1$ and $i \in I$, $V_t^W(i)$ is the membrane potential of neuron i at the previous time step t, defined by

$$V_t^W(i) = \sum_{j=1}^{N} W_{j\to i} \sum_{s=L_t(i)+1}^{t} g_i(t-s) X_s(j).$$

From now on, whenever it is clear with which fixed W we work, we will omit mentioning it and write X_t and V_t instead of X_t^W and V_t^W.

Fix a neuron i and consider its associated sequence of successive spike times

$$\ldots < T_{-n}(i) < \ldots < T_0(i) \leq 0 < T_1(i) < T_2(i) < \ldots < T_n(i) < \ldots, \tag{4.2.2}$$

where

$$T_1(i) = \inf\{t \geq 1 : X_t(i) = 1\}, \ldots, T_n(i) = \inf\{t > T_{n-1}(i) : X_t(i) = 1\},$$

$n \geq 2$, and

$$T_0(i) = \sup\{t \leq 0 : X_t(i) = 1\}, \ldots, T_{-n}(i) = \sup\{t < T_{-n+1}(i) : X_t(i) = 1\},$$

$n \geq 1$.

In neuroscience, $T_{k+1}(i) - T_k(i), k \geq 0$, is referred to as *interspike interval* (ISI). As we mentioned, it has been reported in (73) that successive interspike intervals have negligible correlations. To check weather this feature is

reproduced by our model, we introduce the covariance between successive inter-spike intervals, given a fixed choice of synaptic weights W, by

$$\begin{aligned} Cov^W(T_{k+1}(i) - T_k(i), T_k(i) - T_{k-1}(i)) \\ = E^W[(T_{k+1}(i) - T_k(i))(T_k(i) - T_{k-1}(i))] \\ - E^W(T_{k+1}(i) - T_k(i))E^W(T_k(i) - T_{k-1}(i)), \end{aligned}$$

for any $k \neq 0, 1$.

To obtain estimates for these covariances, we need to make specific assumptions on the graph of interactions. Following (5) we assume that the Erdös-Rényi random graph is slightly super-critical. This means that

$$p_N = \lambda/N, \tag{4.2.3}$$

where

$$\lambda = 1 + \vartheta/N \text{ for some } 0 < \vartheta < \infty\,. \tag{4.2.4}$$

Here, super-critical means that for each neuron, the mean number of post-synaptic neurons related to it has an average strictly greater than one. This implies that most neurons in the network are connected. More details are given in the appendix.

We will show that for most choices of the graph of interactions, the above covariance is exponentially small in N, for large values of N. More precisely, the following theorem holds true.

Theorem 1. *Assume that there exists $\gamma > 0$, such that for all i and for all $s, s' \in \mathbb{R}$,*

$$|\phi_i(s) - \phi_i(s')| \leq \gamma|s - s'|. \tag{4.2.5}$$

Suppose moreover that there exists $\delta > 0$ such that for all $i \in I, s \in \mathbb{R},$,

$$\phi_i(s) \geq \delta. \tag{4.2.6}$$

Then there exists a subset A of realizations of the synaptic weights such that on A,

$$|Cov^W(T_{k+2}(i) - T_{k+1}(i), T_{k+1}(i) - T_k(i))| \leq \frac{3}{\delta^2} N(1-\delta)^{\sqrt{N}}.$$

Moreover,

$$\tilde{P}(A) \geq 1 - 2\frac{e^{2\vartheta}}{\sqrt{N}}.$$

Does the past before the last spike of a neuron influence its future?

The rest of this chapter is devoted to the proof of the above theorem. Before discussing mathematical details let us discuss the intuition behind. The most important point to understand is how a given spike of a given neuron, say i successively influences other neurons which in turn influence other neurons and so on until the moment the effect of this initial spike eventually returns to neuron i (if this ever happens).

The point to be understood is the following. The last spike of neuron i before time $L_t(i)$ affects many neurons different from i. These neurons in turn affect other neurons and so on. How long does it take until this chain of influence returns to the starting neuron i?

To formalize this question we introduce the following sequence of sets:

$$\begin{aligned}\mathcal{V}^1_{i\to\cdot} &= \{j : W_{i\to j} = 1\},\\ &\vdots\\ \mathcal{V}^n_{i\to\cdot} &= \{j : \exists k \in \mathcal{V}^{n-1}_{i\to\cdot} : W_{k\to j} = 1\}, n \geq 2.\end{aligned}$$

Note that $\mathcal{V}^1_{i\to\cdot}$ is the set of neurons influenced by neuron i in one step and $\mathcal{V}^n_{i\to\cdot}$ is the set of neurons influenced by neuron i in n steps. We define

$$\tau^i = \inf\{n : i \in \mathcal{V}^n_{i\to\cdot}\}.$$

Informally speaking this is the first time that an information emitted by neuron i can return to neuron i itself.

Recall (Eq. 4.2.4) that $\lambda = 1 + \vartheta/N$. We have the following bound.

Proposition 2. *For any k the following inequality holds.*

$$\tilde{P}(\tau^i \leq k) \leq \frac{k-1}{N} \exp\left(\vartheta \frac{k}{N}\right).$$

Proof. The proof compares the sequence $(\mathcal{V}^n_{i\to\cdot})_{n\geq 1}$ with a different one in which at each step we choose the links in an independent way, excluding the choice of i itself. Let us call this new sequence $(\tilde{\mathcal{V}}^n_{i\to\cdot})_{n\geq 1}$. The choice of the links appearing at each step is done in such a way that the two sequences, $(\mathcal{V}^n_{i\to\cdot})_{n\geq 1}$ and $(\tilde{\mathcal{V}}^n_{i\to\cdot})_{n\geq 1}$ are coupled together and

$$\cup_{n=1}^m \mathcal{V}^n_{i\to\cdot} \subset \cup_{n=1}^m (\tilde{\mathcal{V}}^n_{i\to\cdot}),\ \forall m < \tau^i\,.$$

The reason to do this is that it is much simpler to compute the probabilities of events related to the sequence $(\tilde{\mathcal{V}}^n_{i\to\cdot})_{n\geq 1}$.

Formally this is done as follows. Let us suppose that $j \neq i$ does not belong to any of the sets $\tilde{\mathcal{V}}^k_{i\to\cdot}$, $k = 1, \ldots, n-1$, but j belongs to $\tilde{\mathcal{V}}^n_{i\to\cdot}$. This means that the choice of the random synaptic weights $W_{j\to j'}$, $\forall j' \neq i$, never interfered in the choices of the previous sets $\tilde{\mathcal{V}}^k_{i\to\cdot}$, $k = 1, \ldots, n$. In this case we are free to choose the original values of the random variables $W_{j\to j'}$ for all $j' \neq i$ to construct the new set $\tilde{\mathcal{V}}^{n+1}$. In opposition, every time a label j appears for the second time or more in the sequence, we are not allowed to use again the original value of $W_{j\to j'}$. What do we do in this case? Every time j appears again we select new random variables $W^m_{j\to j'}$ independently of the first choices. In this notation, m stands for the mth time the random variable is selected independently of past choices. With this notation the original choice of the random variables $W_{j\to j'}$ is now denoted $W^1_{j\to j'}$.

With this construction, the two sequences $(\mathcal{V}^n_{i\to\cdot})_{n\geq 1}$ and $(\tilde{\mathcal{V}}^n_{i\to\cdot})_{n\geq 1}$ are equal up to the first time n at which a label j (including i) appears a second time. Moreover, since in the new sequence $(\tilde{\mathcal{V}}^n_{i\to\cdot})_{n\geq 1}$ we allow for new, independent choices of the random variables $W^m_{j\to j'}$, independently of the first choices, it is clear that $\cup_{n=1}^m \mathcal{V}^n_{i\to\cdot} \subset \cup_{n=1}^m (\tilde{\mathcal{V}}^n_{i\to\cdot})$, for all $m < \tau^i$.

