Bachelor Thesis

Fading Memory in Balanced Neural Networks

Frankfurt Institute for Advanced Studies

Johann Wolfgang Goethe University
Frankfurt am Main
Faculty of Physics

Author
Sebastian Bielfeldt

First Reviewer
Prof. Dr. Jochen Triesch
Frankfurt Institute for Advanced Studies
Goethe University
Frankfurt am Main

Second Reviewer
Prof. Dr. Matthias Kaschube
Frankfurt Institute for Advanced Studies
Goethe University
Frankfurt am Main

Frankfurt am Main, 2. May 2020

Fading Memory in Balanced Neural Networks

Sebastian Bielfeldt

2. May 2020

Abstract

The focus of this Bachelor Thesis is a model of cortical neurons akin to the balanced network developed by Vreeswijk and Sompolinsky (1998). First, some of their findings are reproduced such as a broad firing rate distribution and a linear relationship between external stimulus and network activity. Furthermore, the correlation between the firing rate and the coefficient of variation of interspike intervals is investigated. Also, local and temporal correlations are shown to form for a less than random update order.
Next, the capacity to store information in this type of network is explored. Within a few time steps, the predictive capacity of the network is found to be negligible. The share of correct predictions with a short delay can be improved by slightly increasing the weights of network recurrent excitatory inputs. However, the chaotic nature of the network is shown to prohibit longer storage of information. Even the most similar states are shown to lose any proximity, as measured by the Hamming distance, within a short period of time.

Acknowledgments

First and foremost, I would like to thank Prof. Jochen Triesch and Felix Hoffmann for the outstanding advice and support. Also, I thank Eunhyung **C**ho, Vladimir Denk, *L*eon Goertz, Christian and Lennart Harder, Felix Hoffman and Lars Stoltenberg for your all-seeing eyes and piercing questions.

Contents

1 Introduction **1**
1.1 A Short Anatomy of the Cortex and Neural Tissue 1
1.2 Modeling Cortical Neurons 2
1.3 Information in Neural Networks 3

2 Model and Methods **5**
2.1 Connection Matrix 5
2.2 External Input and Thresholds 6
2.3 Activation Function and Update Mechanism 7
2.4 Measures 7
2.5 The Balanced State 8
2.6 Input Parameters 10
2.7 Output Parameters 11

3 Results and Discussion **13**
3.1 Behavior of the Single and Mean Neuron 13
3.2 Comparison with Rigid Update Order 16
3.3 Differences Between Neurons 18
3.4 Memory Capabilities 21
3.5 Chaotic Behavior 23

4 Conclusion **26**

A Appendix **28**
A.1 Additional Figures 28
A.2 Accessing the Graphical Interface 31

B Bibliography **34**

1 Introduction

Only about 200 years ago, the head of a recently deceased Joseph Haydn was removed from his grave to investigate what enabled him to become a musical genius. To gain insights into the intellectual capabilities of the deceased, the size and shape of the skull were measured. While those methods sound antiquated at best, the desire to decode the inner workings of the human brain is still shared by many.

The inner workings of memory is an especially interesting issue. With finite storage capabilities, the brain has to evaluate which information is valuable enough to keep and which to discard. As a living organ, the brain experiences constant change. Most neurons are connected to thousands of other neurons and are involved in a large range of different processes. Due to its high level of complexity and to it being arguably the most complex organ of the human body, the brain still has not been fully understood.

1.1 A Short Anatomy of the Cortex and Neural Tissue

We will start with a short description of the biological basics underlying the model. The human brain is the central organ of the nervous system. It consists of the cerebrum, the brain stem and the cerebellum. Of those, the cerebrum is the largest, taking up roughly two-thirds of the brain's mass (Britannica, 2020). It is split into a left and a right hemisphere. Each side controls the respective other side of the body.

On the outer layers of the cerebrum lies the cortex. It plays an important role in attention, language, perception, thought and memory (Swenson, 2006; Gilbert and Sigman, 2007). Even though the cortex is only a couple of millimeters thick (in the human brain), it represents half of the mass of the cerebrum or two-fifths of the whole brain's mass (Saladin, 2011). The cortex is structured in folds, which increase its surface area. Larger folds signify the borders between lobes. The cortex contains about 15 to 20 billion neurons (Saladin, 2011; Herculano-Houiel, 2009) of the 100 billion neurons in the brain (Herculano-Houiel, 2009).

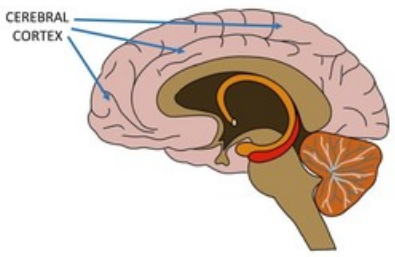


Figure 1.1: The brain and position of the cerebral cortex (Dingman, 2014)

Neurons are cells with specialized features. A neuron consists of a cell body (soma), dendrites and an axon. Most neurons receive signals via the dendrite and transmit signals via the axon. The dendrites branch out widely in an area of hundreds of micrometers. This means that a neuron can receive inputs from a very large quantity of other neurons. In the human brain, the axon can reach lengths of as much as a meter, which allows connections between neurons that are far apart(Herculano-Houiel, 2009). At the tip, an Axon can then transmit an electric signal via a synapse to another neuron's dendrite. If the voltage in a neuron's soma reaches a certain threshold over a short time period, it generates an all-or-nothing electrochemical pulse (or action potential). This travels through the axon and activates a synapse. The synaptic signal can be either excitatory or inhibitory. This means, it can either lead to an increase or decrease, respectively, in the net voltage reaching the next neuron's soma. Next, the research leading to the model at hand is presented.

1.2 Modeling Cortical Neurons

Since Burns and Webb (1976) investigated spiking behavior in live cats, our understanding of cortical neurons has advanced over the last decades. In the early 90s, several research groups (such as Abeles (1991) and Softky and Koch (1993)) have found evidence of highly irregular spiking patterns of cortical neurons over time. The time that passes between spikes, so-called interspike intervals (ISI), resembles a Poisson point process with a short refractory period (Abeles, 1991). A Poisson point process describes a series of events that occur over a continuous time frame at a constant average rate. Additionally, each random event is independent and identically distributed. Accordingly, in an exponential distribution with a refractory period, spikes do not influence one another, apart from a (short) period right after a spike in which a second spike is less likely. However, measurements showed isolated neurons fire predictably and regularly when injected with a constant current. Hence, fluctuations measured in active cortices happen due to changes in input (Holt et al., 1996; Mainen and Sejnowski, 1995).

As stated above, cortical neurons tend to be connected to thousands of other neurons. According to the law of large numbers, independent stochastic processes of this magnitude should only show very small fluctuations. While some correlations have been shown to exist between neurons, the extent of these correlations can only account for a small fraction of the aforementioned fluctuations (Abeles, 1991; Gray et al., 1989). Several mechanisms have been proposed which enhance a neuron's sensitivity to small changes in the incoming current.

Bell et al. (1995) were one of the first to explore the idea of "balancing" the input via a model of a neuron. They achieved this by having the neuron adjust the level from randomly firing excitatory and inhibitory input(i.e.

increasing or decreasing the activation level, respectively). With the right calibration, their model shows irregular output in single output neurons and a reversion to a stable, balanced state.

Tsodyks and Sejnowski (1995) and Vreeswijk and Sompolinsky (1996) built on this by creating models of networks consisting solely of balanced neurons, with some representing excitatory and some inhibitory neurons. Tsodyks and Sejnowski (1995) chose a model with continuous current neurons (so-called integrate-and-fire neurons). At each step of the simulation, the current is increased or decreased according to the input function. When the current surpasses a certain threshold, a spike is registered and the current is reduced to a lower value. To maintain the behavior of the network for different amounts of neuronal connections (and thus the different intensity of input), they altered the spike thresholds. They were able to show that an entire network of balanced neurons also demonstrates irregular spiking patterns. Furthermore, their model is capable of rapidly switching between different states, in other words, it responds quickly to a change in stimulus.

Vreeswijk and Sompolinsky (1996) instead work with a network of binary neurons and let the sum of all excitatory and inhibitory inputs oscillate tightly around the threshold of the activation function. They scale their system by adjusting the weight assigned to each neuron. Neurons are connected randomly with only the mean amount of connections specified (random connectivity) and are updated asynchronously (i.e. one after the other, randomly chosen from the pool of all neurons). Due to relatively high connectivity, cumulative input from both excitatory and inhibitory neurons are large and fluctuations small. The net input, however, displays relatively large fluctuations. They show their model to fire irregularly and are able to reach this state with a wide range of parameters. Unlike previous models, their neurons firing patterns are chaotic and spatial correlations vanish in the limit of an infinitely large network. This makes it a likely assumption that the brain receives balanced input. When the term 'balanced network' is mentioned, it from here on refers to the network described by Vreeswijk and Sompolinsky (1998).

Below, we will reproduce several of the findings made by Vreeswijk and Sompolinsky (1998). We also look at the impact of the update order on spiking patterns.

1.3 Information in Neural Networks

Some neural networks can act as memory devices. Memory can be embedded in the synaptic connections of a network. An external input can then recreate the corresponding part in the memory (Aviel et al., 2005). This is called the associative memory model, as the external input determines which memory is recalled.

However, it is not possible to embed memory in the connections of a bal-

anced network without disregarding the random connectivity assumption in balanced networks. The introduction of some kind of order to the otherwise random connectivity matrix leads to a new critical point at which the asynchronous model becomes unstable (Aviel et al., 2003).

