Mini-project in Neural Networks and Biological Modeling

Modelling the Stroop effect with a Hopfield network

by Barde Paul & Ollitrault Pauline

Ecole polytéchnique fédérale de Lausanne
Laboratory of Computational Neuroscience
Professor Wulfram Gerstner
Assistant: Vasiliki Liakoni
May 30, 2016

Contents

1 Introduction

Reading is a controlled process at an early stage. However, by practicing, it turns into an automatic action [1]. At one point, one does not need to read the sounds off the syllables but the word is automatically recognised. Thus, the process is fast and requires less attention. Therefore, reading can perturbe another task that is contextually related to reading and that is not automatic by conflicting associations. This effect is well illustrated by J. R. Stroop's famous experiment in which patients are asked to tell the color of a word describing another color. For instance on figure 1.01, for the very first word ("Blue"), the patient should say "Red".

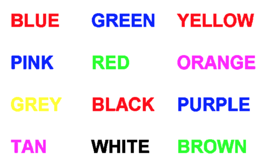


Figure 1.1: The Stroop effect. [2]

Stroop's paper describing the eponyme effect is today the most cited paper in the field of experimental psychology [3]. Many theories were built up in order to explain it. More recently, the development of computer based technologies led to the performance of several studies aiming to model the Stroop effect. These were established using different neural models such as the Hopfield model.

In the first part of this project, we will implement a standard Hopfield network. Pattern retrieval experiments will be performed in order to test it. Certain parameters (flip ratio, number of neurons, number of patterns...) will be varied in order to study the performance of the implemented network.

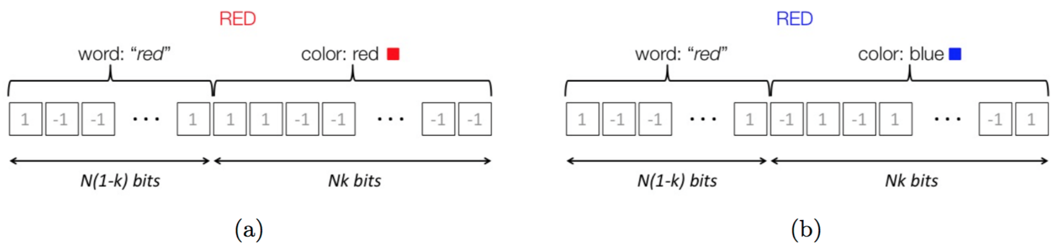


Figure 1.2: Representation of (a) a congruent Stroop pattern: the ink color corresponds to that of the word and (b) an incongruent pattern: the ink and the word colors do not match.

In the second part, we will model the Stroop effect. To do so we will have to create the congruent and incongruent Stroop patterns as shown on figure 1.0.2. The subject's brain will be trained on the congruent patterns before performing the experiment on the incongruent ones. Again several parametters such as k or the number of colors will be varied in order to study their effect on the experiment. Moreover, a physical intuition of the results will be provided. Finally, we will try to understand the limits of our implementation of the Hopfield network and discuss ways to improve it.

2 Exercise I: Standard Hopfield network

2.1 Getting started

Several functions define our Hopfield network and are introduced and explained in the appendix A.1.

2.2 Normalized pixel distance

The retrieval error is defined as the normalized pixel distance (percentage of pixels in the network S which differ from the pattern ξ^μ) of the network state S to the pattern ξ^μ after convergence.
The number of equal neurons is:

$$N_{eq} = \frac{1 + \sum_{i=1}^{N} \xi_i^\mu S_i(t)}{2} \tag{2.1}$$

indeed when for each neurons $\xi_i^\mu \neq S_i(t)$, $N_{eq} = 0$; if for each neurons $\xi_i^\mu = S_i(t)$, $N_{eq} = 1$ and finally if half of $\xi_i^\mu = S_i(t)$ and half $\xi_i^\mu \neq S_i(t)$, $N_{eq} = 0.5$.
Therefore the percentage of different neurons is:

