BCPNN and Sequence Learning

Ramón Martínez

November 14, 2018

1 Introduction

1.1 Sequence Learning

Section with a brief history of sequence learning in computational nueroscience. Key papers, order, structure.

General observations. The problem started with Lashley [Lashley, 1951] Hebb reference.

The other papers

Phenomena that involves time. The idea that if you multiply each of the notes by a frequency the melody still keeps its nature. The melody is more than its parts (relationship among them) [Von Ehrenfels, 1988].

The idea that you can get human movement from the movement of only some dots [Johansson, 1973].

Getting texture information from contact with surfaces (spatio-temporal pattern) [Weber et al., 2013].

Dependence of the on the trained temporal interval [Johnson et al., 2010].

There is this experiment they test the Hippocampus through lesions as the seat of the capacity for path -sequence- disambiguation in rats. Some really interesting result is that is that if a memory delay is imposed upon the rat sequence disambiguation is imparired. Suggestion that the ability to perform such an action is living somewhere in the dynamical activity of the system [Agster et al., 2002].

Cue Triggered Recall

This is a paper about cue-recall in the visual cortex. It fits the idea of recall.

abstract: Cue-triggered recall of learned temporal sequences is an important cognitive function that has been attributed to higher brain areas. Here recordings in both anesthetized and awake rats demonstrate that after repeated stimulation with a moving spot that evoked sequential firing of an ensemble of primary visual cortex (V1) neurons, just a brief flash at the starting point of the motion path was sufficient to evoke a sequential firing pattern that reproduced the activation order evoked by the moving spot.

[Xu et al., 2012]

1.2 Experiments

Phill Paper

Motor area papers

Prefrontal cortex in monkeys. Small ensembles of 3–22 cells, mean 9 cells). The idea is to show that that the patterns in the sequence were active from before. [Averbeck et al., 2002]

Media motor areas, electrophysological recordings. Monkeys (Macaca fuscata).

Abstract: Although a well-known type of premovement activity representing the forthcoming movements was found in the two areas, we found an unexpected type of activity that represented a second-next movement before initiating the first of the two movements. Typically in the pre-SMA, such activity selective for the second-next movement peaked before the initiation of the first movement, decayed thereafter, and remained low in magnitude while initiating the second movement. This type of activity may tentatively hold information for the second movement while initiating the first.

The idea here is that there is some activity on the first unit that influences the second. [Nakajima et al., 2009]

Monkey, (Macaca mulatta). Pre-motor cortex.

quote: We report evidence about the existence of diverse degrees of **metastable dynamics in local cell assemblies**. They seem to be hierarchically organized in time as a web of heterogeneous cortical modules capable to produce fast transitions toward distributed and stereotyped representations of motor-related programs.

This paper seems about motor planning. [Mattia et al., 2013]

Sensory Area Papers

Gustatory cortex, rats. Electrophysiology.

This paper was about trying to show that some activity is lost in the trial average. It concluded that is better to qualify the activity as sequences of states.

quote: reveals that taste processing can be characterized as a progression of reliably stimulus specific sequences of ensemble firing rate

[Jones et al., 2007]

Parietal cortex. Have not checked this paper properly. [Crowe et al., 2010]

Memory

Monkeys (macata fuscata), visual memory task. Long term memory. Temporal cortex.

This paper is about the temporal cortex as a seat of long term visual memory (excitation here evokes imagery)

I could not find the sequences in this paper. [Miyashita, 1988]

Experiments about location in a short-memory tasks. They were actually looking for firing rate sequences to model them as Hidden Markov Model. Monkeys. Recording of single-unit activity from the frontal areas around the upper limb of the arcuate sulcus. [Abeles et al., 1995]

Same structure than the last paper it seems (only read abstract). [Seidemann et al., 1996]

The medial prefrontal cortex of the rat, in a working memory task. sequentially organized and transiently active neurons reliably differentiated between different trajectories of the rat in the maze

[Fujisawa et al., 2008]

Decision Making

Rat. Anterior cingulate cortex. Navigation task. Ensembles of neurons move through different coherent and dissociable states as the cognitive requirements of the task change.

Sequence fail is related to error on behavior. [Lapish et al., 2008]

The posterior parietal cortex (PPC). Optical imaging. perceptual decision and memory-guided navigation in a virtual environment

Distinct sequences of neurons were triggered on trials with opposite behavioural choices and defined divergent, choice-specific trajectories through a state space of neuronal population

During working memory decision tasks, the PPC may therefore perform computations through sequence-based circuit dynamics, rather than long-lived stable states, implemented using anatomically intermingled microcircuits

Tied to beahvior

[Harvey et al., 2012]

Reactivation in sleep

Patterns of firing-rate correlations between neurons in the rat medial prefrontal cortex during a repetitive sequence task were preserved during subsequent sleep, suggesting that waking patterns are reactivated

Tetrodes. [Euston et al., 2007]

We found that spiking patterns not only in the cortex but also in the hippocampus were organized into frames, defined as periods of stepwise increase in neuronal population activity. The multicell firing sequences evoked by awake experience were replayed during these frames in both regions

Rats. [Ji and Wilson, 2007]

This paper I read before, weak evidence if I remember well. [Almeida-Filho et al., 2014]

Murray Paper

1.3 Models

Levi Model

This is a spiking model. It has an assymetric learning rule for STDP which leads to an assymetric connectivity matrix. It forms sequences of cell assemblies that they call distributed synchrony. It has random input. Few sequences. No control of the persistent time [Levy et al., 2001].

Silvia Scarpetta model

Have not read this paper [Scarpetta and Giacco, 2013]

Verduzco Paper

This and that [Verduzco-Flores et al., 2012]

Minai Papers

This and that.

1.4 The BCPNN

Explanation of what is the BCPNN, where does it come from, and some applications.

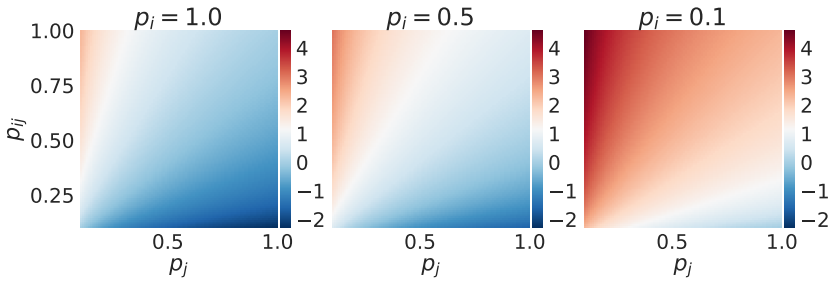


Figure 1: A schematic of how the BCPNN learning rules associates weights to different probability scenarios.

$$p_i = \frac{1}{N} \sum_{\mu=0}^{N} S_i^\mu \tag{1}$$

$$p_{ij} = \frac{1}{N} \sum_{\mu=0}^{N} S_i^\mu S_j^\mu \tag{2}$$

$$\beta_i = \begin{cases} \log(p_i), & p_i > 0, \\ \log(\epsilon), & p_i = 0.0 \end{cases} \tag{3}$$

$$w = \begin{cases} 0, & p_i = 0 \text{ or } p_j = 0, \\ \log(\epsilon), & p_{ij} = 0.0 \\ \log(\frac{p_{ij}}{p_i p_j}), & \text{otherwise} \end{cases} \tag{4}$$

Where ϵ is a small number chosen out of convenience to avoid the fact that *log* is not defined for 0.

1.5 A simple phenomenological theory of sequence recall

Successful sequence recall dynamics can be described as three dynamical qualities and their interplay: fixation, erosion and biased transition.

Fixation is a general mechanism that fixes a state passing a certain point.

Examples of fixation. Attractor dynamics.

Erosion is a mechanisms that eventually suppresses the state after the system has dwell some time on it.

Examples of erosion Spike frequency adaptation [Roach et al., 2016], feedback inhibition [Recanatesi et al., 2017].

Biased Transition, after the state has been eroded the

Examples of biased transition Biased by similiarty [Recanatesi et al., 2017]

2 The BCPNN as a Sequence Learning Network

First we can show that the BCPNN can be used do incremental learning [Sandberg et al., 2002]. And in particular there is a spike model that can be used to learn sequences [Tully et al., 2016].

2.1 Sequence Recall

In order to test test the capabilities of the BCPNN neural network as a sequence learning mechanism we will start with a minimal model of it. The idea here is to show the minimal conditions under which the system can successfully reproduce sequential activity and isolate how the parameters and properties of the network interact with each other. After we are equipped with this knowledge we will add more parts to the model and this in turn will provide new capabilities that we will explore later.

Using the phenomenology presented before, we will explain how the structure of the network in figure 2 and the dynamical equations of the system intertwine to achieve, fixation, erosion and biased transition altogether.

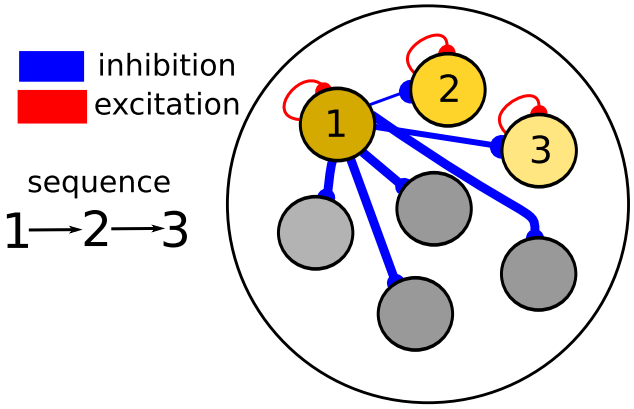


Figure 2: A simple BCPNN network with only one type of connectivity.

First the system achieves fixation by the means of the self-excitatory current depicted in red in figure 2. By itself this mechanism will fix all the patterns at the same time, that is why we need competitive selection. To solve that problem we will use a winner-takes-all mechanism [Yuille and Geiger, 1998] implemented in the form of equation 6. This equation ensures that at any point in time only the unit with the higher input current is activated.

After a particular unit is activated the adaptation current in equations 7 and 5 will be the mechanism responsible for the erosion of the pattern. Once a unit is activate for long enough and in the right parameter regime the adaptation current will surpass the self-excitatory current and the pattern will be suppressed. On this light, the time a particular unit remains activated is mostly dependent on the parameters that determine the dynamics of the adaptation and self-excitatory currents and the competitive balance between them. We will make this last relationship quantitative further down in this document.