We have

$$\tilde{P}(\tau^i > k) = \tilde{P}\left(W_{j\to i} = 0 \ \forall\, j \in \bigcup_{n\leq k-1} \mathcal{V}^n_{i\to\cdot}\right) \geq \tilde{P}\left(W_{j\to i} = 0 \ \forall\, j \in \bigcup_{n\leq k-1} \tilde{\mathcal{V}}^n_{i\to\cdot}\right).$$

Since in the definition of $\tilde{\mathcal{V}}^n_{i\to\cdot}$, no choice $W_{\cdot\to i}$ has been made, we can condition with respect to $\bigcup_{n\leq k-1} \tilde{\mathcal{V}}^n_{i\to\cdot}$, use the fact that for any $j \in \bigcup_{n\leq k-1} \tilde{\mathcal{V}}^n_{i\to\cdot}$, the random variable $W_{j\to i}$ is independent of $\bigcup_{n\leq k-1} \tilde{\mathcal{V}}^n_{i\to\cdot}$, and obtain the following inequality

$$\tilde{P}(\tau^i > k) \geq \tilde{E}\left[(1-p_N)^{|\bigcup_{1\leq n\leq k-1} \tilde{\mathcal{V}}^n_{i\to\cdot}|}\right].$$

We conclude as follows. We observe that for $n \geq 2$, the process $Z_n := |\tilde{\mathcal{V}}^n_{i\to\cdot}|$, where $|\tilde{\mathcal{V}}^n_{i\to\cdot}|$ denotes the number of elements belonging to $\tilde{\mathcal{V}}^n_{i\to\cdot}$, is a classical branching process, starting from $Z_1 = \mathcal{V}^1_{i\to\cdot}$. In this branching process, each element belonging to a given generation gives rise to a random number of offspring elements with mean $\mu = (N-2)\frac{\lambda}{N}$. Here, the factor $N-2$ comes from the fact that any j has $N-2$ choices of choosing arrows $W_{j\to\cdot}$, since j itself and i are excluded.

Write $\Sigma_{k-1} = Z_1 + \ldots + Z_{k-1}$ and let $\tilde{E}(s^{\Sigma_{k-1}}), s \leq 1$, be its moment generating function. Using the convexity of the moment generating function,

we have that

$$\tilde{E}(s^{\Sigma_{k-1}}) \geq 1 + \tilde{E}(\Sigma_{k-1})(s-1).$$

Using that $\tilde{E}(Z_1) = \frac{N-1}{N}\lambda$ and that the offspring mean is $\mu = \frac{N-2}{N}\lambda$, the claim follows from

$$\tilde{E}(\Sigma_{k-1}) = \frac{N-1}{N}\lambda\left[1 + \mu + \ldots + \mu^{k-2}\right] \leq \lambda + \ldots + \lambda^{k-1} \leq (k-1)\lambda^{k-1},$$

since $\mu \leq \lambda$ and $\lambda \geq 1$. Hence, evaluating the above lower bound in $s = 1 - p_N$, we obtain

$$\tilde{P}(\tau^i > k) \geq 1 - p_N(k-1)\lambda^{k-1},$$

and therefore,

$$\tilde{P}(\tau^i \leq k) \leq p_N(k-1)\lambda^{k-1} = \frac{k-1}{N}\lambda^k,$$

since $p_N = \lambda/N$. Using that $\lambda = 1 + \vartheta/N$, we obtain the assertion. □

In what follows, we are going to evaluate the probability that neuron i spikes at at given time, given a fixed past of length k of its own history. To denote such a past, we introduce the following notation. We write a_{-k}^{-1} for the finite sequence $(a_{-1}, \ldots, a_{-k})$, where each a_{-i} is either 1, indicating the presence of a spike, or 0, indicating the absence of a spike, for $1 \leq i \leq k$. In particular, the notation $0^{k-1}1a_{-l}^{-1}$ stands for the sequence given by

$$(0, \ldots, 0, 1, a_{-1}, \ldots, a_{-l}).$$

We write

$$p^{(W,i)}(1|a_{-k}^{-1}) = P^W(X_k(i) = a|X_{k-1}(i) = a_{-1}, \ldots, X_0(i) = a_{-k})$$

for the probability that neuron i spikes, given a fixed choice of synaptic weights W and given its past of length k equals a_{-k}^{-1}. In what follows, conditionings will be read from the left to the right. In particular, we write

$$p^{(W,i)}(a|0^{k-1}1a_{-l}^{-1}) =$$
$$P^W(X_k(i) = a|X_{k-1}(i) = \ldots = X_1(i) = 0, X_0(i) = 1, X_{-1}(i) = a_{-1}, \ldots, X_{-l}(i) = a_{-l}).$$

The following proposition shows that on the event $\{\tau^i > k+l\}$, the two transition probabilities $p^{(W,i)}(1|0^{k-1}1a_{-l}^{-1})$ and $p^{(W,i)}(1|0^{k-1}1)$ necessarily coincide.

Proposition 3. *For any $k \geq 1, l \geq 1$,*

$$\{p^{(W,i)}(1|0^{k-1}1a_{-l}^{-1}) \neq p^{(W,i)}(1|0^{k-1}1)\} \subset \{\tau^i \leq k+l\}.$$

Proof. Let W be fixed. From now on, since we will work for this fixed choice of W, we will omit the superscript W and write for short $p^i(a|a_{-k}^{-1})$ instead of $p^{(W,i)}(a|a_{-k}^{-1})$ and so on. Recall that $\mathcal{V}_{\cdot\to i} = \{j : W_{j\to i} = 1\}$. We have

$$\begin{aligned}
&P(X_k(i) = 1, X_1^{k-1}(i) = 0^{k-1}, X_0(i) = 1, X_{-l}^{-1}(i) = a_{-l}^{-1}) \\
&\quad = \sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{z_1^{k-1}(j)\in\{0,1\}^{k-1}} P(X_k(i) = 1, X_1^{k-1}(i) = 0^{k-1}, X_0(i) = 1, \\
&\qquad\qquad X_{-l}^{-1}(i) = a_{-l}^{-1}, X_1^{k-1}(j) = z_1^{k-1}(j), \forall j \in \mathcal{V}_{\cdot\to i}) \\
&\quad = \sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{z_1^{k-1}(j)\in\{0,1\}^{k-1}} \phi_i\left(\sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{s=1}^{k-1} g_i(k-s)z_s(j)\right) \times \\
&\qquad\qquad \times P(X_1^{k-1}(j) = z_1^{k-1}(j), \forall j \in \mathcal{V}_{\cdot\to i}, X_{-l}^{k-1}(i) = a_{-l}^{-1}10^{k-1}).
\end{aligned}$$

Thus,

$$\begin{aligned}
p^i(1|0^{k-1}1a_{-l}^{-1}) = &\sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{z_1^{k-1}(j)\in\{0,1\}^{k-1}} \phi_i\left(\sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{s=1}^{k-1} g_i(k-s)z_s(j)\right) \times \\
&\times P(X_1^{k-1}(j) = z_1^{k-1}(j), \forall j \in \mathcal{V}_{\cdot\to i} | X_{-l}^{k-1} = a_{-l}^{-1}10^{k-1}).
\end{aligned}$$

The same calculus shows that

$$\begin{aligned}
p^i(1|0^{k-1}1) = &\sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{z_1^{k-1}(j)\in\{0,1\}^{k-1}} \phi_i\left(\sum_{j\in\mathcal{V}_{\cdot\to i}} \sum_{s=1}^{k-1} g_i(k-s)z_s(j)\right) \cdot \\
&\times P(X_1^{k-1}(j) = z_1^{k-1}(j), \forall j \in \mathcal{V}_{\cdot\to i} | X_0^{k-1}(i) = 10^{k-1}).
\end{aligned}$$