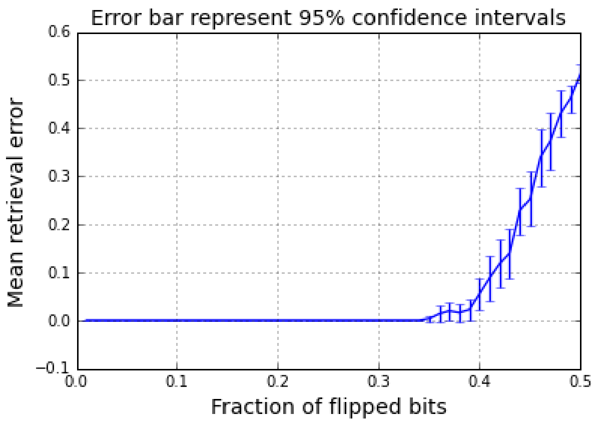
$$d^\mu = 1 - \frac{1}{N}\left(\frac{1 + \sum_{i=1}^{N} \xi_i^\mu S_i(t)}{2}\right) \tag{2.2}$$

which can be rewritten as:

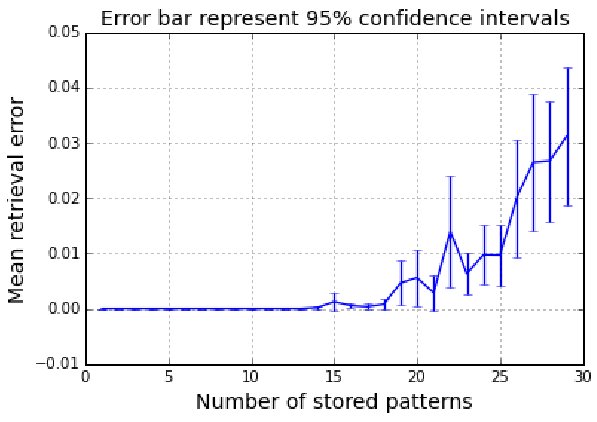
$$d^\mu = \frac{1}{N}\left(\frac{\sum_{i=1}^{N} 1 - \xi_i^\mu S_i(t)}{2}\right) = \frac{1}{2}\Big(1 - m^\mu\Big) \tag{2.3}$$

2.3 Pattern retrieval

We defined a system of N=200 neurons and P=5 patterns. The fraction of flipped pixels between the initial system and the pattern to be retrieved is $0.01 \leq c \leq 0.5$ (0.01 resolution). We tried 40 times to retrieve a randomly chosen pattern μ for each value of c and we plotted the mean retrieval error as a function of c (see figure 2.2.1).
Then we fixed the flipped bits ratio to $c = 0.1$, value at which it won't affect the mean retrieval error. We varied the number of stored patterns $1 \leq P \leq 30$ and 100 realizations were performed in order to obtain the mean retrieval error as a function of P. We chose to set the interval of P to $[1, 30]$ because the memory capacity of a Hopfield network is approximately $C_{store} = \frac{P^{max}}{N} \approx 14\%$. In our case this corresponds to 28 patterns. As the scale is a lot smaller than for the ratio of flipped bits, we performed 100 realizations to get a curve as smooth as possible.
The results of these two experiments are displayed on figure 2.1.1. Figure 2.1.1 (a) shows an expected behavior of the network: the further the initial state is from the pattern to be retrieved, the larger the mean retrieval error. On figure 2.1.1 (b) the fluctuations are more obvious due to the small scale but the expected trend is respected.



(a)



(b)

Figure 2.1: The mean retrieval error with error bars of 95% confidence intervals as a function of (a) the ratio of flipped bits c and (b) the number of stored patterns P.

2.4 Capacity estimation

We know define the *capacity* of a Hopfield network of size N to be the maximal number of patterns $P_{N,max}$ that can be stored such that the retrieval error does not exceed 2%. We can also define the maximal load $\alpha_{N,max} = \frac{P_{N,max}}{N}$.
In order to find the maximal load of our model, we set $c = 0.1$ and we computed $P_{N,max}$ for networks of N=100, 250 and 500 neurons. The results are shown in the following table.

N	Number of realizations	mean $\alpha_{N,max}$	confidence intervals (95%)
100	40	0.1258	6.15e-3
250	40	0.1269	3.59e-3
500	10	0.1272	3.85e-3

Table 2.1: Maximal load values and confidence intervals resulting from the patterns retrieval experiments performed on networks of 100, 250 and 500 neurons.