Finally, this system would jump randomly among the states if it were not for a proper mechanism of biased transition. This is accomplished with differential inhibitory weights (illustrated in figure 2 as different widths for the blue inhibitory connections) that become more and more inhibitory the farther two units are in the sequence. This ensures that once the adaptation currents for a unit becomes big enough the next unit which is the less inhibited one wins the competition and gets prompted to activation by the winner-takes-all mechanism.

$$\tau_s \frac{ds_i}{dt} = g_{beta}\beta_i + \sum_j w_{ij}o_j - g_a a_i - s_i + \xi(t) \tag{5}$$

$$o_i = \delta_{i,argmax(s)} \tag{6}$$

$$\tau_a \frac{da_i}{dt} = o_i - a_i \tag{7}$$

In order to illustrate how the dynamics of the system work together we show an example of a successful sequence recall in figure 3. In the recall process we cue the first unit of the sequence by clamping it by a short period of time ($\sim 100ms$). We then let the system evolve on its own and, given the right combination of parameters, a sequence is effectively recalled if all the units that conform the pattern are activated in the expected order.

It is important to note that in this case we utilized a tailor-made connectivity matrix to clarify the relationship between the different component of the dynamics. Further down we will show how the same effect can be effectively achieved with a matrix that emerges from a self-organized learning process.

2.1.1 Analytical solution

We can solve the deterministic equations analytical with the method of undetermined coefficients. There are two conditions, the unit that is active, and the unit that is not.

$$s(t) = I_{fix} - g_a \left(\frac{C_{charge}}{1-r}\right) e^{-\frac{t}{\tau_a}} + \left(s_0 - I_{fix} + g_a \left(\frac{C_{charge}}{1-r}\right)\right) e^{-\frac{t}{\tau_s}} \tag{8}$$

$$\underset{acti}{s(t)} = \beta + w - g_a + g_a \left(\frac{1-a_0}{1-r}\right) e^{-\frac{t}{\tau_a}} + \left(s_0 - \beta - w + g_a - g_a \frac{1-a_0}{1-r}\right) e^{-\frac{t}{\tau_s}} \tag{9}$$

$$\underset{inact}{s(t)} = \beta + w - g_a \left(\frac{a_0}{1-r}\right) e^{-\frac{t}{\tau_a}} + \left(s_0 - \beta - w + g_a \left(\frac{a_0}{1-r}\right)\right) e^{-\frac{t}{\tau_s}} \tag{10}$$

Where $r = \frac{\tau_s}{\tau_a}$ and $C_{charge} = a_0$ for the non-active case and $C_{charge} = a_0 - 1$ for the active case. Same for $I_{fix} = \beta + w$ for the non-active case and $I_{fix} = \beta + w - g_a$ for the active case.

If there was a previous activation of the unit the quantity a_0 will reflect how much the unit charged (and then discharged) before the current activation of the unit took place.

It is desirable to calculate how long does a unit i becomes activated before the unit j becomes activated. This is called persistent time $T_{persistence}$ and can be calculated from the equations above by making the assumption that the $T_{persistence}$ is far bigger than τ_s, we also need to assume homogeneity in τ_a.

$$\frac{T_{ij}}{\tau_a} = \log\left(\frac{1}{1-B_{ij}}\right) + \log\left(\frac{g_i}{g_j} a_i(0) f_i + (1 - a_j(0)) f_j\right)$$

Where $B_{ij} = \frac{w_{jj} - w_{ij} + \beta_j - \beta_i}{g_j}$ is a measure of inertia or inverse of speed. The close to one it is, the longer the transition takes. Also, we see that B_{ij} needs to be bigger than 0 for the transition to occur, from there we can derive the criteria that $g_j > \delta w + \delta\beta$. We also introduced the short hand $f_i = \frac{1}{1-r_i}$ which is always bigger than 1 for this system and it represents delay effects introduced due to the capacitive nature of the unit. We see that the presence of adaptive current in the unit that is active hastens the transition ($a_j(0) > 0$ makes T_{ij} smaller by reducing the second term inside the second logarithm). On the other hand, the presence of adaptive current on the next unit ($a_i(0) > 0$) in the sequence slows the process by increasing the first term in the second logarithm.

We write now for completion the expressions of homogeneous parameters:

$$\frac{T_{ij}}{\tau_a} = \log\left(\frac{1}{1-B_{ij}}\right) + \log\left(\frac{1}{1-r}\right) + \log\left(a_i(0) + (1 - a_j(0))\right)$$

The second term is of the order of τ_s usually and the third term accounts for memory effects on the adaptation.

We can calculate from this the **backwards time** that is, once you passed through a particular unit.

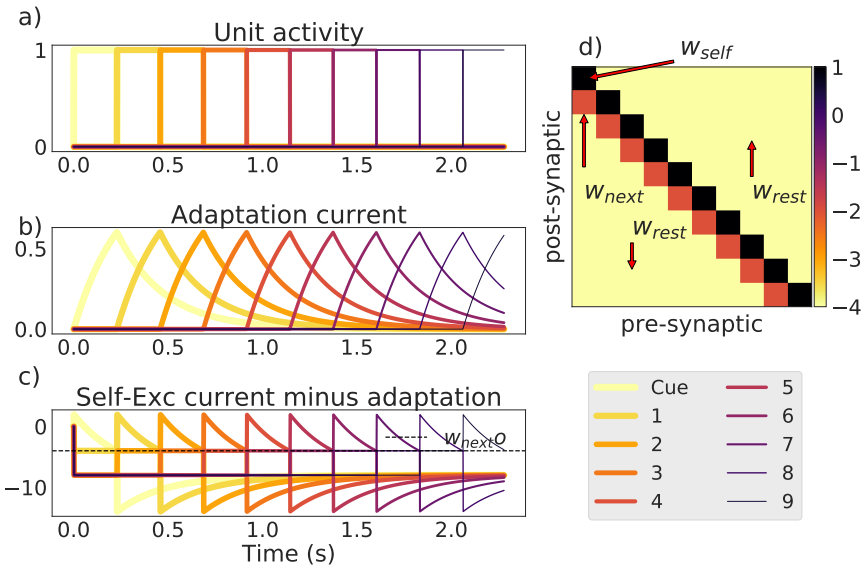


Figure 3: An instance of recall in the simple BCPNN neural network. a) Unit activity starting with the cue. b) the time course of the adaptation for each unit. c) the self-excitatory current minus the adaptation current, note that this quantity crossing the value of w_{next} (depicted here with a dotted line) marks the transition point from one unit to the next. d) The connectivity matrix where we have included pointers to the three most important quantities w_{self} for the self-excitatory weight, w_{next} for the inhibitory connection to the next element and w_{rest} for the rest of the connections.

We will now proceed to characterize the properties of the simplified version of the BCPNN. We will do this in two steps, first we will explain the recall properties of the system and then we will proceed to describe the learning rule and its dynamics. One of the most important quantities of sequence recall network is the **persistence time.** of a a state. It turns out than in this simple system by determining the recall time with regards to certain parameters we can explain both how the network works and when the recall is carried out successfully.

2.1.2 Persistence Time

If we have system with only one type of connectivity, and winner-takes-all selectivity one state will be suppressed in favor of the other as soon as the support of the second state is bigger. In more detail, if we start with a system where the first unit is activated its own support will be $s_1 = gw_{self} - a(t)$ where a is the adaptation current which grows in time. This will continue as long as the first unit is activated and as a consequence the support for the unit will be decreasing. On the other hand the second unit is receiving a constant current to its support $s_2 = gw_{next}$, if this process continues by continuity there will be a point when the support of the first unit will be equal to that of the second:

$$s_{self} = s_{next}$$
$$g_\beta \beta_{self} + g_w w_{self} - g_a(1 - e^{\frac{t}{\tau_a}}) = g_\beta \beta_{next} + g_w w_{next}$$

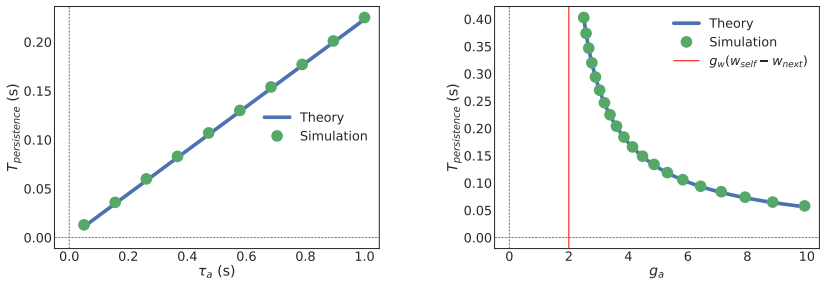
Where we have substituted the proper term for adaptation. We can solve for t in the expression above to obtain the persistent time $T_{persistent}$:

$$T_{persistence} = \tau_a \ln \left(\frac{g_a}{g_a - g_w(w_{self} - w_{next}) - g_{beta}(\beta_{self} - \beta_{self})} \right) \tag{11}$$

We notice that the equation behaves linearly with regards to τ_a and logarithmically with regards to g_a, g_w, $wself$ w_{next}. Even more importantly, certain combinations of the latter parameters will give raise to singularity points. This is important because close to this values the persistence time can be varied greatly with small variations in the respective parameters. In other words we have a wide dynamical variability in $T_{persistence}$ as in [Murray et al., 2017]. We proceed now to describe the scaling for each of the parameters in a more concise way and to compare the theory derived above with simulations.

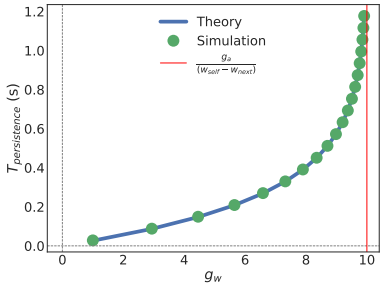
Adaptation current time constant τ_a
We explain here how the persistence time dynamics depend on τ_a the time constant of the adaptation current. A priori, the longer the adaptation time constant the longer it will take to the adaptation time current to erode the pattern. From equation 11 we can observe that the relationship is linear which is exactly what we get in figure 4 a). Note however, that the slope is not very step, that is, we do not get a lot of variation on the persistence time from variations on τ_a.

Adaptation current gain g_a The adaptation time current is actually a limiting factor in the succesfull recall of a sequence. If the adaptation current is not big enough to overcome the difference between the self-excitatory current and the current of the next element then the system will get stuck forever in the same state. We need therefore that the value of the adaptation gain g_a to be bigger the difference in weights multiplied by the weight gain $g_w(w_{self} - w_{next})$ which is just the denominator of the equation 11. Once we have overcome this threshold the behavior becomes obvious, the bigger the gain of the adaptation current the faster this current will overcome the self-excitatory one, as a consequence the persistence time will be smaller. We illustrate this behavior in figure 4 b). The singularity here just reflects the fact when the adaptation is barley enough to overcome the difference in currents it takes an exponential time to actually do it.