Reservoir Computing does not supervise interconnection weights but instead relies upon certain generic properties. A memoryless supervised readout is used to collect information from the network. One of the first applications of reservoir computing in the field of theoretical neuroscience was achieved through liquid state machines (LSM) developed by Maass and Markram (2004). The LSM framework describes a spiking neural network with a large amount of randomly connected neurons that receive time-varying external input. Through the inherent recurrent nature of the connections, external input leads to spatiotemporal activation patterns in the network. Those patterns can then be read out and used to extract information from the system. Unlike in other models, the information does not have to be hard-coded in the network connections. (Mainen and Sejnowski, 1995).

A balanced network is chaotic and, over time, information is lost. Therefore, it will not be able to store information in the long run. Hence, we will explore how well and for how long a balanced network can store information.

2 Model and Methods

In the following, we will define a balanced network in accordance with Vreeswijk and Sompolinsky (1998). As described above, the system consists of populations of excitatory and inhibitory neurons. Unless otherwise specified, all neurons of a given type are interchangeable and share all characteristics. Besides the input from other neurons within the model, all neurons receive external input. We refer to the neurons within the system also as internal or recurrent neurons, to differentiate between them and the external "neurons".

Neurons are always in one of two binary states, i.e. the one and the zero state. The one state will also be addressed as the active or excited state while the zero state is also referred to as the inactive, inhibited or passive state. Apart from those two states, spikes are recorded. A spike occurs if the neuron is active in its current state and was inactive in the previous state. As we will investigate in Chapter 3, the spikes of this model have properties similar to spikes in cortical neurons.

2.1 Connection Matrix

The connection matrix J only distinguishes whether a connection exists between two neurons and the type of both the outgoing and receiving neuron. Only the average number of neural connections per population K is specified. For this reason, one neuron has on average K inhibitory and K excitatory inputs. This is a sparsely connected network, that is to say, K is a lot smaller than the total number of neurons N. As only the average amount of connections is determined, some neurons can have more connections than others. Also because of that, it is possible to have more connections with one population type than the other one. The variation in the two connection numbers leads to differences in spiking patterns as some neurons have a higher (lower) share of input from excitatory neurons leading to a higher (lower) firing rate.

The synaptic weight is thus calculated based on how many connections K the average neuron has and a factor J_{kl} depending on the type of neurons involved as we show in the following:

$$J_{kl}^{ij} = \begin{cases} \frac{J_{kl}}{\sqrt{K}} & \text{if connected,} \\ 0 & \text{otherwise} \end{cases} . \tag{2.1}$$

Here, k and l indicate the population type (represented either by 1,2 or E,I) while i,j show the position inside the respective population. Throughout this thesis, k as well as l and i will be used in this way. The average input from neurons within the model is equal to K times the respective synaptic weight, yielding the following:

$$K \cdot J_{kl}^{ij} = \sqrt{K} J_{kl}. \tag{2.2}$$

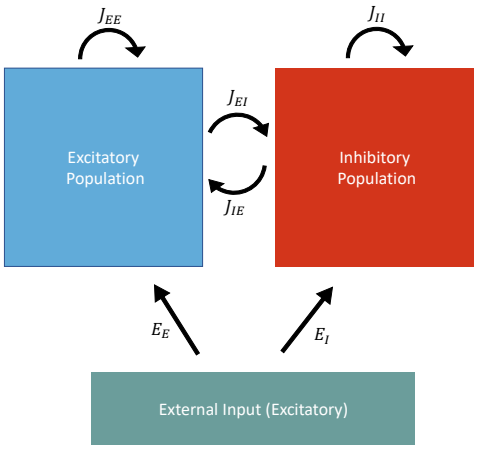


Figure 2.1: Relations between population of neurons

This shows a proportional relationship between the cumulative inputs into a neuron and $\sqrt{K}$ on average.

2.2 External Input and Thresholds

The external input to the neuron mimics a further population of neurons. Therefore, we define the factors E_k for the external input. Similar in size to the synaptic weights J_{kl}, E_k also differ for excitatory and inhibitory neurons. To E_k and J_{kl}, we will collectively refer to as weight factors. The external mean activation rate m_0 is the share of virtual neurons, which are active at a given point in time. The possible values of m_0 range from 0 to 1 and are the same for all neurons. To maintain the same strength ratio of internal and external neurons, we multiply with the root of the connection number $\sqrt{K}$. Otherwise, external input u_k^0 would drown out all information from within the system at some scales and, at the same time, not have any perceivable impact at other scales. With a fixed ratio between external and recurrent inputs, the model maintains consistent behavior for sufficiently large K. Thus, the resulting value for the external input u_k^0 is of the following form:

$$u_k^0 = E_k m_0 \sqrt{K}. \tag{2.3}$$

The thresholds θ_k have to be small compared to $\sqrt{K}$, the reasoning behind this is laid out in Chapter 2.5. A balanced network allows for the thresholds of neurons within the same population to vary. In this thesis, however, the same θ_k is applied to each neuron of a given population.

2.3 Activation Function and Update Mechanism

Now, we look at how a neuron changes its state. As stated above, all neurons are binary, or, put differently, the value of a given neuron σ_k^i is either one or zero. Here, k indicates the population type and i which position the neuron has within this population. The value of the neuron is calculated with the input of the other neurons as well as external input and subtracting the threshold, as can be seen below:

$$u_k^i = u_k^0 + \sum_{l=1}^{2} \sum_{j=0}^{N_l} J_{lk} \sigma_k^i - \theta_k. \tag{2.4}$$

The Heaviside function of the result is calculated to yield the new binary value of state σ_k^i:

$$\sigma_k^i = \Theta(u_k^i). \tag{2.5}$$

Next, we turn to how the neurons interact with each other. The order of updates is asynchronous, meaning that one neuron at a time is changed. The process of choosing the next neuron is twofold. First, the population type is determined with a random process and then, a neuron from within this population is chosen. This is important, as now each population can be updated at its own speed. Hence, there are two mean update times τ_E and τ_I for the excitatory and inhibitory part of the population respectively. We will measure time t in units of τ_E. This means, effectively, τ_E will be fixed to 1. With only τ_I being variable, this will be our measure of the relative update time. The neurons within a population are chosen randomly.

2.4 Measures

In the following, some measures necessary in order to understand and evaluate the system are introduced. The activation level $m_k(t)$ of a given population describes which share of neurons is active at time t. It can be calculated for any point in time t as:

$$m_k(t) = \langle \sigma_k^i(t) \rangle_i = \frac{1}{N_k} \sum_{i=1}^{N_k} \sigma_k^i(t). \tag{2.6}$$

The vector $\vec{\sigma}_k(t)$ contains the binary states of population k (with $k \in \{E, I\}$), N_k represents the size of the population and i enumerates the individual members of the given population. The activation rate is defined as the mean of the activation level over the whole duration of the simulation. It exists for both the whole population as m_k (the mean activation rate) as well as for individual neurons as m_k^i.

Also of interest is the firing rate r_k^i which is a measure of how likely a neuron is to spike in a given time step. The firing rate is a very important

property of the balanced network because, as we will show, the spikes of this model resemble those measured in the brain. For a balanced network, a spike is defined as the event of a neuron switching from the passive to the active state. It is calculated by counting spikes over a certain period of time and determining the average number of spikes per unit of time. It is smaller than the activation rate because, after a spike, a neuron first has to enter the inactive state to be able to spike again. The probability, that a neuron, which is in the active state, switches to the inactive in the next update cycle, is given by inverse activation rate $1 - m_k^i$. Following the same logic, the probability for the deactivated neuron to jump back to the active state is described by m_k^i. Most neurons have low activation rates and stay in the active state only for one or two update cycles. For the same reason, they remain for a long time in the passive state. For low activation rates, the firing rate can be approximated by combining the probabilities of those two steps in a single random event:

$$r_k \approx \frac{m_k(1 - m_k)}{\tau_k} \tag{2.7}$$

which is also taking into account the mean time to update τ_k.

In this case, an exponential distribution describes the time intervals between spikes. The density function of the exponential distribution is given by:

$$f(t, \gamma) = \begin{cases} \gamma e^{-\gamma t} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{2.8}$$

with time t and the rate parameter γ. For an exponential distribution, the ratio of the standard deviation to the mean of the intervals equals one. This ratio is also referred to as the coefficient of variation (CV).

When m_k^i becomes too large, both jumps have to be evaluated independently. This means the interspike intervals cannot be described by a single exponential distribution anymore but rather the addition of two different ones. This means also, that the CV is not constant anymore, but approaches another value for each activation rate m_k^i. We will use the CV to see whether neurons interspike intervals are distributed according to one of those two processes. One should also note, that spikes for very high and very low activation rates behave similarly. In fact, the probabilities of active to passive and passive to active jumps are reversed for $m_k^i > 0.5$, leading to the same spike patterns for the case of $1 - m_k^i$ and m_k^i.

2.5 The Balanced State

In this section, we examine the size of the weight factors. First, we take another glimpse at the input a neuron receives. The mean input yields:

$$u_k = (E_k m_0 + J_{Ek} m_E + J_{Ik} m_I)\sqrt{K} - \theta_k \tag{2.9}$$

when applying Equations 2.2 and 2.3 on Equation 2.4. The model does not allow for the mean activation m_k in the stable state to be at either extreme for any of the two populations. Thus, values of $u_k(t)$ are neither entirely positive or negative. In turn, the mean over time, u_k cannot be $\pm\infty$ and rather has to remain finite (as is θ_K). Only this way the name giving "balance" of the model can be maintained. Next, we transform the matrix and use this information:

$$E_k m_0 + J_{Ek} m_E + J_{Ik} m_I = \frac{(\theta_k - u_k)}{\sqrt{K}} = \mathscr{O}\left(\frac{1}{\sqrt{K}}\right). \tag{2.10}$$

In the large K limit, the right-hand side of Equation 2.10 has to be 0:

$$E_k m_0 + J_{Ek} m_E + J_{Ik} m_I = 0. \tag{2.11}$$

The relation of the six weight factors (three for each, excitatory and inhibitory population) are now completely independent of the connection number K. The following predictions will hold for as long as $\sqrt{K}$ remains large compared to both u_k and θ_k.

