Our main goal is to design a perceptron for binary number clasification. This perceptron takes as an input a vector $\vec{S}$ of N elements in which the first $N/2$ digits correspond to the first number n_T and the $N/2$ last digits correspond to the second number n_B (the elements of the vector will be $S_i = \{\pm 1\}$, being +1 the 1 on binary notation and -1 the 0 on binary notation, so this way the sequence (in the case $N = 20$) -1-1-1-1-1-1-1-1+1+1 would be the 0000000011 on binary, hence the 3 on decimal notation). With this information the perceptron has to give as an output $\sigma = +1$ if $n_T > n_B$ and $\sigma = -1$ if $n_T \leq n_B$ using this algoritm for the output $\sigma(\{S_i^\mu\}, \{J_i\}) = \text{sign}(\sum_{i=1}^N S_i J_i)$. In this exercise, we introduce a randomised parameter r_i following a gaussian distribution of mean $\mu = 0$ and variance $\sigma^2 = 50$ to our learning model and generalise the case of $N = 20$ to $N = 40$ also (in comparison to the previous exercise).

Problem 5 and 6. *Evaluate the corresponding test error* $\epsilon = P_{test}^{-1} \sum_\mu \theta(\sigma(\{S_i^\mu\}, \{J_i\}), \sigma_T^\mu)$ *of the trained student perceptrons with* P_{test} *= 1000. Repeat the procedure 1000 times and evaluate the average test error (and its error bar).*

Comment. After evaluating the test error (in comparison with the results given by a teacher perceptron whose synaptic weights are shown on Equation 1) 1000 times for each one of the trained models (over a sample test of 1000 pair of numbers to be ordered), the averages are shown on the Table 1. As we can see, the average test error is decreasing with the increse of the number of pairs used in the training of the perceptrons, all but the last two, which are exchanged (anomaly which can be reversed within the obtained errors).

$$J_i^* = \begin{cases} 2^{N/2-i} & \text{if } 1 \leq i \leq N/2 \\ -J_{i-N/2}^* & \text{if } N/2+1 \leq i \leq N \end{cases} \tag{1}$$

P	ϵ (%)
1	43.4 ± 1.6
10	36.3 ± 1.5
20	35.8 ± 1.5
50	15.8 ± 1.1
100	13.5 ± 1.1
150	5.1 ± 0.7
200	5.5 ± 0.7

Table 1: Average test error (and error bars) of the $N = 20$ trained models for a sample test of $P_{test} = 1000$ couples to be ordered in comparison with the teacher perceptron.

Problem 7. *Plot* ϵ *vs* P.

Comment. If we plot the obtained test error for each one of the 1000 repetitions for every model vs P, we obtain the graph of Figure 1. As we can see, errors fluctuate around a mean value (shown on Table 1). This fluctuations decrease with the increse of P (which can also be seen at the Table 1), meaning that more trained models, tend to be more precise than less trained ones (as what we would expect).

