Robustness of Recurrent Spiking Networks

Pierre-Étienne Helley Fiquet

A Master Thesis
Presented to the Faculty
of École Normale Supérieure
Département d'Études Cognitives

Advisers:
Sophie Denève
Christian Machens

June 2016

Declarations

Acknowledgements
I would like to thank Sophie and Christian who have been great mentors. It was a privilege to spend time with them and their teams in the Group for Neural Theory in Paris and at the Champalimaud Neuroscience Programme in Lisbon. Teamwork made the daily work delightful. Special thanks to Allan, Gonçalo and Nuno.

Originality
Instantiating the framework of recurrent spiking network has been an opportunity to derive and explain the equations again. By applying them to a new system, the mouse ALM, we obtained positive results that other state of the art models could not reach. On top of demonstrating the expressivity of this type of networks, we expanded the reflection on their robustness and developed a bilateral architecture of excitatory and inhibitory cells.

Contribution
Sophie Denève and Christian Machens defined the scientific question as well as the approach (efficient coding in recurrent spiking networks). Nuno Calaim started probing the robustness of these networks on the fish oculomotor integrator. Pierre-Étienne Fiquet implemented the simulations, suggested the specific dynamical system and network architecture, interpreted the results, generated the figures and wrote the report.

Number of words in the document : 14044

Contents

Chapter 1

Introduction

1.1 Background

A crucial step toward understanding the neurobiological basis of information processing is to bridge the microscopic level of single neurons with the macroscopic level of cognition. To do so, systems neuroscience builds on the convergence of molecular and optical tools [1] with transgenic mouse lines and quantitative behavioral tasks [2]. To establish causal relationships, it is necessary to perturb and manipulate neural circuits in order to reveal their structure and function. Observation only does not suffice to understand the link between patterns of action potentials and animal behavior. This reverse engineering approach searches for the underlying principles of neural design [3] that give rise to the brain as we know it. Importantly, these principles are shaped by metabolic constraints

The team of Karel Svoboda on the Janelia Research Campus successfully developed this method for the study of mouse motor cortices. In a series of studies [4] [5], they have been probing the mouse Anterior Lateral Motor (ALM) cortex, a region that is known to play a role in sensory guided movements. Doing so, they uncovered an intriguing robustness of neuronal dynamics during planning [6]. We will start by introducing these experiments, and then we will motivate a theoretical approach that can suggest underlying mechanisms for the results.

1.2 Experiment

1.2.1 Behavioral task

Data was acquired during a whisker-based object location discrimination task that is composed first of sampling epoch (1.3 seconds), then of a delay epoch (1.3 seconds), and finally of a response epoch that starts at an auditory go cue. During sampling, head-fixed animals are presented with a pole that can appear either in a posterior or an anterior position. Depending on the stimulus location, animals are expected to lick right or left. Although quite advanced for mice, this perceptual decision making task is learnt in less than a week and animals reach a performance plateau of 80%.

1.2.2 Neuronal recordings

The gold standard of neural recordings in animal performing advanced task has been electro-physiology in the head fixed behaving monkey. One example of this is the data acquired by Ranulfo Romo in monkey performing a flutter discrimination task [7]. By recordings several brain regions while the animal performed the same task, results could be compared. Similarly, in the set of experiments we are concerned with, the mouse cortex has been consistently investigated while the animal performed the previously described task. In these head-fixed mouse preparations, experimentalists could take advantage from the genetically targeted sensors and switches to measure and manipulate activity *in vivo*.

Neurons recorded in ALM have a very diverse selectivity. Some respond in advance of movements to the contralateral side and others respond in advance of movements to the ipsilateral side. In that sense, ALM is analogous to pre-motor cortex in the primate pre-frontal cortex.

Interestingly, these recordings showed that a small, sparse subpopulation of pyramidal cells seem to code for object location. These are highly informative neurons that spike overall more than the others. Such cells are already present in a naive mouse, *i.e.*one that has not yet learnt the task.

1.2.3 Optogenetic manipulations

Mouse cre-lines allow detailed circuit analysis by providing genetic functional access to specific cells. The manipulation we are most interested is the unilateral photostimulation of channelrhodopsin-2 in GABAergic interneurons, a perturbation that indirectly inhibits pyramidal neurons. Laser light was shone during the delay epoch and induced a systematic behavioral effect. Performance was not randomly decreased, a distinct ipsi-lateral bias in response. At first, it is not clear why disruption of ALM on one side particularly affects contra-lateral movements. This result is surprising because neurons selective for each side are present in about equal proportions in both hemispheres.

Most importantly, it was observed that if the perturbation occurs at the beginning of the delay epoch, then performance is preserved and neuronal activity rapidly catches up with the baseline trajectory. This robustness was not observed when both hemispheres where disrupted. An additional set of experiment with corpus callosum cut established that the contribution of the contra-lateral hemisphere is required for this robustness to occur. This is depicted in figure 1.1.

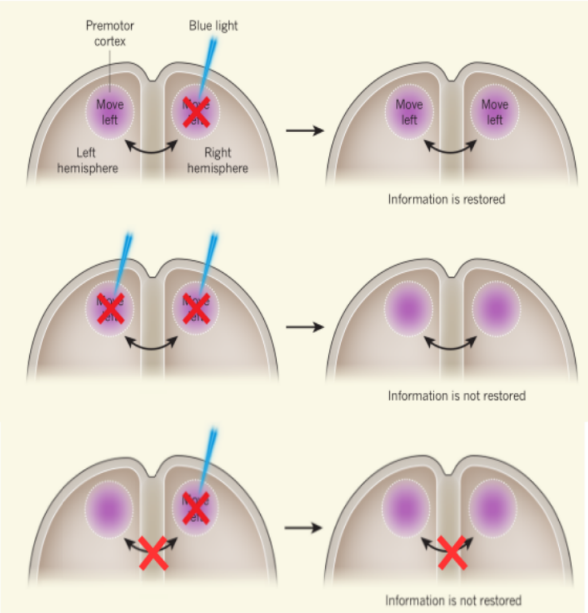


Figure 1.1: Robustness relies on inter-hemispheric cooperation. Top panel : after unilateral inactivation of ALM, information about licking direction is restored. Middle panel : after bilateral inactivation of ALM, information about licking direction is not restored. Bottom panel : if the corpus callosum is cut, unilateral inactivation suffices to block the restoration of information. (Figure adapted from the commentary of Byron M. Yu that accompanied the publication of the results [8]).

1.3 Theory

In order to provide analytical explanations for these experimental findings. We will use a model that is as simple as possible, yet i) interesting, ii) understandable and iii) plausible. Adopting a top-down approach, we will start with optimality principles and derive the corresponding optimal network connectivity. If the model reproduces experimental features, it will provide both a mechanistic explanations and testable predictions. Here we will focus on circuit wiring patterns, but other components such as synaptic plasticity and dendritic computations are also of great interest.

Although the world, our actions and experiences are essentially continuous, neuronal populations essentially communicate via discontinuous spiking. Acknowledging this fact, we will use recurrent spiking networks. More specifically we will present an instantiation of the model developed by Sophie Denve, Christian Machens and colleagues [9]. This work builds upon the theory of both efficient coding [10] and

balanced networks [11]. It is derived from an efficient coding principle : networks should reach the most metabolically efficient representation of information.

Extending previous work [12], we will investigate the effects of *in silico* equivalents for optogenetic inactivation.Our objective is to study the robustness of neuronal networks while they compute and thereby to better understand cortical circuits. As discussed in [6], state of the art models failed to reproduce the experimental findings previously described. Therefore, this master thesis is an occasion to probe the validity of the framework.

All the code used to simulate the network and to generate the plots presented here is available in a companion IPython Notebook document.

Chapter 2

Methods

In this chapter, we describe the recurrent spiking network. The first paragraph of each subsection provides a verbal description of the underlying ideas and it is accompanied by a figure to visualize the geometrical interpretation of the model. Mathematical details can be skipped on a first reading.

2.1 Derivation of the optimal network : coding

In this section, we largely follow the derivation of [13] and [14], but we expand the explanations.

2.1.1 Auto-encoder network

The intuition behind the network of spiking neurons that we use comes from a simple case. Imagine a scenario in which a neural network receives an input signal and has to produce an output signal as close as possible to the input. Note that in the machine learning literature, networks that learn to represent their input are called auto-encoders. Their simple architecture can be used as a building block to construct networks carrying other operations (for example : tracking some dynamics, or holding short-term memory).

Because our modeled neurons communicate only through discrete events called Action Potentials (or spikes), the performance on this task will be imperfect. Indeed a smooth input signal will be approximated by a "jumpy" process. Moreover, the effect of each discrete spike depends on the strength of the connection from the emitter to the receiver neuron. This synaptic strength is represented by the weight of the edge between each pair of units. Weights are values that can be positive, negative or null corresponding respectively to excitatory connections, inhibitory connections and the absence of connection.

Following on the objective described in the introduction, we want to build a functional network that emits as few spikes as possible. To do so we have to determine which is the optimal connectivity, which will be explained step by step in the next

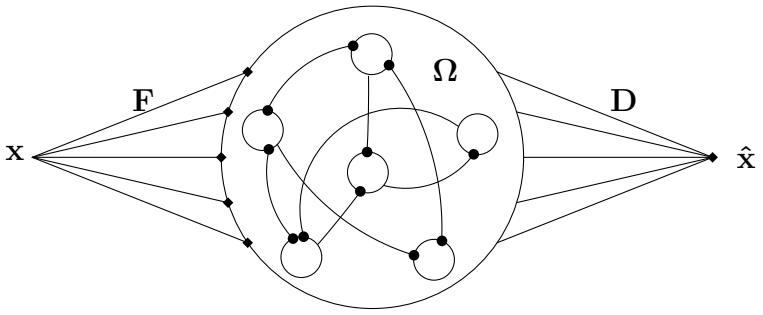


Figure 2.1: A recurrent neural network receiving an input $\mathbf{x}$ through the feed-forward weights $\mathbf{F}$. Units are coupled to one another by recurrent connections $\mathbf{\Omega}$ (note that we will add a second set of recurrent connections in 2.2). The output $\hat{\mathbf{x}}$ is readout from the network activity through the decoding weights $\mathbf{D}$.

sections. Then the spiking rule will be designed to let the network optimize over spike times. The architecture of this recurrent network is depicted in 2.1.

2.1.2 Inputs and outputs

The recurrent network receives task-specific inputs and produces an approximation of a specified target output. Inputs simulate the flow of information from sensory regions and outputs are spikes transmitted to a downstream circuit where they will elicit currents. The integration of the post-synaptic currents decays over time, lending a large weight to the immediate time after the spike and sharply decreasing afterwards. We will model this by adding an exponentially decaying current in the post-synaptic neuron following each pre-synaptic spike. This characteristic shape, represented in the next figure, is an approximation of the biophysical mechanisms occurring on the post-synaptic membrane and their time constant.

These post-synaptic currents can also be interpreted as instantaneous firing rates. Usually rates are computed by sliding a Gaussian window through the spike train. Using an exponential kernel instead has the advantage of increasing the rate only after each spike, thereby enforcing a strict causality of events. This operation, called a convolution, results in a smoothed, or filtered, spike train.

The network receives a set of time-varying inputs $\mathbf{c}(t) = (c_1(t), ..., c_j(t), ..., c_J(t))$, where J is the dimension of the input signal and $c_j(t)$ is the j^{th} input. It then produces spike $\mathbf{o}(t) = (o_1(t), ..., o_n(t), ..., o_N(t))$, where N is the dimension of the input signal. The spike train of the n^{th} neuron is given by $o_n(t) = \sum_l \delta(t - t_l^n)$, where $\{t_l^n\}$ are the spike times of that neuron. Throughout this document we will refer to these two vector spaces respectively as the signal space ($\mathbb{R}^J$) and the activity space ($\mathbb{R}^N$). In

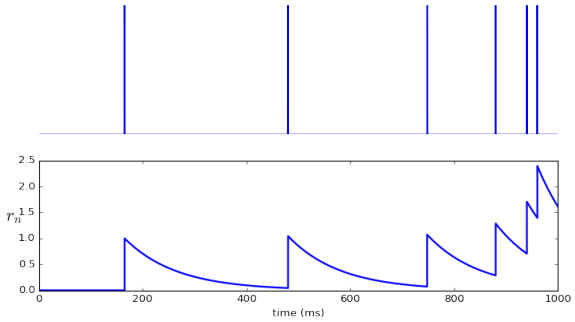


Figure 2.2: A spike train from a single model neuron (top), and the normalized synaptic current r_n that it generates (bottom)

figure 2.1 the signal space is outside the circle and the activity space inside it. The connectivity matrices **F** and **D** make the transition between the two.