This shows that in order to ensure that $p^i(1|0^{k-1}1a_{-l}^{-1}) = p^i(1|0^{k-1}1)$, it is sufficient to have

$$\begin{aligned}
&P(X_1^{k-1}(j) = z_1^{k-1}(j), \forall j \in \mathcal{V}_{\cdot\to i} | X_0^{k-1}(i) = 10^{k-1}) = \\
&\qquad = P(X_1^{k-1}(j) = z_1^{k-1}(j), \forall j \in \mathcal{V}_{\cdot\to i} | X_{-l}^{k-1}(i) = a_{-l}^{-1}10^{k-1}), \quad (4.2.7)
\end{aligned}$$

for all possible choices of $z_1^{k-1}(j), j \in \mathcal{V}_{\cdot\to i}$, which is implied by $\tau^i > k+l$. □

We are now able conclude the proof of Theorem 1. We call $\tilde{\Omega}$ the probability space where all synaptic weights W are realized, with $\tilde{P}$ the corresponding probability measure. Condition (4.2.6) allows to introduce a sequence of independent Bernoulli random variables $(\xi_t(i), i \in I, t \in \mathbb{Z})$ of parameter δ, such that positions and times (i,t) with $\xi_t(i) = 1$ are spike times for any realization of the chain. Write $l = \sup\{n < T_2(i) : \xi_n(i) = 1\}$ and $r = \inf\{n > T_2(i) : \xi_n(i) = 1\}$. Since the label i is fixed, in what follows, we write for short T_n for $T_n(i)$. Put

$$A = \{\tau^i > 2k(N)\},$$

where $k(N)$ is such that $k(N) \to \infty$ as $N \to \infty$ and $k(N) \leq N$. We will fix the choice of $k(N)$ later. We have for any realization of $W \in A$,

$$\begin{aligned} &E^W[(T_3 - T_2)(T_2 - T_1)] \\ &\qquad \leq E^W[(r - T_2)(T_2 - l)1_{\{l<T_2-k(N)\}\cup\{r>T_2+k(N)\}}] \\ &\qquad\quad + E^W[(T_3 - T_2)(T_2 - T_1)1_{\{l\geq T_2-k(N);r\leq T_2+k(N)\}}]. \end{aligned} \tag{4.2.8}$$

Using that conditionally on T_2, $r - T_2$ and $T_2 - l$ are independent and geometrically distributed, we obtain a first upper bound

$$\begin{aligned} &E^W[(r - T_2)(T_2 - l)1_{\{l<T_2-k(N)\}\cup\{r>T_2+k(N)\}}] \\ &\qquad\qquad \leq \frac{1}{\delta^2}(k(N)+2)(1-\delta)^{k(N)}. \end{aligned} \tag{4.2.9}$$

Here, we have used that for a geometrically distributed random variable T of parameter δ, $E(T1_{\{T>k\}}) = (1-\delta)^k \frac{1}{\delta}$.

We now consider the second term and use that $\tau^i > 2k(N)$. We have

$$\begin{aligned} &E^W[(T_3 - T_2)(T_2 - T_1)1_{\{l\geq T_2-k(N);r\leq T_2+k(N)\}}] \\ &\qquad = \sum_t E^W[(T_3 - t)(t - T_1)1_{\{l\geq t-k(N);r\leq t+k(N)\}}1_{\{T_2=t\}}] \\ &= \sum_t E^W\left[(t - T_1)1_{\{l\geq t-k(N)\}}1_{\{T_2=t\}}E^W[(T_3 - t)1_{\{r\leq t+k(N)\}}|X^t_{t-k(N)}(i)]\right]. \end{aligned} \tag{4.2.10}$$

Now, since $T_3 \leq r$,

$$E^W[(T_3 - t)1_{\{r \leq t+k(N)\}} | X^t_{t-k(N)}(i)] = \sum_{n=1}^{k(N)} n \times P^W(T_3 - t = n; r \leq t + k(N) | X^t_{t-k(N)}(i)) \leq \sum_{n=1}^{k(N)} n \times P^W(T_3 - t = n | X^t_{t-k(N)}(i)).$$

Notice that

$$P^W(T_3 - t = n | X^t_{t-k(N)}(i)) = p^i(0|1X^{t-1}_{t-k(N)}(i)) \times p^i(0|01X^{t-1}_{t-k(N)}(i)) \times \ldots \times p^i(0|0^{n-2}1X^{t-1}_{t-k(N)}(i)) p^i(1|0^{n-1}1X^{t-1}_{t-k(N)}(i)).$$

Now we use Proposition 3. Since we are working on $\{\tau^i > 2k(N)\}$, we have

$$p^i(0|1X^{t-1}_{t-k(N)}(i)) = p^i(0|1), \ldots, p^i(1|0^{n-1}1X^{t-1}_{t-k(N)}(i)) = p^i(1|0^{n-1}1),$$

for all $n \leq k(N)$. Therefore,

$$\begin{aligned} E^W[(T_3 - t)1_{\{r \leq t+k(N)\}} | X^t_{t-k(N)}(i)] &\leq \sum_{n=1}^{k(N)} n \times p^i(0|1)p^i(0|01) \times \ldots \times p^i(0|0^{n-2}1)p^i(1|0^{n-1}1) \\ &\leq \sum_{n=1}^{\infty} n \times p^i(0|1)p^i(0|01) \times \ldots \times p^i(0|0^{n-2}1)p^i(1|0^{n-1}1) \\ &= E^W(T_3 - T_2). \quad (4.2.11) \end{aligned}$$

We conclude that on A, using successively (4.2.8)–(4.2.11),

$$E^W[(T_3 - T_2)(T_2 - T_1)] \leq \frac{1}{\delta^2}(k(N) + 2)(1 - \delta)^{k(N)} + E^W(T_3 - T_2)E^W(T_2 - T_1).$$

In a second step, we are seeking for lower bounds. We start with

$$E^W[(T_3 - T_2)(T_2 - T_1)] \geq E^W[(T_3 - T_2)(T_2 - T_1)1_{\{l \geq T_2 - k(N); r \leq T_2 + k(N)\}}]. \quad (4.2.12)$$

Then on $\{T_2 = t\}$,

$$\begin{aligned}
&E^W[(T_3 - t)1_{\{r \le t+k(N)\}} | X^t_{t-k(N)}(i)] \\
&\qquad = \sum_{n=1}^{k(N)} n \times P^W(T_3 - t = n; r \le t + k(N) | X^t_{t-k(N)}(i)) \\
&\qquad\qquad \ge \left(\sum_{n=1}^{k(N)} n \times P^W(T_3 - t = n | X^t_{t-k(N)}(i)) \right) \\
&\qquad\qquad\quad - k(N)^2 P^W(r > t + k(N) | X^t_{t-k(N)}(i)) \\
&\qquad = \left(\sum_{n=1}^{k(N)} n \times P^W(T_3 - t = n | X^t_{t-k(N)}(i)) \right) - k(N)^2 (1-\delta)^{k(N)}.
\end{aligned}$$

Now, on $\{T_2 = t\}$,

$$\begin{aligned}
&\sum_{n=1}^{k(N)} n \times P^W(T_3 - t = n | X^t_{t-k(N)}(i)) \\
&= E^W(T_3 - T_2; T_3 - T_2 \le k(N)) = E^W(T_3 - T_2) - E^W(T_3 - T_2; T_3 - T_2 > k(N)) \\
&\qquad \ge E^W(T_3 - T_2) - E^W(r - T_2; r - T_2 > k(N)) \\
&\qquad\qquad \ge E^W(T_3 - T_2) - \frac{1}{\delta}(k(N) + 2)(1-\delta)^{k(N)}.
\end{aligned}$$

Therefore, for any realization $W \in A$,

$$\begin{aligned}
&E^W[(T_3 - T_2)(T_2 - T_1)] \ge \\
&\qquad E^W(T_3 - T_2) E^W(T_3 - T_2) - [\frac{2}{\delta^2}(k(N) + 2) + k(N)^2](1-\delta)^{k(N)}.
\end{aligned}$$

Putting things together and supposing that $k(N) + 2 \le k(N)^2$, we obtain finally

$$|E^W[(T_3 - T_2)(T_2 - T_1)] - E^W(T_3 - T_2) E^W(T_3 - T_2)| \le \frac{3}{\delta^2} k(N)^2 (1-\delta)^{k(N)}.$$

It remains to find an upper bound for $\tilde{P}(A^c)$. Clearly, applying Proposition 2, since $k(N) \le N$, we have

$$\tilde{P}(A^c) \le e^{2\vartheta} \frac{2k(N)}{N}.$$

It is enough to choose $k(N) = \sqrt{N}$ to conclude the proof. •

Chapter 5

But time is continuous!

