Department of Mathematics and Computer Science
University of Southern Denmark, Odense

November 4, 2022
Marco Chiarandini

DM573 - Introduction to Computer Science

Training Session, Weeks 45-46, Autumn 2022

I: To be solved during the exercise class in week 45

Exercise 1. k-Nearest Neighbors: Prediction

Suppose you are trying to predict a continuous response y to an input x and that you are given the set of training data $D = [(x_1, y_1), \ldots, (x_{11}, y_{11})]$ reported and plotted in Figure 1.

$$D = \begin{bmatrix} (8, & 8.31) \\ (14, & 5.56) \\ (0, & 12.1) \\ (6, & 7.94) \\ (3, & 10.09) \\ (2, & 9.89) \\ (4, & 9.52) \\ (7, & 7.77) \\ (8, & 7.51) \\ (11, & 8.0) \\ (8, & 10.59) \end{bmatrix}$$

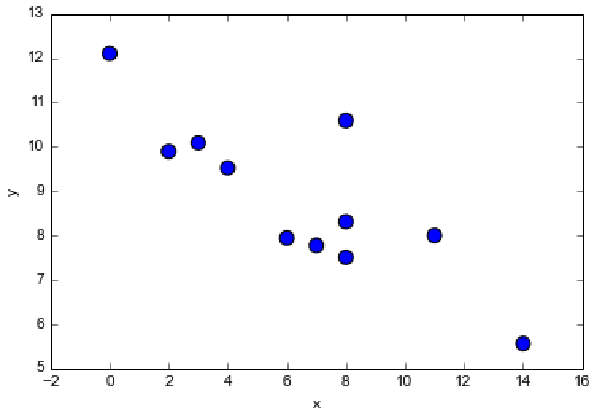

Figure 1: The data for Exercise 1.

Using 5-nearest neighbors, what would be the prediction on an new input $x = 8$?
What form of learning is this exercise about?

- Supervised learning, regression
- Supervised learning, classification
- Unsupervised learning
- Reinforcement learning

Exercise 2. k-Nearest Neighbors: Prediction

Suppose you are trying to predict the class $y \in \{0, 1\}$ of an input (x_1, x_2) and that you are given the set of training data $D = [((x_{1,1}, x_{1,2}), y_1), \ldots, ((x_{11,1}, x_{11,2}), y_{11})]$ reported and plotted in Figure 2.
Using the 5-nearest neighbors method, what would the prediction be on the new input $\vec{x} = (5, 10)$?
What form of learning is this exercise about?

- Supervised learning, regression
- Supervised learning, classification
- Unsupervised learning
- Reinforcement learning

$$D = \begin{bmatrix} ((10, & 2), & 1) \\ ((15, & 2), & 1) \\ ((6, & 11), & 1) \\ ((2, & 3), & 0) \\ ((5, & 15), & 1) \\ ((5, & 14), & 1) \\ ((10, & 1), & 0) \\ ((1, & 6), & 0) \\ ((17, & 19), & 1) \\ ((15, & 13), & 0) \\ ((19, & 9), & 0) \end{bmatrix}$$

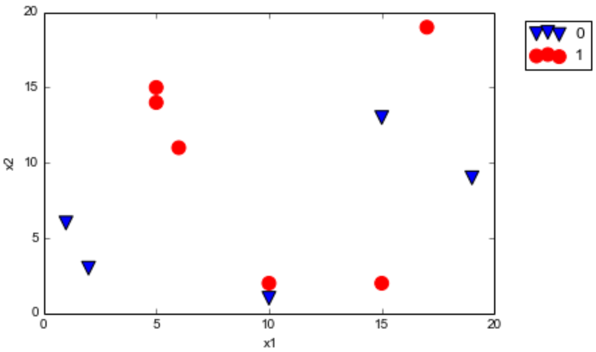

Figure 2: The data for Exercise 2.