We furthermore define convolved versions of the input and output signals. The instantaneous firing rate of neuron n is given by $r_n(t) := o_n(t) * h(t)$, where we assume that $h(t)$ is non-negative and is normalized by $\int_0^\infty h(t)dt = 1$. We chose to convolve the spike train with an exponential filter, such that :

$$r_n(t) := o_n(t) * e^{-\lambda_d t} = \int_0^\infty e^{-\lambda_d t'} s_k(t - t')dt',$$

in other words each spike contributes a decaying exponential kernel to the firing rate.

This operation produces a filtered spike train, which can also be captured by the following differential equation (in vector notation for the population) :

$$\dot{\mathbf{r}}(t) = -\lambda_d \mathbf{r}(t) + \mathbf{o}(t) \tag{2.1}$$

Similarly the filtered input signal, $\mathbf{x}(t)$, is given by

$$\dot{\mathbf{x}}(t) = -\lambda_d \mathbf{x}(t) + \mathbf{c}(t), \tag{2.2}$$

where the parameter λ_d controls the time constant of the decay.

2.1.3 Voltage dynamics

Models of membrane dynamics have been proposed at several levels of biophysical realism. Conductance-based models provide a biophysical representation of excitable cells that allow a very detailed description of the cellular mechanisms. Yet the mathematical tool on which they rely are fairly advanced and require enormous computational resources to simulate populations of neurons. More flexible and expressive approaches such as rate models have been widely used to describe the activity of neural populations. But, as discussed in the introduction, this class of model misses the fundamentally discrete nature of synaptic transmission.

We will therefore focus on an intermediary level, namely current-based models, and more specifically leaky integrate-and-fire (LIF) neurons. Indeed these incorporate crucial features of biological neurons, such as operation in continuous time, spike generation and reset, while also maintaining some degree of analytical tractability. A neuron's state is described by its membrane potential, which performs a leaky integration of its input (both feed-forward and recurrent). As soon as the membrane potential exceeds its threshold, the neuron emits an action potential and resets its value. These spikes are discrete events and, contrary to Hodgkin-Huxley neurons for example, they do not have a characteristic shape. We will have to add vertical lines on the voltage traces to show the time steps at which spikes are emitted. We furthermore assume that the resting potential is halfway between the reset potential and the threshold, and take 0 for simplicity.

Let us start with a simplified case where units are not constrained to be excitatory or inhibitory. It is known that biological network have separate populations of neurons for excitation and inhibition, which is referred to as Dale's principle. The separation of excitation and inhibition will be introduced and analysed in section B. Note that we assume that the number of neurons exceeds the number of dimensions of the signal, *i.e.*$N \geq J$.

Let us consider a recurrent network of N LIF neurons. The spiking rule can be written $V_n(t) > T_n$, where V_n is the time varying membrane potential of neuron n and T_n its fixed threshold. The dynamics of the membrane potential of neuron n are given by :

$$\dot{V_n}(t) := \frac{\partial V_n}{\partial t}(t) = -\lambda_V V_n(t) + \sum_{j=1}^{J} F_{nj} c_j(t) + \sum_{k=1}^{N} \Omega_{nk} o_k(t) + \sigma_V \eta_n(t), \tag{2.3}$$

where F_{nj} is the connection strength from input j to neuron n and Ω_{nk} the connection strength from neuron k to neuron n. The time constant of the voltage leak is set by λ_V. The reset values are included in the diagonal elements of the recurrent connectivity matrix. One could think of them as an autapse. After each spike, the voltage is reset to $V_n \to T_n + \Omega_{nn}$, where Ω_{nn} is a negative number. The variance of the synaptic background noise $\eta_n(t)$ is controlled by σ_V. From now on we will drop the time dependencies to simplify notations.

2.1.4 Readout and objective

We will assume that a downstream area reconstructs an estimate of the input signal from a simple weighted sum of the filtered output spike trains, corresponding to a synaptic integration. This type of linear readout is a classic method used to analyse electro-physiology data [15] [16]. By extension this is also how we understand the readout of an incoming signal by a downstream circuit. This operation can be written

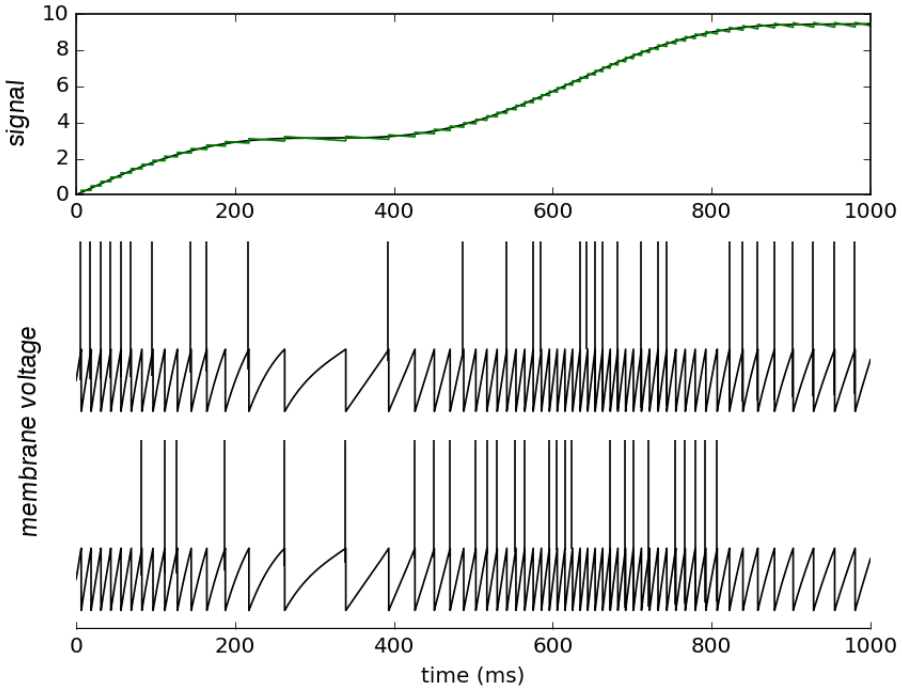


Figure 2.3: **Top panel** Signal space : in black, the time evolution of a one dimensional input signal and in green, the "jumpy" reconstruction achieved by the network. **Bottom panel** Voltage dynamics : voltage traces (arbitrary units) of the two identical LIF neurons that compete to represent the signal. The voltages fluctuate around resting potential. Neuron integrate their input up to threshold, at which point they fire an action potential. After spiking, the membrane is decreased to a reset value. Note also that the respective spike trains of these neurons are irregular.

:

$$\hat{\mathbf{x}} = \sum_{n=1}^{N} \mathbf{D}_n r_n, \tag{2.4}$$

where $\mathbf{D}_n$ is the n-th column of the decoder matrix. This J-dimensional vector characterises the position of neuron n in the activity space and can be interpreted as the the feature represented by this neuron. Note that by applying equation 2.1, we can deduce the dynamics of the estimate signal : $\frac{\partial \hat{\mathbf{x}}}{\partial t} = -\lambda_d \hat{\mathbf{x}} + \mathbf{D}\mathbf{o}$.

Now recall that the objective of this network is to have $\hat{\mathbf{x}} \approx \mathbf{x}$ at low metabolic cost, as shown in the upper panel of 2.3. To quantify how good such a readout performs, we can simply average a loss function over time. Such a loss function is typically composed of two terms :

$$l(\mathbf{x}, \mathbf{r}) = D(\mathbf{x}, \hat{\mathbf{x}}) + C(\mathbf{r}),$$

where the first term is the distance from the real input to its reconstruction, and the second one is a cost over neural activity. Neural responses can be obtained by minimizing this loss over the activity, and this is how we will chose the spiking rule. For mathematical convenience we will use the Euclidean distance and a linear combination of L_2 (quadratic) and L_1 (linear) costs terms. The role of these costs will be discussed in section 2.1.7. The loss function that we use is :

$$l = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \mu \|\mathbf{r}\|_2^2 + \nu \|\mathbf{r}\|_1 , \tag{2.5}$$

where μ and ν control the respective weight of these costs. The performance of the readout over a period T is given by $\mathcal{L} = \langle l(t) \rangle_t$. Provided that T is long enough all possible inputs are seen and this average over time is equivalent to an average over the distribution of the input.

2.1.5 Choice of decoders

Having said that the cost function quantifies the quality of the reconstruction, let us add that it can also be used to determine the best possible decoder. Minimizing the loss over the decoder is a linear regression problem that can be solved. But in some situations we might not have such a simple solution. In general, the optimal decoders (or features) are those minimizing the expected value of the loss. Note that decoders could also be fitted to neural data (see [12]).

For sake of simplicity, we will start by choosing the decoder matrix such that neurons pave the signal space. For example in the next figures, decoders will be evenly distributed on a circle. For signals living in higher dimensions, one can simply sample points from multi-dimensional Gaussian distributions. Algorithms of linear complexity such as Poisson Disk sampling allow to randomly sample points in space, while preserving a certain regularity (*i.e.*avoiding regions with abnormally high or low sample density). In any case, recall that $\mathbf{D} \in \mathbb{R}^{J \times N}$ and that each column $\mathbf{D}_n$ contains the information on the position of neuron n in the J-dimensional signal space.

2.1.6 Connectivity and thresholds

Having defined an objective function and some decoders, we will now derive the feed-forward and recurrent connectivity that will optimally fulfil this objective. In doing so, we take inspiration from the formalism of associative memory models, and more specifically from the energy functions developed by Hopfield [17] [18]. Indeed, in this kind of top-down approaches, one first states an energy function from which the connectivity can then be deduced. Then the network relaxes to states which are local minima of the energy function (also called objective function, or loss function).

Illustration is provided in figure 2.4 where three important things can be remarked. First, we can see that after each spike, the membrane voltage of the emitter neuron and of its neighbours is decreased. After deriving the optimal connectivity, we will

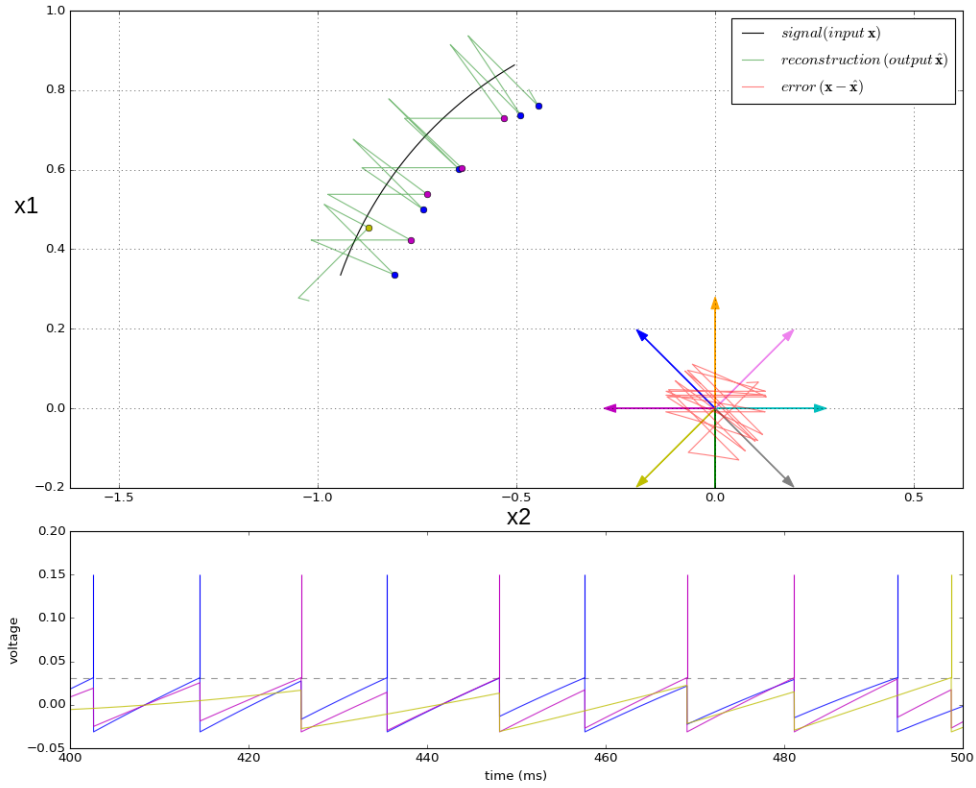


Figure 2.4: Encoding a 2D oscillator, the example of a circular signal and of 8 neurons evenly distributed on a circle (without costs nor noise) **Top panel** : two dimensional signal space. The trajectory of the input signal during a 100 ms time window is represented in black. Its corresponding reconstruction is represented in green. The error (their difference) is represented in red. Colored arrows indicate the position in space of each neuron's decoder and colored dots stand for their respective spikes. **Bottom panel** : voltage traces of the neurons involved in the coding of this signal during this 100 ms time window (using the same color code). Neurons have the same threshold value (dashed black line).

understand that the function of recurrent connection is indeed to spread local information to other parts of the network.

