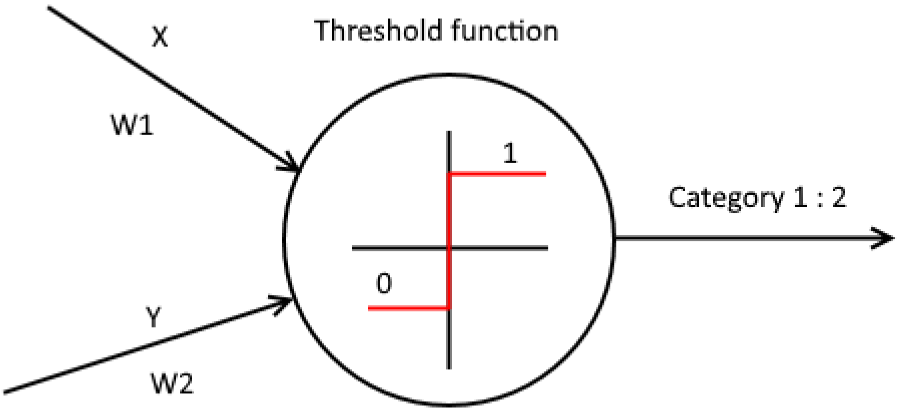
Exercise 6

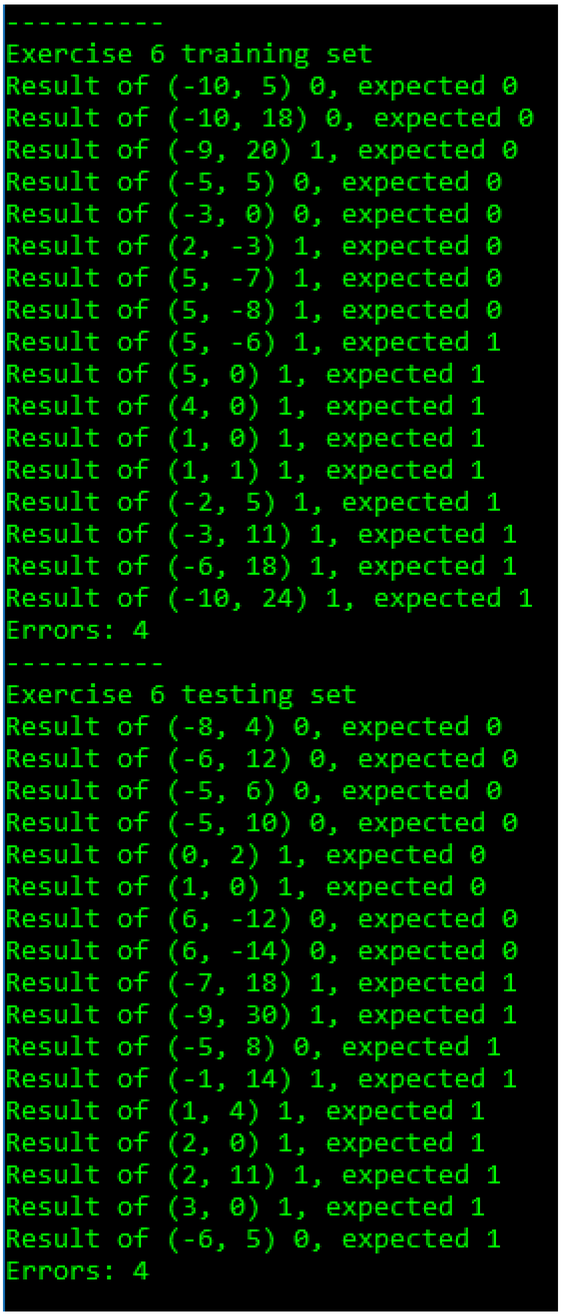
a) I imagine the perceptron will look like this:



It has two inputs, X and Y, and returns 0 or 1 based on the set it belongs to. 0 corresponds to category 1, 1 corresponds to category 2. The weights are randomly generated in the beginning, and are adjusted every time an error is detected. A learning parameter is introduced to the weight calculation.