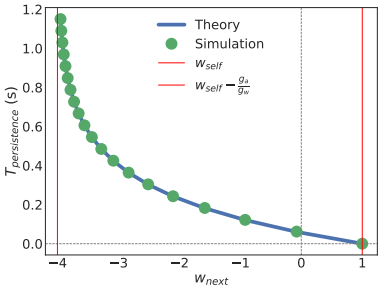


(a) τ_a

(b) g_a



(c) g_w



(d) w_{next}

Figure 4: Persistence time relationship with the parameters. a) We can appreciate that the persistence time grows linearly with τ_a, the adaptation current time constant. b) Here we depict the logarithimic dependence of $T_{persistence}$ on g_a. c) The same dependence for g_w. d) We illustrate here the effects of making the weight differential bigger (note that $w_{self} = 1$ in this plot).

Weight Gain If we look at the denominator of equation 11 we can interpret g_w as an amplifier of the weight differential ($w_{self} - w_{next}$, this in turn is a proxy for the amount of current that the adaptation current has to overcome. In the light of this is not surprising the the persistence time increases with g_w. Moreover, once the weight differential gets amplified enough the adaptation current will take an exponential amount of time to overcome this difference given rise to a singularity as well.

Next weight value In order to quantify the effects of varying w_{next} we fixed the value of w_{self} to 1. With this perspective on mind, the farther w_{next} is from 1 (the red line on the right) the biggest is the weight differential and using the same reasoning that we used for the effects of g_w above we conclude increasing persistent time and a singularity when the difference becomes big enough (red line to the left).

2.2 Sequence Learning

2.2.1 Off-line learning rule

In order for the BCPNN to learn from temporal patterns a first step is to generalize the rules given in equations 2 and 2 to cope with continuous signal. We propose here the following equations:

$$p_i = \frac{1}{T}\int_0^T S_i(t)dt \tag{12}$$

$$p_{ij} = \frac{1}{T}\int_0^T S_i(t)S_j(t)dt \tag{13}$$

Where T is the total presentation time. Note that this rules are a direct generalization of the discrete rules and can be reduced to them under the appropriate conditions, nevertheless they can deal directly deal with patterns of different duration and with incomplete activation in a way that is consistent with the normal rule. Once we have calculated the probabilities from the complete signal in the equations above we can use the p_i and p_ij quantities to estimate w and β using equations 4 and 3.

In figure 2.2.1 we illustrate how a collection of patterns presented in succession (left) look as continuous signals over time.

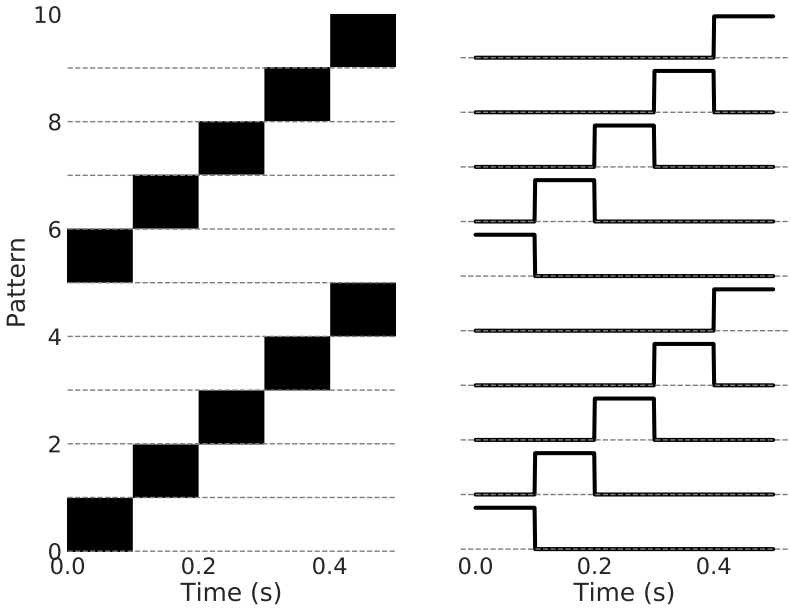


Figure 5: left: a collection of patterns presented to the neural network. right) The same set of patterns

In figure 2.2.1 we compare the both the weights w and the co-activations $p_i j$ estimated from both the rule for discrete signals and the continuous signals. As we can appreciate the results are the same in both cases.

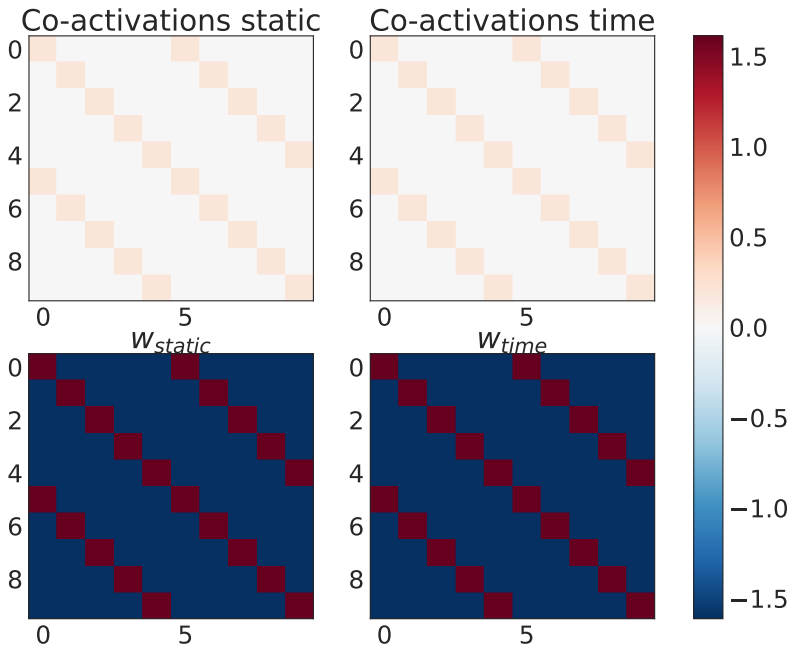
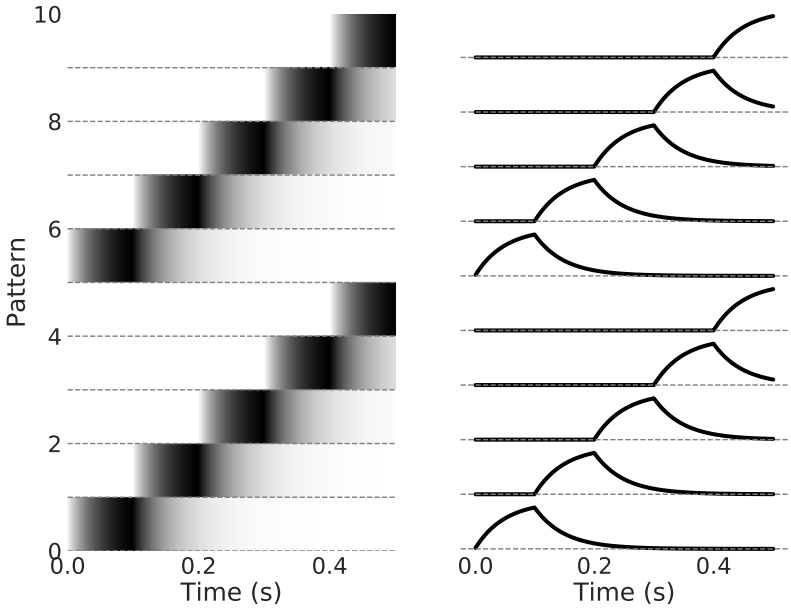


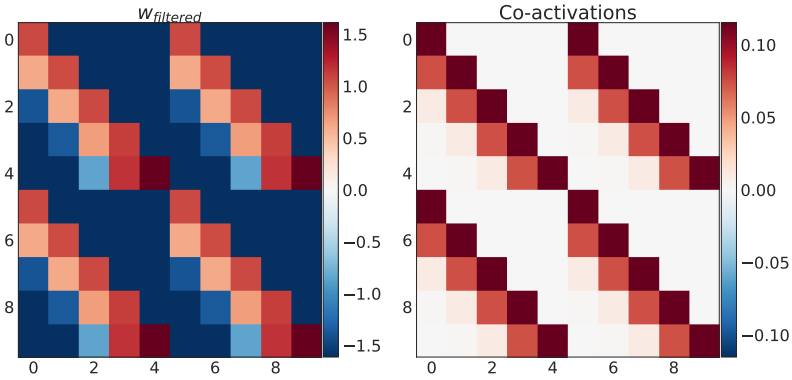
Figure 6: left) estimation of both co-activations and w for the discrete presentation of patterns. right) estimation of both co-activations and w for the continuous signal.

Finally if we assume that we are estimating the probabilities from signals whose information is delay and interpreted through a window of time

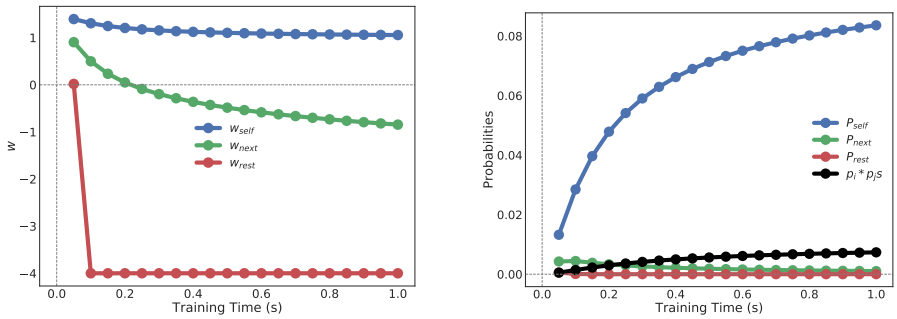
$$z(t) = \frac{1}{\tau_z} \int_{-\infty}^{t} s(\tau) e^{-\frac{t-\tau}{\tau_z}} d\tau \tag{14}$$



An example of such filtering of the signal is illustrated on figure 2.2.1. In the figure we can appreciate that the signal "leaks' on time and as a consequence there is some overlap between patterns that are contiguous on time where before there was none. It follows from there that the co-activations and in consequence the connectivity w will connect patterns that are close on time as illustrated on figure 2.2.1