Second, we can visualize the spiking rule geometrically. This figure shows that, as soon as the projection of the reconstruction error exceeds half of a decoder norm, the corresponding neuron emits a spike. Each spike pushes the error toward the origin and the estimate away from it - while the membrane leak does the opposite. We will show in the derivation that voltages are indeed projections of the error onto the decoders.

Third, note that neurons do not strictly take turn. This can be surprising given that every region of the circle is best encoded by one unit. Yet, for a signal that

is larger than the decoders, a combination of decoders can provide a more accurate estimate that any single unit. Indeed when several neurons are active at almost the same time, then their instantaneous firing rate are strictly positive. As a result the readout will combine their respective contribution. This situation does not arise when signal and decoders have similar norms because in that case neurons are only active at distant times.

Following the definitions, the effect of neuron n emitting a spike at time t is an instantaneous increase in the filtered spike train, $r_n \to r_n + 1$, and an update of the signal estimate $\hat{\mathbf{x}} \to \hat{\mathbf{x}} + \mathbf{D}_n$, where $\mathbf{D}_n$ is the n-th column of the decoder matrix $\mathbf{D}$. In that case we can rewrite the loss function 2.5 as :

$$l(t|n \text{ spiked}) = \|\mathbf{x} - \hat{\mathbf{x}} - \mathbf{D}_n\|_2^2 + \mu \|\mathbf{r} + \mathbf{e}_n\|_2^2 + \nu \|\mathbf{r} + \mathbf{e}_n\|_1 ,$$

where $[\mathbf{e}_n]_i = \delta_{ni}$. We want each individual neuron to fire only if that minimizes the overall value of the loss function. To do so, we apply a greedy minimization principle. It means that the minimization is done at every time step without consideration for the future impact of changes induced by a spike. Note that in continuous time, two neurons will never spike simultaneously. But when simulating the network, we have to enforce that at most one spike can be emitted per time step (which is chosen sufficiently small).

This spiking condition can be written $l(t|n \text{ spiked}) < l(t|n \text{ did not spike})$. By multiplying out and cancelling, it directly follows that:

$$\mathbf{D}_n^\top(\mathbf{x} - \hat{\mathbf{x}}) - \mu r_n < \frac{1}{2}(\|\mathbf{D}\|_2^2 + \mu + \nu)$$

Notice that all terms on the left hand side are time dependant, while all terms on the right hand side are constant. It is therefore straightforward to interpret the former ones as a time-varying voltage and the latter ones as a fixed threshold. Such that :

$$V_n = \mathbf{D}_n^\top(\mathbf{x} - \hat{\mathbf{x}}) - \mu r_n \tag{2.6}$$

$$T_n = \frac{1}{2}(\|\mathbf{D}\|_2^2 + \mu + \nu). \tag{2.7}$$

In other words the membrane voltage of neuron n is the projection of the reconstruction error onto its decoder plus a quadratic cost term, and its threshold is half the norm of its decoder plus a penalty. Importantly, this means that each neuron contributes to the global minimisation objective having access to local quantities only. Rephrasing these equations into a geometrical statement, we have that : a neuron fires when the reconstruction error aligns with its decoder.

We could work with this voltage equation, but to describe its voltage dynamics, we have to differentiate it. This operation allows us to comply with the known LIF

model and to use the related terminology. The temporal derivative of V_n is given by :

$$\begin{aligned}\dot{V_n} &= \mathbf{D}_n^\top \dot{\mathbf{x}} - \mathbf{D}_n^\top \mathbf{D}\dot{\mathbf{r}} - \mu \dot{\mathbf{r}}_n \\ &= \mathbf{D}_n^\top(-\lambda_d \mathbf{x} + \mathbf{c}) - \mathbf{D}_n^\top \mathbf{D}(-\lambda_d \mathbf{r} + \mathbf{o}) - \mu(-\lambda_d r_n + o_n) \\ &= -\lambda_d \mathbf{V}_n + \mathbf{D}_n^\top \mathbf{c} - (\mathbf{D}_n^\top \mathbf{D} + \mu \mathbf{e}_n)\mathbf{o},\end{aligned}$$

where we successively applied equations 2.2, 2.1 and 2.6.

Finally we can conclude on the network connectivity by comparing the previous equation with 2.3. This yields the following optimal network connectivity (in matrix notation) :

$$\begin{aligned}\mathbf{F} &= \mathbf{D}^\top \\ \mathbf{\Omega} &= -\mathbf{D}^\top \mathbf{D} - \mu \mathbf{I}\end{aligned}$$

Note that the threshold equation 2.7 could be rewritten $\mathbf{T} = \frac{1}{2}(-\text{diag}(\mathbf{\Omega}) + \nu)$ and that the decay time constant of the membrane and of the filtered spike train are equal, $\lambda_V = \lambda_d$.

Let us now interpret this optimal connectivity. First, let us discuss the optimal feed-forward weights. As expressed by equation 2.6, a neuron only sees a projection of the reconstruction error. And according to the architecture shown in figure 2.1, the error is measured through projection on the feed-forward connectivity $\mathbf{F}$. But, after spikes are emitted, the reconstruction changes according to the readout connectivity $\mathbf{D}$. It is therefore natural that optimaly connected networks both measure and correct the error in the same direction.

Second, let us stress that the optimal recurrent weights have one very important property. To understand it, we have to observe that recurrent weights are symmetric and directly related to the decoding weights. Neurons with orthogonal decoders are not connected to each other, indeed their dot product is null. For example, figure 2.4 shows that the yellow voltage trace is not influenced by spikes from the blue neuron because these two are orthogonal to each other.

Following on the same geometrical interpretation of the dot product, we see that oppositely tuned neurons will excite each other via these connections. Conversely, similarly tuned neurons inhibit each other, which means that a spike from one of them acts as a reset on the others. Note that the impact of a spike on the other parts of the network is instantaneous, recurrent connections act without delay. Therefore these connections can be called fast connections.

Knowing that voltages are projections of the reconstruction error, modifying their value is influencing the global coding objective. When a neuron inhibits another one, it notifies him that the error in the direction they both code for has already been corrected. A neuron that might have participated as well in the reconstruction is prevented to do so, and only few spikes are emitted. This is how the network satisfies

the efficiency objective. Conversely, when a neuron excites an oppositely tuned one, it notifies him that the error it codes for has increased. This second neuron is then more likely to spike.

In conclusion, the property of fast recurrent connections is to instantaneously spread local information to all the members of the network.

Overall, the result is that membrane voltages vary around zero. Therefore, these fast recurrent connections enforce a balance between the excitatory and inhibitory currents. To put it shortly, the efficient coding spiking rule leads to balanced networks.

2.1.7 Costs and regularization

Several combinations of neuronal activity can minimize the distance between the signal and its reconstruction. In other words the solution space to this task is degenerate. Let us consider two extreme solutions : the sparse and the dense regime. In the sparse regime, a small subset of neurons contributes to the task and emit many spikes, which results in high firing rates. In the dense regime, a large fraction of neurons participate and each one emits a small number of spike, which results in low firing rates. The costs, that we introduced in the loss 2.5, allow to regularize the solution and to bring the network in the desired regime.

Note nevertheless that this degeneracy is a desirable feature of the model. When many neurons compete to represent a low-dimensional signal (high ratio N/J), one out of several combination of spike trains will be realized. And when running the network again, a different combination of activity will be visited. Therefore, this inter-trial variability is not noise, it is only a consequence of the degeneracy. The model suggests that the variability observed in recordings is due to the realisation at each trial of one out of the many possible responses.

The linear cost controlled by ν only raises the threshold of all the neurons (equation 2.7). In other words it makes spiking more difficult, such that responses are sparser. Note that this cost can also be used to solve the pathological ping-pong effect. This case occurs when the spike of one neuron brings the error exactly up to the threshold of an oppositely tuned neuron. At the next time step, this other neuron immediately replies by another spike. This new will then have the same effect on the first neurons, and, in the absence of cost, this would go on.

The quadratic cost controlled by μ also increases thresholds, but it has an additional negative impact on the recurrent weights. This second component reduces the amount of information that a spiking neuron spreads to other members of the network. By doing so, it permits a wider set of the neurons to fire. But a side effect of this quadratic cost is to downscale the readout. Indeed two requirement have to be accommodated, namely representing the signal accurately and emitting few spikes.

The loss function shows the trade-off between these two objectives :

$$l = \sum_{j=1}^{J}(x_j - D_j r_j)^2 + \mu \sum_{n=1}^{N} r_n^2 + \nu \sum_{n=1}^{N} |r_n|$$

where the first objective scales with the dimensionality of the signal and the second with the number of neurons in the network. Therefore one has to chose the values controlling this trade-off according to a clear policy. One possible approach is to keep the reset value constant with respect to network size, indeed biophysical properties might be unchanged by the number of neurons. Another possible approach is to keep the trade-off between the two objectives constant for all network sizes. We opt for this second option, and therefore scale the norm of the decoders by N.

2.2 Derivation of the optimal network : computing

We have seen that the efficient coding principles results in networks that are tightly balanced. But this constraint does not allow to carry more general computations. To build functional network, we will now separate the time scales for encoding and for computing. More specifically we will introduce a set of connections that operate on the rates, that is to say on a slower time scale. From now on, when a neuron in the network fires an action potential, it contributes both fast and slow synaptic currents to other members of the network. Using the previously described auto-encoder as building block, let us now derive a network architecture capable of solving dynamics.

2.2.1 Linear dynamical systems

Animal behaviors are usually described by a set of symbolic instruction. But rules and goals of the task can not be processed as such by the previously described framework. Yet, one can reformulate them in terms of continuous dynamics, which motivates an extension of the framework to make it able to track a system of differential equations. In previous work [13], slow connections were added and applied directly on the convolved activity. The specificity of these connection is that their effect lasts several time steps after a spike. Note that this idea is also present in a different form in the Hopfield networks literature [19].

The voltage dynamics of neuron n mow includes a slow component and is given by :

$$\frac{\partial V_n}{\partial t}(t) = -\lambda_V V_n(t) + \sum_{j=1}^{J} F_{nj} c_j(t) + \sum_{k=1}^{N} \Omega_{nk} o_k(t) + \sum_{k=1}^{N} \Phi_{nk} r_k(t) + \sigma_V \eta_n(t), \quad (2.8)$$

where Φ_{nk} is the weight of the slow connection from neuron n to neuron k, the other terms are similar to the previous voltage dynamic (equation 2.3).

In analogy with the previous section, we will now derive the optimal value for these connections. The idea is simply to feed in a dynamical system in place of $\dot{\mathbf{x}}$.

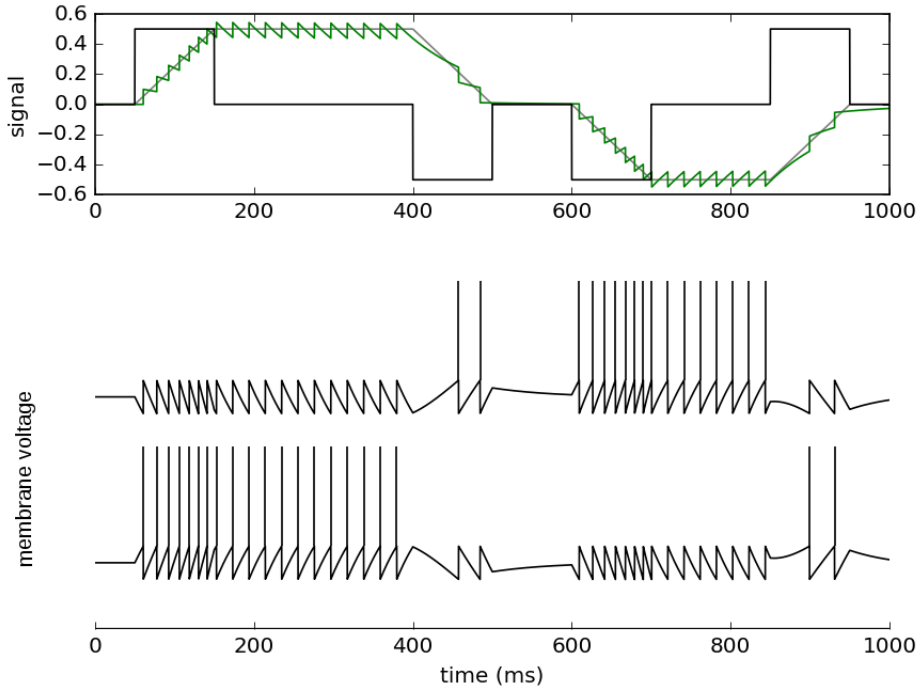


Figure 2.5: Perfect integrator network. **Top panel** Signal space : in black, the time evolution of a one dimensional input signal and in green, its integration by the network. The readout is close to the perfect integration (target in grey). **Bottom panel** Voltage dynamics : in black, voltage traces of the two oppositely tuned neurons. The neuron on the top codes for negative values, and the one on the bottom for positive values. In accordance with the geometrical interpretation of the fast recurrent connections, we see that each spike resets the emitter neuron and excites the oppositely tuned neuron.

