Neural Networks
Group 04
Andrea Jemmett (2573223)
Bas van der Drift (1707507)

Assignment 1 - Perceptron

Section A

To create a linearly separable dataset, a function called get_data was written. This function takes two arguments: the amount of data that should be generated and the scale of these points. Based on the point's x and y a class of either 1 or -1 is assigned. The code for this can be found in the file 'get_data.m'.
Below, in figure 1, the dataset resulting from get_data(1000, 10) is displayed in a scatterplot.

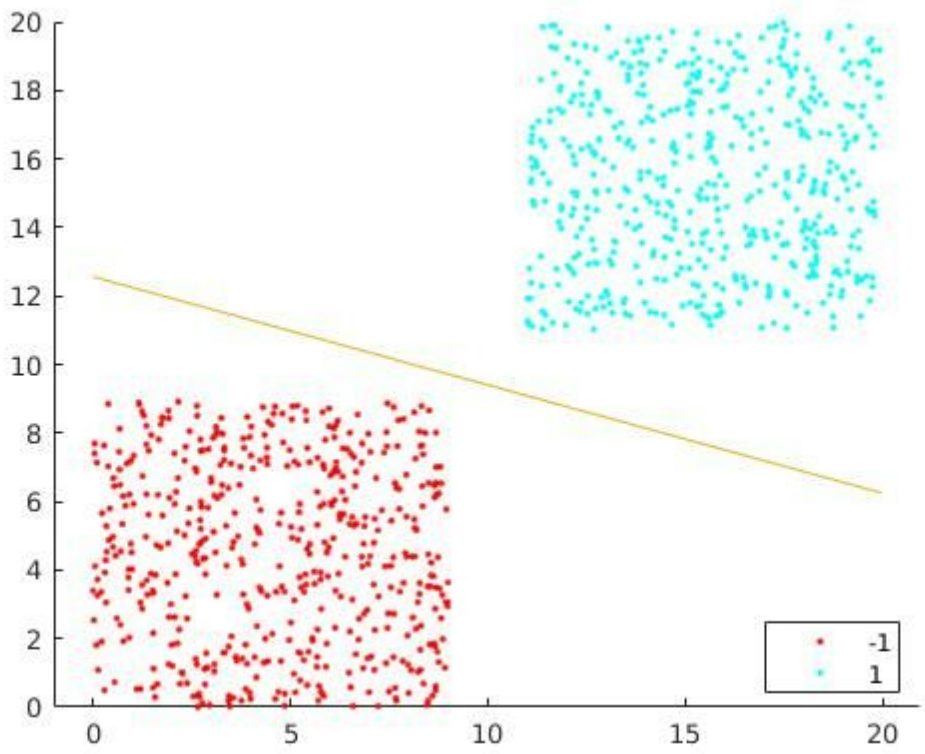

Figure 1. Scatter plot of the generated dataset (1000 points, scale 20) with the decision boundary found by the perceptron.

Section B

A perceptron function was written that takes three arguments and returns three vectors. Its arguments are the dataset to use, the learning rate and the maximum number of iterations to train for. The output is a vector of weights, the error at each epoch and the number of iterations it took to reach convergence (otherwise the set maximum iteration number). The code for this function can be found in the file 'perceptron.m'.

The error measurement used is the mean squared error of the perceptron. It is calculated by squaring the difference between the desired output and the actual output and dividing this by the total number of observations:

$$\frac{1}{N}\sum_{i=1}^{N}(d_i - y_i)^2$$

where d is the actual class and y is the network output. Once the actual influence of the learning rate on the output of the perceptron is investigated, it becomes clear that the learning rate only influences the scale of the weights, which results in the same output boundary.

Section C

The perceptron is able to classify all cases correctly, as shown in Figure 1. The scatter plot and the decision boundary are drawn by the 'scatterboundary.m' function. To plot the boundary we created a linear space for a variable (x_2 in the equations below) and put the equation of the boundary from the general form:

$$w_1 + w_2 x_2 + w_3 x_3 = 0$$

to the slope-intercept form:

$$x_3 = -\frac{w_1 + w_2 x_2}{w_3}$$

All red cases are situated below and to the left of the decision boundary line, whereas the blue ones are
above it and to the right. Since our dataset is randomly generated, the plot may vary, but the classification task is always carried out completely successfully within a maximum of two learning iterations.