It is difficult to compare our results with litterature. Indeed, the majority of articles define the memory capacity in a different way: they consider the number of learned pattern that are stable if they are taken to initialize the network. In our case, we look for patterns that can be retrived from a initial pattern containing 10% of flipped bits. The litterature predicts a maximal load $0.135 \leq \alpha_{max} \leq 0.15$ [4][5]. The maximal load seems to increase with increasing number of neurons: the capacity does not linearly increase with the number of neurons.
We can consider these results as a validation of our model and we can thus use it to model the Stroop effect.

3 Exercise II: Modelling the Stroop effect

3.1 Comment on the implementation

The functions implemented for this exercise are described in appendix A.2. The different retrieval and convergence criteria are also stated there.

3.2 Getting started

After several trials with the given parameters we see that the subjects identifies the ink color or the word color with approximately the same probability. This is understandable since we give the same amount of information on the ink than on the word and that there is no part of the implementation that asks the patient to focus more on the ink color than on the word color.

There is a non negligible number of trials where the subjects identifies a different color than the ink color or the word color, from now on we will call these errors "unexpected errors". This does not seem realistic since it would imply that a person looking at the word "red" written in blue sometimes answers yellow. Such an answer is rather unlikely if the subject is not under a large amount of stress and focuses a minimum. However, we might eventually imagine a configuration where this could happen. Imagine the word "purple" written in red, mixing these two information one might want to answer pink. We think that such results arises when we have a lot of stored patterns compared to the number of neuron. Hence, there is more possible overlap between the congruent patterns (we never require them to be orthogonal). Thus the initial incongruent pattern might be closer to a congruent pattern coding a totally different color but which ink and word parts put together are closer than the one of the congruent patterns involved in the incongruent pattern used to initialize the network. Finally, in exercise I we derived a capacity estimation around 13% so 13 storable patterns for a network of 100 neurons like the one considered here. Yet, this has been derived for a flip ration of 0.1; here the flip ratio is not explicit but should somehow scale like $(k-1)$ the ratio of the Wordpattern bits : 0.5 in this case.