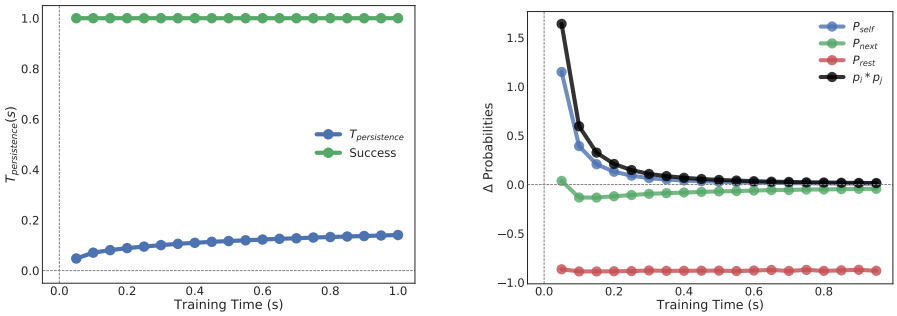


Training time



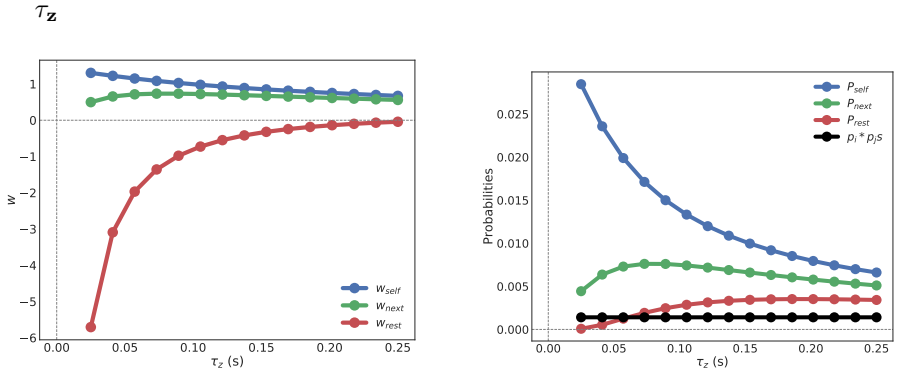
(a) (b)

Figure 7: Results of training with different training times. a) The weights after training. b) probabilities after training.



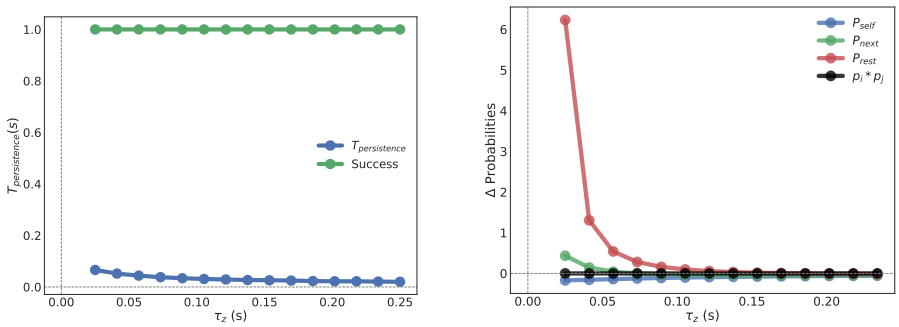
(a) (b)

Figure 8: Results of training with different training times. a) The weights after training. b) probabilities after training.



(a) (b)

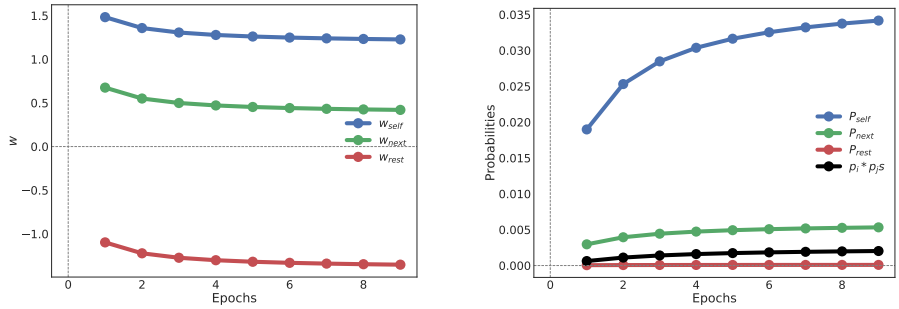
Figure 9: Results of training with different values of τ_z. a) The weights after training. b) probabilities after training.



(a) (b)

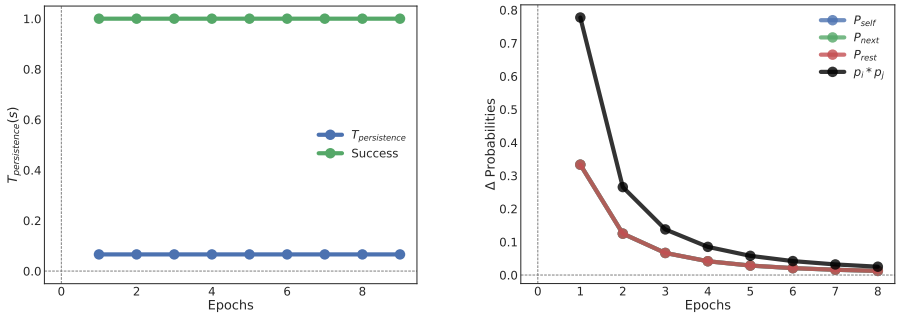
Figure 10: Results of training with different training times. a) The weights after training. b) probabilities after training.

Epochs



(a) (b)

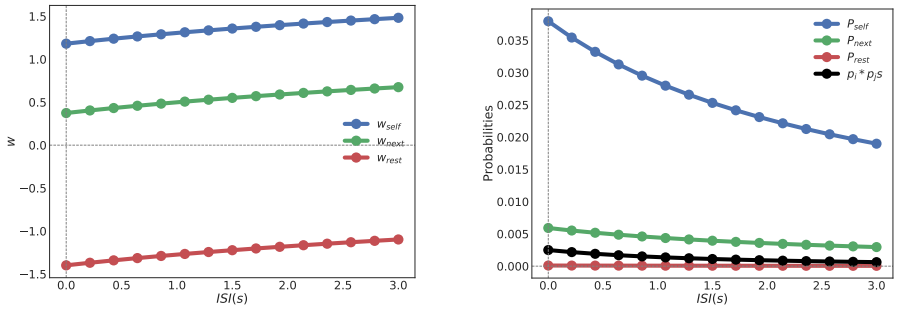
Figure 11: Results of training with different number of epochs. a) The weights after training. b) probabilities after training.



(a) (b)

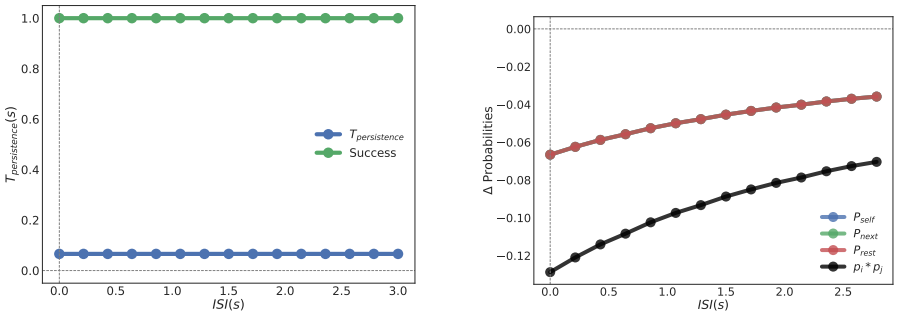
Figure 12: Results of training with different training times. a) The weights after training. b) probabilities after training.

Inter Sequence Interval



(a) (b)

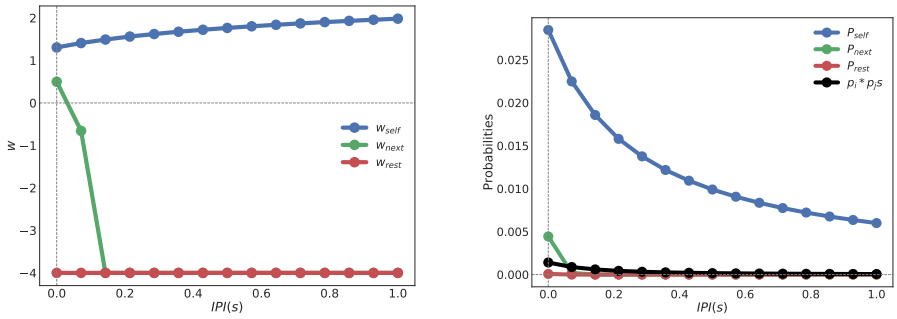
Figure 13: Results of training with different values of the inter sequence interval. a) The weights after training. b) probabilities after training.



(a) (b)

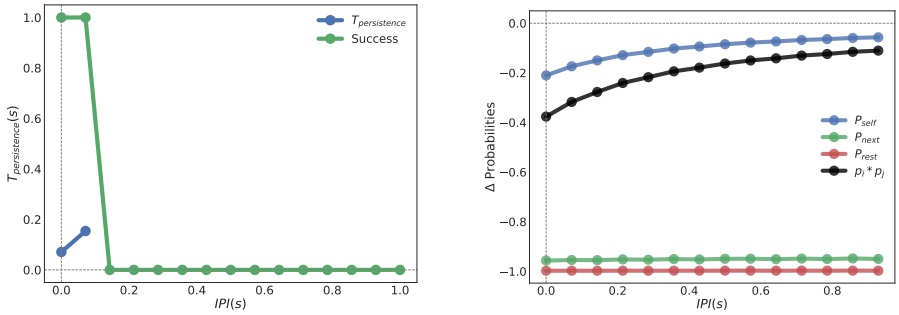
Figure 14: Results of training with different training times. a) The weights after training. b) probabilities after training.

Inter Pulse Interval



(a) (b)

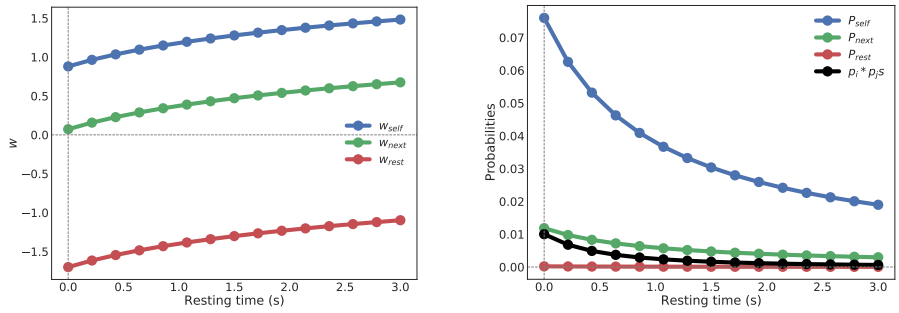
Figure 15: Results of training with different values of the inter pulse interval. a) The weights after training. b) probabilities after training.