Let us consider a linear dynamical system described by :

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{c}(t), \tag{2.9}$$

where $\mathbf{A} \in \mathbb{R}^{J \times N}$ is the state transition matrix and $\mathbf{c}(t)$ is the control signal. Two simple cases can already be described. When $\mathbf{A} = -\lambda\mathbf{I}$, the system filters the signal $\mathbf{c}(t)$. This first example recovers the auto-encoder network discussed in the previous section. The second example, shown in figure 2.5, is a one dimensional integrator where $x(t) = \int_0^t c(t')dt'$. This integration is perfect because there is no leak ($i.e. A = 0$, which corresponds to an infinite time constant). Nevertheless, it is relevant to add a leak to model biological systems.

Following the same logic as before, let us differentiate the voltage equation 2.6, we now obtain :

$$\begin{aligned}\dot{V}_n &= \mathbf{D}_n^\top \dot{\mathbf{x}} - \mathbf{D}_n^\top \dot{\hat{\mathbf{x}}} - \mu \dot{\mathbf{r}}_n \\ &= \mathbf{D}_n^\top (\mathbf{A}\mathbf{x} + \mathbf{c}) - \mathbf{D}_n^\top (-\lambda_d \hat{\mathbf{x}} + \mathbf{D}\mathbf{o}) - \mu(-\lambda_d r_n + o_n) \\ &= \mathbf{D}_n^\top \mathbf{A}\mathbf{x} + \mathbf{D}_n^\top \mathbf{c} + \lambda_d \mathbf{D}_n^\top \hat{\mathbf{x}} - (\mathbf{D}^\top \mathbf{D}_n + \mu \mathbf{e}_n)^\top \mathbf{o} + \mu \lambda_d r_n,\end{aligned}$$

where we used equations 2.9, 2.4 and 2.1 in the second line. We can now recognize $\mathbf{\Omega}$ and use $\mathbf{D}_n^\top \hat{\mathbf{x}} = \mathbf{D}_n^\top \mathbf{x} - V_n - \mu r_n$ from the voltage equation 2.6. This yields :

$$\begin{aligned}\dot{V}_n &= \mathbf{D}_n^\top \mathbf{A}\mathbf{x} + \mathbf{D}_n^\top \mathbf{c} + \lambda_d (\mathbf{D}_n^\top \mathbf{x} - V_n - \mu r_n) + \mathbf{\Omega}_n \mathbf{o} + \mu \lambda_d r_n \\ &= -\lambda_d V_n + \mathbf{D}_n^\top \mathbf{c} + \mathbf{\Omega}_n \mathbf{o} + \mathbf{D}_n^\top (\mathbf{A} + \lambda_d \mathbf{I})\mathbf{x}.\end{aligned}$$

But neuron n could not possibly have access to the signal $\mathbf{x}$. Therefore, we need to continue the derivation and replace this signal by a local quantity (that is a quantity to which the neuron has access). We saw in the previous section that when fast connections balance the network, then the encoding objective is fulfilled. Assuming that this condition is met, we can use $\mathbf{x} \approx \hat{\mathbf{x}}$ and apply equation 2.4 again. The next subsection discusses the exact solution. We finally get the following approximate solution :

$$\dot{V}_n \approx -\lambda_d V_n + \mathbf{D}_n^\top \mathbf{c} + \mathbf{\Omega}_n \mathbf{o} + \mathbf{D}_n^\top (\mathbf{A} + \lambda_d \mathbf{I})\mathbf{D}\mathbf{r}.$$

Comparing this approximation with equation 2.8, we can now conclude that the optimal slow connections take the form (in matrix notation) :

$$\mathbf{\Phi} = \mathbf{D}^\top (\mathbf{A} + \lambda_d \mathbf{I})\mathbf{D}$$

This set of slow connection resembles the fast connections, except for the introduction of the state transition matrix and of the leak. The major difference is that slow connections are applied on rates, so that their effect lasts several time step after the emission of a spike.

2.2.2 Exact solution

In order to correctly replace $\mathbf{x}$ in the voltage dynamics, we have to use the voltage equation 2.6. We can write (in vector notation) : $\mathbf{D}^\top \hat{\mathbf{x}} = \mathbf{D}^\top \mathbf{x} - \mathbf{V} - \mu \mathbf{r}$. The matrix $\mathbf{D}^\top$ being rectangular, we need to compute its left pseudo inverse $\mathbf{L}$. We have $\mathbf{L} = (\mathbf{D}\mathbf{D}^\top)^{-1}\mathbf{D}$, which yields for neuron n :

$$\mathbf{x} = \mathbf{L}_n V_n + \hat{\mathbf{x}} + \mu \mathbf{L}_n r_n$$

Using this new equation, we obtain the exact voltage dynamics :

$$\dot{V}_n = \mathbf{D}_n^\top \mathbf{A}\mathbf{L}_n V_n - \mu \mathbf{D}_n^\top (\mathbf{A} + \lambda_d \mathbf{I})\mathbf{L}_n r_n + \mathbf{D}_n^\top \mathbf{c} + \mathbf{\Omega}_n \mathbf{o} + \mathbf{\Phi}_n \mathbf{r},$$

where we used the slow connections as derived previously.

Nevertheless, this solution is not biologically plausible because it assumes a leak term that depends on the voltages of other neurons. This equation is also unusable when implementing inactivation. Indeed for the same reason, disrupting the voltage of some neurons will directly disrupt the voltages of other neurons. Overall the voltage leak should remain a local quantity. We will therefore stick to the approximate solution. Moreover an asymptotic argument can be made to ensure that this solution is valid for large networks.

2.3 Implementation of transient inactivations

As described in the introduction, we are mainly interested in the robustness of the network. Modeling optogenetics inactivation is very straightforward, we force the membrane potential to remain at resting potential. This manipulation prevents inactivated neurons to cross threshold.

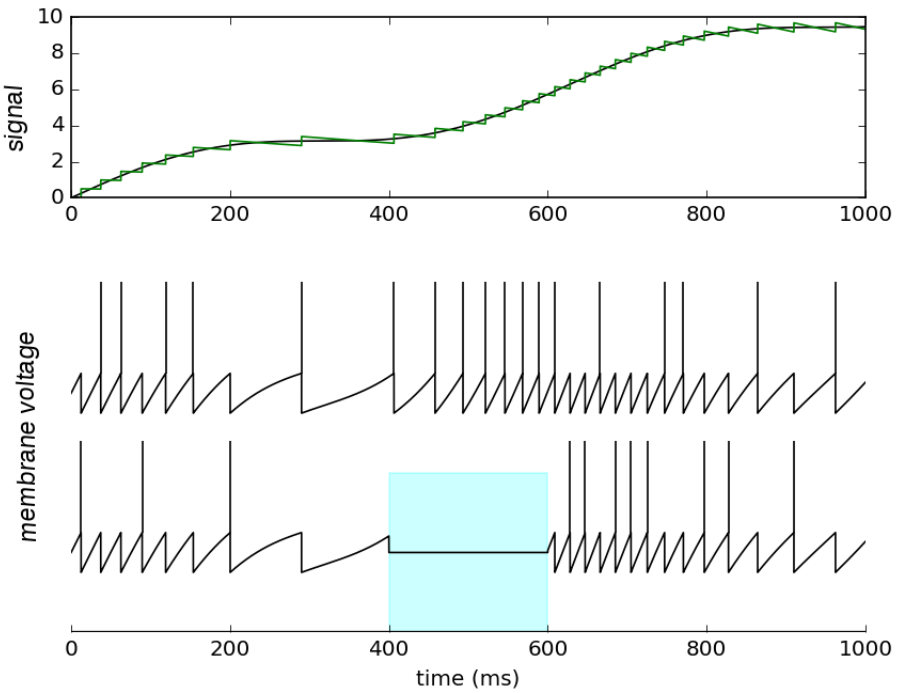


Figure 2.6: Inactivation **Top panel** Signal space : in black, the time evolution of a one dimensional input signal and in green, its reconstruction by the network. The accuracy of the encoding is not compromised by the inactivation. **Bottom panel** Voltage dynamics : in black, voltage traces of the two identical neurons. The shaded cyan area indicates the time window of inactivation. The remaining neuron compensates for the silence of the inactivated one.

Figure 2.6 presents one implementation of inactivation, but it could be simulated in other ways. For example, rather than clipping the membrane voltage to zero, one could strongly hyperpolarise it. But, once the inactivation window ends, it then takes unreasonable time to reach resting potential again. This is the reason why we favoured the first method. Note that by construction, manipulating the membrane dynamics or the threshold is equivalent up to a convolution. Note also that the effect of using a smaller λ_d compared to figure 2.3 and 2.5 is a longer decay time constant and less spikes.

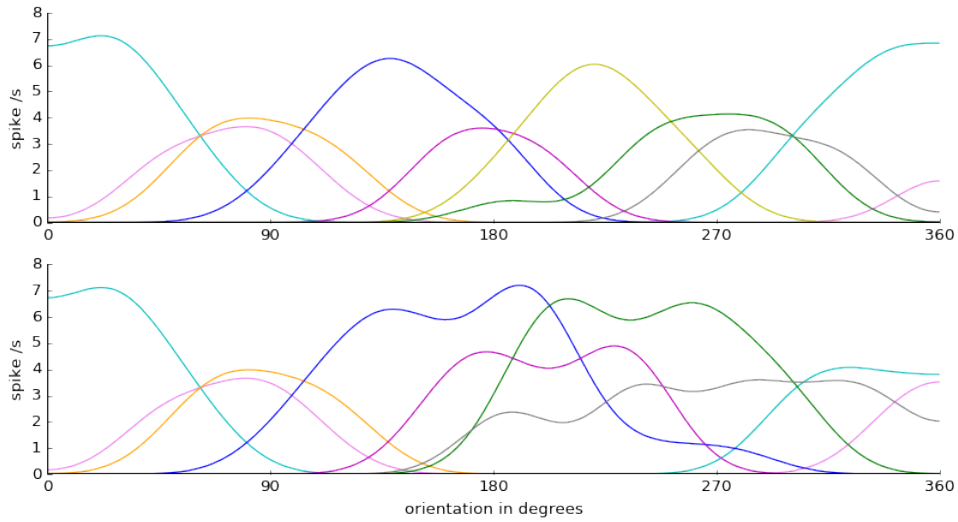


Figure 2.7: Tuning curves are a temporary solution to an optimisation problem. Same set-up and color code as in figure 2.4 : a two-dimensional circular signal encoded by eight neurons evenly distributed on a circle. **Upper panel** : tuning curves in the standard case. Each neuron codes for a signal feature, depending on the direction of its decoder in the signal space. **Lower panel** : tuning curves after the inactivation of one neuron. The yellow neuron has been silenced, therefore its neighbour have to broaden their tuning and fire more to compensate for its absence.

Figure 2.7 shows that an external perturbation can modify the tuning curves of the neurons. In general, this framework suggests that tuning curves of biological neurons are not invariant. Indeed when presenting several different inputs to the network, it is necessary to keep learning the optimal decoders in order to reach the most metabolically efficient representation (not shown in this document). In this case, tuning curves are constantly updated.

Chapter 3

Results

We will now turn to the mouse pre-motor system and present an instantiation of the recurrent spiking network. Following the three steps we took when introducing the data in section 1.2, let us discuss the results of our simulations in the same order. We will first describe theoretical analogues to the task, then to the neural recordings and eventually to the manipulations. In turn, these results will motivate expansions on the model, and we will introduce an approach to make its architecture more realistic.

3.1 Set-up

Applying the model requires us to interpret the task in terms of continuous dynamics. Indeed this is the kind of element that can be processed by the network. Stimulus can be presented in two different location, which is easily described as a step function. We will represent anterior pole location by a positive step, and posterior location by a negative step. The corresponding contingency is : lick right for anterior stimulation and lick left for posterior stimulation of the whiskers.

Now, from a network that receives an input step, we want to generate responses analogue to the recordings. To do so we have to design a dynamical system that captures most of the trajectory of the neural responses and then let the network solve these dynamics. It has been observed that the task elicits variable yet characteristic neural responses. From these we can extract the principal components and assign each of them to one dimension of the dynamical system.