5.1 Introduction

Time is intrinsically continuous, but measurements are made with finite precision through observations within discrete time bins. The stochastic chain introduced in Chapter 3 was designed to model the sequence of discrete time observations produced by experimental measurements.

From a mathematical viewpoint there are many technical differences between models in discrete and those in continuous time. Discrete time models can be introduced in a more elementary way, while continuous time requires immediately more sophisticated mathematical tools.

Since before recording, data are continuous, it is experimentally legitimate to investigate the effect of time discretization on the conclusions drawn from experiments. For instance, the information given by the precise ordering between spike times of different neurons is important, but could be missed with time discretization. An example where this would be a problem is when we do the statistical analysis of spike trains with the goal to identify the graph of interactions between neurons. We will come back to this topic in Section **??** below.

Time discretization can also misguide the neurobiologists when they describe the qualitative behavior of the system. For instance spurious synchronisation behavior can be induced by too large discretization steps in deterministic models (see for instance (51)).

For all these reasons as well as others that will be presented in the sequel, it is clear that we need to develop continuous time models.

5.2 Basic continuous time model

In continuous time it is natural to model networks of spiking neurons as systems of interacting point processes (see also Section **??**). Informally speaking, a point process is an increasing sequence of times. In our case we are interested in the sequence of spiking times of each neuron in the system such that each neuron i is described by its own point process

$$\ldots < T_{-n}(i) < \ldots < T_0(i) \leq 0 < T_1(i) < T_2(i) < \ldots < T_n(i) < \ldots,$$

where the $T_n(i)$ are the successive spiking times of neuron i.

We may associate a counting process to this sequence of times by

$$N^i(]s,t]) = \sum_n 1_{\{s<T_n(i)\leq t\}} \tag{5.2.1}$$

for any $i \in I$ and for any $s \leq t$.

We may also adopt the viewpoint of network to describe the entire system. The network generates an event any time one of its neurons spikes. We write $T_n, n \in \mathbb{Z}$, for the sequence of successive spiking times of the system, where the successive times are again ordered such that

$$\ldots < T_{-n} < \ldots < T_0 \leq 0 < T_1 < T_2 < \ldots < T_n < \ldots.$$

Moreover we keep track of the index of the spiking neuron by introducing a sequence of random marks $I_n \in I, n \in \mathbb{Z}$, where I_n denotes the index of the neuron that is spiking at time T_n. For any $-\infty < s < t < \infty$, we can then introduce

$$N(]s,t]) = \sum_n 1_{\{s<T_n\leq t\}} \tag{5.2.2}$$

which is the total number of spikes during the interval $]s,t]$. We have

$$N(]s,t]) = \sum_{i\in I} N^i(]s,t]).$$

The sequence $(T_n, I_n)_{n\in\mathbb{Z}}$ is called a *marked point process* in the literature. $(I_n)_{n\in\mathbb{Z}}$ is the sequence of *marks* associated to the point process.

Both descriptions, that is considering either the family of point processes $(T_n^i)_{i\in I, n\in\mathbb{Z}}$ or considering the marked point process $(T_n, I_n)_{n\in\mathbb{Z}}$ are equivalent if simultaneous spikes of different neurons do not happen, that is, if $T_n^i \neq T_m^j$ for all $i \neq j, n, m$.

To define the process, we use the same ingredients as in the discrete time setting. But in continuous time, the functions ϕ_i are allowed to take any

positive value, that is, any $\phi_i : \mathbb{R} \to \mathbb{R}_+$ is a non-decreasing function, not necessarily bounded. The model will be defined in such a way that different neurons never spike at the same time.

Let us introduce for any $t \in \mathbb{R}$ the quantity

$$L_t(i) = \sup\{T_n(i) : T_n(i) \leq t\}$$

which is the last spiking time of neuron i before time t. We shall always work under conditions ensuring that this last spiking time exists. From a mathematical point of view this means that we are sure that there exists an infinity of spiking times for each neuron. From a computational point of view that requires that we initialise our model with a piece of past history in which all the neurons have spiked at least once. From the statistical point of view we can only start the analysis once each neuron has spiked at least once.

In our model, the membrane potential $V_t(i)$ of neuron i at time t is given by

$$V_t(i) = \sum_{j \in I} w_{j \to i} N^j(]L_t(i), t]).$$

Let us start with an informal description of our model. We start at some time t and a piece of past evolution long enough such that every neuron has spiked at least once before time t.

1. We compute $V_t(i)$ for every neuron i.

2. For any small time increment $\Delta > 0$ and given the past, neuron i spikes within $]t, t + \Delta]$, independently of the others, with probability

$$\begin{aligned} P(N^i(]t, t+\Delta]) = 1 \,|\, V_t(i)) &= \phi_i(V_t(i))\Delta + o(\Delta), \qquad (5.2.3) \\ P(N^i(]t, t+\Delta]) = 0 \,|\, V_t(i)) &= 1 - \phi_i(V_t(i))\Delta + o(\Delta), \\ P(N^i(]t, t+\Delta]) \geq 2 \,|\, V_t(i)) &= o(\Delta). \end{aligned}$$

In the above formulas (5.2.3), the expression $o(\Delta)$ means a quantity depending on Δ which decreases to 0 as Δ approaches 0 faster than Δ, that is, $\lim_{\Delta \to 0} \frac{o(\Delta)}{\Delta} = 0$.

Clarification for biologists. The above description is informal. The quantity Δ corresponds to the time resolution we are considering. The goal of the continuous time model is to be able to precisely observe in which order different neurons spike. This cannot be achieved if our observations

have a fixed time precision Δ. The length Δ should be a decreasing function of the number of neurons that we want to observe. Experimentalists refer to Δ as the sampling period, that is, the inverse of the sampling rate. By sampling rate we mean the rate of the clock of the digitization system used by the experimentalist to record the data. Experimentally we cannot increase as much as we would like the sampling rate implying that some fine time differences are unavoidably lost upon data acquisition. We discuss this issue further in the Appendix section **??**.

We also have to ensure that Δ is sufficiently small such that the quantity $\phi_i(V_t(i))\Delta$ is smaller than 1 which is a necessary condition for our formulas to have a probabilistic meaning. Therefore, Δ must be a decreasing function of both the number of neurons we observe and the maximal spiking rate of the observed population of neurons.

We now go a step forward in this description. Between two spike times of the system, each neuron has a fixed membrane potential value. Therefore, the waiting time until the next spike of neuron i is a geometrically distributed random variable with parameter $\phi_i(v(i))$. Obviously, this is only true until the next spiking time of the total system. At this spike time, all the neurons update the values of their membrane potentials, and we return to the previous situation. As a consequence we can describe the entire system in the following way.

For simplicity we assume that the set of neurons is given by $I = \{1, \ldots, N\}$, where N is an integer greater or equal to 2.