(a) (b)

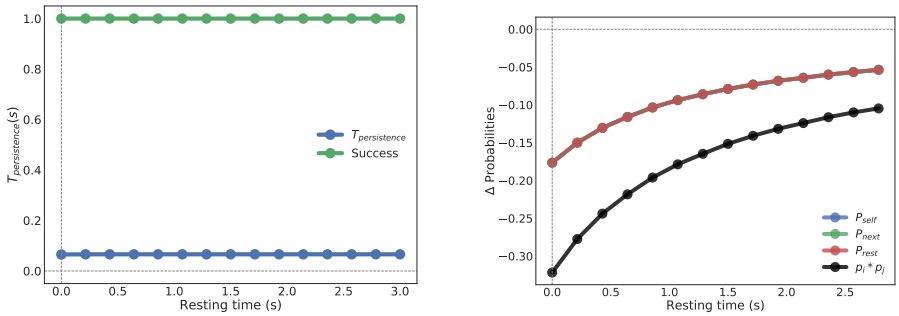
Figure 16: Results of training with different training times. a) The weights after training. b) probabilities after training.

Resting Time



(a) (b)

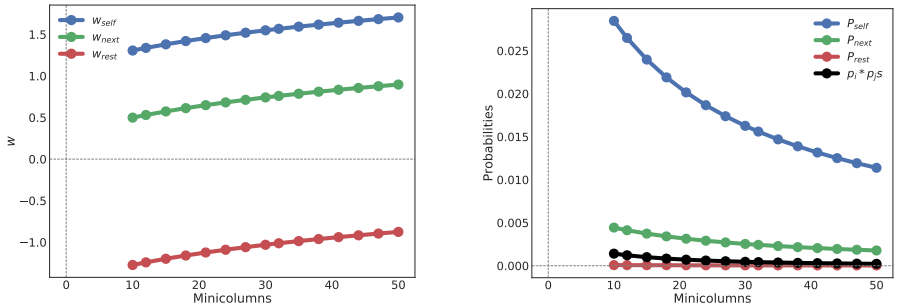
Figure 17: Results of training with different values of resting time. a) The weights after training. b) probabilities after training.



(a) (b)

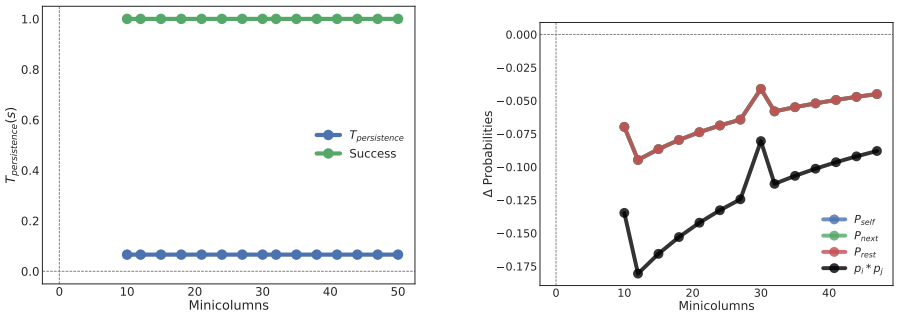
Figure 18: Results of training with different training times. a) The weights after training. b) probabilities after training.

Minicolumns



(a) (b)

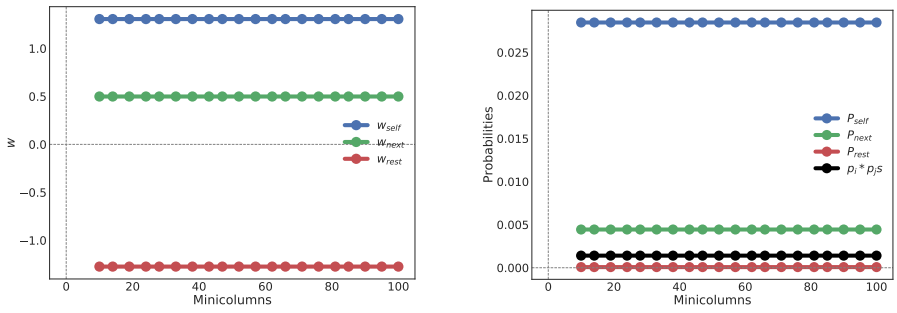
Figure 19: Results of training with different values of resting time. a) The weights after training. b) probabilities after training.



(a) (b)

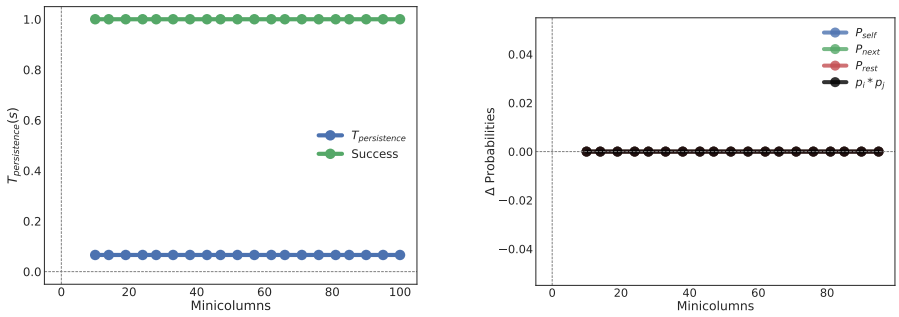
Figure 20: Results of training with different training times. a) The weights after training. b) probabilities after training.

Minicolumns - fixed sequence



(a) (b)

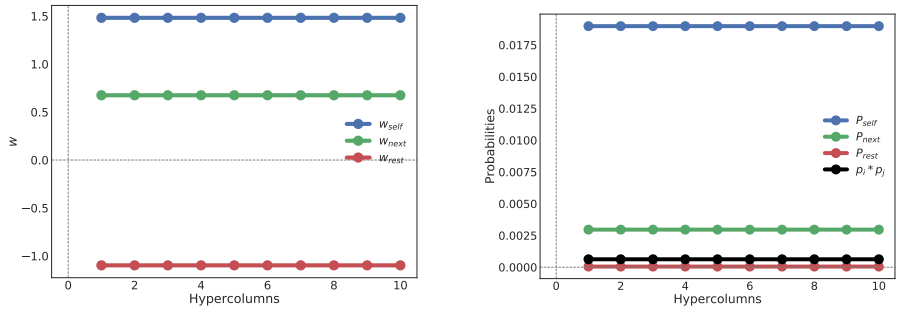
Figure 21: Results of training with different values of resting time. a) The weights after training. b) probabilities after training.



(a) (b)

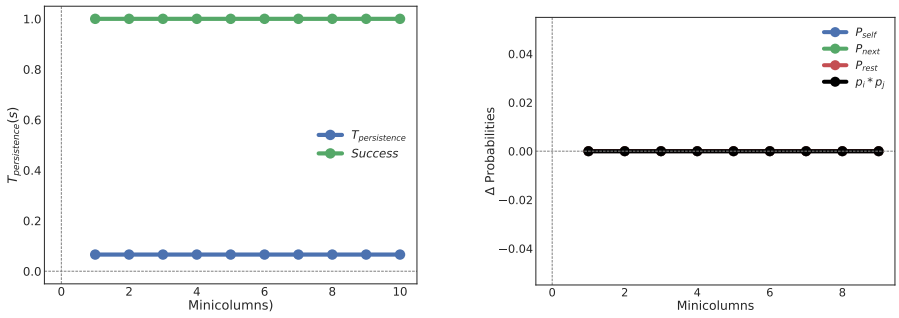
Figure 22: Results of training with different values of resting time. a) The weights after training. b) probabilities after training.

Hypercolumns



(a) (b)

Figure 23: Results of training with different values of resting time. a) The weights after training. b) probabilities after training.



(a) (b)

Figure 24: Results of training with different values of resting time. a) The weights after training. b) probabilities after training.

2.2.2 On-line vs off-line filters

Here we show the equivalence between the formulation of the filters as differential equations and the filters as convolutions.

$$\begin{aligned}
\tau_z \frac{dz}{dt} &= s(t) - z(t) \\
\tau_z \mathcal{F}\{\frac{dz}{dt}\} &= \mathcal{F}\{s(t)\} - \mathcal{F}\{z(t)\} && \text{Applying the Fourier transform} \\
\tau_z i\nu \mathcal{F}\{z(t)\} &= \mathcal{F}\{s(t)\} - \mathcal{F}\{z(t)\} && \text{Using the derivate rule} \\
\mathcal{F}\{z(t)\}(1 + \tau_z i\nu) &= \mathcal{F}\{s(t)\} \\
\mathcal{F}\{z(t)\} &= \mathcal{F}\{s(t)\} \left(\frac{1}{1 + \tau_z i\nu}\right) && \text{Factorizing and rearranging} \\
z(t) &= \mathcal{F}^{-1}\Big\{\mathcal{F}\{s(t)\} \cdot \left(\frac{1}{1 + \tau_z i\nu}\right)\Big\} && \text{Inverse} \\
z(t) &= \mathcal{F}^{-1}\Big\{\mathcal{F}\{s(t)\}\Big\} \circledast \mathcal{F}^{-1}\Big\{\frac{1}{1 + \tau_z i\nu}\Big\} && \text{Convolution} \\
z(t) &= \frac{1}{\tau_z} s(t) \circledast \mathcal{F}^{-1}\Big\{\frac{1}{\frac{1}{\tau_z} + i\nu}\Big\} && \text{Identity and rearranging} \\
z(t) &= \frac{1}{\tau_z} s(t) \circledast e^{-\frac{t}{\tau_z}} u(t) && \text{Heaviside function u(t)} \\
z(t) &= \frac{1}{\tau_z} \int_{-\infty}^{\infty} s(\tau) e^{-\frac{t-\tau}{\tau_z}} u(t-\tau) d\tau && \text{Definition} \\
z(t) &= \frac{1}{\tau_z} \int_{-\infty}^{t} s(\tau) e^{-\frac{t-\tau}{\tau_z}} d\tau && \text{Heaviside argument positive}
\end{aligned}$$

Analytical probabilities

We can calculate the probability of a single unit being activated with the following formula:

$$T_{total} * p_i = T_{activation} + \tau_z e^{\frac{T_{start} - T_{total}}{\tau_z}} \left(e^{\frac{T_{activation}}{\tau_z}} - 1 \right)$$

Where $T_{activation}$ is the the time that unit was activated during the course of the training. T_{start} is the time at which the unit was activated, and it takes into account frontier effects because of the filter. Finally T_{total} is the total training time.