Exercise 3. Linear Regression: Prediction

As in Exercise 1. you are trying to predict a response y to an input x and you are given the same set of training data $D = [(x_1, y_1), \ldots, (x_{11}, y_{11})]$, also reported and plotted in Figure 3. However, now you want to use a linear regression model to make your prediction. After training, your model looks as follows:

$$g(x) = -0.37x + 11.22$$

The corresponding function is depicted in red in Figure 3. What is your prediction $\hat{y}$ for the new input $x = 8$?

$$D = \begin{bmatrix} (8, & 8.31) \\ (14, & 5.56) \\ (0, & 12.1) \\ (6, & 7.94) \\ (3, & 10.09) \\ (2, & 9.89) \\ (4, & 9.52) \\ (7, & 7.77) \\ (8, & 7.51) \\ (11, & 8.0) \\ (8, & 10.59) \end{bmatrix}$$

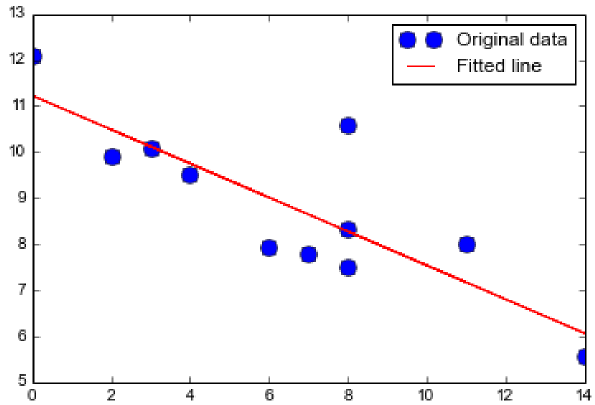

Figure 3: The data for Exercise 3.

Exercise 4. Linear Regression: Training

Calculate the linear regression line g for the set of points:

$$D = \begin{bmatrix} (2, 2) \\ (3, 4) \\ (4, 5) \\ (5, 9) \end{bmatrix}$$

Calculate also the *loss* of using g to predict the data from D.
Plot the points and the regression line on the Cartesian coordinate system.
[You can carry out the calculations by hand or you can use any program of your choice. Similarly, you can draw the plot by hand or get aid from a computer program.]

Exercise 5. Logical Functions and Perceptrons

Perceptrons can be used to compute the elementary logical functions that we usually think of as underlying computation. Examples of these functions are AND, OR and NOT.

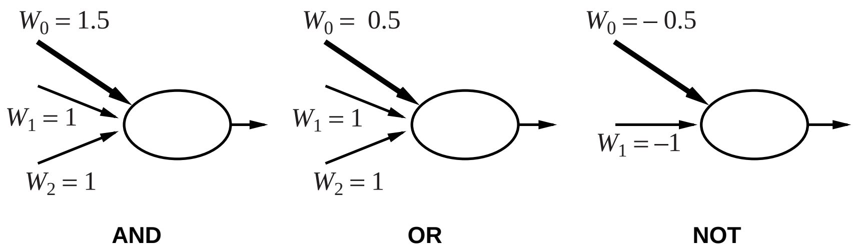

Figure 4: Logical functions and perceptrons. Exercise Exercise 5..

In class, we carried out the verification that the left most perceptron in Figure 4 is a correct representation of the AND operator.

- Verify that the perceptrons given for the OR and NOT cases in Figure 4 are also correct representations of the corresponding logical functions.
- Design a perceptron that implements the logical function NAND.

Later in this Assignment Sheet we will see that there are also Boolean functions that cannot be represented by a single perceptron alone.

Exercise 6. Multilayer Perceptrons

Determine the truth table of the Boolean function represented by the perceptron in Figure 5:

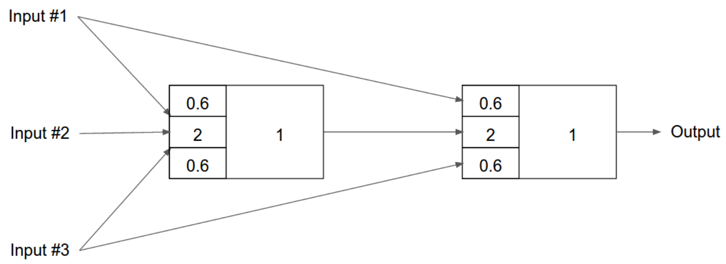

Figure 5: . The multilayer perceptron of Exercise 6.