1. We assign a real number $v_0(i)$ (the value of its membrane potential) to each neuron i. Moreover we put $T_0 = 0$.

2. For $n = 1, \ldots, M$, do

   (a) For each neuron i, independently of the others, we choose a geometric random variable τ_n^i of parameter $\phi_i(v_{n-1}(i))\Delta$.

   (b) We put $\tau_n = \min \tau_n^i$ and $T_n = T_{n-1} + \tau_n$.

   (c) We put $\mathcal{I}_n = \{i : \tau_n^i = \tau_n\}$.

   (d) For all $i \in \mathcal{I}_n$, we put $v_n(i) = 0$. Moreover, for all $i \in \mathcal{I} \setminus \mathcal{I}_n$, we update
   $$v_n(i) = v_{n-1}(i) + \sum_{k \in \mathcal{I}_n} w_{k \to i}.$$

3. Print $(v_n, T_n), n = 1, \ldots M$.

We use the above procedure to define a continuous time process $(V_t)_{t\in\mathbb{R}_+}$ by putting $V_t = v_n$ for all $t \in [T_n, T_{n+1}[$, $n = 0, \ldots, M-1$. In the above description, simultaneous spikes are still possible. We now want to take Δ sufficiently small such that with overwhelming probability, $\mathcal{I}_n$ is a singleton for each $n = 1, \ldots, M$. Mathematically speaking this means that we have to work in the limit as Δ goes to 0 sufficiently fast as a function of the number of neurons N and of the maximal value of ϕ.

After passing to the limit $\Delta \to 0$, the time evolution of the entire system we want to define can be described as follows

1. There are no simultaneous spikes of different neurons.

2. For the entire system the times between two successive spikes are exponentially distributed random variables with parameter $\sum_{i\in I} \phi_i(v(i))$, where $v(i)$ is the value of the membrane potential of neuron i after the last spike. Moreover, the successive waiting times between the spikes of the system are independent.

3. Finally, given that a spike occurs, the probability that it is neuron i that is spiking is given by
$$\frac{\phi_i(v(i))}{\sum_{j\in I} \phi_j(v(j))}.$$

We resume the above description in the following pseudo-code.

1. We start with a piece of past evolution long enough such that every neuron has spiked at least once before time 0, including 0.

2. We compute $V_0(i)$ for every neuron i.

3. We choose an exponential time T_1 with parameter $\sum_{i\in I} \phi_i(V_0(i))$.

4. We choose $I_1 = i$ with probability $\frac{\phi_i(V_{T_1}(i))}{\sum_{j\in I} \phi_j(V_{T_1}(j))}$.

5. We update the values of $V_{T_1}(j)$ for every neuron j, namely we put $V_{T_1}(i) = 0$ and $V_{T_1}(j) = V_0(j) + w_{i\to j}$, for each $j \neq i$.

6. We iterate the previous steps, starting from step 2. at time T_n and with V_{T_n} already computed. This means that we choose an exponentially distributed waiting time $T_{n+1} - T_n$, independent of anything else, with parameter $\sum_{i\in I} \phi_i(V_{T_n}(i))$, and that we choose $I_{n+1} = i$ with probability $\frac{\phi_i(V_{T_n}(i))}{\sum_{j\in I} \phi_j(V_{T_n}(j))}$.

We can rewrite the above dynamic as follows. Recalling that $L_t(i)$ denotes the last spiking time of neuron i before time t, we have that

$$V_t(i) = \begin{cases} \sum_{j\in I} w_{j\to i} N^j(]L_t(i), t]), & \text{if } T_1^i \le t, \\ v_0(i) + \sum_{j\in I} w_{j\to i} N^j(]0, t], & \text{if } T_1^i > t. \end{cases}$$

The process $(V_t(i))_{i\in I}$ is a Markov jump process taking values in $\mathbb{R}^I$. Being a Markov process means that at any time $t > 0$, it is sufficient to know all values $V_t(i), i \in I$, to predict the future evolution of the process; we do not need to keep the memory of any of the values $V_s(i), s < t$. A Markov jump process is a Markov process that is piecewise constant and does only evolve through jumps, that is, it jumps from one configuration to another after exponentially distributed waiting times.

Markov processes are often represented by means of their associated generator, a concept that we are going to explain now. Suppose that we start at time $t = 0$ from initial potential values $v = (v(i), i \in I)$. If we take a bounded test function $f : \mathbb{R}^I \to \mathbb{R}$, we may calculate

$$Lf(v) = \lim_{t\to 0} \frac{E(f(V_t)) - f(v)}{t} = \lim_{t\to 0} \frac{E(f(V_t)) - f(V_0)}{t}.$$

We obtain

$$Lf(v) = \sum_{i\in I} \phi_i(v(i)) \left[f(v + \Delta_i(v)) - f(v) \right], \tag{5.2.4}$$

where

$$(\Delta_i(v))_j = \left\{ \begin{array}{ll} w_{i\to j} & j \neq i \\ -v(i) & j = i \end{array} \right\}. \tag{5.2.5}$$

The application that maps f to Lf is called the *generator* of the process. Equation (5.2.4) means the following. Whenever the process V is in configuration $x \in \mathbb{R}^I$, then it stays in this configuration up to the next jump time. This jump time is exponentially distributed with parameter $\bar{\phi}(v) := \sum_{i\in I} \phi_i(v(i))$. At the next jump, we choose the index of the jumping particle i with probability $\phi_i(v(i))/\bar{\phi}(v)$ and replace the former configuration v by $\Delta_i(v)$.

5.2.1 Adding leakage in continuous time

Up to now we have modeled the membrane potential as a *piecewise constant process*, only jumping when it receives presynaptic inputs or when it is reset due to the neuron's own spiking activity. This is a simplification that sets aside the leakage of actual biological membranes. The latter, made

of lipid bilayers, are far from being perfect electrical insulators and a membrane voltage (V) deviating from the equilibrium value (0 in our setting) will generate a trans-membrane current that will eventually bring the potential back to the equilibrium value–this is what *leakage* means in that context. The actual membrane current is proportional to the voltage deviation, leading to a differential equation of the form: $dV/dt = -V/\tau$. An elementary modification of our basic model, making it more in line with experimental observations, consists therefore in adding such a leakage effect.

In our model without leakage, we suppose that each neuron has its own leakage time constant $\tau_i > 0$. In other words, if the potential of neuron i equals $v(i)$ at time 0, then up to the next spiking time of the system, its value decreases according to $V_t(i) = e^{-t/\tau_i} v(i)$. As before, the system is described by a sequence $(T_n, I_n)_{n\geq 1}$, where T_n is the $n-$th spiking time of the whole system after time 0, and where $I_n = i \in I$ if it is neuron i that spikes at this time. When adding leakage, we have to change the way we choose the successive waiting times up to the next spike. This is done through an acceptance/rejection algorithm in which some of the proposed spiking times will be rejected. This is a consequence of the fact that the parameter $\sum_{i\in I} \phi_i(V_t(i))$ does not remain constant between successive spiking times but decreases due to the leakage effect, see Figure **??**.

Rejected spiking times will be labelled with †. The random variables $I_n, n \geq 1$, now take values in $I \cup \{\dagger\}$.