We can also calculate the probability of the joint probability between two

units with the more complicated formula.

$$
\begin{aligned}
p_{21} &= \frac{A+B-C}{T_{total}} \\
A &= M_1\tau_{pre}\left(e^{\frac{-(T_s-T_1)}{\tau_{pre}}} - e^{\frac{-(T_s+T_2-T_1)}{\tau_{pre}}}\right) \\
B &= M_1\tau_p\left(e^{\frac{T_1}{\tau_{pre}}+\frac{T_s}{\tau_{post}}-\frac{T_s+T_2}{\tau_p}} - e^{\frac{T_1}{\tau_{pre}}+\frac{T_s}{\tau_{post}}-\frac{T_s}{\tau_p}}\right) \\
C &= M_1M_2\tau_p\left(e^{\frac{T_1}{\tau_{pre}}+\frac{T_s+T_2}{\tau_{post}}-\frac{T_{total}}{\tau_p}} - e^{\frac{T_1}{\tau_{pre}}+\frac{T_s+T_2}{\tau_{post}}-\frac{T_S+T_2}{\tau_p}}\right) \\
M_1 &= 1-e^{-\frac{T_1}{\tau_{pre}}} \\
M_2 &= 1-e^{-\frac{T_2}{\tau_{post}}} \\
\tau_p &= \frac{\tau_{pre}\tau_{post}}{\tau_{pre}+\tau_{post}}
\end{aligned}
$$

Where T_1 is the time that the first unit is activated, T_2 is the time that the second unit remained activated, T_s is the time at which the second unit become activated (If second unit is activated immediately after the first one then $T1 = T_s$), finally T_{total} is the total training time.

We can also calculate the joint probability of a unit with itself with the following formula

$$
\begin{aligned}
p_{11} &= \frac{A+B}{T_{total}} \\
A &= T_1 - M_{pre}\tau_{pre} - M_{post}\tau_{post} + M_p\tau_p \\
B &= \tau_p M_{pre}M_{post}\left(1-e^{-\frac{T_{total}-T_1}{\tau_p}}\right) \\
M_{pre} &= 1-e^{-\frac{T_1}{\tau_{pre}}} \\
M_{post} &= 1-e^{-\frac{T_1}{\tau_{post}}} \\
M_p &= 1-e^{-\frac{T_1}{\tau_p}} \\
\tau_p &= \frac{\tau_{pre}\tau_{post}}{\tau_{pre}+\tau_{post}}
\end{aligned}
$$

2.2.3 On-line rule

Once we know the dynamics of the system given a certain matrix the natural question to consider is whether we can learn the weight matrix. As described in [Sandberg et al., 2002] with the help of traces we can add on-line learning capabilities to the BCPNN neural network.

$$\tau_z \frac{dz_i}{dt} = o_i - z_i \tag{15}$$

$$\tau_p \frac{dp_i}{dt} = z_i - p_i \tag{16}$$

$$\tau_p \frac{dp_{ij}}{dt} = z_i z_j - p_{ij} \tag{17}$$

$$w_{ij} = \log(\frac{p_{ij}}{p_i p_j}) \tag{18}$$

$$\beta_i = \log(p_i) \tag{19}$$

We can understand the Bayesian nature of this learning rule by focusing our attention in equation 18. The argument of the logarithm weights co-activations of the two units against the base rate activation of the units multiplied by each other. When that two units are activated at the same time the co-activation p_{ij} of the units increases, if this is bigger than the base rate of the activation of each unit multiplied ($p_i p_j$) then the weight increases, otherwise it decreases. We translate this to a problem of linking on time by using the z-traces as described in equation 15 and illustrated in equation 25. If two units are close together in time there will be co-activation of the traces (illustrated in red) within a time scale of size τ_z. Finally we use a second low-pass filter with longer time constant to preserve the learning for a longer time-scale τ_p.

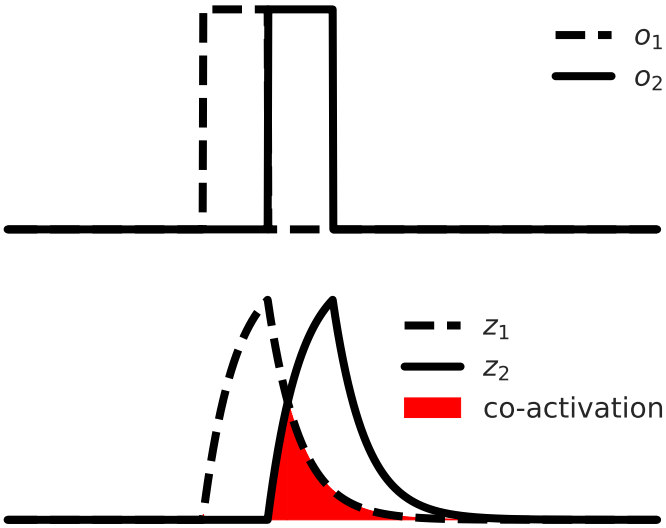


Figure 25: In red the intersection of two traces (co-activation) weighted against the base activation rate of each unit is responsible for the increase or decrease on the connectivity weight.

The training protocol:
Here we describe the training protocol. The first to note is that we train the neural network by clamping a given set of units in the order of a given sequence for a given time. More specifically a training protocol is characterized by the following quantities. First there is the **training time** for a given element of the sequence, that is, the time that element remains activated. It may well be that the an element of the sequences does not follow the other immediately, in order to account for this possibility we have the **inter pulse interval (IPS)** which is just the time elapsed between a pattern and its successor. It is also the case that a given sequence of elements is often presented more than once, we call **epoch** to a particular instance of a sequence pretension. Finally, we usually leave some time between each of the epochs during the presentation of the training protocol, this is know as the **inter sequence interval (ISI)**. We present a schematic of the training protocol in figure 26.

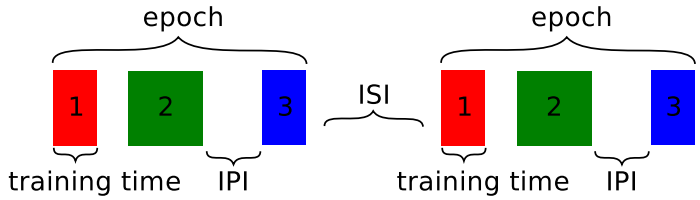


Figure 26: The training protocol. IPI stands for inter pulse interval and ISI for inter sequence interval. Explanations in the text.

In order to illustrate how this looks in practice we illustrate with a successful training example in figure 27. In this case we pass the training protocol consisting of two epochs and half a second of ISI as illustrated in b). Regarding the evolution of the weight dynamics, every time that there is a coincidence on the traces as in a) we will have a respective increase in the connectivity matrix as we see in c) marked with red shading. If the training process works properly we end up with a matrix as the one in f) which posses similar structure as the on in figure 3 d). The recall protocol consists on clamping the first pattern of the sequence for a given amount of time ($T_{cue} = 100$ (ms) in this case) and then letting the dynamics of the network evolve on its own. In this case we can appreciate that every element is recalled in the correct order.

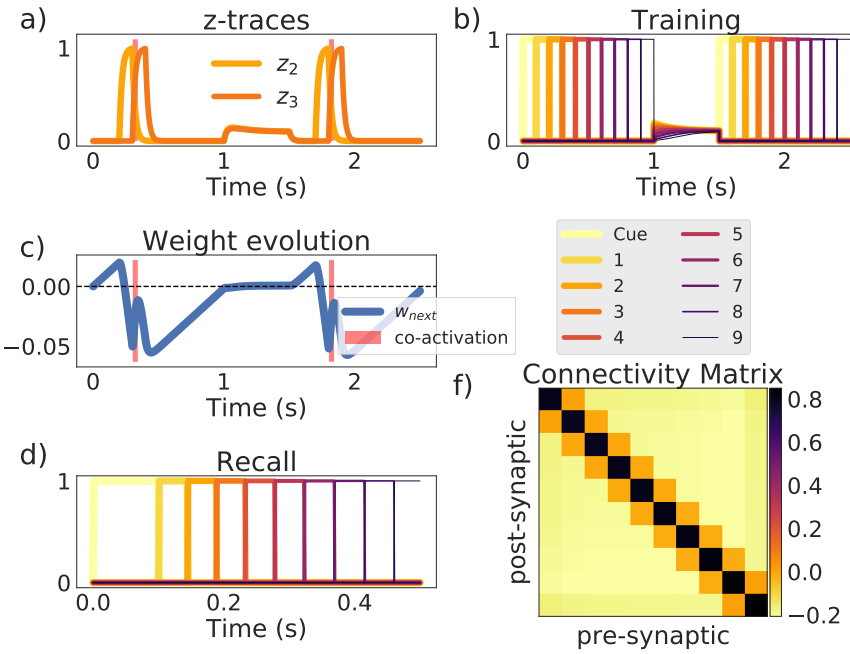


Figure 27: An example of a successful training and recall. .

Now that we have the training protocol we will study how the different weights vary with the training protocol parameters. We ran the training protocol

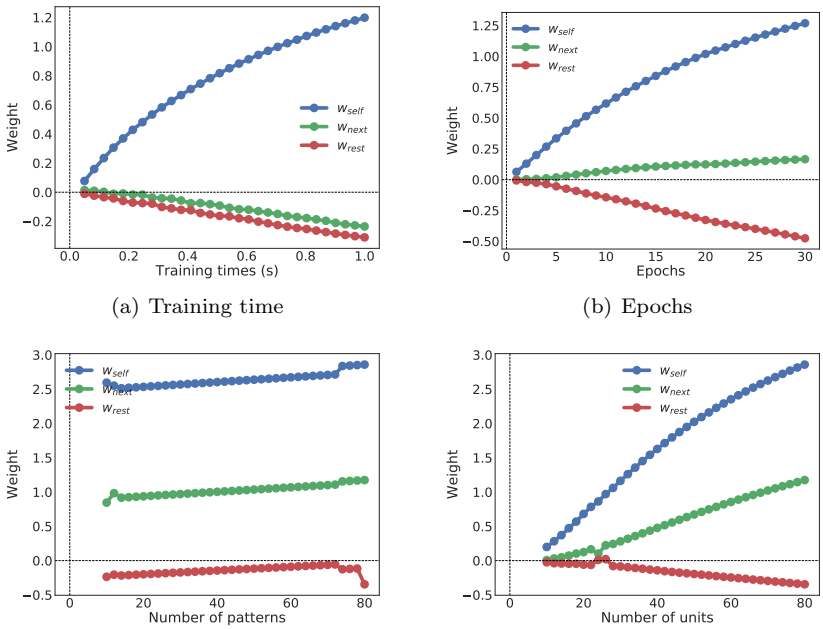
with a given set of parameters and we examine the values of the connectivity matrix at the end of the training protocol. We characterized the w_{self} as the self-connection of unit 2 with itself, w_{next} on the other hand is given by the weight of the connection between units 2 and 3. Finally we calculated w_{rest} as the average of the connections weights from unit two to all the other units. We studied the effects on learning of the training time, the number of epochs, the number of patterns and finally the number of units.