Exercise 7. Single Layer Neural Networks: Prediction

In Exercise 2. we predicted the class $y \in \{0, 1\}$ of an input (x_1, x_2) with the 5-nearest neighbors method using the data from set D. We used those data to train a single layer neural network for the same task. The result is depicted in Figure 6. (We use the convention $x_0 = -1$ in the linear combination of the inputs.)

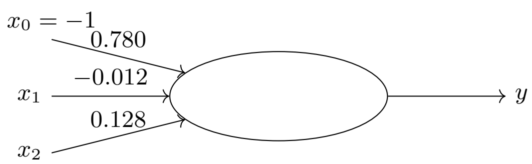

Figure 6: A single layer neural network for the task of Exercise 7.

- Calculate the prediction of the neural network for the new input $\vec{x} = (5, 10)$. Assume a step function as activation function in the unit (i.e., a perceptron).
- Calculate the prediction of the neural network for the new input $\vec{x} = (5, 10)$. Assume a sigmoid function as activation function in the unit (which is therefore a sigmoid neuron).

- Compare the results at the previous two points against the result in Exercise 2. Are they all consistent? Is this expected to be always the case? Which one is right?
- In binary classification, the loss can be defined as the number of mispredicted cases. Calculate the loss for the network under the two different activation functions. Which one performs better according to the loss?
- Derive and draw in the plot of Exercise 2. the decision boundaries between 0s and 1s that is implied by the perceptron and the sigmoid neuron. [See Section 2.1.3 of the Lecture Notes.] Are the points linearly separable?

Exercise 8. Expressivness of Single Layer Perceptrons

Is there a Boolean (logical) function in two inputs that cannot be implemented by a single perceptron? Does the answer change for a single sigmoid neuron?

II: To be solved at home before the exercise class in week 46

Exercise 1. k-Nearest Neighbors: Prediction

Suppose you are trying to predict a continuous response y to an input x and that you are given the set of training data $D = [(x_1, y_1), \ldots, (x_8, y_8)]$ reported and plotted in Figure 7.

$$D = \begin{bmatrix} 11. & 6.36 \\ 14. & 5.98 \\ 6. & 8.02 \\ 10. & 7.17 \\ 18. & 5.51 \\ 18. & 4.77 \\ 3. & 10.25 \\ 4. & 9.45 \end{bmatrix}$$

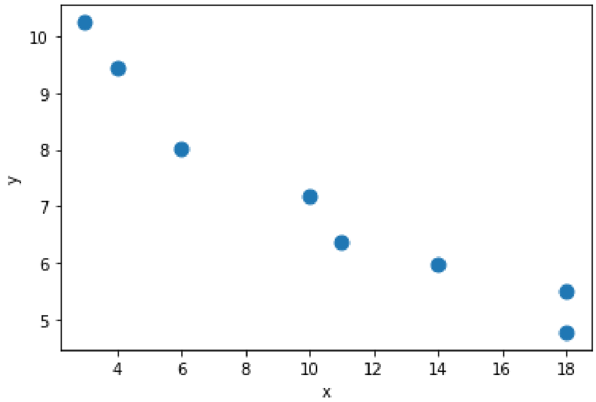

Figure 7: The data for Exercise 1.

Using 3-nearest neighbors, what would be the prediction on an new input $x = 12$?
What form of learning is this exercise about?

- Supervised learning, regression
- Supervised learning, classification
- Unsupervised learning
- Reinforcement learning

Exercise 2. k-Nearest Neighbors: Prediction

Suppose you are trying to predict the class $y \in \{0, 1\}$ of an input (x_1, x_2) and that you are given the set of training data $D = [((x_{1,1}, x_{1,2}), y_1), \ldots, ((x_{9,1}, x_{9,2}), y_9)]$ reported and plotted in Figure 8.

$$D = \begin{bmatrix} ((4, & 2), & 0) \\ ((16, & 7), & 0) \\ ((10, & 17), & 1) \\ ((12, & 10), & 1) \\ ((8, & 6), & 0) \\ ((3, & 8), & 1) \\ ((1, & 6), & 0) \\ ((9, & 3), & 0) \\ ((17, & 7), & 1) \end{bmatrix}$$