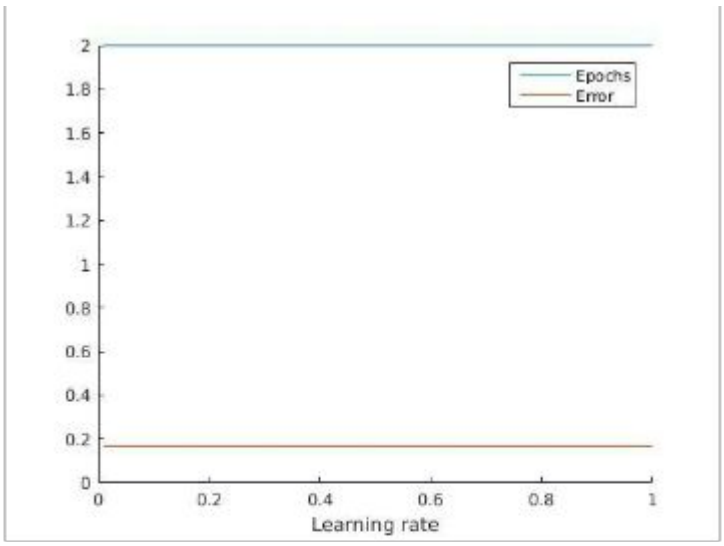

Figure 2. Number of epochs required for convergence and overall error for the linearly separable case.

As explained before, the learning rate does not influence the overall error produced by the network. This is due to the learning rate merely affecting the scale of weights and not the resulting output, since the signum function categorises everything above 0 as a 1, and everything below it as a -1. In this light it does not matter whether the signum function receives, for example, a value of 5 or a value of 0.5 as a weight, which is the only thing that reflects a change in learning rate. Moreover we can see that the error signal in the perceptron learning rule (d - y in our implementation) can only output values of -2, -1, 0, 1 or 2 being y the output of a signum function and d the target (which has

values -1 or 1); this discrete error measure makes the learning rate a scale factor that doesn't affect the learning process.

Section D

The perceptron functions as expected with a non-linearly separable dataset. A decision boundary is found, but the algorithm does not classify more than 169 out of 250 training examples correctly (68%). While this is higher than chance level and thus one could say the perceptron has some use, much better performance could be gained from using, for example, a linear regression approach (or a continuous activation function).
Convergence is never reached, no matter the learning rate or number of iterations chosen. The learning rate does not influence this non-converging behaviour or the overall error because of what was already explained in section C. Figures 3 and 4 display this behaviour graphically.

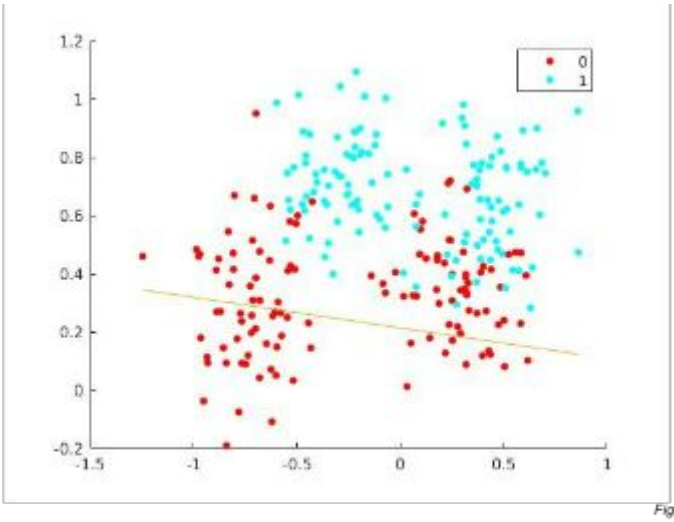

Figure 3. Non linearly separable dataset with decision boundary.

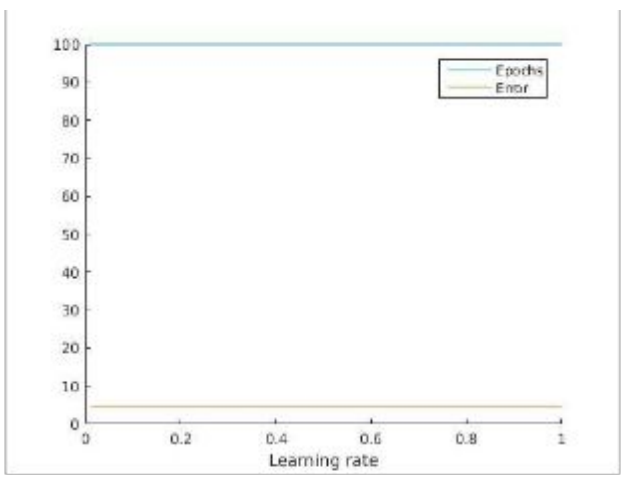

Figure 4. Number of epochs and overall network error for different learning rates. Since convergence is never reached, the number of epochs is maximal and the error is constantly higher.