With six parameters to dispose of and two equations (for $k = E$ and $k = I$), there are too many degrees of freedom. The relative information between the three inputs to a given population (J_{Ek}, J_{Ik} and E_k for $k \in \{E, I\}$) is most relevant. Thus, we set the factors of input originating from excitatory neurons (i.e. J_{EE} and J_{EI}) to 1 and are consequently still able to fully control the composition of inputs. The sign of the connection weights originating from inhibitory neurons J_{Ik} is negative. To avoid confusion, we, therefore, use the absolute value and show the actual negative sign. With this, we can rewrite Equation 2.11 to show the mean network activity for excitatory

$$m_E = \frac{|J_{II}|E_E - |J_{IE}|E_I}{|J_{IE}| - |J_{II}|} m_0 \tag{2.12}$$

and inhibitory neurons

$$m_I = \frac{E_E - E_I}{|J_{IE}| - |J_{II}|} m_0. \tag{2.13}$$

The fractions have to be positive as negative activation levels $m_k(t)$ cannot exist. This means one of the two following cases has to hold, either

$$\frac{E_E}{E_I} > \left|\frac{J_{IE}}{J_{II}}\right| > 1 \tag{2.14}$$

or

$$\frac{E_E}{E_I} < \left|\frac{J_{IE}}{J_{II}}\right| < 1. \tag{2.15}$$

Now, the behavior outside of the long term stable state will be investigated. Here too, the model should not allow for one of the two activation levels m_k to permanently remain in either complete activation or inhibition. This would be achieved in the case of $m_k = 1$ and u_k positive, or $m_k = 0$ and u_k negative. Indeed, Equation 2.15 allows for m_E to be 0 while at the same time u_k remains negative, leading to a solution of m_k being 1 or 0. Therefore, the first option must hold.

On top of that, if both $|J_{Ik}|$ are smaller than their respective J_{Ek}, there is a long term solution for u_k with $m_E = m_I = 1$ and $m_0 = 0$:

$$u_k = (J_{Ek}m_E - |J_{Ik}|m_I)\sqrt{K} > 0. \tag{2.16}$$

Consequently, as we fixed J_{Ek} to 1, it should hold that $J_{Ik} > 1$.

To summarize, as long as the following two conditions are met:

$$|J_{IE}| > 1, \tag{2.17}$$

and

$$\frac{E_E}{E_I} > \left|\frac{J_{IE}}{J_{II}}\right| > 1 \tag{2.18}$$

the system will regulate itself and move towards a balanced state.

As can be seen in Equations 2.12 and 2.13, the balanced state depends on the factors of the weight matrix (including E_k) as well as m_0. So, changes to those necessarily also shift the stable state. We will look At this in Chapter 3.4.

2.6 Input Parameters

In this section, we will present the parameter values used throughout this thesis. The size parameter N refers to the size of one population and the connection number K refers to the average number of connections one neuron has to each population. The size of both N and K is kept constant throughout all the tests presented in this work. As no differing values in-between populations are introduced, subscripts were not added. Time is measured in units of τ_E or the average amount of cycles it takes to update all excitatory units once. The mean external activation m_0 is constant for the most part. However, in Chapter 3.4 we also look at a case in which m_0 randomly changes back and forth between two different values. There, we also vary the values of J_{EE} and E_E. All other parameters are the same throughout this thesis. Where possible and appropriate, we choose the same parameters as Vreeswijk and Sompolinsky (1998) did for their model. This allows for a direct comparison of our results with those of the paper. The parameters are listed in Table 1.

The update order is random within a population and the choice between the populations depends on the ratio between τ_E and τ_I. However, we also take a look at more deterministic update orders.

N	10,000
K	1,000
m_0	0.1*
t	200*

Effect on:	J_{Ek}	J_{Ik}	E_k	θ_k	τ_k
Excitatory E	1*	-2.0	1.0*	1	1
Inhibitory I	1	-1.8	0.7	0.7	0.9

Table 1: Paramter Overview
Left-hand side shows all parameters for the whole simulation. Right-hand side shows all parameters which are different for each population.
** for parameters that have different values in some figures.*
J_{lk} synaptic weights (first subscript shows originating population, second target population), E_k factor of external input weight, θ_k threshold, τ average time of updates in a time step, K average number of connections per population, N size per population, m_0 external excitation rate, t time steps of simulation (times an average excitatory neuron has been updated)

2.7 Output Parameters

In this implementation, we extract four arrays containing information from the system. Firstly, we extract the timestamps, when a given neuron is activated. The timestamp is calculated on the basis of the average update times of excitatory neurons. In the simulations in Chapter 3, we round the timestamps of events of spiking to integers and describe them in discrete time steps that are of size 1 (with respect to the normalized time τ_E). The cumulative number of updates of excitatory neurons is divided by the number of excitatory neurons. Effectively, this is an array with the amount of all neurons N on the first axis and the individual events on the second axis.

Since most m_k^i are a lot smaller than 0.5, this way of saving information is more memory efficient than an array of size $N \times$ time t. Another reason for this method is that it is possible, for a neuron is activated twice in one time step due to the randomized update order. At the same time, there is no concrete information on whether a cell was updated at all during a given time step because the inactive states are not recorded.

We also record the neuronal firing separately. All events contained within the measure of neuronal firing can be found in the measure of neuronal activation as well. However, as explained in the definition of the firing rate, if the neuron was already active, it cannot fire. Thus, in this case, no timestamp is recorded.

For some plots, more information on the individual neuron level is needed. Thus, for a (customizable) subset of neurons, the following data points are recorded:

- cumulative excitatory input
- cumulative inhibitory input
- net input (excluding the threshold)

- threshold
- internal excitatory input (or short, excitatory input)
- external excitatory input (or short, external input)

The cumulative excitatory input is equal to the sum of the internal and external excitatory input. Lastly, changes in the mean activation of the external input m_0 are recorded. For more information about the code, which can be visualized with a graphical interface, see Appendix, Chapter A.2.

3 Results and Discussion

With the parameters defined in Chapter 2.6, we will begin reproducing some of the results published by Vreeswijk and Sompolinsky (1998), such as the behavior of a single neuron, its interspike times and the networks firing rates. Then we move to some other questions, such as the effects of a different update order, the distribution coefficients of variation and chaotic state of the system. Lastly, we alter the setup, change the excitatory input and try to predict past inputs.

3.1 Behavior of the Single and Mean Neuron

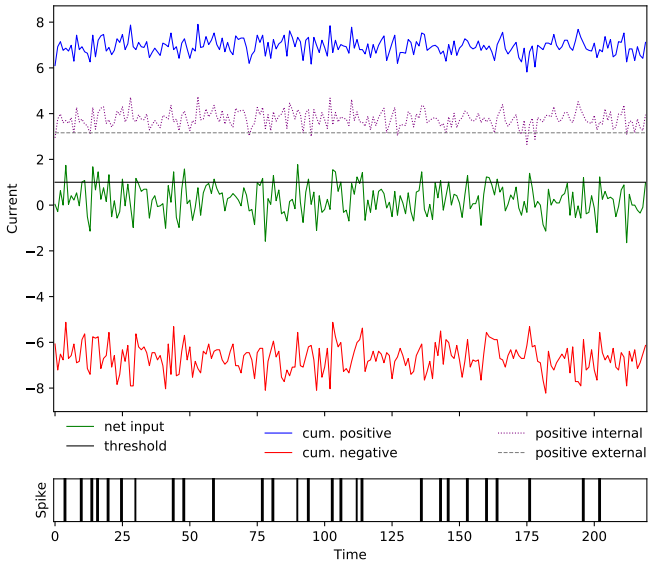
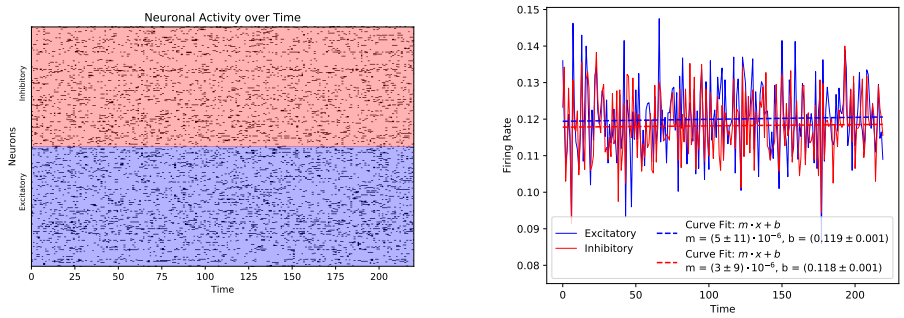


Figure 3.1: The temporal structure of all input streams of an excitatory neuron. Displaying the balance of excitation and inhibition in neurons. In the upper part, the inputs are displayed. Dashed line: input from external neurons. Blue dotted line: input from internal excitatory input to the neuron. Solid blue line: input from all excitatory (i.e. positive) inputs to the neuron. Red line: Inhibitory (i.e. negative) input to the neuron. Solid black line: Threshold for excitation. Solid green line: the sum of all excitatory and inhibitory (i.e. net) inputs to the neuron. In the lower part, the output is displayed. When the total sum crosses the threshold, the neuron is activated and a spike is recorded (in black).
Time in units of τ_E and current in units of (excitatory) thresholds θ_E.