b) Link to source code: https://source.uit.no/aja073/Exercise_6_AI/tree/master

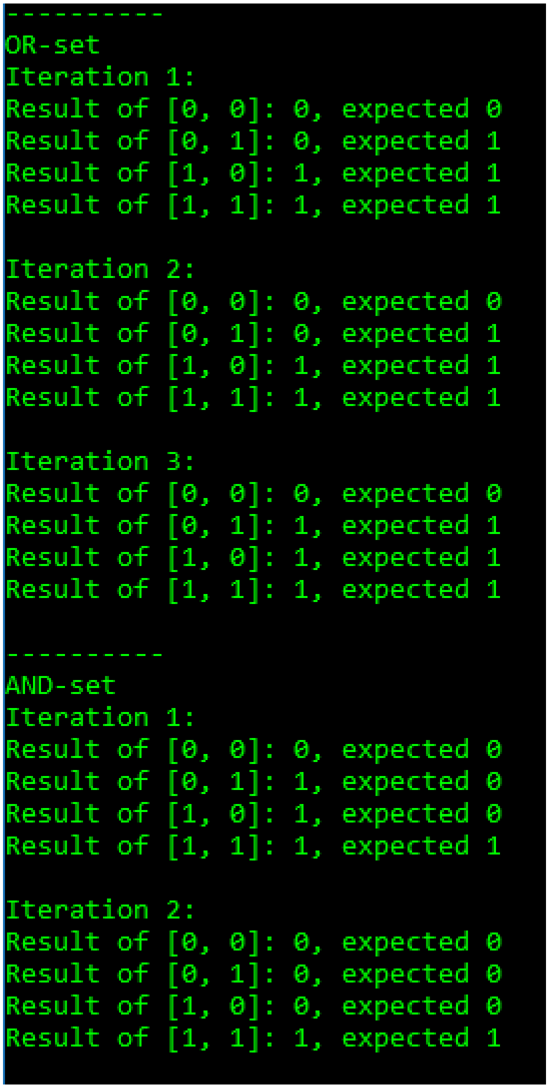
Result of one run:



```
----------
Exercise 6 training set
Result of (-10, 5) 0, expected 0
Result of (-10, 18) 0, expected 0
Result of (-9, 20) 1, expected 0
Result of (-5, 5) 0, expected 0
Result of (-3, 0) 0, expected 0
Result of (2, -3) 1, expected 0
Result of (5, -7) 1, expected 0
Result of (5, -8) 1, expected 0
Result of (5, -6) 1, expected 1
Result of (5, 0) 1, expected 1
Result of (4, 0) 1, expected 1
Result of (1, 0) 1, expected 1
Result of (1, 1) 1, expected 1
Result of (-2, 5) 1, expected 1
Result of (-3, 11) 1, expected 1
Result of (-6, 18) 1, expected 1
Result of (-10, 24) 1, expected 1
Errors: 4
----------
Exercise 6 testing set
Result of (-8, 4) 0, expected 0
Result of (-6, 12) 0, expected 0
Result of (-5, 6) 0, expected 0
Result of (-5, 10) 0, expected 0
Result of (0, 2) 1, expected 0
Result of (1, 0) 1, expected 0
Result of (6, -12) 0, expected 0
Result of (6, -14) 0, expected 0
Result of (-7, 18) 1, expected 1
Result of (-9, 30) 1, expected 1
Result of (-5, 8) 0, expected 1
Result of (-1, 14) 1, expected 1
Result of (1, 4) 1, expected 1
Result of (2, 0) 1, expected 1
Result of (2, 11) 1, expected 1
Result of (3, 0) 1, expected 1
Result of (-6, 5) 0, expected 1
Errors: 4
```