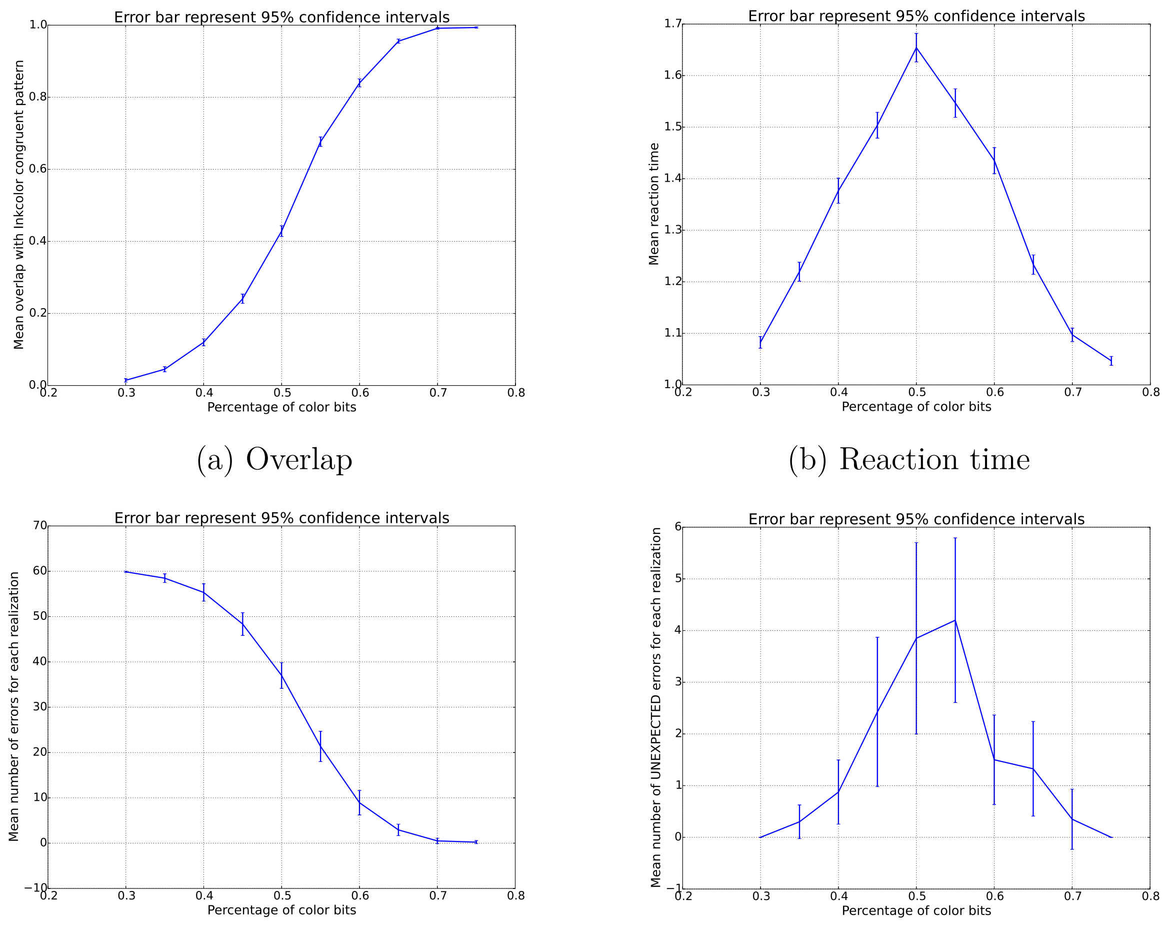
3.3 The impact of k

In order to have enough experiments to perform meaningful statistical analysis we ran the test on 40 subjects ($N_{realization} = 40$) for which we create a new network and learn new congruent patterns. Then, for each of these subjects, we present them 60 coloured words ($N_{trial} = 60$) or in other terms 60 incongruent patterns. We counted the errors as discussed previously: the unexpected errors are counted with the regular errors. However, we also counted separately these unexpected errors to see what might influence them. The statistical quantities rely on two different populations:

- On the first hand, the overlap and the reaction time are based on all the realizations and all the trials for one given percentage of ink color bits k. Therefore, at fixed k we look at $N_{realization} \times N_{trial}$ events. We average over all the subjects and over all their answers.

- On the second hand, the errors are averaged only over the realizations. This means that for a given k, we count the error made by each subject on their N_{trial} words and then we average over the number of subjects. The average is done on $N_{realization}$ events representing the number of mistakes of each subject.

The results are shown in the following figure:



(a) Overlap

(b) Reaction time

(c) Regular Errors

(d) Unexpected errors

Figure 3.1: Influence of the percentage of ink color bits

Let's first give a physical interpretation to the percentage of ink color bits. If it is low it means that proportionally we give more information about the word than about its color. For example, a clearly written word with a very pale color. On the other hand, a high percentage of ink color represents a poorly written word with a flashy color.

From the evolution of the overlap curve we see that the more information we give about the ink color the more it is retrieved. This is easily understandable. When the color is pale, the subject focuses on the word and retrieves its color, hence the overlap with the ink color is close to zero since the congruent patterns are randomly chosen and thus have small overlaps. When the color is strong the subject retrieves it easily and the overlap is close to 1. Notice that one would expect the mean overlap to be 0.5 at $k = 0.5$ meaning that when the information is evenly distributed between the word and the color the subject retrieves either the color (overlap of 1) or the word (overlap of 0) with the same probability, hence a mean overlap of 0.5. However, we see that a mean overlap of 0.5 is archived for a k above 0.5. This shows that we have to account for the "unexpected" errors that diminishes the mean overlap for a k close to 0.5. Indeed, in this range we see that 4 out of 60 words (fig (d)) yield unexpected errors and therefore a ink overlap close to 0. In brief, unexpected errors give an additional cause for error and hence a diminution of the mean overlap.

The analysis of the number of errors is quite similar than the one of the overlap. Here again we notice the effect of the unexpected errors (in fact unexpected errors are counted as regular errors in this plot). Indeed, one would again expect that for $k = 0.5$ the subject would make only 30 errors, meaning that he retrieves the ink color as often as the word. Nevertheless, we see that the subject is in reality wrong about 35 times out of 60. We can observe in (d) that this 5 additional errors are unexpected errors. Moreover, we notice in (d) that unexpected errors only occur for medium values of k (close to 0.5) and this is why the limits of the errors curve are 60 (all wrong) and 0 (all right): in these limits the unexpected errors do not affect the subject.

Why are the unexpected errors more present for k around 0.5? Well in this range the subject gets the same amount of information about the ink and the word. Therefore, the initial incongruent pattern is highly similar to the two congruent patterns, hence the subject is more likely to be influenced by any resemblance with a congruent pattern of another color.