First, we will look at a single neuron and see how the balance looks in practice. An example is displayed in Figure 3.1. In the upper part,

we see all the relevant inputs to the neuron. Both cum. excitatory and inhibitory inputs are large compared to the threshold. They also scale with the connectivity ($\propto \sqrt{K}$), as we discussed in Chapter 2. The external input, as well as the threshold, are both constant. Both the excitatory and the inhibitory input are large compared to the threshold, as they should be. Their values seem to fluctuate in a similar way compared to their mean value. As we will see soon in Figure 3.2b, the mean activation levels and the range in which fluctuations occur are similar for both populations. The largest difference between the two populations is the connection weight J_{lk} leading to different levels of mean input and the same level of fluctuation adjusted for the mean. The update mechanism works largely the same for both neuron populations. The net input fluctuates with absolute values comparable to the inhibitory input, but its mean is a lot smaller (in absolute terms). This leads to very a high level of fluctuations in comparison to its mean. Whenever the net input surpasses the threshold the neuron is in the active state, which is displayed in the lower part. The spikes do not form an instantly apparent pattern. Small changes in fluctuations of the overall neuron activity lead to irregular spikes, as is the case in cortical neurons.



(a) Overview of Activity Levels

(b) Mean Activation over Time

Figure 3.2: On the left-hand side, there is an overview over neural activity of the whole network over time. Every point on the y-axis represents one neuron with excitatory neurons in the lower half and inhibitory neurons in the upper half. The right-hand side shows the mean activation rate over time together with a linear fit.

Now, we will look at the behavior of the system as a whole, as shown in Figure 3.2. The figure on the left displays the activation status of all neurons over time (similar to the lower part of Figure 3.1). This is meant to give an overview of how the system behaves. This illustrates that there is little that differentiates a given time step qualitatively or, more succinctly, there is no discernible vertical correlation. One can, however, make out some lines showing neurons active for long time periods. This will be investigated more in detail in Chapter 3.3. There seems to be no apparent structure or

synchronization between different neurons' spiking behavior. In the figure on the right, the information of (a) is condensed into the mean activation of the system over time. Both mean inhibitory and mean excitatory activation fluctuate heavily in a range of several percentage points. As the linear regression shows, the slope of either trendline is only negligibly different from 0. This shows that the average of the mean activation does not change over time (for unchanging external input). Instead, the mean activation level of both populations quickly settles into a relatively stable state. There is little difference between the mean activation levels for excitatory and inhibitory neurons and both rise and fall at the same time steps. The mean activation of excitatory neurons is higher by a small but significant amount compared to the inhibitory neurons. This is due to the parameters chosen and if activity levels were switched no qualitative difference would emerge in the system.

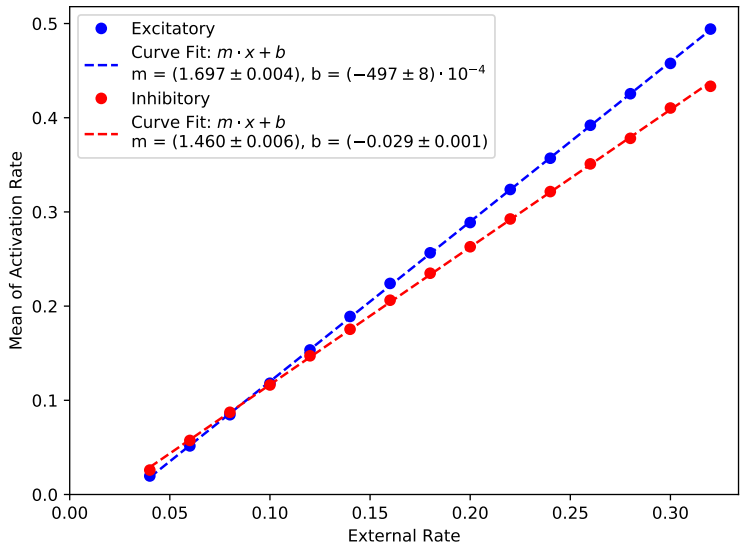


Figure 3.3: Relationship between external mean activity m_0 and excitatory and inhibitory mean activation m_E and m_I. Measuring mean activity over 20 time steps for each datapoint. Other parameters as defined in Table 1.

In Figure 3.3, we investigate the behavior of the mean activation rates, when the level of external activation m_0 is changed. In the calculations in Chapter 2.5, we already showed that the overall activation of the network is linearly dependent on the external input. We ran our model for 20 time steps at many levels of external activation m_0 and measured the mean activity for each population(m_E and m_I). Such a short time span is sufficient because the mean over time is very stable as we can see in Figure 3.2b. As predicted in , 2.5 both excitatory and inhibitory activation rates rise in a linear fashion with m_0. The linear regression shows the slopes to be $m_E = 1.7 \cdot m_0$ and

$m_I = 1.46 \cdot m_0$ with K=1000. For the case of finite K, the threshold θ_K has a not negligible effect. The thresholds $\theta_E = 1$ and $\theta_I = 0.7$ reduce u_k^i, yielding less cases in which u_k^i is positive and, thus, a lower activation rate. This results in the difference between the measured gradients and those calculated for the infinite K scenario (with $m_E = 2 \cdot m_0$ and $m_I = 1.5 \cdot m_0$). Also, these results are similar to the approximation for the case of $K = 1000$ by Vreeswijk and Sompolinsky (1998).

3.2 Comparison with Rigid Update Order

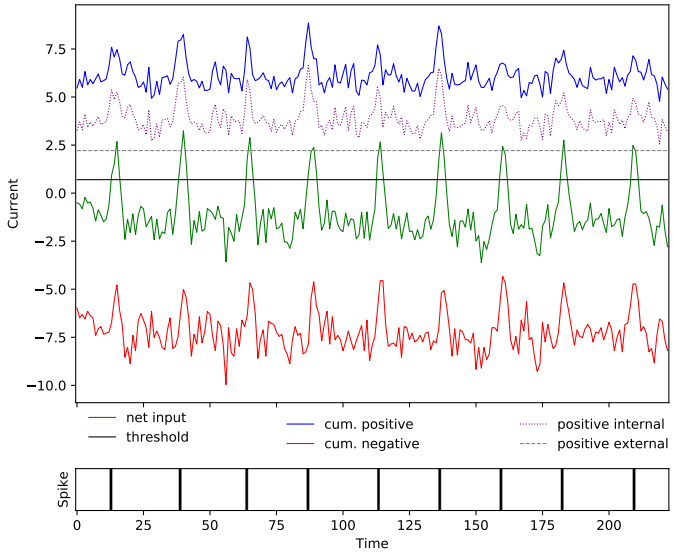
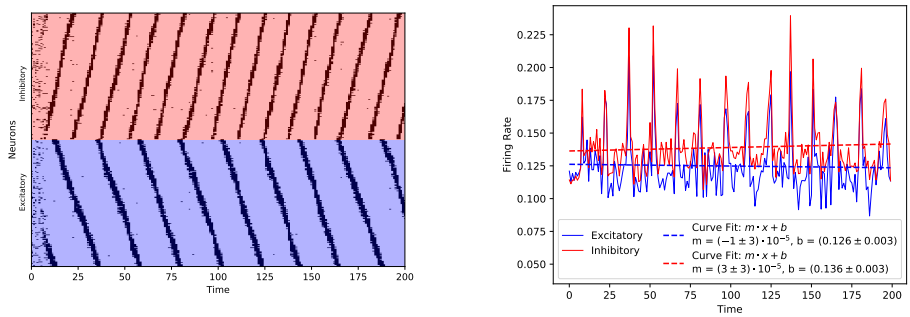


Figure 3.4: Input streams to neuron over time with rigid update order. Rigid update order leads to oscillations. Here, the update order within a population is fixed. There is still a randomized decision whether a member of the excitatory or inhibitory population is updated. All other conditions are the same as in Figure 3.1.

Now, we consider the case of holding the update order within a population static. In this scenario, it still is random (with likelihoods determined by the values of τ_I) whether a neuron of the excitatory or inhibitory population is chosen. But if, for example, σ_E^{17} is preceded by σ_E^{254} once, anytime σ_E^{254} is updated, the next time an excitatory neuron is to be updated, it will be σ_E^{17}.

We first look at how the inputs into a neuron behave, as in the chapter before. In Figure 3.4, for most time steps the behavior of the network is similar to the behavior in Figure 3.1. However, in a fixed periodic interval of roughly 25 time steps, input from excitatory neurons increases at the

same time as input from inhibitory neurons decreases. This leads to a very large increase in the net input. Both excitatory and inhibitory input quickly return to their initial position and fluctuate as in Figure 3.1.



(a) Overview of Activity Levels

(b) Mean Activation over time

Figure 3.5: Activation of individual neurons and mean activation over time for neurons with rigid update order. In (a), neurons within a population are sorted by position in update order. Definition of rigid update order see Figure 3.5 or Chapter 3.2. Otherwise, conditions as in Figure 3.2.

Now, we look at all how neurons behave over time. The results displayed in Figure 3.5 are markedly different from the case of a random update order. In Subfigure (a), neurons in each population are sorted by update order. We see that after a short period of time, the neurons' activity is highly correlated with the activity of neurons updated right before and after them. In the first few time steps, the pattern seems similar to the case in Figure 3.2a. After some time has passed, there remain only a few cases of neurons switching to the active state without their neighbors being active as well. The vast majority of events happen in close synchrony. Still, all neurons are not active at the same time, instead, a certain share is at any given point in time. This leads to diagonal lines of activation to propagate through each population. They move at different speeds for excitatory and inhibitory neurons. This is due to the fact, that the inhibitory neurons are updated at a faster rate ($\tau_I > \tau_E$).