Training time: in figure 28 a) we show the results of learning with different training times on the connectivity matrix. Using the Bayesian interpretation of the learning rule we can explain why the w_{self} is increasing with the training time. The longer the training time the longer the unit is activated which means the co-activation of the unit with itself is bigger, which in turn leads to a bigger weight. Using the reverse of this statement is easy to see why w_{next} and w_{rest} are decreasing. The BCPNN learning rules weights the co-activation of the units against their individual activation. As the training time grows for different units the latter becomes bigger and bigger and therefore the decreasing effect on weights that we observe on the graph.

Epochs: in figure 28 b) we show the results that training with different number of epochs has on the connectivity matrix. This graph is better understood in terms of the dynamics of learning reaching to their natural equilibrium. The co-activation to independent activation ratio is dynamical process that follows the equations, the longer the dynamics run the closer this dynamic approaches its steady-state behavior. In the case of w_{self} and w_{next} there are definitive ratios of activation given by the dynamic and the system converges to it. The w_{rest} quantity however, does not. This reflects the fact that the units never activate together and the learning rule keeps accumulating evidence of it.

Number of patterns: in figure 28 c) we depict the outcome that having a different number of patterns in the sequence has on the connectivity matrix. In terms of the Bayesian nature of the learning rule this graphs can be explained in the following way. When we increase the number of patterns we are making the overall probability space bigger (number of possible ways in which the network could possible be). This makes the any two units co-activation every more meaningful, this is the reason why w_{self} and w_{next} increase. On the other hand, the network has more and more time to accumulate evidence of the lack of co-activation for the rest of the units and therefore w_{rest} decreases.

Number of minicolumns in figure 28 d) we illustrate the effects of the number of units on the connectivity matrix. The reasoning is analogue to the one in the number of pattern, but the effect is more pronounced because we are making the network bigger overall.



(a) Training time

(b) Epochs

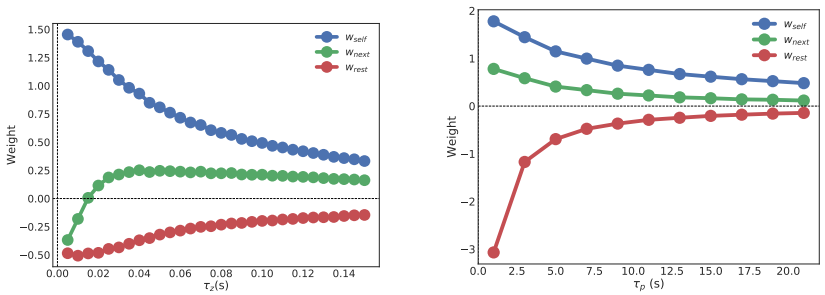
(c) Number of patterns

(d) Number of units

Figure 28: Dynamics of weight learning. a) effects of the training time on learning. b) effects of the number of epochs on training. c) effects of the number of patterns on learning. d) effects of the number of units on learning. See text for explanation.

Effects of learning on the persistent time: in terms of the dynamics of the persistence time we can sketch out how the different training regimes will affect the behavior of the former quantity. The effect of an increased training time increases the distance between w_{self} and w_{next}, as we haven seen in figure 4 d) this increases the persistent time $T_{persistence}$.

Learning Dynamic Parameters: We have studied above the effects of the training protocol we can study the effects of the time constants in equations 15 and 16. We show both of this results in figure 29.



(a) (b)

Figure 29: Connectivity matrix dependency on the learning parameters. a) Dependence on τ_z. We appreciate than the distance between w_{self} and w_{next} becomes smaller as τ_z increases. b) Dependence on τ_p. In this case the relationship between w_{self} and w_{next} is not so pronounced, although there is a little bit of a change.

2.3 Robustness of the simple model to noise

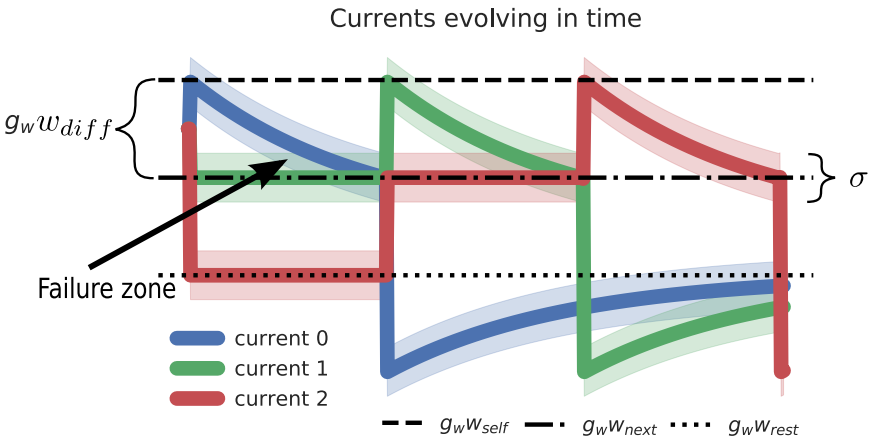
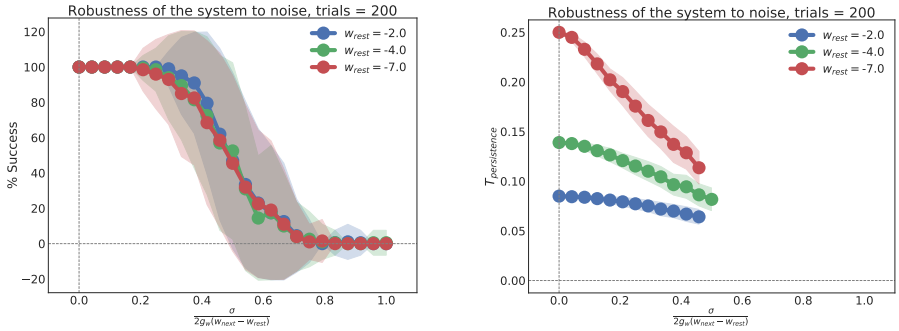
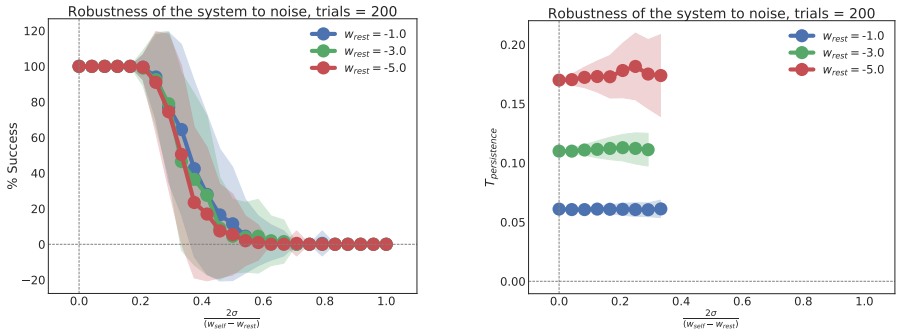


Figure 30: A schematic of how noise works in the system.



(a) (b)

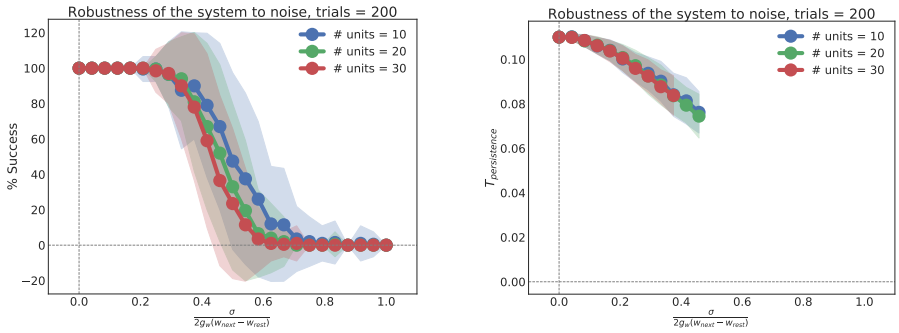
Figure 31: The effects of noise for different connectivity matrices.



(a) (b)

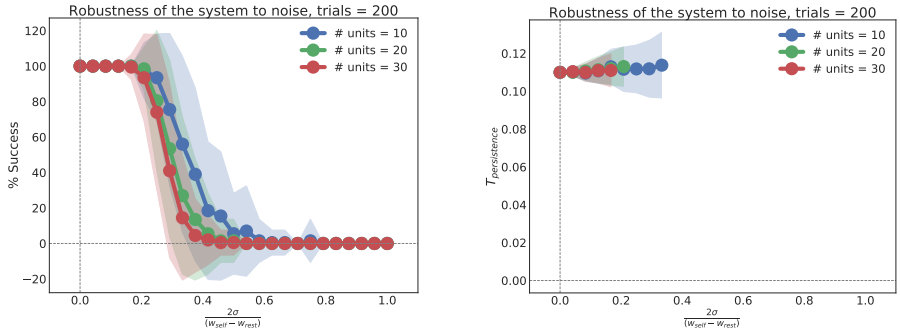
Figure 32: The effects of noise for different connectivity matrices.

Noise on scale



(a) (b)

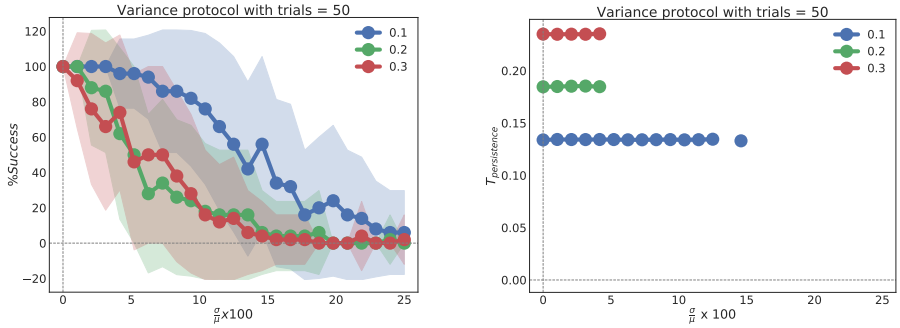
Figure 33: The effects of noise for different connectivity matrices.