Finally, regarding the reaction time we notice that it is really small for extreme values of k. In this case, the initial pattern is close to a pattern known by the subject and hence a small effort is needed (few iterations) to retrieve it. However, for partial information, the time to retrieve a given pattern is higher since more effort is required, the initial pattern is far from anything known.

Globally, the behavior of the network is close to what we would expect in a Stroop experiment. Nevertheless, there is important difference that should be stressed: in a real experiment the subject is asked to retrieved the **ink** color, thus there is a bias in his choice and the information on the ink color is more important, weights more. This focus on the ink color is not implemented in our model and both ink and word information weight alike in the decision. Hence, in reality we would expect the overlap curve to be squeezed upward (less overall errors) and the error curve squeezed downward. We would also expect the reaction time to be monotonously decreasing since the subject does not want to make a mistake and therefore makes time consuming effort to retrieve a pattern from partial information.

3.4 The impact of the number of colors

Following a similar routine as previously we get:

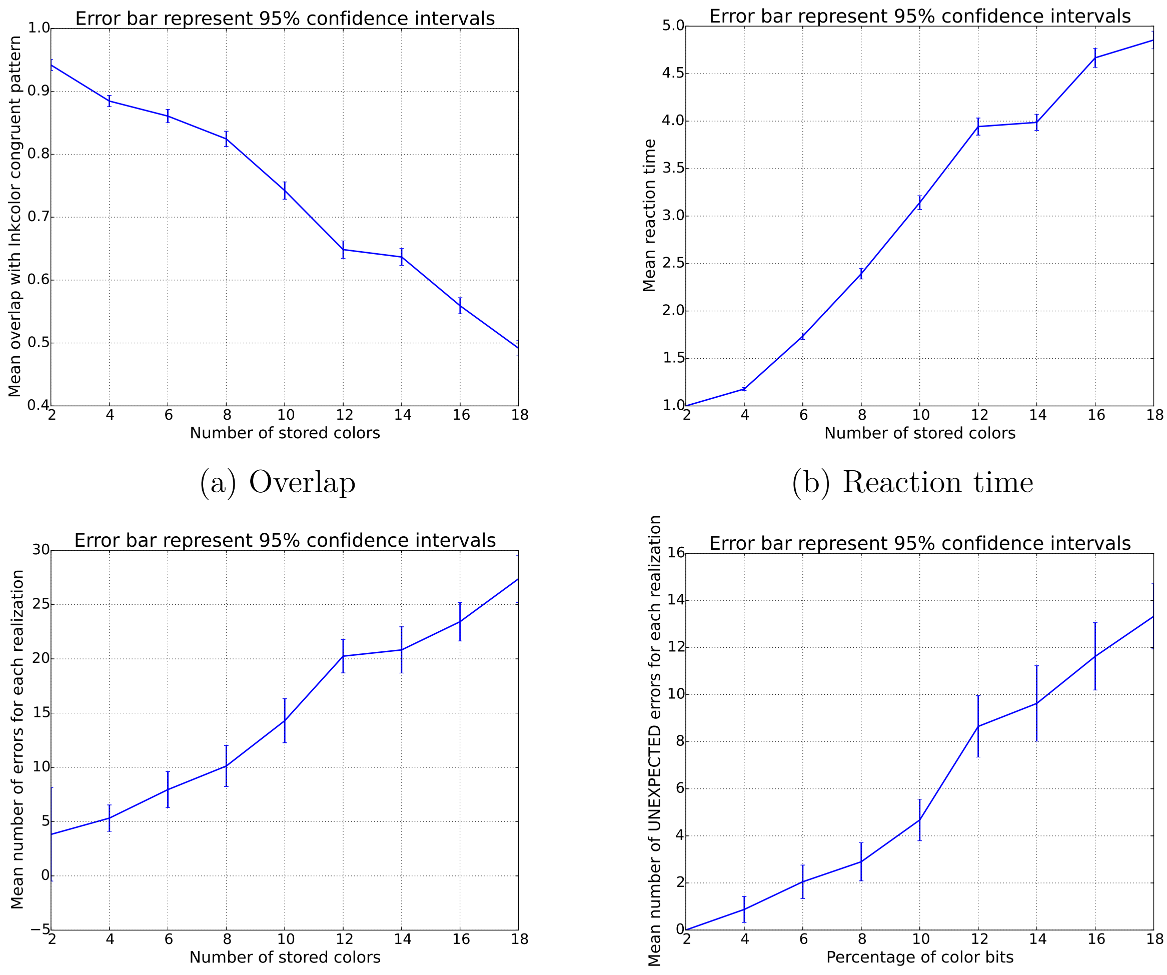


Figure 3.2: Influence of the number of learned colors

There is no real surprise in the results we get. As the number of learned colors increase for a given number of neurons the subject has more difficulties retrieving the correct color. Hence, the number of errors and the reaction time increase. The overlap on the other hand diminishes. With more colors there is more similar patterns and the subject is more likely to retrieve a different pattern than the congruent ink color. Finally, it should be noted that the unexpected errors account for approximately half of the total number of errors when the number of colors is above 10. Even with only 2 colors the subject does not manage to retrieve the correct color at every trial. This might be due to the low information percentage of ink color, $k = 0.6$.

3.5 Asynchronous vs synchronous updates

If one uses the synchronous update and initializes the network with always the same incongruent pattern, he gets always the same outcome. Either always the same correct answer or the same mistake. In fact, in this case, there is no randomness and everything is deterministic. This behavior tends to remain if one uses always the same ink color to build the incongruent pattern: if the subject is able to retrieve the color in one incongruent pattern this is likely to happen even for different incongruent word colors. However, the result for a given initialization is always the same, overlap and reaction time included.

Dissimilarly, using the asynchronous update one gets different outcomes for a same initial ink color. This is due to the additional randomness introduced by the update. The errors are not always the same, we sometime get unexpected errors and the value of the overlap and the reaction time are variable.

Finally, no matter the outcome (correct or not) the asynchronous update tends to converge faster. This may be explained by the fact that as the update progresses the neuron gets more and more signals from neurons that are already updated.

Conclusion With this general conclusion we will discuss question *2.5 Speculation.*