In Figure 3.5b, we see that the mean activity in most time steps is in a band between 10 and 15 %. This is the same the range in which the activation moved in Figure 3.2b as well. Here too, excitatory neurons have a higher level of excitation. However, there are periodic spikes in mean activation. They are consistent in their difference in time, but not in their intensity.

The mostly constant update order still has the random element of whether the neuron to be updated is excitatory or inhibitory. If several inhibitory neurons are updated shortly after a rise in activity, this could lead to lower peaks. In fact, the diagonally moving spike trains do not necessarily need

to translate to spikes in the mean activation. Further research could look into the behavior of such systems.

Here, we saw different lines for each population. It would be interesting to see whether this behavior could be replicated also for a larger amount of subgroups. For the case of splitting each population into two subgroups (applying the same combination of fixed and random update order), would four independent diagonal lines emerge?

If one has a completely unchanging update order the frequency becomes even smaller. It is also possible to update all excitatory and then all inhibitory neurons in a row. Then the system simply oscillates between excitation and inhibition every other time step, as described.

3.3 Differences Between Neurons

Next, we turn our attention to the randomness in the system. We want to see neurons behave unpredictably and random, for this we will see whether neurons behave differently compared to each other and whether the behavior of each neuron was likely to be predictable.

As was stated above, only the average number of connections between neurons is specified but the actual number of connections is not. As a result, the inputs are unequally distributed and each neuron has a distinct number of inputs from excitatory and inhibitory sources. On the one hand, this leads to a less predictable behavior of the system due to many different starting conditions. On the other hand, depending on the synaptic weights applied, this could lead to some neurons to be highly active while others rarely fire at all. If many neurons barely fire while some fire in rapid succession, a lot less information can be stored in the system. This effect would not be reflected in the mean activation rate and therefore not visible in Figure 3.2b. In Figure 3.2a, we already saw that a sizeable share of neurons does fire. Yet a large group of inactive neurons could be hidden among the spiking ones.

Indeed, the distribution of firing rates in the model reveals almost all neurons spike and only a few do not fire at all (see Figure 3.6). For a longer duration of the simulation, most likely, all neurons would spike. The different starting conditions of neurons are visible as there is a broad range of different firing rates, which are similar to the predictions by Vreeswijk and Sompolinsky (1998).

Figure 3.7a shows the distribution of the coefficient of variation. Most neurons are concentrated around a CV of one. The likelihood of CVs larger than one decreases rapidly. However, there exist a lot more neurons with CVs lower than one. To understand this, we look how at how firing rates affect the coefficient of variation (see Figure 3.7b). Indeed, for most firing rates, the CV does not vary a lot, as it was described in Chapter 2.4. Only neurons with the lowest firing rates have coefficients of variation which are

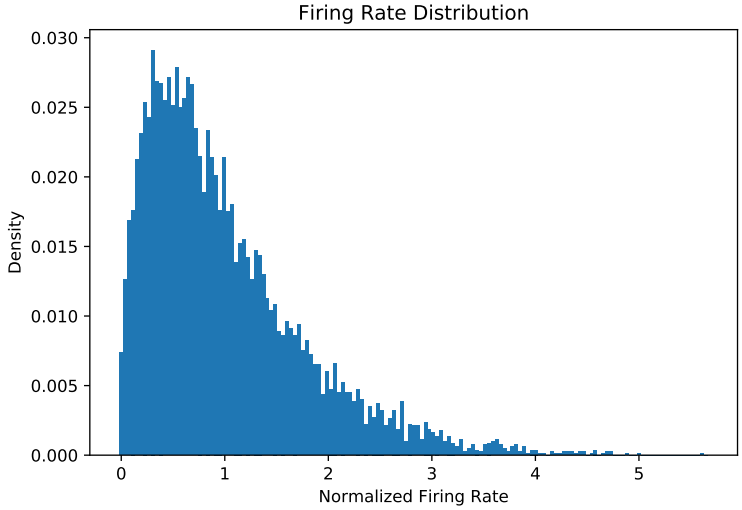


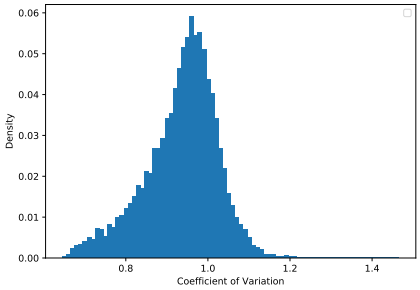
Figure 3.6: Distribution of Firing Rates. Firing rates are based on a single neuron each. The normalized firing rate is calculated by dividing each rate through the mean firing rate.

widely spread out. This is the case because those neurons only spiked on very few occasions and, in turn, each has only a small number of interspike intervals as well. The mean interval is very large for all neurons but, as only a few values were sampled, their standard deviation may vary a lot. This explains the larger variation of the coefficient of variation for small firing rates and also why it becomes more concentrated with rising firing rates.

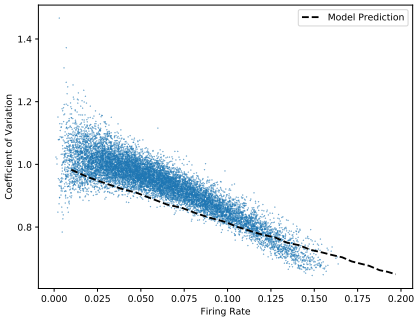
Furthermore, random spikes, as described in Chapter 2.4, are also simulated and their CV is displayed for comparison. Both the model prediction and the actual CVs measured decrease for larger firing rates. While the CVs of the balanced network do not decrease as linearly as the prediction, the behavior of the network can largely be explained by that behavior. Also, the approximation with a single exponential distribution seems to hold for values below a firing rate of 0.05 (see Equation 2.7).

To visualize this, interspike intervals of two neurons from the same simulation are displayed in Equation 3.7c and Equation (d). In (c) an average neuron with a CV close to one can be seen. The most intervals are found close to 0 and with their occurrence decreasing rapidly. Sometimes, one bin has a lot more or less than both his neighbors, implying that there is still a certain amount of fluctuation visible. Apart from that, the distribution of interspike intervals does overall resemble an exponential one.

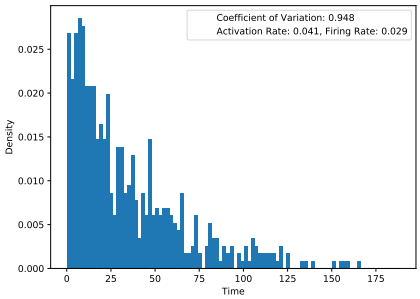
Figure 3.7d shows a neuron that spikes with a firing rate of 0.14 which is more than twice the average rate. This neuron is also an example of a neuron that reaches an activation rate above 0.50 (in a population with a



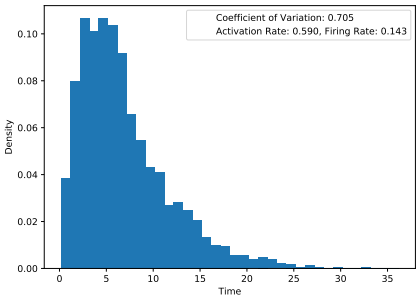
(a) Distribution of coefficients of variation



(b) Scatterplot of coefficients of variation and Firing Rates



(c) Interspike Interval of an irregular spiking neuron



(d) Interspike Interval of a fast spiking neuron

Figure 3.7: Distribution of CV and ISI in the network. Simulation over 20,000 time steps.
(a) Distribution of the coefficients of variation based on the individual neurons' interspike intervals;
(b) CV vs firing rates, Firing rate and CV calculated for each neuron; The CV estimate is obtained by simulating a random neuron spiking 10,000 times. In each update cycle, it is active with a likelihood of p and inactive otherwise. The time between two updates is drawn from an exponential distribution with a rate of 1. For different likelihoods of activation, we calculate both the firing rate and coefficient of variation;
(c) and (d) show Interspike Intervals of two neurons. (c) shows one with a CV close to one, representing the majority of neurons and (d) shows a neuron with particularly high firing rate (or CV)

mean activation rate of 0.12). Compared to (c), the distribution of (d) looks comparatively smooth. This can be explained as, over the same duration, this neuron recorded a lot more spikes and thus also more interspike intervals, which decreases the variance. Besides, a refractory period is visible, as for the first few time steps, a spike is less likely to occur. After the refractory period, spikes become less likely for larger intervals, with a distribution

comparable to (c). The distribution is comparable to that of a random variable whose values are drawn from two different exponential distributions (an example can be seen in the Appendix, Figure A.1.

3.4 Memory Capabilities

In this section, we analyze the impact, that oscillating input values have on the spiking behavior of the system. We use the same balanced network as before, except for the external mean activation m_0, which now changes over time. It is either in a high or low state, which is determined randomly in each time step. For the following, we choose inputs of $m_0 = 0.1$ and $m_0 = 0.3$.

We train a linear classifier to identify whether the input is high or low in the current period. Furthermore, it is also trained to identify the input level for previous periods (i.e. with delay). The results are shown in Figure 3.8. The readout correctly classifies the input level of the current period in each case. For predictions with a delay of one, on average 70 % of predictions are correct. However, there is a high level of fluctuation. For a larger time interval between label and prediction, the readout is not better than chance.