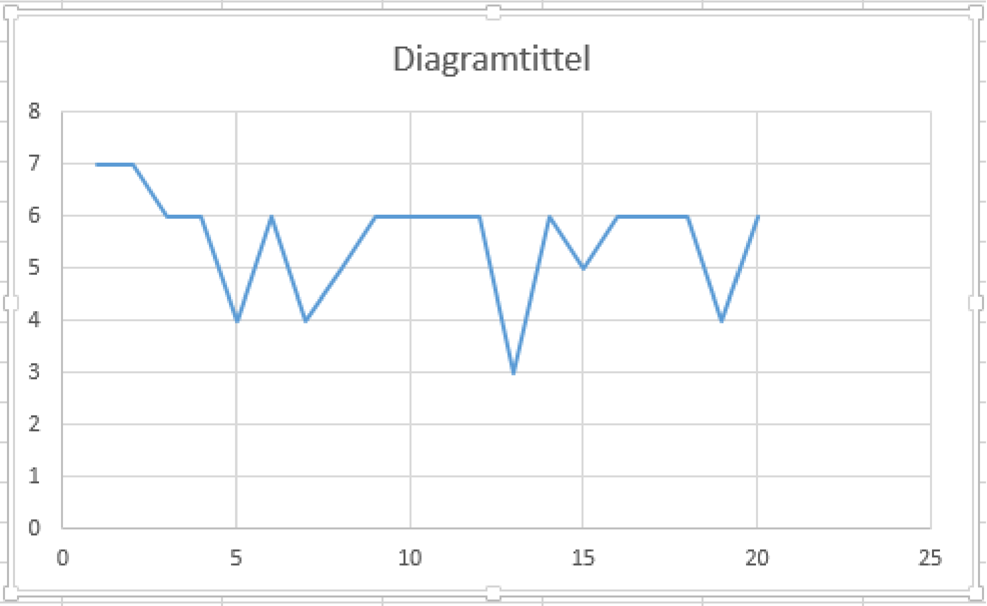
This particular run produced 4 errors

The perceptron has also learned about OR and AND:



```
----------
OR-set
Iteration 1:
Result of [0, 0]: 0, expected 0
Result of [0, 1]: 0, expected 1
Result of [1, 0]: 1, expected 1
Result of [1, 1]: 1, expected 1

Iteration 2:
Result of [0, 0]: 0, expected 0
Result of [0, 1]: 0, expected 1
Result of [1, 0]: 1, expected 1
Result of [1, 1]: 1, expected 1

Iteration 3:
Result of [0, 0]: 0, expected 0
Result of [0, 1]: 1, expected 1
Result of [1, 0]: 1, expected 1
Result of [1, 1]: 1, expected 1

----------
AND-set
Iteration 1:
Result of [0, 0]: 0, expected 0
Result of [0, 1]: 1, expected 0
Result of [1, 0]: 1, expected 0
Result of [1, 1]: 1, expected 1

Iteration 2:
Result of [0, 0]: 0, expected 0
Result of [0, 1]: 0, expected 0
Result of [1, 0]: 0, expected 0
Result of [1, 1]: 1, expected 1
```

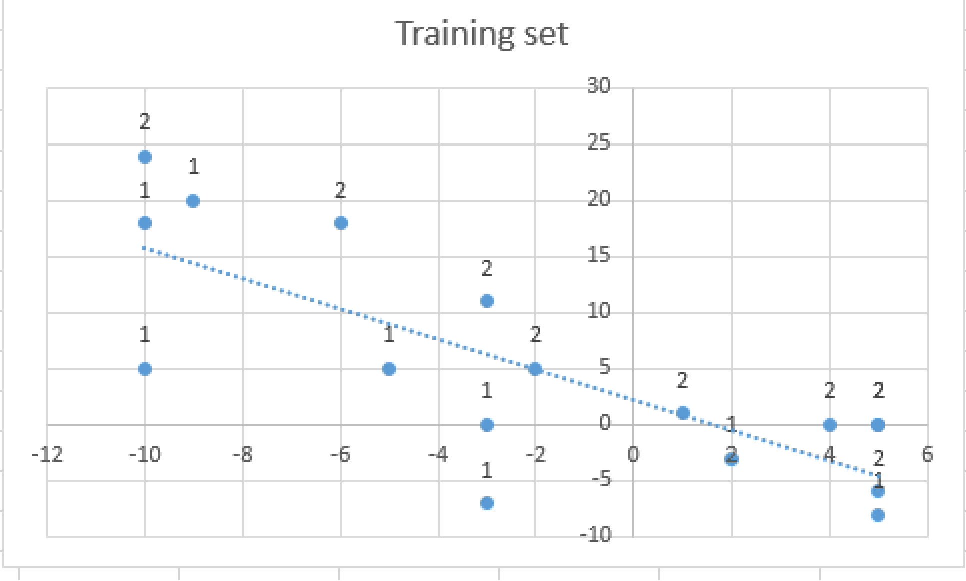
c)