We saw during the second exercise that the reaction time as a function of the ratio of ink bits is not monotonously decreasing. This means that when the level of information about the ink is very low the patient will not take his time to give the right answer but will answer as fast as possible leading to numerous mistakes. Moreover, we observed the *unexpected errors* that may be, as discussed previously, a reflect of stress or lack of attention. It seems that the level of attention is therefore an important parameter to consider in order to improve the accuracy of the Hopfield model. How could this be done? We ended up with a couple of speculations to address this question.
In the second exercise, it was mentioned that there is no part in the implementation of our Hopfield network in which we specify to the subject to focus more on the ink color than on the word color. Therefore, when the ratio of ink bits equals the ratio of word bits the patient has more than 50% chance to make a mistake. However, this is not representative of the real Stroop experiment in which the subject knows that he must retrieve the ink color. Thus, the number of errors must also be affected by a factor of attention paid to the retrieval of the ink color. Doubling the weights of the ink color pixels in the network (set $\omega_{ij}^{ink} = 2$) would be a way to implement this extra level of attention paid to the ink color.

Another way to model the level of attention of the subject, that may be more straightforward, is to introduce voluntary mistakes which probability goes as a given percentage of attention. If the level of attention is 100% percent, no extra mistake is added.
Another important factor to consider, in the Stroop effect, is the stress, the precipitation of the patient to answer (given that the experiment is timed) that necessarily affects the result. Indeed, if the subject had all the time he needed, it would be easier for him not to make mistakes. We came out with two ideas in order to take this factor into consideration. Firstly, the level of attention of the

subject could be proportional to the remaining time. This implies that as the time goes, the subject gets stressed and his level of attention on the given task decreases, being more paid to the time than the color retrieval. Secondly, the T_{max} of the dynamics can be shortened, forcing the subject to answer quickly. With all these factors, the results our model could be more realistic and maybe match the experimental results collected by Stroop in 1935.

A Appendix

A.1 Implementation exercise I

- ___init___(self, N)
  Initializes the system to N neurons.

- make_pattern(self, P=1, ratio=0.5)
  This function has two roles. Firstly, it creates the different patterns. Secondly, it trains the patient's brain with these patterns.
  P patterns of N neurons are created. Each one is built in the following way. A N-dimensional array containing only "OFF" pixels is created ($S_i = -1$). The number of pixels to be turned on (*idx*) is determined by multiplying the ratio (input) with the number of neurons (N). The first *idx* neurons of the N-dimensional array are then turned on ($S_i = 1$) and all the neurons of the array are randomly permuted in order to have an homogeneous distribution of "ON" and "OFF" pixels.
  The subect's brain training corresponds to the definition of the weights w_{ij} as stated in equation 1.0.4.