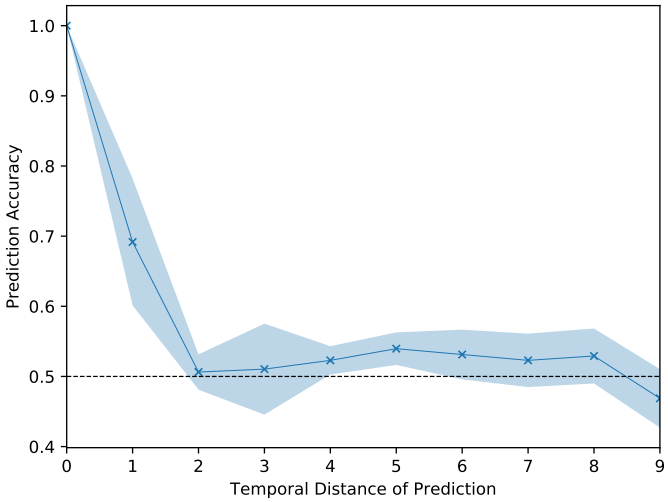
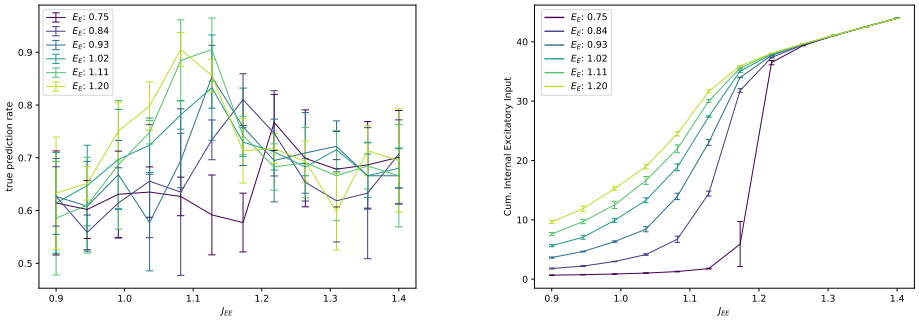


Figure 3.8: Memory Capacity of Network Drops Rapidly. External mean activation m_0 randomly changes between 0.1 and 0.3 (All other Parameters as defined in Table 1). Share of times logistic regression correctly labels activation as high or low depending on how many time steps have passed between labeled event and prediction. Individual measurements represented by $\times$ and the shaded area is within one standard deviation of the mean. The black dotted line represents a prediction accuracy of chance.



(a) Prediction Quality with Delay of One Time step

(b) Cumulative Excitatory Internal Input

Figure 3.9: Variation of Ratio Internal/External Excitatory Input.
(a) Prediction of high or low level of external activation m_0 after one time step has passed. Correctness of Linear Readout in relation to the strength of the excitatory to excitatory connection strength J_{EE}. The colors of lines represent different levels of strength for the external Input E_E. Errorbars measure the statistical error of the mean using five independent measurements for each datapoint. Otherwise setup like in Figure 3.8.
(b) Measures the excitatory input (in units of the excitatory threshold) instead of prediction accuracy. Otherwise conditions of (a) apply.

Next, we investigate the question of whether the relative strength of external and recurrent input affects the quality of the prediction. To answer the question, whether a higher share of recurrent input leads to a higher level of information contained in the system. We varied the weight factors for the input from excitatory neurons to excitatory neurons J_{EE} as well as for the external input to excitatory neurons E_E. We see the highest share of correct predictions for recurrent weights around $J_{EE} = 1.1$. Here, lower values of E_E increase the predictive capabilities of the classifier and also extend them to lower values of J_{EE}. For values of J_{EE}, which are not close to the peak, E_E does not change the number of correct predictions. For low values of J_{EE}, correct predictions occur more or less six out of ten times which is slightly better than chance. For very high values of J_{EE} on the other hand, predictions occur roughly in 70 percent of cases which is a bit better.

To better understand these results, we take a look at how the cumulative internal excitatory input changes with J_{EE}. Here, it is immediately apparent, why external input was only relevant in some cases. For low and high values of J_{EE}, external input is a lot more similar than around the peak. For high values of J_{EE}, one can, in fact, see, that the recurrent input is the same for different levels of external input. This happens because almost all neurons in the system are constantly active (as can be quickly determined when comparing the cumulative excitatory recurrent input with the result of

Equation 2.2). Apparently, changes in external input can still be registered by some neurons leading to the classifier perform better than chance. However, the external input in a given period is smaller than internal excitatory input by almost an order of magnitude. Therefore, the external input is drowned out in many cases.

If both J_{EE} and E_E are small, J_{IE} is comparatively large. So large that for the lowest value of E_E, there is no excitatory recurrent input until $J_{EE} > 1.1$. For network combinations with E_E than that, the external input is as large or larger than all recurrent positive input. Therefore, small changes in excitatory input (due to past changes in external input) are drowned out by the current external input.

For levels of J_{EE} which yielded the most correct prediction, varying E_E has the strongest impact on excitatory recurrent input. This means, changes to the external input in general influence the level of excitatory input. Furthermore, neither excitatory nor inhibitory effects dominate the positive recurrent input. In turn, external effects from the previous period can lead to changes in the activation level. This enables the system to correctly predict the state of the previous period.

However, the difficulties of storing information in the longer term are also visible here. External input has to be sufficiently strong (compared to the inhibitory input) to be stored by the system. At the same time, excitatory input has to be strong to carry information from one time step to the next. The stronger external input in the current period is, the stronger has the recurrent input to be to remember any previous input. At the same time, if the sum of all positive inputs becomes too large, the system is constantly activated and no information is stored.

3.5 Chaotic Behavior

In this section, we take a look at two different initializations of the system which are far away from the equilibrium state. We take a fully activated system and a fully inhibited system, meaning that the initial vector $\sigma_i(t = 0)$ equals either 0 or 1 for all units (with parameters as in Chapter 3 and 3.3). We track the mean firing rate over time. Excitatory and inhibitory neurons behave the same way and are therefore not shown separately. The fully activated network, by definition, starts with all neurons firing. In the first time step, the firing rate drops to zero and remains close to zero for the next time step. The fully activated network reaches the long term mean activation of approximately 0.15, already after three time steps. The fully inhibited network jumps to a firing rate of 0.2 and then swiftly decays towards the long term stable state, over a period of two to three time steps.

The fully activated system cannot fire in the first time step, as it needs the neurons to deactivate first before it can fire again. The updating process happens randomly, thus, some neurons still have not been updated in the

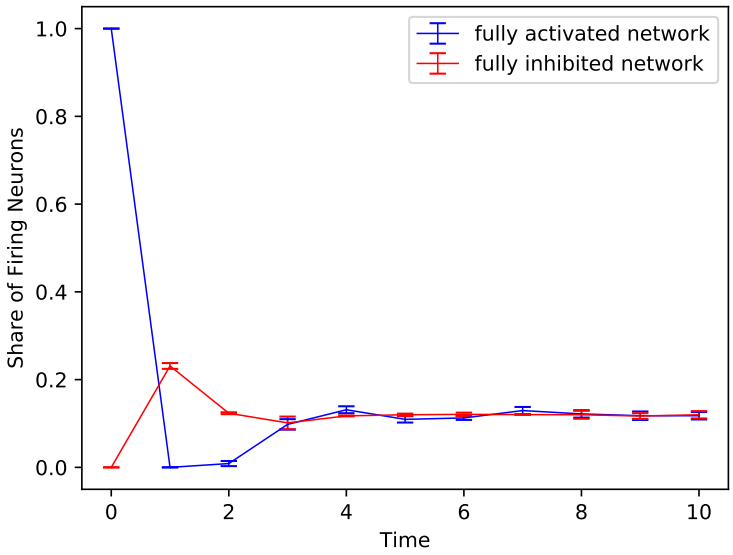


Figure 3.10: Outside the Balanced State. Initializing values to a fully inhibited or fully excited network. Measuring mean activation over time for both excitatory and inhibitory neurons. Errorbars show the standard deviation in the sample. All parameters as in Table 1.

second time step. The stable balanced state contains only a few active neurons. Therefore, some neurons remain active, which does not lead to new spikes, and thus, they keep the system suppressed.

Both systems are far away from the long term balanced state. The immediate reaction is a swing to the other side of the stable state. Then, they approach the stable state reasonably quickly. In Chapter 2.5, we chose the weight factors, so that the balanced network moves towards the same stable state in long term. One should take note of how quickly the system is able to perform this move. In the case of a maximal distance to the stable state m_k, the mean activation level $m_k(t)$ returns to m_k within three time steps.

Now, we move from very different starting conditions to very similar starting conditions. For this, the same state is initialized twice (σ and $\hat{\sigma}$) and for $\hat{\sigma}$ one neuron's state is flipped (i.e. changed from active to passive or vice versa). Then, both are updated in the same order. We estimate how similar they are in each time step using the Hamming distance. It is a measure of how different any two binary vectors (or states of the network) are. Precisely, it is defined as follows:

$$d_H(t) = \frac{1}{N} \sum_{i=1}^{N} \left(\sigma_i(t) - \sigma_i'(t)\right)^2 . \tag{3.1}$$

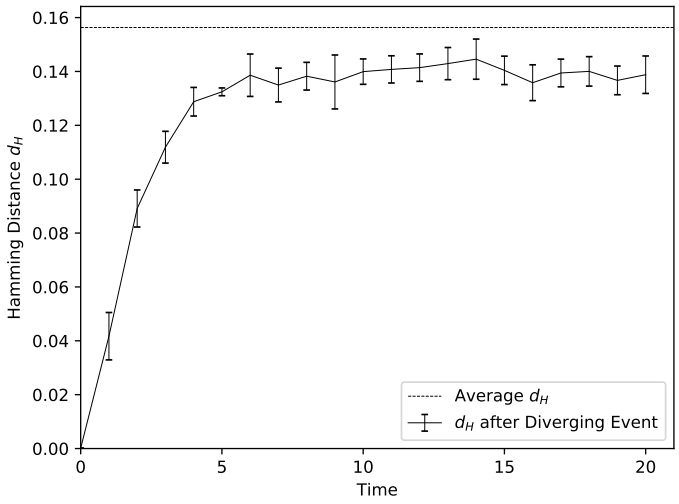


Figure 3.11: Hamming Distance after Single Diverging Event. A diverging event is defined as switching one value from active to inactive (or vice versa). Tracking of states σ and $\hat{\sigma}$ over time with the same parameters and update-order. From this, the Hamming distance is calculated every time step. Errorbars show the standard deviation in the sample. The dotted line represents the average Hamming distance of unrelated states of the system. Parameters as described in Chapter 2.