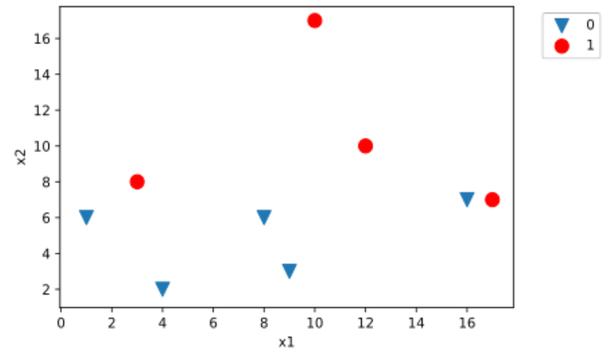

Figure 8: The data for Exercise 2.

Using the 3-nearest neighbors method, what would the prediction be on the new input $\vec{x} = (14, 8)$?
What form of learning is this exercise about?

- Supervised learning, regression
- Supervised learning, classification
- Unsupervised learning
- Reinforcement learning

Exercise 3. Linear Regression: Training

Calculate the linear regression line g for the set of points:

$$D = \begin{bmatrix} 5 & 44 \\ 16 & 16 \\ 13 & 20 \\ 19 & 3 \\ 9 & 29 \end{bmatrix}$$

Calculate also the *loss* of using g to predict the data from D.
Plot the points and the regression line on the Cartesian coordinate system.
[You can carry out the calculations by hand or you can use any program of your choice. Similarly, you can draw the plot by hand or get aid from a computer program.]

Exercise 4. Feed-Forward Neural Networks: Single Layer Perceptron

Determine the parameters of a single perceptron (that is, a neuron with step function) that implements the majority function: for n binary inputs the function outputs a 1 only if more than half of its inputs are 1.

Exercise 5. Single Layer Perceptrons

Can you represent the two layer perceptron of Figure 9 as a single perceptron that implements the same function? If yes, then draw the perceptron.

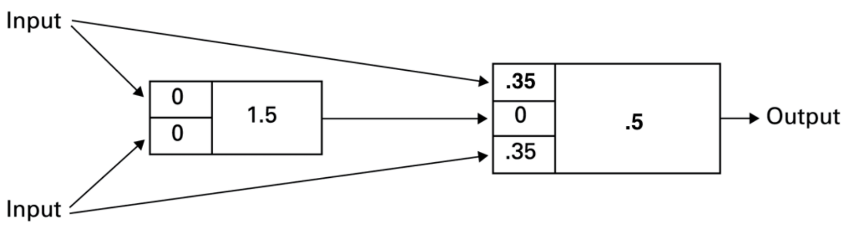

Figure 9: A two layer neural network

Exercise 6. Logical Functions and Neural Networks

The NAND gate is universal for computation, that is, we can build any computation up out of NAND gates. We saw in Exercise 5. that a single perceptron can model a NAND gate. From here, it follows that using networks of perceptrons we can compute any logical function.
For example, we can use NAND gates to build a circuit which adds two bits, x_1 and x_2. This requires computing the bitwise sum, x_1 XOR x_2, as well as a carry bit which is set to 1 when both x_1 and x_2 are 1, i.e., the carry bit is just the bitwise product x_1x_2. The circuit is depicted in Figure 10.

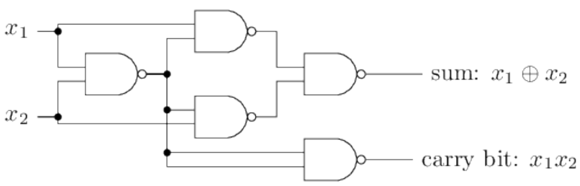

Figure 10: The adder circuit of Exercise 6.. All gates are NAND gates.

Draw a neural network of NAND perceptrons that would simulate the adder circuit from the figure. [You do not need to decide the weights. You have already discovered which weights for a single perceptron would implement a NAND function in Exercise 5.]

What is the advantage of neural networks over logical circuits when representing Boolean functions?