Three main modes emerge from the analysis of the recordings in ALM. First, there is a large burst of activity during the sampling epoch. This flow of cortical activity corresponds to the reception of the sensory information. This mode can therefore be described by a simple encoding of the stimulus. The second mode is a sustained activity during the delay period. This activity is crucial to hold the information after the stimulation stopped and until the response. Integration of the sensory evidence is a natural way to describe it. The third component of the activity is the steady increase of all firing rates throughout the task. This ramping mode can be captured by a squared integration. Indeed, integrating the persistent activity is a way to obtain an increasing function. The resulting three dimensional dynamical system is depicted

in the left panel of next figure. In analogy with the experimental work, we will give names to the three different periods of the task. The "sampling epoch" goes from 0 to 250 ms, it corresponds to stimulus encoding. The "delay epoch", from 250 to 700 ms, corresponds to the waiting time. And finally the "response epoch" goes from 700 to 1000 ms.

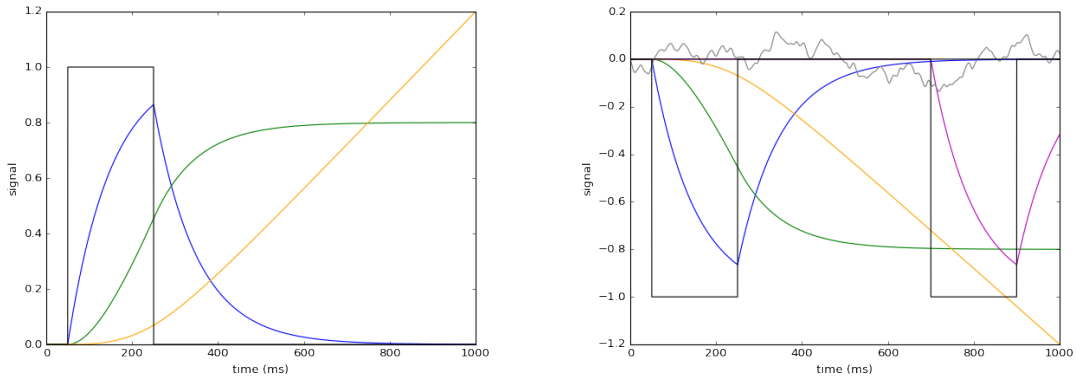


Figure 3.1: Dynamical system that captures modes of the neuronal activity. **Left panel** : simple three dimensional system. The stimulus is represented by the black step and its encoding by the blue trace. Integration (green) and squared integration (orange) of this trace correspond respectively to persistent activity and to ramping activity. **Right panel** : a five dimensional system. On top of the three modes previously described, encoding of the go cue (magenta) and noise (grey) correspond respectively to response burst and to background activity. Note that the input command is negative, corresponding to a lick left trial.

Let us write down the corresponding linear dynamical system. Following on the methods (equation 2.9), we simply have to choose the relevant state transition matrix. This yields :

$$\dot{\mathbf{x}} = \begin{pmatrix} -1 & 0 & 0 \\ \alpha & 0 & 0 \\ 0 & \beta & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} stimulus \\ 0 \\ 0 \end{pmatrix},$$

where α and β are values between 0 and 1. Indeed, the first dimension of the system encodes the stimulus, the second one integrates the stimulus scaled by a factor α and the third dimension integrates the second one scaled by a factor β.

It is of course possible to design complex dynamics that would account for more features of the recordings. For example, there is another burst of activity at the go cue that continues during the response. This can be described as another step input and its encoding. Moreover, there is ongoing activity in ALM, that could be approximated by noise. Indeed this region might be involved in many other computations. A system incorporating these additional dimensions is depicted in the right panel of figure 3.1.

Another project would be to design a network able to compute a binary decision (corresponding to lick left or right response) from the sensory evidence it received. This would require non-linear operations that are not supported by the architecture discussed here. Exploratory analysis of dendritic non-linearities indicate that the framework could be further expanded to carry such tasks. Note that letting the persistent activity and ramping activity fade away as soon as the response has been emitted would bring the model even closer to the experiment. For now, we will present results obtained from the simple three dimensional dynamical system only.

3.2 Neuronal activity

Having shown that the trajectory of the neural responses can be captured by a set of dynamics, let us feed them to the network and analyse the resulting activity. To do so, we simply have to set the connectivity of the network to its optimal value (as described in the methods). Because the membrane voltage equation contains a noise term, running the task several times will yield different output spike trains. Repeating this operation for each conditions of the task allows to compute statistics of the population activity. Parameter values and decoders used in the simulations are available in A.1.

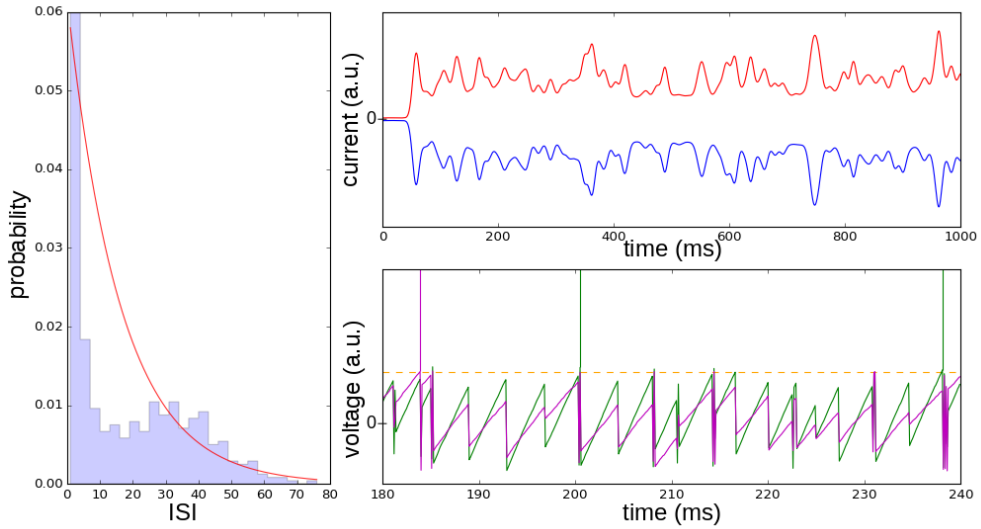


Figure 3.2: Neural activity is variable and the network is balanced. **Left panel** : the inter-spike intervals distribution is (almost) close to the exponential distribution, which is a signature of Poisson-like variability (Fano factor = 0.99). **Upper right panel** : input currents to each neuron are tightly balanced. Blue and red traces correspond respectively to inhibitory and excitatory currents received by the example neuron. **Lower right panel** : membrane voltage of two similarly tuned neurons. The dashed orange line represents the spiking threshold. Even though voltages are strongly correlated, the spike trains differ.

We observe in figure 3.2 that, alike recordings, modeled neural responses are highly variable and that the network is tightly balanced. The model provides underlying mechanisms for these key features. As explained in the first paragraphs of section 2.1.7, variability is not noise. It is only a consequence of the redundancy of the network. At each run, one out of the many possible response patterns is realized. And as explained in the last paragraphs of 2.1.6, balance is a consequence of the efficient coding objective. Fast recurrent connections spread local information across the network.

Let us now investigate individual cells properties. We will consider the raster plots and peri-stimulus time histogram of four example neurons. Each line of the raster plot corresponds to the activity of one given cell across several trials. We present results for forty runs of the network in each of the conditions : left or right stimulus, and standard or inactivation trial (160 runs in total). We show neurons that are selective for the lick right trials, but the same goes for left selective neurons.

Figures 3.3 to 3.6: Four example neurons during lick left and lick right trials (respectively coded in red and in blue). Top (resp. bottom) of each figure shows the raster plot (resp. the PSTH). Dashed lines indicate sample, delay and response epochs.

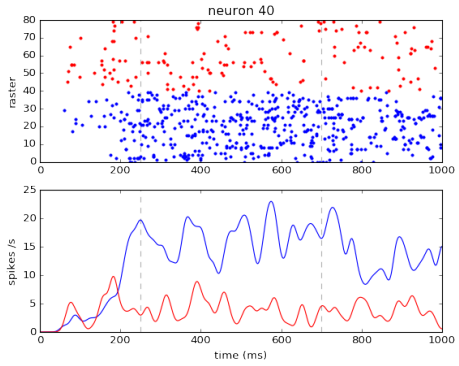


Figure 3.3: this right selective neuron shows persistent activity

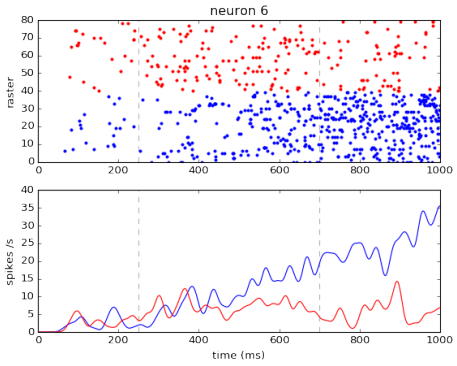


Figure 3.4: this right selective neuron shows ramping activity

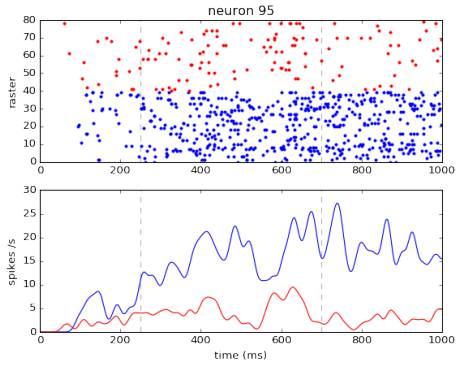


Figure 3.5: this right selective neuron shows persistent activity

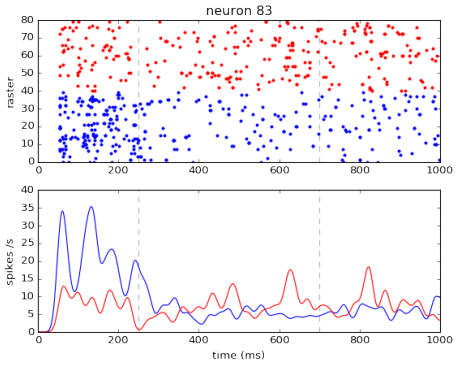


Figure 3.6: this right selective neuron shows encoding activity

To better understand the results shown in figures 3.3 to 3.6, we again have to refer to the methods. Each modeled neuron codes for a feature of the signal, and we saw that this selectivity depends on the neuron's position in the signal space. These positions are described by the columns of the decoder matrix. We derived from the efficient coding objective that a neuron should fire whenever the reconstruction error is in the direction of its decoder (equation 2.6). In other words, neurons should be recruited when the signal is in the direction for which they code.

Here we sampled decoders randomly on the surface of the unit sphere (see A.1 for illustration). This means that their decoders have three non-zero components (except for those falling on the axis planes). Therefore each neuron will fire for a specific combination of the signal dimensions. In other words, each neuron has a mixed selectivity, and the observed activity is the solutions to an optimization problem solved at the population level.

In conclusion, modeled neurons are able to reproduce the recorded patterns of activity both at the population and at the individual level. And the model suggests an explanation to this intermingled activity in terms of position of the decoders in the signal space.

3.3 Compensatory mechanisms

Having shown that the network is able to generate the desired activity, let us now probe its robustness. In experiments, transient optogenetic inactivation has been applied during the delay epoch to one of the anterior lateral motor cortex. To mimic this manipulation, we will clip to zero the membrane voltages the first half of the neurons during the delay. Let's assume that neurons 0 to 49 (resp. 50 to 99) stand for the left hemisphere (resp. right hemisphere).

Note that we do not add any further components to the network. We previously designed a dynamical system to reproduce recorded activity, and here we simply assess the robustness of the network as it is.

3.3.1 The network is able to compensate

To evaluate the performance of the network, we have to consider two things : first, the quality of the readout (distance between the reconstructed and the desired signal) and second, the efficiency of the code (number of spikes emitted). We want to see if the encoding, the persistent and the ramping activity modes are preserved. The standard and the perturbed network should have equivalent behaviors, except for the time window of inactivation in which some cells are silent and others compensate.