A Hamming distance d_H of 0 means that the states are identical and $d_H = 1$ shows that not a single state is the same. Therefore, a value of d_H of 0.5 means that the two systems are independent of each other.

When running the simulation, we see that the difference can be extinguished if the flipped neuron is one of the first to be updated. However, this occurs rarely and it has been excluded from the displayed data. Furthermore, we initialize a range of independent states σ with similar levels of activation and calculate their Hamming distance. With this method, we can calculate the expected ceiling of d_H. Figure 3.11 shows that the Hamming distance increases rapidly after the first state is changed. In a short period of time, d_H has reached a level slightly below the ceiling and does not significantly change its value anymore. This shows how quickly the balanced network diverges and destroys past information. Even the greatest similarity vanishes within a few time steps.

This indicates that in this balanced network, neither the activity of specific neurons nor the general level of activation can be used to predict previous input levels beyond short time periods.

4 Conclusion

We were able to replicate a balanced network as described by Vreeswijk and Sompolinsky (1998) and reproduce some of their major findings. We showed in Figure 3.1 that while activity in the network as a whole is relatively stable, net input to a given neuron varies strongly with each time step. This leads to a seemingly random spiking behavior for most neurons. As in the human brain, the amount of connections is not completely uniform but instead varies to some degree. In this model too, neurons can have more excitatory connections and a lesser amount of inhibitory connections or vice versa which leads to an increased or decreased firing rate, respectively. We show in Figure 3.6, that the firing rates differ widely between neurons. [1] Furthermore, we were also able to demonstrate a linear correlation between external activation levels and network activation levels (see Figure 3.3), in line with the calculations in Chapter 2.5.

Building on the work of Vreeswijk and Sompolinsky (1998), we find a correlation between neurons' firing rates and their interspike intervals' coefficient of variation. We are able to explain this correlation with a neuron which randomly selects either the active or inactive state with a given probability. For lower firing rates, neurons' interspike intervals can be approximated with a single exponential distribution. Furthermore, we found that reducing the randomness in the update order leads to periodic spiking patterns in the individual neuron. Apart from the temporal correlation, there were also strong spatial correlations (see Figure 3.4 and Figure 3.5). In this setup, neurons that follow one another in the update order tend to spike together.

In Figure 3.8, we saw that the capacity of the network to store information fades with time progressing. After two time steps (i.e. updating, on average, every neuron twice), a linear readout was about as good as chance. We found that varying the weights of excitatory input from within the network and outside (varying J_{EE} and E_E) could increase correct predictions after one time step from roughly 0.7 with the conditions as proposed by Vreeswijk and Sompolinsky (1998) to above 0.9 (see Figure 3.9a). However, we did not analyze the impact of all parameters of the network. Future research could, therefore, alter further parameters and possibly find more promising results within a short time frame. Yet, for larger time differences, we do not see a path leading to a significant increase in correct predictions because of the chaotic nature of the network. The level of activation reverts rapidly to the stable state and even almost identical states lose all similarities within a few time steps (see Figure 3.11 and Figure 3.10, respectively). Depending on the parameter set, the time scales can be adapted to some degree. Nevertheless, those mechanisms are an essential part of the system.

[1] Best order of the last three sentences?

Thus we can conclude, that balanced networks are not suitable for storing information. However this model could serve another purpose for which it is favorable to have short-lived signals. A possible example could be perception, which also is processed in the cortex. Here, a structure similar to a balanced network could store incoming information for a short period, while it is being processed.

In this frame of work, we always looked at external input as uniform for all neurons within one population. Further research could vary external input for neurons within a population. In this scenario, a more complex structure of the network could emerge, which, in turn, could lead to high levels of correct predictions for longer time frames. Additionally, will the mean level of external input still be proportional to the mean network activity? On a different note, while spikes ought to occur at the same rate for very high and very low activation rates, one could also explore whether the system as a whole behaves the same. Fortunately, to answer these questions, no dead musicians are required.

A Appendix

A.1 Additional Figures

In this section, we will present some figures, which were not used in the main text. First, we present a simulation of random spikes with a likelihood p = 0.6 to be active. The interspike intervals of the model can be imitated by calculating the sum of two different random variables drawn from two different exponential distributions with the argument being p and $1-p$ (for $p \in [0,1]$).

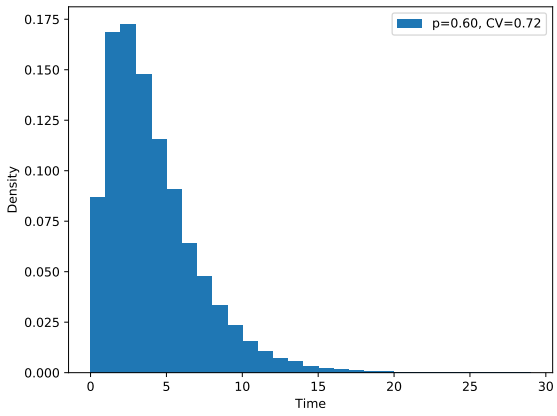
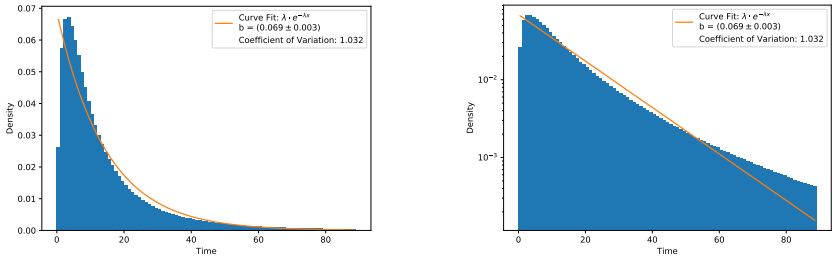


Figure A.1: simulation of random spiking neuron

In Figure A.2, we see a histogram of interspike intervals of all neurons. Here one can see a dip in the beginning. However, an exponential distribution still could explain the results reasonably well.



(a) Histogram of all interspike intervals in the whole network

(b) Histogram of all ISI with log-scale

Figure A.2: Random nature of individual firing behavior. Histogram of interspike intervals of all neurons. Collecting the time differences between two spikes with time measured in τ_E. In the beginning, there is a short refractory period preventing a high frequency of spiking. For most of the data, however, the likelihood of larger spiking intervals falls exponentially. The orange line represents an exponential fit. (a) shows graph without log-scale, (b) with log-scale.

The Figure A.3, displays the ratio of internal and external input. It is thus similar to Figure 3.9a. When dividing/multiplying the external input u_k^0, one can convert between the two.

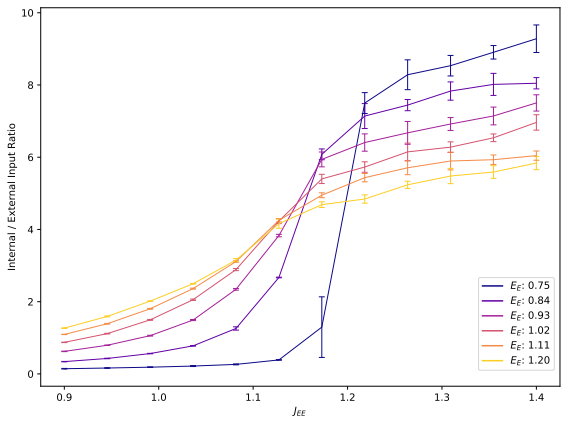


Figure A.3: Drastic change in Input Structure. Internal Excitatory Input in dependence of Internal Excitatory to Excitatory Factor J_{EE}. In units of the excitatory threshold. The colors of lines stand for different levels of strenght for the External Excitatory Input. Errorbars show the standard deviation in the sample. All parameters identical to Figure 3.8.

The variation of the strength factors J_{EE} and E_E, for delays larger than one, does not contain any set of parameters that achieves a high share of correct predictions. The results can be seen in Figure A.4.

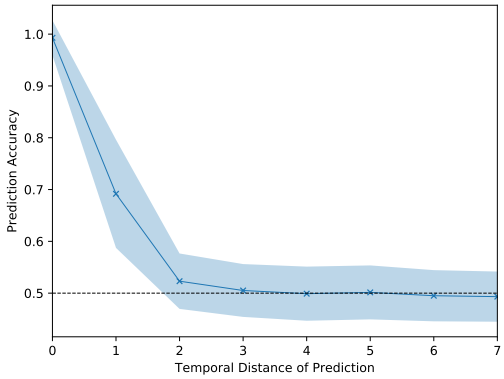


Figure A.4: Memory Capacity of Network for different ratios of external to internal input. Variation of J_{EE} from 0.9 to 1.4 and E_E from 0.75 to 1.20 (same as Figure A.3. External mean activation m_0 randomly changes between 0.1 and 0.3 (All other Parameters as defined in Table 1). Share of times logistic regression correctly labels activation as high or low depending on how many timesteps have passed between labeled event and prediction. Individual measurements represented by $\times$ and shaded area is within one standard deviation of the mean. Black dotted line represents a prediction accuracy of chance.