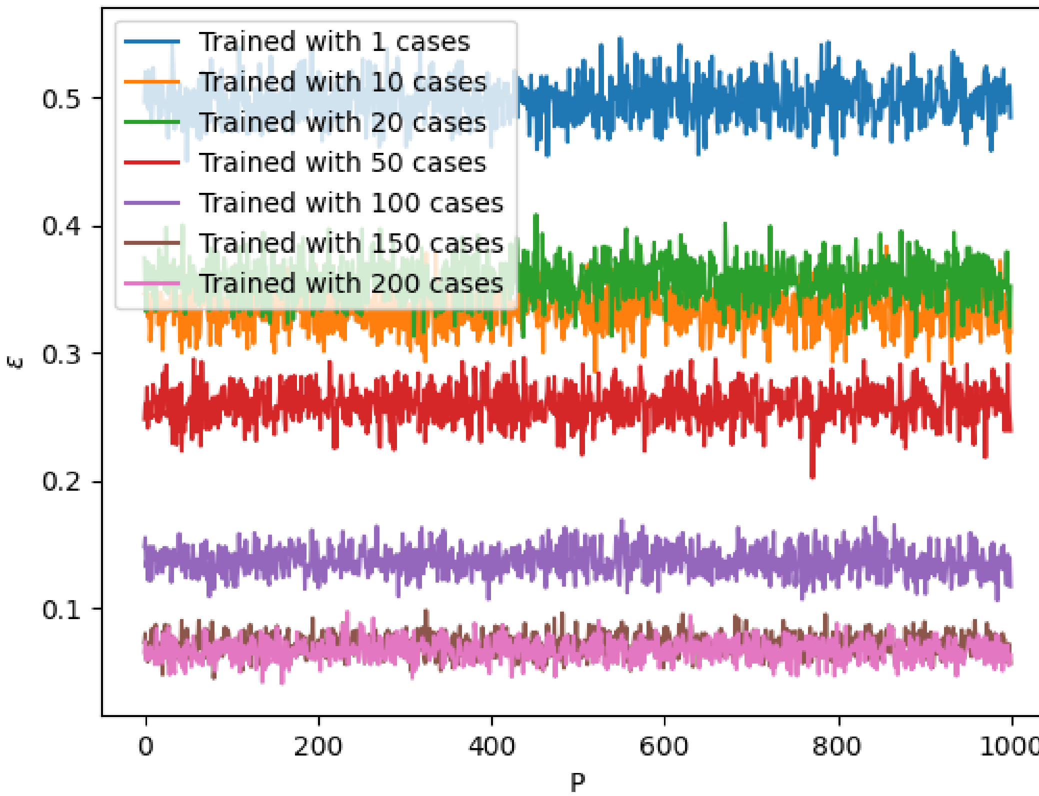

Figure 1: Test error vs P for 1000 points of the different trained models ($N = 20$) in comparison to the predictions of the teacher perceptron.

Problem 8. *Repeat the entire study (points from (3) to (6)) for $N = 40$ and $P = 1, 2, 20, 40, 100, 200, 300$.*

Comment. For the case of $N = 40$ (meaning that the numbers which the perceptrons should order are now between 0 and $2^{20} - 1$ instead of 0 and $2^{10} - 1$) and $P = 1, 2, 20, 40, 100, 200, 300$, the average test errors are shown in Table 2. In this table we can see more than one anomaly. The first one is that for P = 200, the error is exactly zero, which means that the perceptron trained with 200 well ordered pairs by the teacher hasn't made any fail, but then the one of P = 300 is not also zero as we would expect. Then we have also an "exchange" of positions between P = 20 and P = 40 values, this time not beingreconcile within the error ranges. Other than this, precisions also tend to increse with the increse of P.

Problem 9. *Plot ϵ vs P for $N = 40$ together with the data obtained with $N = 20$.*

P	ϵ (%)
1	79.6 ± 1.3
2	62.3 ± 1.6
20	2.0 ± 0.4
40	3.3 ± 0.6
100	1.0 ± 0.3
200	0.0 ± 0.0
300	0.6 ± 0.2

Table 2: Average test error (and error bars) of the N = 40 trained models for a sample test of $P_{test} = 1000$ couples to be ordered in comparison with the teacher perceptron.

Comment. If we graph the test errors for all the trained perceptrons, we obtain the plot of the Figure 3 (Even though some colors get repeated, we can distinguish them with the test error averages shown in the Tables 1 and 2. As we could see, perceptron trained with $N = 20$ have the 6, 7, 8, 9, 10, 11 and 12 positions (given that a higher position means a higher probability of fail under a couple ordering) while perceptron trained with $N = 40$ have the 1, 2, 3, 4, 5, 13 and 14 positions. This remark the fact that (in comparison) training data size is more important on the perceptrons of $N = 40$ than in the ones of $N = 20$. For example, one couple trained perceptron for $N = 20$ has a failure rate of less than 50