1. The initial values $V_0(i) \in \mathbb{R}$ are given for any $i \in I$.

2. We compute for every neuron i the maximal spiking rate λ_i given by

$$\lambda_i = \begin{cases} \phi_i(V_0(i)) & \text{if } V_0(i) \geq 0, \\ \phi_i(0)) & \text{if } V_0(i) < 0. \end{cases}$$

3. We choose an exponential time S_1 with parameter $\bar{\lambda} = \sum_{i\in I} \lambda_i$.

4. We choose

$$K_1 = \begin{cases} i & \text{with probability } \phi_i(e^{-S_1/\tau_i} V_0(i))/\bar{\lambda}, \text{ for each } i \in I, \\ \dagger & \text{with probability } 1 - \sum_{i\in I} \phi_i(e^{-S_1/\tau_i} V_0(i))/\bar{\lambda}. \end{cases}$$

5. If $K_1 = i \in I$, we put $T_1 = S_1$, $I_1 = i$, and we update the values of the membrane potentials as follows.

$$V_{T_1}(i) = 0,$$
$$V_{T_1}(j) = e^{-S_1/\tau_j} V_0(j) + w_{i\to j}, \text{ for each } j \neq i.$$

6. If $K_1 = \dagger$, we update $V_{S_1}(j) = e^{-S_1/\tau_j} V_0(j)$ for all $j \in I$.

7. In both cases, we iterate the previous steps, starting from step 2. at time T_n and with V_{T_n} already computed.

The above algorithm allows to define a continuous time process $(V_t)_{t\in\mathbb{R}_+} = ((V_t(i))_{t\in\mathbb{R}_+}, i \in I)$ taking values in $\mathbb{R}^I$ through

$$V_t(i) = \sum_{n=0}^{\infty} 1_{\{T_n \le t < T_{n+1}\}} e^{-(t-T_n)/\tau_i} V_{T_n}(i).$$

Having in mind the above algorithm, we can also rewrite the value of the membrane potential of neuron i at time t as

$$V_t(i) = \sum_{j\in I, j\neq i} \sum_{k: L_t(i) < T_k(j) \le t} e^{-(t-T_k(j))/\tau_i}. \tag{5.2.6}$$

This formula holds true if neuron i has already spiked before time t, otherwise, we have to replace $L_t(i)$ by time 0.

A process evolving as in (5.2.6) is called *Piecewise deterministic Markov process* (PDMP). PDMPs evolve in a deterministic manner in between successive jumps. Jumps arrive after random waiting times, which are extended exponential random variables. These jumps might have random amplitude as well, but this is not the case in our model. The reader can find more details about such processes in the Appendix Section **??** .

5.2.2 Python implementation

A simple implementation of the above algorithm in Python is given below.
BOX PYTHON

5.3 Complements and exercices.

We will prove in the Appendix Section **??** the following theorem.

Theorem 4. *The process $(V_t)_t = (V_t(i), i \in I)_t$ given by (5.2.6) is a piecewise deterministic Markov process having generator which is given for any bounded test function $f \in C^1(\mathbb{R}^I)$ by*

$$Lf(v) = \sum_{i\in I} \phi_i(v(i)) \left[f(v + \Delta_i(v)) - f(v)\right] - \sum_{i\in I} \alpha_i \frac{\partial f}{\partial v(i)}(v) v(i), \tag{5.3.7}$$

where the jump term $\Delta_i(v)$ is given in (5.2.5) above.

Neurobiologists do not need to know how to prove this theorem. But it is useful for them to learn how to interpret the formula of the infinitesimal generator. It is a sum of two terms, corresponding to the two different aspects of the process' time evolution. The first sum describes how the process jumps together with the infinitesimal probabilities that these jumps occur within very small time intervals, see equation (5.2.3) above. The second sum describes the leakage effect that each neuron undergoes continuously in time.

Exercice 1: Simultaneous spikes do not occur

Show that (5.2.3) implies that

$$P(N^i(]t,t+\Delta]) = 1, N^j(]t,t+\Delta]) = 1 | V_t(i), V_t(j)) = o(\Delta).$$

Exercice 2

In this exercice we show how passing to the limit $\Delta \to 0$ in (5.2.3) naturally leads to the introduction of exponentially distributed inter-spike intervals.

Suppose we know $V_0(i)$ for every neuron i. Write $\mathcal{F}_k := \sigma\{V_t, 0 \le t \le k\delta\}$ for the sigma field that is generated by the membrane potential values up to time $k\delta$. Then (5.2.3) implies that (ignoring the terms of order $o(\Delta)$)

$$\begin{aligned} P(T_1 > n\Delta) &= E\left(1_{\{T_1>(n-1)\Delta\}} P(N^i(](n-1)\Delta, n\Delta]) = 0,\ \forall\, i \in I | \mathcal{F}_{(n-1)\Delta)}\right) \\ &= E\left(1_{\{T_1>(n-1)\Delta\}} \prod_{i\in I} \left(1 - \phi_i(V_{(n-1)\Delta}(i))\Delta\right)\right). \end{aligned}$$

1. Show that this gives

$$P(T_1 > n\Delta) = E\left(1_{\{T_1>(n-1)\Delta\}} \prod_{i\in I} (1 - \phi_i(V_0(i))\Delta)\right).$$

2. Prove that iterating this formula yields

$$P(T_1 > n\Delta) = \prod_{i\in I} (1 - \phi_i(V_0(i)\Delta)^n.$$

3. Pass to logarithm and use that $\log(1-x) \sim -x$ as $x \to 0$, to obtain

$$P(T_1 > n\Delta) = e^{\sum_{i\in I} n \log(1-\phi_i(V_0(i)))} \sim e^{-n\Delta \sum_{i\in I} \phi_i(V_0(i))},$$

as $\Delta \to 0$. Thus, T_1 is exponentially distributed with parameter $\sum_{i\in I} \phi_i(V_0(i))$.

4. Compute $P(I_1 = i | T_1 = n\Delta)$.

Exercice 3

Suppose that all rate functions ϕ_i are bounded. Prove equation (5.2.4) in this case.

5.4 Discussion and bibliographic comments

The quantity $t - L_t(i)$ measures the *time elapsed since the last spike* of neuron i, it is also sometimes called the *age* of the process, see (**?**). We have $t = L_t(i)$ if and only if i has a spike at that time, and by definition, its membrane potential equals 0 at this time.

Bibliography

[1] André, M. A result of metastability for an infinite system of spiking neurons. *J Stat Phys 117* (2019), 984–1008.

[2] André, M., and Planche, L. The effect of graph connectivity on metastability in a stochastic system of spiking neurons. *Stochastic Processes and their Applications 131* (2021), 292 – 310.

[3] Bak, P. *How nature works*. Springer New York, 1996.

[4] Baladron, J., Fasoli, D., Faugeras, O., and Touboul, J. Mean-field description and propagation of chaos in networks of Hodgkin-Huxley and Fitzhugh-Nagumo neurons.

[5] Beggs, J., and Plenz, D. Neuronal avalanches in neocortical circuits. *J. Neurosc 23* (2003), 11167–11177.

[6] Bollobás, B. *Random graphs.* Cambridge Studies in Advanced Mathematics. 73. Cambridge: Cambridge University Press., 2001.

[7] Bollobás, B., Janson, S., and Riordan, O. The phase transition in inhomogeneous random graphs. *Random Structures Algorithms 31*, 1 (2007), 3–122.

[8] Braitenberg, V., and Schüz, A. *Cortex: Statistics and Geometry of Neuronal Connectivity*, 2nd ed. Springer, 1998.

[9] Brémaud, P., and Massoulié, L. Stability of nonlinear Hawkes processes. *Ann. Probab. 24*, 3 (1996), 1563–1588.

[10] Bressloff, P. C. Stochastic neural field theory and the system-size expansion. *SIAM J. Appl. Math. 70*, 5 (2009), 1488–1521.

[11] Brillinger, D. The identification of point process systems. *Ann. Probab. 3*, 6 (12 1975), 909–924.

[12] Brillinger, D. Maximum likelihood analysis of spike trains of interacting nerve cells. *Biol. Cybern. 59*, 3 (1988), 189–200.

[13] Brillinger, D., Bryant, H., and Segundo, J. Identification of synaptic interactions. *Biol. Cybern. 22* (1976), 213–228.

[14] Brochini, L., Costa, A., Abadi, M., Roque, A., Stolfi, J., and Kinouchi, O. Phase transitions and self-organized criticality in networks of stochastic spiking neurons. *Scientific Reports 6: 35831* (2016).