Here, we present the same scenario as in Figure 3.8, but with a variation of strength factors J_{EE} and E_E as in Figure 3.9. The main difference between the two is, that, in some cases, the linear readout fails to correctly predict the current external input state. After two time steps, here too, the amount of correct answer is the same as a coin flip.

A.2 Accessing the Graphical Interface

In the following, it will explained how to access the code with which the simulations in this thesis have been made. Clear instructions how to operate the graphical interface are provided. First, to access the code, the following is necessary:

- Fork or download the code from github under: `https://github.com/cvazz/bachelor_thesis`
- An installation of python 3.7 or more recent.
- and the following pytbon packages: numpy, scipy, matplotlib, pathlib, pickle and pyqt5

The actual model is located in the 'neuron.py' file. To run simulations on it, several examples are presented in the 'initialize.py' module. The plots, which can be performed on a single simulation of the network, are handled via the 'plots.py' file and can be accessed via a graphical interface. The graphical interface is created by running the 'graphical.py' file. An environment such as the following should appear.

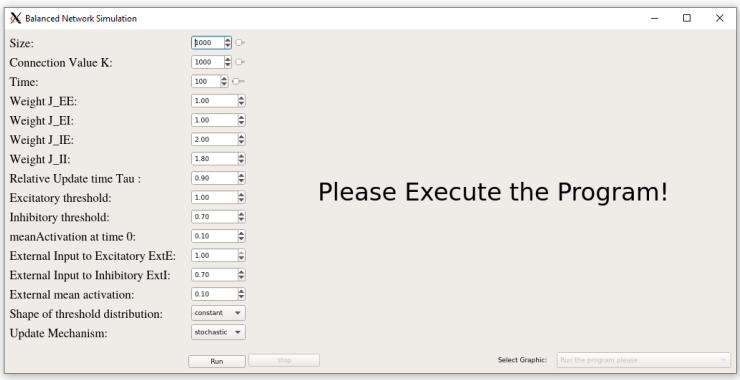


Figure A.5: Program on Start-up

On the left-hand side of the program, the parameters are presented in the parameter panel and can be customized. On the right-hand side, plots will be shown as soon as the first simulation has been run. A simulation can be run by pressing the 'run' button below the parameters. Then, the parameters are taken and passed to the simulation. While the simulation takes place, values cannot be changed anymore. A 'stop' button is included on the right side of the 'run' button. It halts the simulation and unfreezes the parameters panel. Then one can start a new simulation or look at the figures of the last completed simulation. If one has successfully run a

simulation, one can use the drop-down menu in the bottom right to select the desired visualization.

By default, the parameters are set to the values used in Chapter 2, except for the duration of the simulation and size of the network. If possible, it is recommended to increase the amount of detail visible in the graphic. But large values of either carry a large computational cost and demand long calculation times. For this reason, a simulation which takes too long can be canceled.

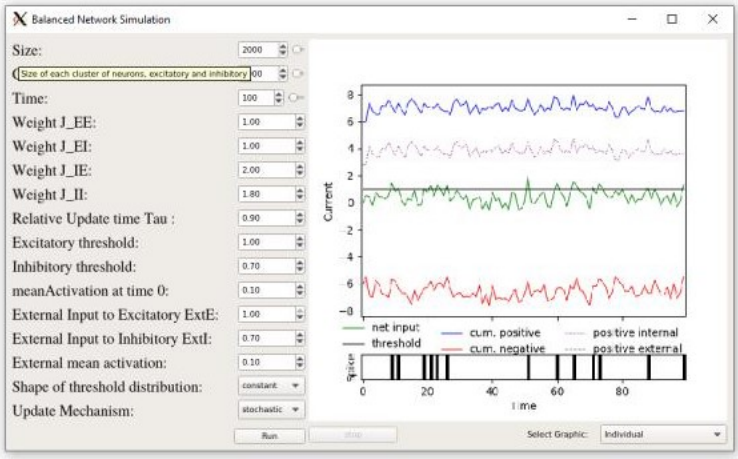


Figure A.6: Additional Information can be gained by pointing the mouse at any parameter

The figures in Chapter 3.1 and Chapter 3.3 can be calculated with preset parameters (as well as Figure A.2). If one needs more information about a given parameter, it suffices to simply hover above the parameter name and a more in-depth description will appear. It is possible to also simulate networks that are constantly active or other such scenarios, by choosing different parameters for the connection weights.

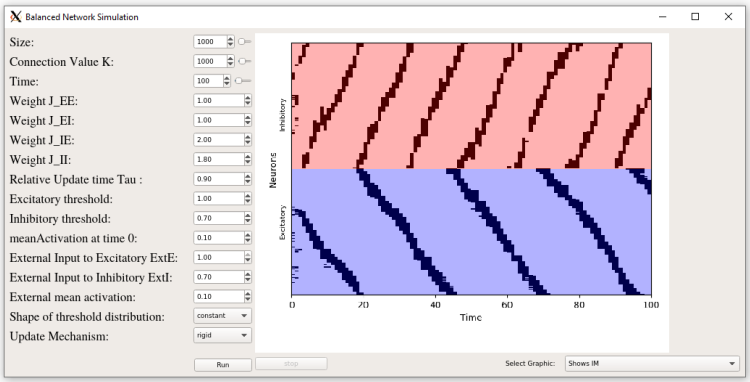


Figure A.7: The update order can be controlled

For the plots in Figure 3.7, it is necessary to increase the time parameter to above 1000 to see interesting results. It is possible to set the update order to 'rigid' which allows one to see all plots for the version of the network in Chapter 3.2. Also, it is possible to explore the behavior of the network with different configurations of the threshold distribution such as a uniform distribution over an interval or a Gaussian distribution around a given mean. An example can be seen in Figure A.8.

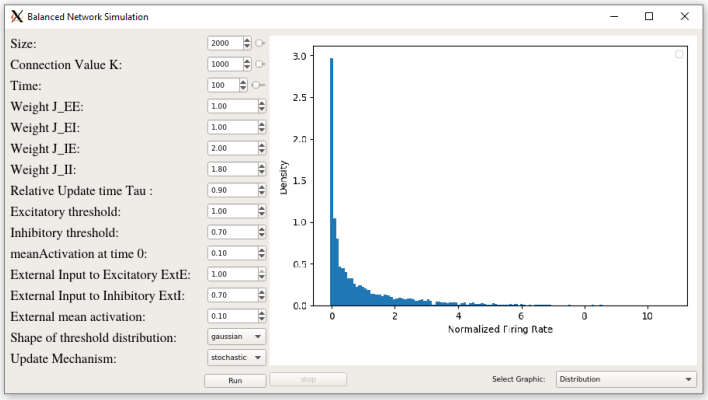


Figure A.8: Also the threshold distribution can be manipulated

B Bibliography

M. Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.

Y. Aviel, C. Mehring, M. Abeles, and D. Horn. On embedding synfire chains in a balanced network. *Neural computation*, 15(6):1321–1340, 2003.

Y. Aviel, D. Horn, and M. Abeles. Memory capacity of balanced networks. *Neural Computation*, 17(3):691–713, 2005.

A. Bell, Z. F. Mainen, M. Tsodyks, and T. J. Sejnowski. "balancing" of conductances may explain irregular cortical spiking. *La Jolla, CA: Institute for Neural Computation Technical Report INC-9502*, 1995.

Britannica. *Cerebrum*. Encyclopædia Britannica, inc., 5th edition, 2020. URL `https://www.britannica.com/science/cerebrum`.

B. D. Burns and A. Webb. The spontaneous activity of neurones in the cat's cerebral cortex. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 194(1115):211–223, 1976.

M. Dingman. Cerebral cortex. *Neuoscientifically Challenged*, 2014. URL `https://www.neuroscientificallychallenged.com/blog/know-your-brain-cerebral-cortex`.

C. D. Gilbert and M. Sigman. Brain states: top-down influences in sensory processing. *Neuron*, 54(5):677–696, 2007.

C. M. Gray, P. König, A. K. Engel, and W. Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338(6213):334–337, 1989.

S. Herculano-Houiel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in Human Neuroscience*, 31(3), 2009. doi: https://doi.org/10.3389/neuro.09.031.2009.

G. R. Holt, W. R. Softky, C. Koch, and R. J. Douglas. Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. *Journal of neurophysiology*, 75(5):1806–1814, 1996.

W. Maass and H. Markram. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4):593 – 616, 2004. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2004.04.001. URL `http://www.sciencedirect.com/science/article/pii/S0022000004000406`.

Z. F. Mainen and T. J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506, 1995.

K. Saladin. *Human Anatomy*. McGraw-Hill, 2011. ISBN 9780071222075. URL https://books.google.de/books?id=DfESQgAACAAJ.

W. R. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *Journal of Neuroscience*, 13(1):334–350, 1993.

R. Swenson. Review of clinical and functional neuroscience. *Dartmouth Medical School*, 2006. URL https://www.dartmouth.edu/~rswenson/NeuroSci/chapter_11.html.

M. V. Tsodyks and T. Sejnowski. Rapid state switching in balanced cortical network models. *Network: Computation in Neural Systems*, 6(2):111–124, 1995.

C. v. Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274(5293):1724–1726, 1996.

C. v. Vreeswijk and H. Sompolinsky. Chaotic balanced state in a model of cortical circuits. *Neural computation*, 10(6):1321–1371, 1998.