- dynamic(self)
  This function executes one timestep of the dynamics as defined in equations 1.0.1 and 1.0.3 with an *asynchronous* update. It means that a random neuron is picked and its state variable is updated. Then another neuron is picked randomly and updated within the same time step and can therefore feel the influence of the previous neurons' updates. To do so an array (*idx*) is created containing the idexes of all neurons of pattern μ. All the components of that array are randomly permuted so the final vector defines the (random) order of update.

- overlap(self, μ)
  Returns the overlap of the current state with pattern number μ. The overlap is given by:

$$m^{\mu} = \frac{1}{N} \sum_{i=1}^{N} \xi_i^{\mu} S_i(t) \tag{A.1}$$

- retrival_error(self, μ)
  Question 1.2
  Returns the retrieval error of the current state with the pattern number μ as defined in question 1.2.

- run(self, t_max=20, mu=0, flip_ratio=0)
  First of all, the run function initializes the network close to a chosen pattern μ but with a proportion of flipped pixels. The system is set on pattern μ exactly. Then the neurons in the system's array are randomly permuted. The number of neurons to be flipped is $idx = flip\_ratio \times N$. The first idx neurons of the randomly permuted array are flipped. Secondly, for each time steps, provided that $t < t\_max$, the dynamic is performed (by calling the dynamic function). The overlap with pattern μ is stored. The run is stopped if the state S of the system does not change between two time steps or if t reaches t_max. This function returns the retrieval error of pattern μ defined above.

A.2 Implementation exercise II

Modelling the Stroop effect and the subject's brain
In order to build P congruent and incongruent Stroop patterns we do the following:

- We generate P random patterns of size $\text{int}(Nk)$ for the coding of the ink-color, these are the Inkpatterns.
- We generate P random patterns of size $N - \text{int}(Nk)$ for the coding of the word-color, these are the Wordpatterns.
- We concatenate the Inkpatterns with the Wordpatterns without permutations: the first generated Inkpattern goes with the first Wordpattern, the second with the second and so on. This gives us the congruent Stroop patterns.
- We randomly pick one Wordpattern and we concatenate it with one random Inkpattern. This gives now an incongruent Stroop pattern from which we initialize the network. We do this at each network realization (i.e. each time we present a new coloured word to the subject) hence, if we do enough realizations we should cover every possible ink color and word combination.

Convergence, decision and reaction time

- **Convergence criterion:** We stop the simulation as soon as the updated state is the same as the previous one. Therefore, we let the network evolve until it reaches a fixed equilibrium point. To avoid infinite simulations that could arise for example if the network is oscillating around a equilibrium position, we used a maximal number of updates that is set to 40.
- **Decision criterion:** We count that the subject's answer is a success only if he retrieved the **correct** ink color. Meaning that the overlap with the congruent pattern of the initial ink color must be higher than the overlap with the congruent pattern of the initial word color but also that it has to be greater than the overlap with any other congruent pattern. This means

that the patient is considered to be wrong not only if its guess is closer to the word color than the ink color but also if he retrieved a color different than the ink color.

- **Reaction time:** The reaction time is given by the number of times that the network is updated. Therefore, if we initialized the network to a learned pattern and that the network does not change we get 0 reaction time.

References

[1] Amir Raz, Irving Kirsch, Jessica Pollard, and Yael Nitkin-Kaner. Suggestion reduces the stroop effect. *Psychological Science*, 17(2):91–95, 2006.

[2] Susan Songstad. Stroop Effect Tester. `https://faculty.washington.edu/chudler/colors3.html`. [Online; accessed 1-May-2016].

[3] Colin M MacLeod. Half a century of research on the stroop effect: an integrative review. *Psychological bulletin*, 109(2):163, 1991.

[4] Abu-Mostafa, Yaser S, Jacques, and Jeannine-Marie St. Information capacity of the hopfield model. *Information Theory, IEEE Transactions on*, 31(4): 461–464, 1985.

[5] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition.* Cambridge University Press, 2014. ISBN 9781107060838. URL `https://books.google.ch/books?id=D4j2AwAAQBAJ`.