The mechanism for the compensation shown in figure 3.7 relies on the redundancy of the network. Indeed, when several neurons code for the same feature, removing one of them induces the remaining neurons to compensate, so that the readout is overall unaffected. This was shown in figure 2.6 for the one-dimensional case with two neurons, and the same applies here. Here, redundancy is naturally built in the

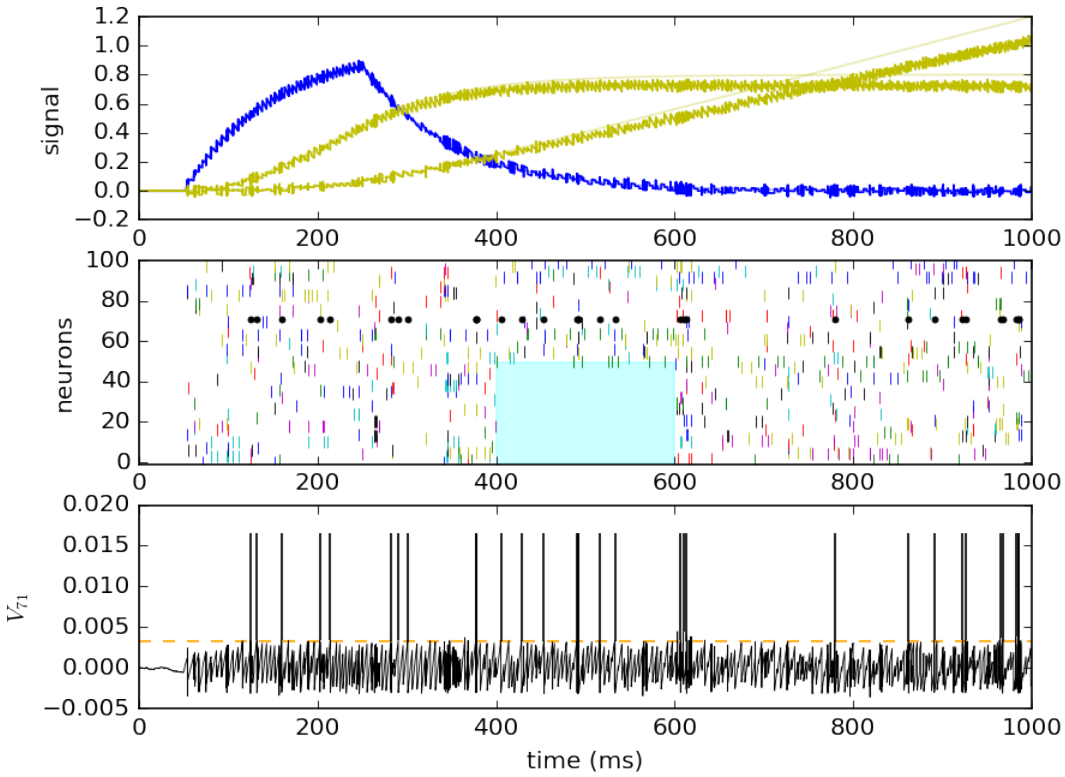


Figure 3.7: The network performs the task in spite of perturbation.
Top panel : "jumpy" output signal. In blue the encoding of the step input, and in yellow its integration and squared integration. The thin lines of the same color are the target signal. Due to the leak, the reconstruction progressively deviates from the target. The accuracy of the readout is not affected by the perturbation. **Middle panel** : raster plot. The shaded cyan area represents the inactivation. This manipulation has no other effect on activity than silencing half of the neurons. **Bottom panel** : voltage trace of the neuron which emitted most spikes (highlighted in the raster plot). We recognise a LIF neuron, fluctuating around resting potential and emitting spikes when hitting threshold (dashed orange line).

network. Given that one hundred neurons span a three dimensional signal space, each dimension of the signal can still be accounted for when some neurons are silenced (see A.1 for illustration). By following the prescribed spiking rule, remaining neurons naturally and optimally adapt their activity.

But there is a limit to the compensation ability of the network. Following on the same interpretation of the model, as soon as the signal enters a region of the space that is not covered by any neuron, the network will fail. Indeed, neurons only see projections of the error onto their decoders. If no decoder points in the direction of the error, then this error is not seen and can therefore not be corrected. For example in figure 2.5, there are two oppositely tuned neurons integrating a one dimensional

input. If one of them was silenced, then the other one would not be able to replace it and the task would not be realized. In general, the symptom of this situation is the imbalance of the input received by the neurons. Here, as expected, when all neurons are silenced, the network is unable to restore information and it fails.

3.3.2 Neurons return to their initial trajectories

The critical part of the experimental results is that after inactivation of one hemisphere, the activity of neurons returns to its initial trajectory. This is true for both hemispheres, that is to say for both the compensating and the perturbed neurons. As discussed in [6], state of the art models failed to reproduce this effect. Let us look at the results of our model.

Figures 3.8 to 3.11: Same four right-selective example neurons during lick right trials. Comparison of the trials without inactivation (coded in black) to the trials with inactivation (coded in cyan). During inactivation trials, the first half of the network is silenced between 400 and 600 ms, and the second half compensates for this loss.

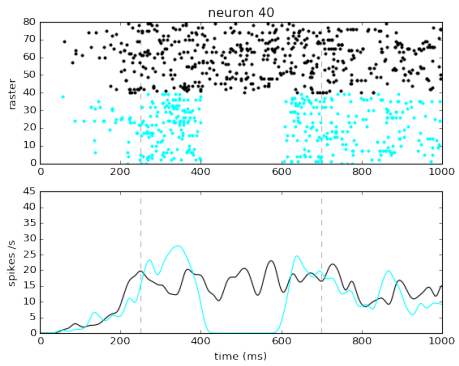


Figure 3.8: this silenced neuron returns to its persistent activity

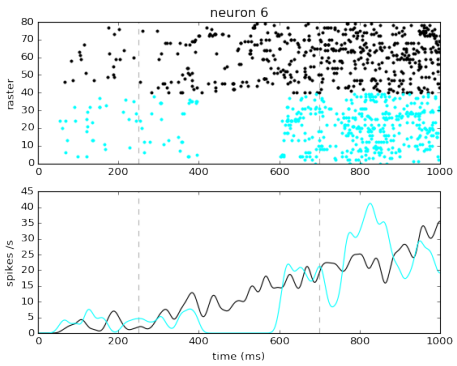


Figure 3.9: this silenced neuron returns to its ramping activity

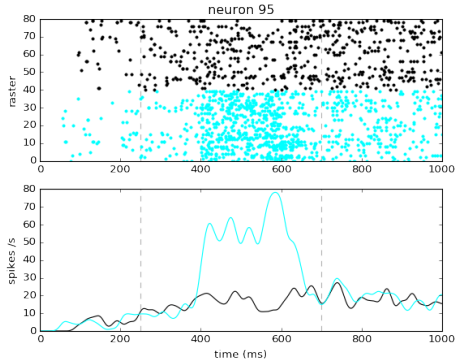


Figure 3.10: this neuron is enhanced during the perturbation and then returns to its trajectory

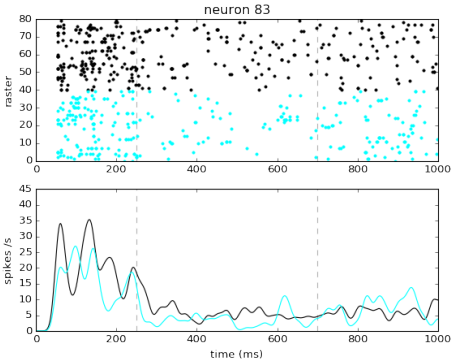


Figure 3.11: this neuron is depressed during the perturbation and then returns to its trajectory

Figures 3.8 to 3.11 show that, in accordance with data, the activity of the modeled neurons returns to its initial trajectory. But here, activity catches up very rapidly after inactivation, while it takes tens of milliseconds in the data presented in [6]. It should be possible to chose the parameters of the simulation to reproduce this compensation time constant. Yet, we preferred to avoid any parameter tuning and chose to present the network as it is.

Following on the discussion of the variability of the network, the presence of costs were critical to obtain this result. Indeed, costs regularize the optimisation problems by decreasing the space of solution. In other words, they control the number of possible redundant network solutions. Without costs, the network would always visit new combinations of activities to solve the task. This is due to the high neuron over dimension ratio of our case ($N/J = 100/3$, see A.1 for illustration).

3.3.3 Compensation is heterogeneous

Interestingly, recordings revealed that the compensating activity of the remaining neurons is very heterogeneous. When inactivation is applied, some cells fire more compared to baseline, but others fire less. This might be counter intuitive given that more spikes are often associated with more information. Let us see whether compensating cells from our network are able to reproduce the zoo of activity described by experimentalists.

Figure 3.12 shows that compensatory activity is also heterogeneous in the network. To characterise this activity, we once again have to understand that it depends on the relative position of the decoders (see A.1 for illustration). In the baseline case, each region of space is covered by many neurons. But after inactivation, this can change, and some regions might be less well covered. Let us consider two cases.

In the first case, after inactivation, a region of the signal space is covered by a few remaining neurons only. Then, when the reconstruction error enters this part of the space, these few neurons have to fire more than in baseline. In baseline, the responsibility of correcting the error was shared among more neurons, and each one was recruited less often.

In the second case, the opposite is true. Imagine that after inactivation, one direction of the signal space is poorly covered. Then, neurons pointing in the opposite direction should fire less. Indeed these neurons are less likely to cross threshold because the neurons that should send them excitation are silent (recall that oppositely tuned neurons excite each other through fast connections, lasts paragraphs of section 2.1.6).

In conclusion, remaining neurons that are similarly tuned to the silenced ones fire more than in baseline, and conversely for oppositely tuned neurons. The network was able to reproduce the diversity of compensatory mechanisms observed in the data and suggests that it is due to the relations between signal features.

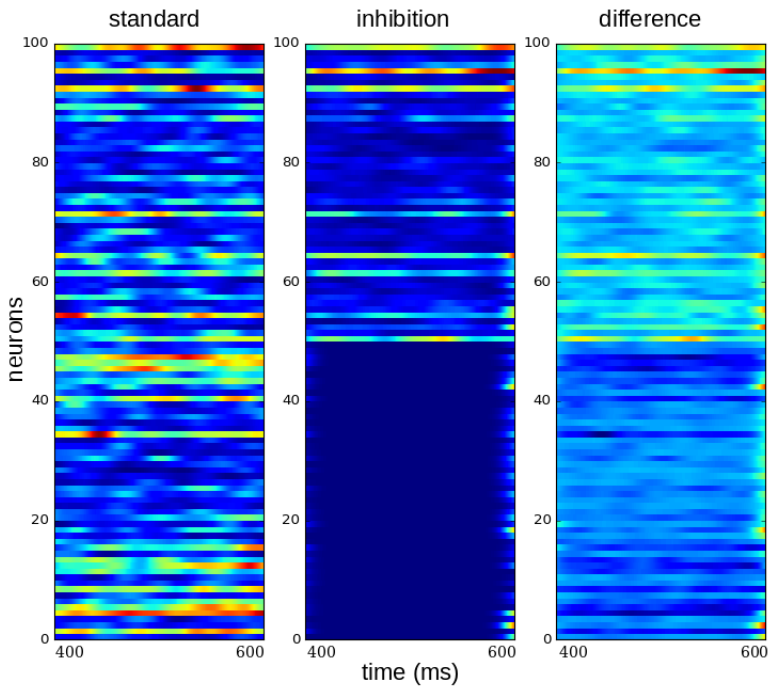


Figure 3.12: PSTH of all the neurons during the 200 ms time window of inactivation. Comparison of the case without inactivation (left panel) to the case with inactivation (middle panel). Neurons corresponding to the left hemisphere (0 to 49) have been silenced. The difference between the two conditions (right panel) shows that compensation is heterogeneous. Indeed some of the remaining neurons fire more (positive values in red) and some other fire less (negative values in blue).

3.4 Anatomical constraints on the model

Results presented so far were obtained with a network of neurons that send both excitatory and inhibitory projections. But this is not biologically plausible. Not only it does not comply with Dale's principle, is also assumes unrealistically fast communication through the corpus callusum of both excitatory and inhibitory projections. The first problem can be solved using the approach outlined in B. For the second one, we could develop an architecture where each hemisphere has its private inhibitory pool of neuron that does not communicate with the other side.

Such an architecture is shown in 3.13 and could be simulated as well. It would also open the door for more advanced applications, such as replicating the lateralized bias in licking induced by the unilateral perturbation. To do so, we would simply have to inject some asymmetry in the network by differently biasing the pool of sampled decoders for each of the two hemisphere. Such an approach seems to have support in experimental data which show that the two hemispheres do not compute exactly the same thing.

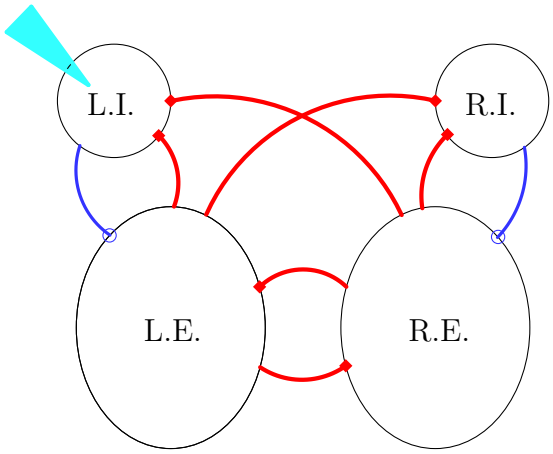


Figure 3.13: Private inhibition. In red excitatory projections, in blue inhibitory projections (both fast and slow). Unilateral inactivation can be implemented by depolarizing one pool of inhibitory neurons. L. R. I. and E. describes the neuronal populations and respectively stand for Left, Right, Inhibitory, Excitatory.