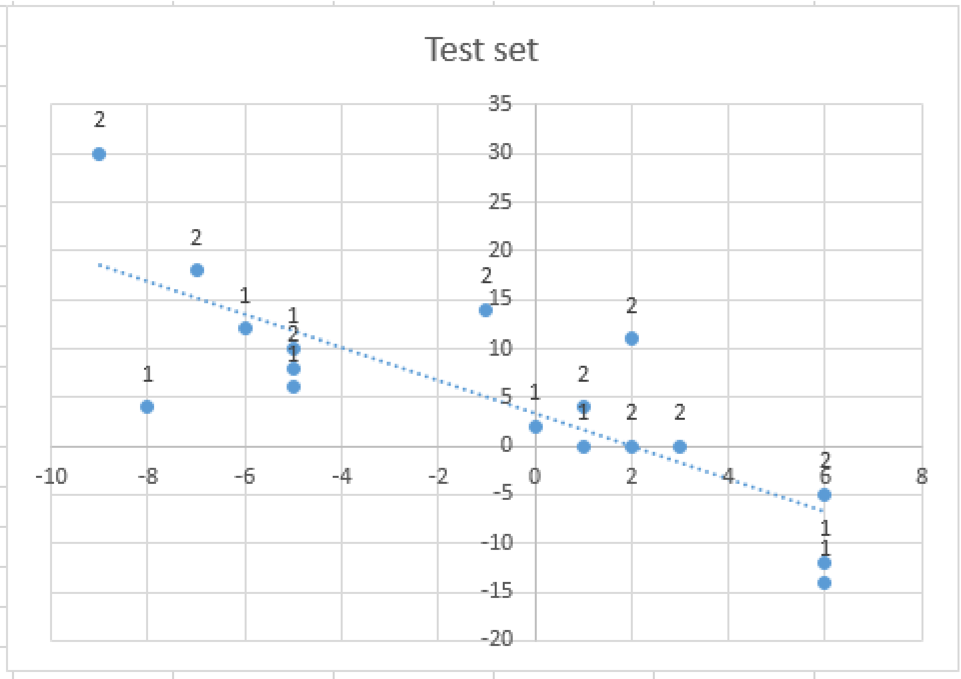
This graph illustrates the error rate after X cycles. It can be seen that around cycle 13 the error rate is 3, which is a global minima. Running the program for 100, 500 and 1000 cycles produces similar global minima after less than 50 cycles.

d) The smallest error produced by the perceptron is 3, as shown in the graph above. The error rate is the same for the test set. This is similar to the error obtained by Renate Karlsen.

e)



The regression line for the training set reflects the number of errors produced by the perceptron. As seen, there is some noise that makes it harder to get a clear separation between the two categories.



This also holds for the testing set.

f) The major differences between my perceptron and a feed forward back propagation network is that my solution only consists of one single perceptron and isn’t really a network with multiple layers. The feed forward network passes the data forwards to the next layer, and after the output layer comes up with an answer, the error is propagated back through the layers to the input layer.

g) Deep learning implies multi-layered neural networks processing large amounts of information. Deep learning is a way of machine learning connected to artificial intelligence.