[15] Bullmore, E., and Bassett, D. Brain graphs: Graphical models of the human brain connectome. In *Annual Review of Clinical Psychology*, S. NolenHoeksema, T. Cannon, and T. Widiger, Eds., vol. 7. Annual Reviews, 2011, pp. 113–140.

[16] Burkitt, A. N. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics 95*, 1 (Apr 2006), 1–19.

[17] Castelfranco, A. M., and Hartline, D. K. Evolution of rapid nerve conduction. *Brain Research 1641* (2016), 11 – 33.

[18] Cessac, B. A discrete time neural network model with spiking neurons: II: Dynamics with noise. *Journal of Mathematical Biology 62* (2011), 863–900.

[19] Cessac, B. Statistics of spike trains in conductance-based neural networks: Rigorous results. *J. Math. Neurosci. 1*, 8 (2011), 1–42.

[20] Chevallier, J., Cacéres, M. J., Doumic, M., and Reynaud-Bouret, P. Microscopic approach of a time elapsed neural model. *Mathematical Models and Methods in Applied Sciences 25*, 14 (2015), 2669–2719.

[21] Chornoboy, E. S., Schramm, L. P., and Karr, A. F. Maximum likelihood identification of neural point process systems. *Biological Cybernetics 59*, 4-5 (Sep 1988), 265–275.

[22] Cohen, L., Celnik, P., Pascual-Leone, A., Corwell, B., Faiz, L., Dambrosia, J., Honda, M., Sadato, N., Gerloff, C., Catalá, M., and Hallett, M. Functional relevance of cross-modal plasticity in blind humans. *Nature 389*, 6647 (1997), 180–183.

[23] Cooley, J. W., and Dodge, F. A. Digital computer solutions for excitation and propagation of the nerve impulse. *Biophys J 6*, 5 (Sep 1966), 583–599.

[24] Cormier, Q., Tanré, E., and Veltz, R. Long time behavior of a mean-field model of interacting neurons. *arXiv:1810.08562 [math]* (May 2019).

[25] Cox, D. R., and Lewis, P. A. W. *The Statistical Analysis of Series of Events*. 1966.

[26] Cuesta-Albertos, J., del Barrio, E., Fraiman, R., and Matran, C. The random projected method in goodness of fit for functional data. *Comp. Stat. and Data Analy. 51* (2007), 4814–4831.

[27] Davis, M. Piecewise-deterministic Markov processes: A general class of non- diffusion stochastic models. *J. R. Stat. Soc., Ser. B 46* (1984), 353–388.

[28] De Masi, A., Galves, A., Löcherbach, E., and Presutti, E. Hydrodynamic limit for interacting neurons. *J. Stat. Physics 0-3* (2015).

[29] Deco, G., and Jirsa, V. K. Ongoing cortical activity at rest: Criticality, multistability, and ghost attractors. *Journal of Neuroscience 32*, 10 (2012), 3366–3375.

[30] Delarue, F., Inglis, J., Rubenthaler, S., and Tanré, E. Global solvability of a networked integrate-and-fire model of McKean-Vlasov type. *Ann. Appl. Probab 25* (2015), 2096–2133.

[31] Ditlevsen, S., and Löcherbach, E. Multi-class Oscillating Systems of Interacting Neurons. *Stochastic Processes and their Applications 127*, 6 (June 2017), 1840–1869.

[32] Duarte, A., Galves, A., Fraiman, R., Ost, G., and Vargas, C. D. Context tree retrieval for eeg data. *Mathematics 7* (2019), 427.

[33] Duarte, A., Galves, A., Löcherbach, E., and Ost, G. Estimating the interaction graph of stochastic neural dynamics. *Bernoulli 25*, 1 (2019), 771–792.

[34] Duarte, A., and Ost, G. A model for neural activity in the absence of external stimuli. *Markov Process. Related. Fields 22* (2016), 37–52.

[35] Duarte, A., Ost, G., and Rodríguez, A. A. Hydrodynamic Limit for Spatially Structured Interacting Neurons. *Journal of Statistical Physics 161*, 5 (Dec. 2015), 1163–1202.

[36] Eguiluz, V., Chialvo, D., Cecchi, G., Baliki, M., and Apkarian, V. Scale-free brain functional networks. *Phys. Rev. Lett. 94* (2005), 018102.

[37] Ethier, S., and Kurtz, T. *Markov Processes. Characterization and Convergence*. Wiley Series In Probability And Statistics, 2005.

[38] Faugeras, O., Touboul, J., and Cessac, B. A constructive mean-field analysis of multi-population neural networks with random synaptic weights and stochastic inputs. *Frontiers in Comp. Neuroscience 3* (2009), 1–28.

[39] Fernández, R., Ferrari, P. A., and Galves, A. Coupling, renewal and perfect simulation of chains of infinite order, Notes for a minicourse given at the Vth Brazilian School of Probability. *Manuscript http://www.univ-rouen.fr/LMRS/Persopage/Fernandez/resucoup.html* (2001).

[40] Fournier, N., and Löcherbach, E. On a toy model of interacting neurons. *Ann. Inst. H. Poincaré Probab. Statist. 52*, 4 (11 2016), 1844–1876.

[41] Galves, A., and Löcherbach, E. *Stochastic chains with memory of variable length.* TICSP Series vol. 38, 2008, pp. 117–133.

[42] Galves, A., and Löcherbach, E. Infinite systems of interacting chains with memory of variable length - a stochastic model for biological neural nets. *J. Stat. Phys. 151* (2013), 896–921.

[43] Galves, A., and Löcherbach, E. Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *Journal of Statistical Physics 151*, 5 (2013), 896–921.

[44] Galves, A., and Löcherbach, E. Modeling networks of spiking neurons as interacting processes with memory of variable length. *Journal de la Société Française de Statistiques 157* (2016), 17–32.

[45] Galves, A., Löcherbach, E., Pouzat, C., and Presutti, E. A system of interacting neurons with short term synaptic facilitation. *J Stat Phys 178* (2020), 869–892.

[46] Gerstein, G. L., and Kiang, N. Y.-S. An approach to the quantitative analysis of electrophysiological data from single neurons. *Biophys J. 1*, 1 (1960), 15–28.

[47] Gerstein, G. L., and Mandelbrot, B. Random walk models for the spike activity of a single neuron. *Biophysical Journal 4*, 1 (Jan 1964), 41–68.

[48] Goldberg, J. M., Adrian, H. O., and Smith, F. D. Response of neurons of the superior olivary complex of the cat to acoustic stimuli of long duration. *J. Neurophysiol.* (1964), 706–749.

[49] Greenwood, P. E., and Ward, L. M. *Single Neuron Models*. Springer International Publishing, 2016.

[50] Hagiwara, S. Analysis of interval fluctuation of the sensory nerve impulse. *The Japanese Journal of Physiology 4* (1954), 234–240.

[51] Hansel, D., Mato, G., Meunier, C., and Neltner, L. On numerical simulations of integrate-and-fire neural networks. *Neural Computation 10*, 2 (1998), 467–483.

[52] Hawkes, A. G. Point spectra of some mutually exciting point processes. *J. R. Stat. Soc., Ser. B 33* (1971), 438–443.

[53] Hawkes, A. G. Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika 58*, 1 (1971), 83–90.

[54] Hodara, P., and Löcherbach, E. Hawkes processes with variable length memory and an infinite number of components. *Advances in Applied Probability 49*, 1 (2017), 84–107.

[55] Hodgkin, A. L., and Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology 117*, 4 (1952), 500–544.

[56] Höpfner, R., Löcherbach, E., and Thieullen, M. Ergodicity for a stochastic hodgkin-huxley model driven by ornstein-uhlenbeck type input. *Annales de l'IHP 52*, 1 (2016), 483–501.