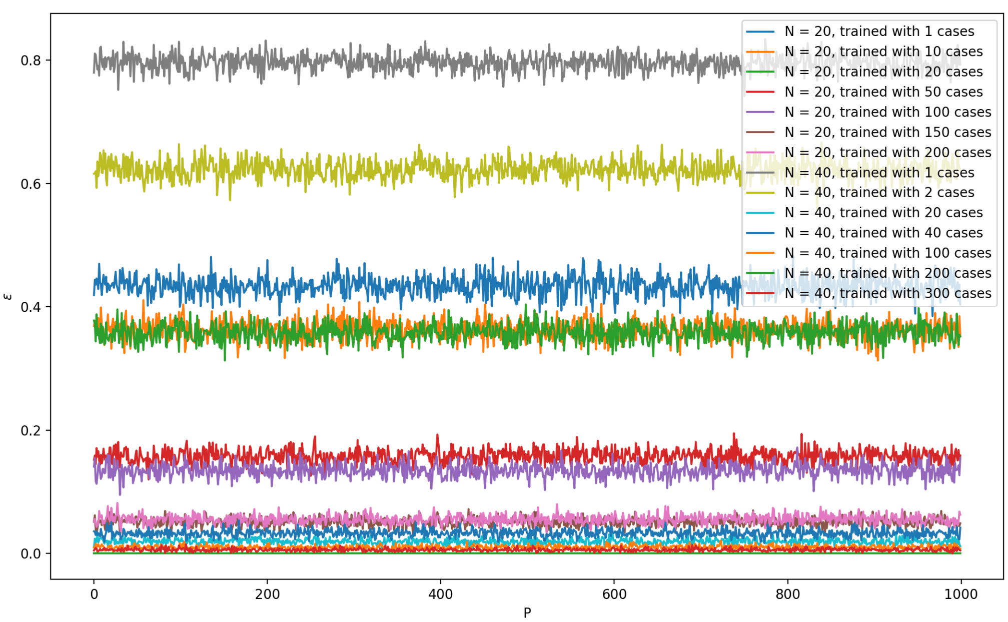

Figure 2: Test error vs P for 1000 points of the different trained models ($N = 20$ and $N = 40$) in comparison to the predictions of the teacher perceptron.

Problem 9. *Find a way to represent the data so that they superimpose and comment on* *what done.*

Comment. If we superimpose the data, we would lose information about the rate of failure of each of the models, but this overlap would give us information about the noise of the tase of failure. We could think that the models which in some points reach zero error have their noise limitated, but due to the fact that it is not posible to have negative faiulure rate (or over 100%) this limitation of the noise is naturally seen as precision of the model. So if we subtract each point of the plot by their averages and then we sum them 0.5, all noise graphs will be centered at 50%. Doing this we obtain the plot of the Figure **??**. Is important to remark that the plot contain errors diferent to the ones of the rest of the exercise (because the code was re-runned in order to obtain this las plots due to some errors in the code).

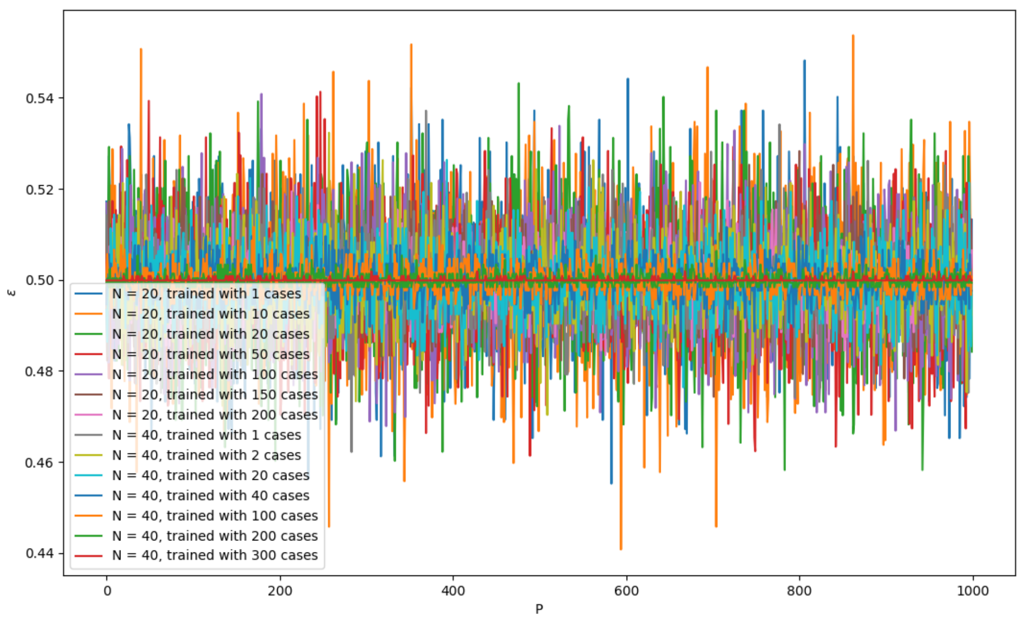

Figure 3: Test error vs P for 1000 points of the different trained models ($N = 20$ and $N = 40$) in comparison to the predictions of the teacher perceptron centered at 50%.