Chapter 4

Conclusion

4.1 Discussion

We have shown that recurrent spiking networks can be derived from an efficient coding objective and that they are able to perform interesting tasks such as solving linear dynamics. These networks are also able to reproduce two features that are ubiquitous in cortex : the response variability and the tight excitation - inhibition balance. The first is a consequence of degeneracy and does not equate with noise, while the second is a signature of efficient coding. Overall, properties of single neurons are temporary network solutions to an optimization problem solved at the population level

Moreover, we established that these networks are robust to perturbations, provided that neurons are redundant. This allowed to reproduce specific features of the data, namely that after perturbation, neurons return to their initial trajectories and that the compensation activity of the remaining neuron is heterogeneous.

Compared to other approaches to the same kind of problems, this model has the advantage of reproducing statistics of neural activity and of being able to compute at the same time, while being understandable in simple geometrical terms.

More importantly, this model did not need any additional component to account for the robustness of neural activity. In this respect it has an advantage over the alternative option presented in [6], where robustness was obtained only after adding a redundant modules. Plus our model also shows intra-hemispheric robustness.

Nevertheless, the assumption giving rise to these results are not exactly in line with the modeled system. The network is highly recurrent and employs fast connection. This is a fair approach to describing local microcircuits, but accounting for long range communication between cortical areas requires substantial adaptation of their architecture. We outlined a network architecture that could incorporate some important anatomical constraints.

More generally, the theoretical framework used here has its own limitations. Because it relies on fully connected networks, it lacks the sparsity of connections observed

in cortex. Given that the network is built on the basis of these many symmetrical connections, there is no easy fix to this problem. Nonetheless it does not discredit the ideas developed, and in high dimensional spaces this problem disappears (indeed, in that case the dot product of two random vectors will be often be null).

One real limitation to this approach is that it provides no information at all on how representation are constructed. By focusing on the decoding of the information, we neglected the question of encoding. However, predicting neural activity from the state of the animal and of its environment is a very important challenge.

4.2 Perspectives

On top of the network architecture modifications suggested in the text, it would be interesting to expand this work in several directions. Here is a list of intriguing possibilities.

(1) Letting the network learn its connectivity using synaptic plasticity rules instead of ascribing the optimal connectivity from start. Ongoing work [14] shows that neurons in the learning network simply act to balance their input, which spares the disputable assumption of predictive neurons that spike only when the predicted impact of a spike is to minimize the global objective function.

(2) Expanding the functional repertoire of the network computational abilities by exploring the role of dentrites in parallel computing to solve non-linear differential equations. Ongoing work suggests that sigmoidal dendritic non-linearities as described in the Poirazi model [20] would be useful.

(3) Introducing synaptic delays and asymmetries in connectivity in order to relax the assumption of instantaneous communications [21]. It would allow to explore the role of different time constants for the inhibitory neurons, and more generally to design several cell subtypes (not just two).

(4) Deriving the connectivity from a different cost function, changing the type distance and of costs and analyse the properties of the resulting network. Also one could reinterpret the loss in a probabilistic framework, where the distance would be derived from the likelihood of the stimulus, and the costs from the prior distribution over features.

(5) Analysing the experimental data that have been discussed (available data online on CRCNS). Applying a demixed principal component analysis [22] would help to understand the activity modes, and to expand the dynamical system. This would also help to shed light on the referential restoration of certain components of the activity that has been documented in [6].

Finally, this model makes a critical prediction that could be empirically tested to decide of its value. Neurons tuned to the same feature, that is to say neurons that receive correlated inputs, should inhibit each other - while oppositely tuned neurons

should excite each other. This preferential connectivity should be observable in the emerging connectivity data that uses on genetic barcoding technologies.

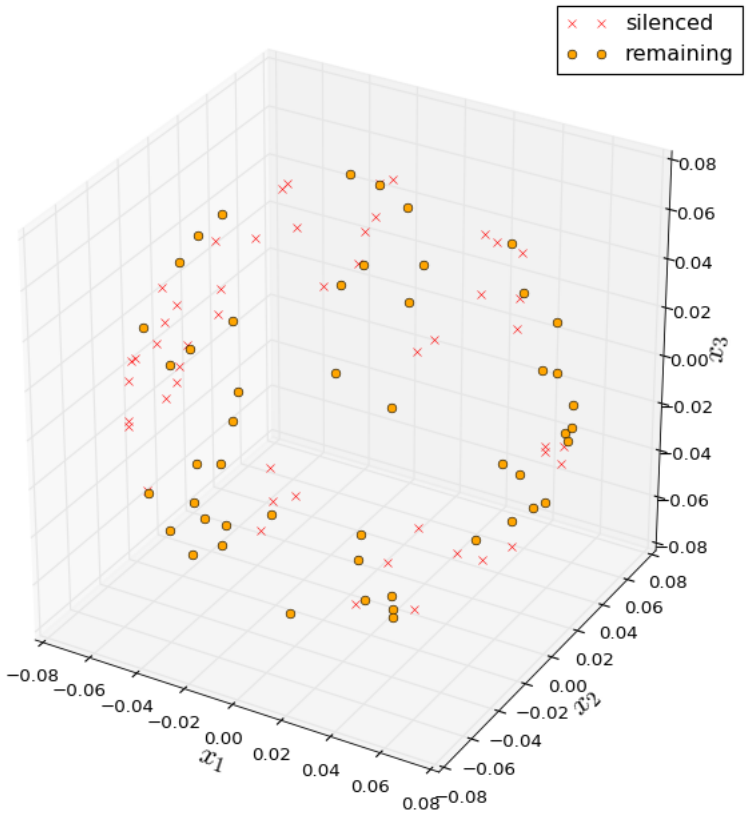
Appendix A

Simulation parameters

Table A.1: List of paramters used for the simulations of the figures throughout the third chapter (results)

Parameters	**Values**
Time step dt	$10^{-4}s$
Number of neurons N	100
L2 cost μ	10^{-5}
L1 cost ν	10^{-5}
Standard deviation of membrane noise σ_V	10^{-5}
Membrane leak λ_V	2
Decoder leak λ_d	2
Dimension of input J	3
Scaling factor α	0.4
Scaling factor β	0.2

Figure A.1: Overcomplete set of decoders. One hundred neurons span a three dimensional signal space. Each decoder is a vector that goes from the origin to the point drawn on the figure. Red crosses show neurons that are silenced during the inactivation. Orange dots show the remaining neurons that will have to compensate for this loss. We can see that the remaining neurons still cover the full space. Given that this set of remaining neurons is still overcomplete, the network could face even more drastic perturbation (four neurons minimum). But this would come at the cost of unrealistic firing rates.



Appendix B

Separation of Excitatory and Inhibitory units

In this chapter, we will shortly outline how networks with separate pools of neurons for excitation and for inhibition can be built. This section is not required to understand the first part of the results, it can therefore be skipped in a first reading.

Although the cells that naturally follow from the efficient coding objective are mixed, it is possible to separate these two components by adding some constraints and some new units. Simply forcing the recurrent connections to be strictly excitatory does not solve the question. Indeed, in that case, neurons with similar tuning would not be able to balance each other.

Therefore, on top of restraining the existing network to be excitatory, we have to introduce an additional population of M inhibitory neurons that will have to fill the remaining role, namely to balance the excitatory population. To do so, inhibitory neurons will have to encode the activity of the excitatory population. This new population will reset similarly tuned excitatory neurons each time an excitatory spike is emitted. By adding this extra population, we do not change the core architecture of the network, input is projected only on the excitatory neurons and output is also readout from them only.

The new architecture is represented in figure B.1 and can be expressed in a new set of separated voltage dynamics :

$$\frac{\partial V_n^E}{\partial t} = -\lambda_V V_n^E(t) + \mathbf{F}_n^E \mathbf{c}(t) + \mathbf{\Omega}_n^{EE}\mathbf{o}^E(t) + \mathbf{\Omega}_n^{EI}\mathbf{o}^I(t) + \mathbf{\Phi}_n^{EE}\mathbf{r}^E(t) + \mathbf{\Phi}_n^{EI}\mathbf{r}^I(t) + \sigma_V\eta(t),$$

$$\frac{\partial V_m^I}{\partial t} = -\lambda_V V_m^I(t) + \mathbf{\Omega}_m^{II}\mathbf{o}^I(t) + \mathbf{\Omega}_m^{IE}\mathbf{o}^E(t) + \sigma_V\eta(t),$$

where the elements of $\mathbf{\Omega_{EI}}$ and $\mathbf{\Omega_{II}}$ are assumed to be negative and the elements of $\mathbf{\Omega_{IE}}$ and $\mathbf{\Omega_{EE}}$ are assumed to be positive up to one exception : the self-connection weight of excitatory neurons is assumed to remain negative. Indeed this connection is

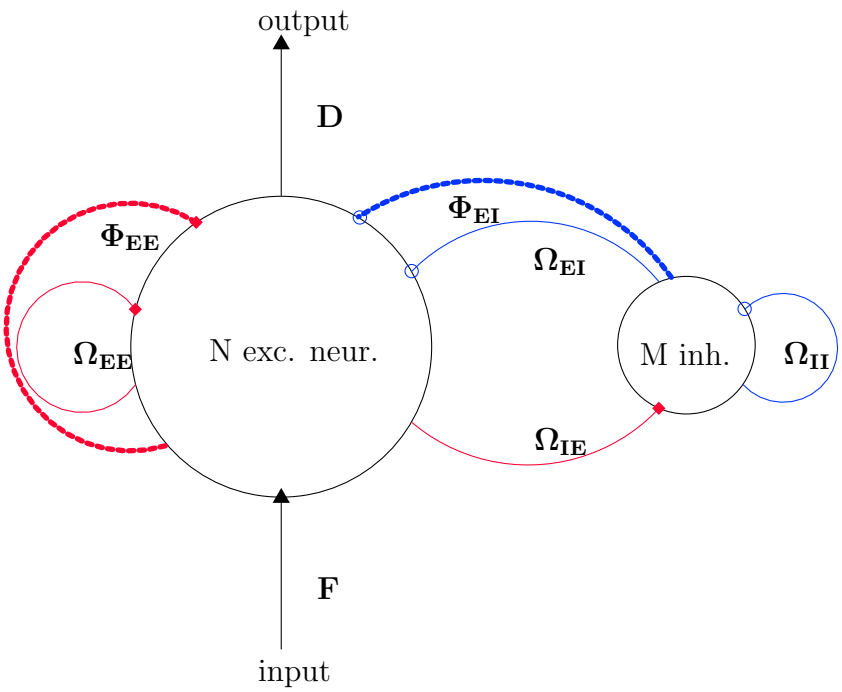


Figure B.1: EI network. Excitatory projections are shown in red with diamonds synapses, and inhibitory projections are shown in blue with open circles synapses. Fast recurrent connections are represented by continuous lines and slow connections are represented by dashed lines.

not an autapse, but the value of the neurons reset potential. For sake of simplicity, we will assume that the time constants are homogeneous between the two populations.

Notice that the feed-forward and decoding weights are not constraint to be positive. But this should not be a concern as it can be solved by working in the positive quadrant of the signal space only.

B.1 Derivation of the optimal connectivity

Given that the recurrent weights are directly related to the decoding weights, we can simply separate these decoders into a strictly positive and a strictly negative component. This gives : $\mathbf{D} = \mathbf{D}_+ - \mathbf{D}_-$. Further separating the recurrent connectivity, we get:

$$\begin{aligned}\mathbf{\Omega r} &= -(\mathbf{D}_+ - \mathbf{D}_-)^\top(\mathbf{D}_+ - \mathbf{D}_-)\mathbf{r} \\ &= (\mathbf{D}_-{}^\top\mathbf{D}_+ + \mathbf{D}_+{}^\top\mathbf{D}_-)\mathbf{r} - (\mathbf{D}_+{}^\top\mathbf{D}_+ + \mathbf{D}_-{}^\top\mathbf{D}_-)\mathbf{r}\end{aligned}$$

where the first member has strictly positive weights and the second one strictly negative weights. Therefore this second member can not be accounted for by excitatory neurons, this is where the new pool of inhibitory neurons comes into play. As described previously, we want these inhibitory neurons to track the activity of the

excitatory neurons. This mew readout objective be written :

$$\hat{\mathbf{r}}_E = \widetilde{\mathbf{D}}\mathbf{r}_I, \tag{B.1}$$

where $\mathbf{r}_E$ and $\mathbf{r}_I$ are the firing rate respectively of the excitatory neurons and of the inhibitory neurons, and $\widetilde{\mathbf{D}} \in \mathbb{R}^{N\times M}$ are the decoding weights of the inhibitor neurons.