(a) (b)

Figure 34: Effect of weight noise on the scaling of the network

2.4 Robustness of the simple model to different training times



(a) (b)

Figure 35: Variance on the training protocol

3 The problem Of Complex Sequences

Here it is one paper [Guyon et al., 1988]

4 z-traces as activations

If we assume that the currents adapt instantaneously to the currents ($\tau_s \to 0$) at the point where the unit n is activated we have

$$s_{n+1}(t) = c_{n+1}(t) + \beta_{n+1}$$
$$s_n(t) = c_n(t) + \beta_n - a(t)$$

We need the find at which this currents are equal $s_{n+1}(t) = s_n$ which is equivalent to the point in which the difference on the currents $c(t)$ are equivalent to the adaptation.

$$g_a(1 - e^{\frac{-t}{\tau_a}}) = \Delta\beta + \underbrace{(c_n(t) - c_{n+1}(t))}_{A}$$

We want an expression for A, we know that that the terms of $z_k(t) = 0$ for $k > n$ (those units have not been activated before) so we have:

$$A = \sum_{k=0}^{n} (w_{n,k} - w_{n+1,k}) z_k(t)$$
$$A = (w_{self} - w_{next})(1 - e^{\frac{-t}{\tau_z}}) + \underbrace{\alpha}_{w_{rest} - w_{next}} e^{\frac{-t}{\tau_z}} (1 - e^{\frac{-T_{per}}{\tau_z}}) \underbrace{\sum_{k=0}^{n} e^{\frac{kT_{per}}{\tau_z}}}_{geometric\ series}$$

The term in the geometric series is equal to $\frac{(1-e^{\frac{-nT}{\tau_z}})}{(1-e^{\frac{-T}{\tau_z}})}$. Her we used the fact that the connectivity matrix produced with the BCPNN learning rule is linear in the differences between the weights (α).

We therefore have:

$$g_a(1 - e^{\frac{-t}{\tau_a}}) = \Delta\beta + (w_{self} - w_{next})(1 - e^{\frac{-t}{\tau_z}}) + \alpha e^{\frac{-t}{\tau_z}} (1 - e^{\frac{-nT}{\tau_z}})$$

If we assume that n is big enough that the term in the right becomes 1 we can solve the following equation for getting the persistent time:

$$g_a(1 - e^{\frac{-t}{\tau_a}}) = \Delta\beta + (w_{self} - w_{next})(1 - e^{\frac{-t}{\tau_z}}) + \alpha e^{\frac{-t}{\tau_z}}$$

For small persistent times with respect to τ_z and τ_a this can be approximated by:

$$T_{per} = \frac{B + \frac{\Delta w_{rest}}{g_a}}{\frac{1}{\tau_a} + \frac{\Delta w_{rest}}{g_a \tau_z}}$$

Decomposing only one of the currents

The solution for the deterministic trajectories of the currents as is

We need to find an expression for the term $c_n(t) = \sum_{k=0}^{N} w_{n,n-k} z_{n-k}(t)$ to find the current at each point in time at the n pattern in a sequence. Where N is the sequence length

We can decompose the term in the following way

$$c_n(t) = w_s(1 - e^{\frac{-t}{\tau_z}}) + w_n(1 - e^{\frac{-T_{per}}{\tau_z}})e^{\frac{-t}{\tau_z}} + w_{rest}(1 - e^{\frac{-T_{per}}{\tau_z}}) \left(\sum_{k=1}^{n-1} k e^{-kT_{per}} \right) e^{\frac{-t}{\tau_z}}$$

Solve the differential equation for small τ_z

The general expression for the current if the T_{per} is big compared to τ_z.

$$\begin{aligned} s_{act}(t) &= \beta + w_s - g_a - \frac{X_1}{1 - r_z} e^{\frac{-t}{\tau_z}} + g_a e^{\frac{-t}{\tau_a}} \\ &\quad + \left(s_0 - \beta - w_s + g_a + \frac{w}{1 - r_z} - \frac{g_a}{1 - r_a} \right) e^{\frac{-t}{\tau_s}} \end{aligned} \tag{20}$$

$$s_{inac}(t) = \beta + w_n - \frac{X_2}{1 - r_z} e^{\frac{-t}{\tau_z}} + \left(s_o - \beta - w_s + \frac{X_z}{1 - r_z} \right) e^{\frac{-t}{\tau_s}} \tag{21}$$

Where $X_1 = w_s - w_n + w_n e^{\frac{T}{\tau_z}}$ and $X_2 = w_n - w_r + w_r e^{\frac{-t}{\tau_a}}$

References

[Abeles et al., 1995] Abeles, M., Bergman, H., Gat, I., Meilijson, I., Seidemann, E., Tishby, N., and Vaadia, E. (1995). Cortical activity flips among quasi-stationary states. *Proceedings of the National Academy of Sciences*, 92(19):8616–8620.

[Agster et al., 2002] Agster, K. L., Fortin, N. J., and Eichenbaum, H. (2002). The hippocampus and disambiguation of overlapping sequences. *Journal of Neuroscience*, 22(13):5760–5768.

[Almeida-Filho et al., 2014] Almeida-Filho, D. G., Lopes-dos Santos, V., Vasconcelos, N. A., Miranda, J. G., Tort, A. B., and Ribeiro, S. (2014). An investigation of hebbian phase sequences as assembly graphs. *Frontiers in neural circuits*, 8:34.

[Averbeck et al., 2002] Averbeck, B. B., Chafee, M. V., Crowe, D. A., and Georgopoulos, A. P. (2002). Parallel processing of serial movements in prefrontal cortex. *Proceedings of the National Academy of Sciences*, 99(20):13172–13177.

[Crowe et al., 2010] Crowe, D. A., Averbeck, B. B., and Chafee, M. V. (2010). Rapid sequences of population activity patterns dynamically encode task-critical spatial information in parietal cortex. *Journal of Neuroscience*, 30(35):11640–11653.

[Euston et al., 2007] Euston, D. R., Tatsuno, M., and McNaughton, B. L. (2007). Fast-forward playback of recent memory sequences in prefrontal cortex during sleep. *science*, 318(5853):1147–1150.

[Fujisawa et al., 2008] Fujisawa, S., Amarasingham, A., Harrison, M. T., and Buzsáki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature neuroscience*, 11(7):823.

[Guyon et al., 1988] Guyon, I., Personnaz, L., Nadal, J., and Dreyfus, G. (1988). Storage and retrieval of complex sequences in neural networks. *Physical Review A*, 38(12):6365.

[Harvey et al., 2012] Harvey, C. D., Coen, P., and Tank, D. W. (2012). Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, 484(7392):62.

[Ji and Wilson, 2007] Ji, D. and Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100.

[Johansson, 1973] Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211.

[Johnson et al., 2010] Johnson, H. A., Goel, A., and Buonomano, D. V. (2010). Neural dynamics of in vitro cortical networks reflects experienced temporal patterns. *Nature neuroscience*, 13(8):917.

[Jones et al., 2007] Jones, L. M., Fontanini, A., Sadacca, B. F., Miller, P., and Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proceedings of the National Academy of Sciences*, 104(47):18772–18777.

[Lapish et al., 2008] Lapish, C. C., Durstewitz, D., Chandler, L. J., and Seamans, J. K. (2008). Successful choice behavior is associated with distinct and coherent network states in anterior cingulate cortex. *Proceedings of the National Academy of Sciences*.

[Lashley, 1951] Lashley, K. (1951). The problem of serial order in behavior. In *Cerebral mechanisms in behavior*, pages 112–136.

[Levy et al., 2001] Levy, N., Horn, D., Meilijson, I., and Ruppin, E. (2001). Distributed synchrony in a cell assembly of spiking neurons. *Neural networks*, 14(6-7):815–824.

[Mattia et al., 2013] Mattia, M., Pani, P., Mirabella, G., Costa, S., Del Giudice, P., and Ferraina, S. (2013). Heterogeneous attractor cell assemblies for motor planning in premotor cortex. *Journal of Neuroscience*, 33(27):11155–11168.

[Miyashita, 1988] Miyashita, Y. (1988). Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, 335(6193):817–820.

[Murray et al., 2017] Murray, J. M. et al. (2017). Learning multiple variable-speed sequences in striatum via cortical tutoring. *eLife*, 6:e26084.

[Nakajima et al., 2009] Nakajima, T., Hosaka, R., Mushiake, H., and Tanji, J. (2009). Covert representation of second-next movement in the pre-supplementary motor area of monkeys. *Journal of neurophysiology*, 101(4):1883–1889.

[Recanatesi et al., 2017] Recanatesi, S., Katkov, M., and Tsodyks, M. (2017). Memory states and transitions between them in attractor neural networks. *Neural computation*, 29(10):2684–2711.

[Roach et al., 2016] Roach, J. P., Sander, L. M., and Zochowski, M. R. (2016). Memory recall and spike-frequency adaptation. *Physical Review E*, 93(5):052307.

[Sandberg et al., 2002] Sandberg, A., Lansner, A., Petersson, K., and Ekeberg, O. (2002). A bayesian attractor network with incremental learning. *Network: Computation in neural systems*, 13(2):179–194.

[Scarpetta and Giacco, 2013] Scarpetta, S. and Giacco, F. (2013). Associative memory of phase-coded spatiotemporal patterns in leaky integrate and fire networks. *Journal of Computational Neuroscience*, 34(2):319–336.

[Seidemann et al., 1996] Seidemann, E., Meilijson, I., Abeles, M., Bergman, H., and Vaadia, E. (1996). Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task. *Journal of Neuroscience*, 16(2):752–768.

[Tully et al., 2016] Tully, P., Lindén, H., Hennig, M., and Lansner, A. (2016). Spike-based bayesian-hebbian learning of temporal sequences. *PLoS computational biology*, 12(5):e1004954.

[Verduzco-Flores et al., 2012] Verduzco-Flores, S. O., Bodner, M., and Ermentrout, B. (2012). A model for complex sequence learning and reproduction in neural populations. *Journal of computational neuroscience*, 32(3):403–423.

[Von Ehrenfels, 1988] Von Ehrenfels, C. (1988). On "gestalt qualities.". *B. Smith (Ed. & Trans.), Foundations of Gestalt theory*, pages 82–117.

[Weber et al., 2013] Weber, A. I., Saal, H. P., Lieber, J. D., Cheng, J.-W., Manfredi, L. R., Dammann, J. F., and Bensmaia, S. J. (2013). Spatial and temporal codes mediate the tactile perception of natural textures. *Proceedings of the National Academy of Sciences*, page 201305509.

[Xu et al., 2012] Xu, S., Jiang, W., Poo, M.-m., and Dan, Y. (2012). Activity recall in a visual cortical ensemble. *Nature neuroscience*, 15(3):449.

[Yuille and Geiger, 1998] Yuille, A. L. and Geiger, D. (1998). Winner-take-all mechanisms, the handbook of brain theory and neural networks.