[57] Höpfner, R., Löcherbach, E., and Thieullen, M. Strongly degenerate time inhomogeneous SDEs: densities and support properties. Application to a Hodgkin-Huxley system with periodic input. *Bernoulli 23* (2017), 2587–2616.

[58] Huntsman, M., Porcello, D., Homanics, G., DeLorey, T., and Huguenard, J. Reciprocal inhibitory connections and network synchrony in the mammalian thalamus. *Science 283* (1999), 541–543.

[59] Izhikevich, E. *Dynamical systems in neuroscience: The geometry of excitability and bursting.* MIT Press, 2009.

[60] Jahn, P., Berg, R. W., Hounsgaard, J., and Ditlevsen, S. Motoneuron membrane potentials follow a time inhomogeneous jump diffusion process. *Journal of Computational Neuroscience 31* (2011).

[61] Janert, P. K. *Data Analysis with Open Source Tools.* O'REILLY, 2011.

[62] Janert, P. K. *Gnuplot in Action. Understanding data with graphs.*, 2nd ed. Manning Publications Co., 2016.

[63] Kilpatrick, Z. P. *Wilson-Cowan model, In: Encyclopedia of Computational Neuroscience.* Springer New York, 2015, pp. 3159–3163.

[64] Kinouchi, O., and Copelli, M. Optimal dynamical range of excitable networks at criticality. *Nature physics 2* (2006), 348–351.

[65] Klemens, B. *21st Century C*, 2nd ed. O'REILLY, 2016.

[66] Koch, C., Poggio, T., and Torre, V. Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing. *Proceedings of the National Academy of Sciences 80*, 9 (1983), 2799–2802.

[67] Krumin, M., Reutsky, I., and Shoham, S. Correlation-based analysis and generation of multiple spike trains using Hawkes models with an exogenous input. *Front Comput Neurosci. 4* (2010), 147.

[68] Lapicque, L. Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *J. Physiol. Pathol. Gen.* (1907), 620–635.

[69] Liggett, T. *Interacting Particle Systems.* Springer Berlin Heidelberg, 1985.

[70] Löcherbach, E., and Monmarché, P. Metastability for a system of interacting neurons. *arXiv 6* (2020), 325–356.

[71] Luo, L. *Principles of Neurobiology*, 1st ed. Garland Science, 2016.

[72] McCulloch, W. S., and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5*, 4 (1943), 115–133.

[73] Nawrot, M. P., Boucsein, C., Rodriguez-Molina, V., Aertsen, A., Grün, S., and Rotter, S. Serial interval statistics of spontaneous activity in cortical neurons in vivo and in vitro. *Neurocomputing 70* (2007), 1717–1722.

[74] Nicolelis, M., and Ribeiro, S. Seeking the neural code. *Sci. Am. 295*, 6 (2006), 70–77.

[75] Pakdaman, K., Thieullen, M., and Wainrib, G. Fluid limit theorems for stochastic hybrid systems with application to neuron models. *Advancements in Applied Probability 42*, 3 (2010), 761–794.

[76] Perge, J. A., Niven, J. E., Mugnaini, E., Balasubramanian, V., and Sterling, P. Why do axons differ in caliber? *Journal of Neuroscience 32*, 2 (2012), 626–638.

[77] Perkel, D. H., Gerstein, G. L., and Moore, G. P. Neuronal spike trains and stochastic point processes. I the single spike train. *Biophys. J. 7* (1967), 391–418.

[78] Phi, T., Muzy, A., and Reynaud-Bouret, P. Event-scheduling algorithms with kalikow decomposition for simulating potentially infinite neuronal networks. *SN COMPUT. SCI. 35* (2020).

[79] Potjans, T. C., and Diesmann, M. The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Model. *Cerebral Cortex 24*, 3 (12 2012), 785–806.

[80] Potjans, T. C., and Diesmann, M. The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Model. *Cerebral Cortex 24*, 3 (12 2012), 785–806.

[81] Potjans, T. C., and Diesmann, M. The cell-type specific cortical microcircuit: Relating structure and activity in a full-scale spiking network model. *Cereb Cortex 24* (2014), 785–806.

[82] Rall, W. *Handbook of physiology, cellular biology of neurons*. American Physiological Society, Bethesda, MD, 1977, ch. Core conductor theory and cable properties of neurons., pp. 39–97.

[83] Reynaud-Bouret, P., Rivoirard, V., Grammont, F., and Tuleau-Malot, C. Goodness-of-fit tests and nonparametric adaptive estimation for spike train analysis. *The Journal of Mathematical Neuroscience 4*, 1 (2014), 3.

[84] Riedler, M. G., Thieullen, M., and Wainrib, G. Limit theorems for infinite-dimensional piecewise deterministic markov processes. Applications to stochastic excitable membrane models, 2012.

[85] Rissanen, J. A universal data compression system. *IEEE Trans. Inform. Theory 29*, 5 (1983), 656–664.

[86] Robert, P., and Touboul, J. On the dynamics of random neuronal networks. *J. Stat. Phys 165* (2016), 545–584.

[87] Sacerdote, L., and Giraudo, M. T. *Stochastic Integrate and Fire Models: A Review on Mathematical Methods and Their Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 99–148.

[88] Schulte, E., Davison, D., Dye, T., and Dominik, C. A multi-language computing environment for literate programming and reproducible research. *Journal of Statistical Software 46*, 3 (2012), 1 – 24.

[89] Shimoura, R. O., Kamij, N. L., Pena, R. F., Cordeiro, V. L., Ceballos, C. C., Romaro, C., and Roque, A. C. [Re] The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *ReScience 4*, 1 (May 2018), #2.

[90] Spitzer, F. Interaction of Markov Processes. *Adv. Math. 5*, 2 (1970), 246–290.

[91] Sporns, O. The non-random brain: Efficiency, economy, and complex dynamics. *Frontiers in Computational Neuroscience 5* (2011), 5.

[92] Tass, P. Phase and frequency shifts in a population of phase oscillators. *Phys. Rev. E 56* (Aug 1997), 2043–2060.

[93] Tuckwell, H. C. *Introduction to Theoretical Neurobiology. Volume 1: Linear cable theory and dendrite structure*. Cambridge University Press, 1988.

[94] Tuckwell, H. C. *Introduction to Theoretical Neurobiology. Volume 2: Nonlinear and stochastic theories.* Cambridge University Press, 1988.

[95] van Vreeswijk, C., and Sompolinsky, H. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science 274* (1996), 1724–1726.

[96] Vargas, C., Aballea, A., Rodrigues, E., Reilly, K., Mercier, C., Petruzzo, P., Dubernard, J., and Sirigu, A. Re-emergence of hand-muscle representations in human motor cortex after hand allograft. *Proc. Natl. Acad. Sci. USA 106*, 17 (2009), 7197–7202.

[97] Veltz, R., and Faugeras, O. Local/global analysis of the stationary solutions of some neural field equations. *SIAM Journal on Applied Dynamical Systems 9*, 3 (2010), 954–998.

[98] Verveen, A., and Derksen, H. Fluctuation phenomena in nerve membrane. *Proceedings of the IEEE 56*, 6 (1968), 906–916.

[99] Wainrib, G., Thieullen, M., and Pakdaman, K. Reduction of stochastic conductance-based neuron models with time-scales separation. *Journal of Computational Neuroscience 32*, 2 (Aug. 2011), 327–346.

[100] Wilson, H. R., and Cowan, J. D. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal 12*, 1 (1972), 1 – 24.

[101] Wilson, H. R., and Cowan, J. D. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik 13*, 2 (1973), 55–80.

[102] Yaginuma, K. A stochastic system with infinite interacting components to model the time evolution of the membrane potentials of a population of neurons. *Journal of Statistical Physics 163*, 3 (2016), 642–658.

[103] Yarom, Y., and Hounsgaard, J. Voltage fluctuations in neurons: Signal or noise? *Physiological Reviews 91*, 3 (Jul 2011), 917–929.