Then the link to the mixed network is given by :

$$\mathbf{\Omega r} = \mathbf{\Omega}_{EE}\mathbf{r}_E - \mathbf{\Omega}_{EI}\widetilde{\mathbf{D}}\mathbf{r}_I,$$

And we now have an additional objective, which is simply a new version of the loss function discussed in the first section of this chapter. Which is:

$$l_I = \left\|\mathbf{r}_E - \widetilde{\mathbf{D}}\mathbf{r}_I\right\|_2^2 + \mathbf{C}(\mathbf{r_I}).$$

There is a strict hierarchy between the two objective functions, indeed the decoding from the excitatory population will be correct only if the inhibitory population satisfied its objective function in the first place.

From there it is then straightforward to add the slow connectivity. Following the same logic as before, we simply split $\mathbf{\Phi}$ into a positive and a negative part, respectively $\mathbf{\Phi}^+$ and $\mathbf{\Phi}^-$, such that the new slow connectivity is :

$$\mathbf{\Phi r} = \mathbf{\Phi}_{EE}\mathbf{r}_E - \mathbf{\Phi}_{EI}\mathbf{r}_I, \tag{B.2}$$

where the first member is implemented by the excitatory neurons and the second part by the inhibitory neurons.

B.2 Replication of the mixed network

To run this new network, we now need to decide on the decoders for the new population. As discussed previously, in general : $\widetilde{\mathbf{D}} = \arg\min_{\widetilde{\mathbf{D}}} \langle l_I \rangle$.

Let us start with the simplest case, each excitatory neuron has its corresponding inhibitor that is exactly tuned to track it. That is to say $\widetilde{\mathbf{D}} = \mathbf{I}$.

Figure B.2 shows it is possible to separate a mixed network into and EI network and yet obtain an equivalent behavior. Nevertheless to obtain this match it has been necessary to double the number time step for the inhibitory population. Indeed, all excitatory neurons need to receive the information about the error that has been corrected exactly when a spike is emitted. This temporary solution needs to be complemented by a more systematic investigation of the role of delays in the network.

Note also that it is possible to change the proportion of the inhibitory neurons by changing their decoders. Nevertheless this requires a clever choice of decoders because the task is to span a N dimensional space with only M neurons. It is possible to fulfil this objective when the rate of the excitatory population only explore a subspace of the activity space $\mathbb{R}^N$.

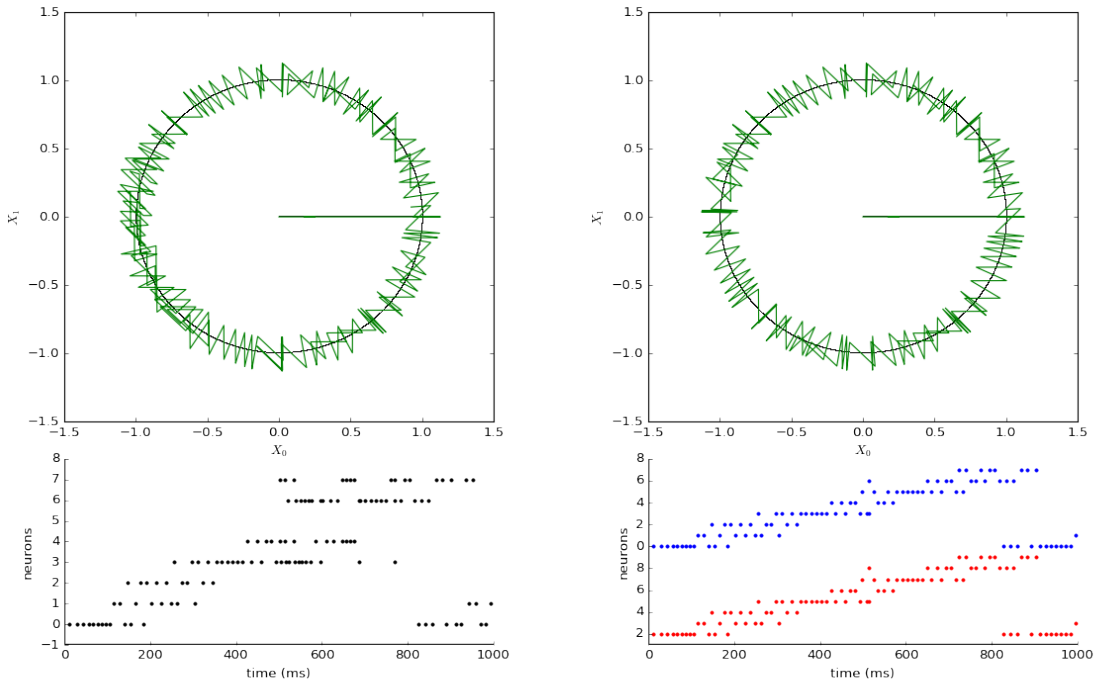


Figure B.2: Replication of the previous results in a network of excitatory and inhibitory neurons. On top of each panel, the input signal in black and its reconstruction in green. Bottom of each panel, raster plot. **Left panel** Mixed network **Right panel** EI network, in red excitatory units, and in blue inhibitory units.S

Appendix C

Pre-registration document

Predictive coding and robustness of representation
in recurrent spiking networks
March 15th 2016

Sketch of my research project CogMaster M2, defence in English in June

Supervisors : Sophie Denve (ENS, LNC, Group for Neural Theory, Paris)
Christian Machens (Champalimaud Center for the Unknown, Lisbon)

Reviewers suggestions : JP. Nadal, V. Hackim, R. Brette, S. Ostojic, M. Chalk

C.1 Background and rationale

The variability and heterogeneity of cortical activity is both puzzling and fascinating. Indeed it makes the question of how the brain implements cognitive functions even harder. The nervous system has to combine high accuracy of information representation, high computational complexity of processing, robustness to perturbation and constrained metabolic cost. In order to better understand these properties one can study the neural code and search for the underlying principles that give rise to the brain as we know it.

The theoretical framework developed by Sophie Denve, Christian Machens and colleagues suggests a recurrent network model of integrate-and-fire neurons that exhibits such qualities. This work relies on two main assumptions. The first one is that information encoded by the sensory system of interest can be decoded linearly. This means that after complex processing within one layer, a simple synaptic integration of the output spike trains is transmitted to the next processing stages. In this respect the properties of single neurons are temporary network solutions to an optimization problem. The second main assumption is that each action potential provides new

information, which implies that spiking dynamics are crucial. Given that neural responses are shaped by metabolic constraints, there are good reasons to think neurons tend to reduce information redundancy. Here the efficiency is achieved through predictive spiking neurons whose membrae potential track the difference between the input signal and its representation. This mechanism is implemented by fast and slow lateral connections and allows to track dynamical variables spike per spike.

Building upon the theory of both efficient coding and balanced networks, this approach accounts for the robustness and flexibility of the code. It also suggests that neurons collaborate to represent information and compute in highly recurrent networks. By doing so it provides arguments for sparse coding and it indicates that high variability of neural responses does not result from noise but from the degeneracy of the representation, as several patterns can code for the same variable. This framework gives importance to the precise spike timing and most importantly, it provides a functional explanation for the tight balance between excitatory and inhibitory potentials received by each neuron.

C.2 My project

The brain has an impressive ability to withstand neural damage. In order to better understand this ability I probe the predictions of the aforementioned theoretical framework on real data, the variable I am interested in is the robustness of the neural representation to external perturbation. In doing so I am trying to provide an analytical explanation for a bewildering experimental finding.

The data was collected in Karel Svoboda lab (Janelia farm) and consists of mice Anterior Lateral Motor (ALM) cortex recordings, a region that is known to play a role in sensory guided movements. The study asks how neurons within ALM drive movements, that is to say how preparatory activity translates in actual motor commands. Data was acquired during a whisker-based object location discrimination task that is composed first of sampling period, then of a delay period that ends with an auditory go cue, and finally of a response period. Depending on the location of the pole they are presented with, animals are expected to lick right or left and do so correctively on about 80% of the trials. Recordings in this region show that neurons have diverse selectivity, some respond in advance of movements to the contralateral side and others respond in advance of movements to the ipsilateral side. Unilateral photostimulation of channelrhodopsin-2 in GABAergic interneurons in ALM during the delay period induces an ipsilateral bias in response. That is to say, disruption of ALM on one side particularly affects contralateral movements, while neurons selective for each side are present in about equal proportions in both hemispheres.

To tackle that disconnect I write Python simulations of the recordings previously described and investigate the effects of *in silico* equivalents for optogenetic inactivation. This allows me to study the evolution of the cellular dynamics and to describe

the performance changes of the agent as a function of the inactivation. The aim is to characterize the compensatory activity of the intact part of the network : remaining neurons change their firing patterns in various ways to compensate, some firing more and others less depending on their selectivity. I can then compare the read-out of this activity to the behavior of the animal. I try to reproduce the Tuning Curves of the recorded neurons, and more importantly, after silencing units, I try to reproduce the changes in the tuning curves of the leftover kernels. These simulations should reproduce the mixed selectivity and heterogeneous activity described *in vivo*. By simulating a network with two sub-populations corresponding to each hemisphere, we hope to reproduce the ipsilateral bias in response induced by a unilateral perturbation.

This work can be expanded in various directions, a first one would be to study the learning mechanisms that regulate the aforementioned tight balance, a second one would be to investigate the sparsity of the connectivity structure of the network. Through this project we hope to better understand sensory decision making and persistent activity mechanism of working memory.

Bibliography

[1] Valentina Emiliani, Adam E Cohen, Karl Deisseroth, and Michael Häusser. All-optical interrogation of neural circuits. *The Journal of Neuroscience*, 35(41): 13917–13926, 2015.

[2] Daniel H O'Connor, Daniel Huber, and Karel Svoboda. Reverse engineering the mouse brain. *Nature*, 461(7266):923–929, 2009.

[3] Peter Sterling and Simon Laughlin. *Principles of neural design*. MIT Press, 2015.

[4] Zengcai V Guo, Nuo Li, Daniel Huber, Eran Ophir, Diego Gutnisky, Jonathan T Ting, Guoping Feng, and Karel Svoboda. Flow of cortical activity underlying a tactile decision in mice. *Neuron*, 81(1):179–194, 2014. URL `https://crcns.org/data-sets/motor-cortex/alm-1`.

[5] Nuo Li, Tsai-Wen Chen, Zengcai V Guo, Charles R Gerfen, and Karel Svoboda. A motor cortex circuit for motor planning and movement. *Nature*, 519(7541): 51–56, 2015. URL `https://crcns.org/data-sets/motor-cortex/alm-2`.

[6] Nuo Li, Kayvon Daie, Karel Svoboda, and Shaul Druckmann. Robust neuronal dynamics in premotor cortex during motor planning. *Nature*, 532(7600):459–464, 2016.

[7] Ranulfo Romo and Emilio Salinas. Flutter discrimination: neural codes, perception, memory and decision making. *Nature Reviews Neuroscience*, 4(3):203–218, 2003.

[8] M Yu Byron. Neuroscience: Fault tolerance in the brain. *Nature*, 2016.

[9] Sophie Denève and Christian K Machens. Efficient codes and balanced networks. *Nature neuroscience*, 19(3):375–382, 2016.

[10] Bruno A Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[11] Michael N Shadlen and William T Newsome. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *The Journal of neuroscience*, 18(10):3870–3896, 1998.

[12] David Barrett, Sophie Deneve, and Christian Machens. Optimal compensation for neuron death. *bioRxiv*, 2015.

[13] Martin Boerlin, Christian K Machens, and Sophie Denève. Predictive coding of dynamical variables in balanced spiking networks. *PLoS Comput Biol*, 9(11): e1003258, 2013.

[14] Wieland Brendel, Ralph Bourdoukan, Pietro Vertechi, David Barrett, Christianand Machens, and Sophie Denève. Optimal spike-timing-dependent plasticity for maximizing representational efficiency. in press, 2016.

[15] Donald R Humphrey, EM Schmidt, and WD Thompson. Predicting measures of motor performance from multiple cortical spike trains. *Science*, 170(3959): 758–762, 1970.

[16] Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[17] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8): 2554–2558, 1982.

[18] John Hertz, Anders Krogh, and Richard G Palmer. *Introduction to the theory of neural computation*, volume 1. Basic Books, 1991.

[19] Haim Sompolinsky and I Kanter. Temporal association in asymmetric neural networks. *Physical Review Letters*, 57(22):2861, 1986.

[20] Panayiota Poirazi, Terrence Brannon, and Bartlett W Mel. Pyramidal neuron as two-layer neural network. *Neuron*, 37(6):989–999, 2003.

[21] Matthew Chalk, Boris Gutkin, and Sophie Denève. Neural oscillations as a signature of efficient coding in the presence of synaptic delays. *bioRxiv*, page 034736, 2015.

[22] Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Ranulfo Romo, Xue-Lian Qi, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *eLife*, 5:e10989, 2016.

[23] Sophie Denève and Matthew Chalk. Efficiency turns the table on neural encoding, decoding and noise. in press, 2016.