# Task Sheet 1, Solution

Nico Schmidt (nico.schmidt@uzh.ch), Mathias Weyland (mathias.weyland@uzh.ch)

## Solutions

1. Refer to chapter 2 of the script for a detailed discussion of the five basic concepts of neural networks.
2. Refer to chapter 1.2 of the script for a detailed discussion of differences between conventional computers and brains.
3. The *weight* corresponds to the synaptic strength. The *output* corresponds to spikes in the axon. The *inputs* corresponds to spikes in dendrites, and the *connections* correspond to meeting points between dendrites and axons.
4. The solution to the NOR problem is depicted in the following table:

| $\xi_0$ | $\xi_1$ | $\xi_2$ | $w_0$ | $w_1$ | $w_2$ | $\sum w_i\xi_i$ | $o$ | $\zeta$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | −0.4 | −0.5 | 0.5 | −0.4 | 0 | 1 |
| 1 | 0 | 1 | 0.2 | −0.5 | 0.5 | 0.7 | 1 | 0 |
| 1 | 1 | 0 | −0.4 | −0.5 | −0.1 | −0.9 | 0 | 0 |
| 1 | 1 | 1 | −0.4 | −0.5 | −0.1 | −1.0 | 0 | 0 |
| 1 | 0 | 0 | −0.4 | −0.5 | −0.1 | −0.4 | 0 | 1 |
| 1 | 0 | 1 | 0.2 | −0.5 | −0.1 | 0.1 | 1 | 0 |
| 1 | 1 | 0 | −0.4 | −0.5 | −0.7 | −0.9 | 0 | 0 |
| 1 | 1 | 1 | −0.4 | −0.5 | −0.7 | −1.6 | 0 | 0 |
| 1 | 0 | 0 | −0.4 | −0.5 | −0.7 | −0.4 | 0 | 1 |

5. There are four different inputs to a binary function: (0,0), (0,1), (1,0), (1,1). The question now is, how many distinct sets of outputs can be assigned to these four inputs. Visualizing these four inputs as rows, similar to a truth table, the first row can have either 0 or 1 as output. The same holds for the remaining rows, leading to $2 \cdot 2 \cdot 2 \cdot 2 \cdot = 2^4 = 16$ possibilities.

6. By trial, one can separate each of the resulting functions except two: The XOR and the XNOR function. These functions are shown in figure 1.


Figure 1: XOR (left) and XNOR (right) functions. A filled circle represents 1 and a hollow circle a 0.

7. The decision boundary ($3 + \xi_1 + 3\xi_2 = 0$) of a perceptron with weight vector $w^\mathsf{T} = (w_0, w_1, w_2) = (3, 1, 3)$ is shown in figure 2. Anything above the red line is classified as 1, anything below as 0.

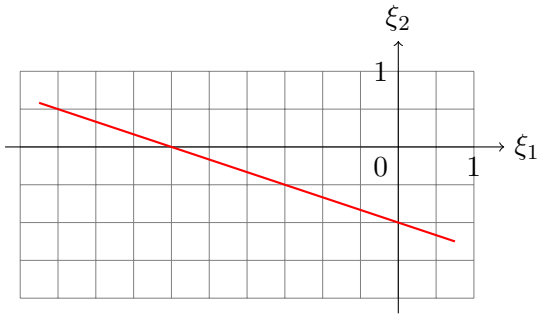


Figure 2: Decision boundary

8. Both of these pattern sets are linearly dependent and not linearly separable. See the discussion for techniques on how to figure this out.[1]

9. True or false:

   a) *Adalines and Perceptrons use the same activation function.* This is not true since perceptrons use a threshold function whereas adalines use a linear function.

[1] Note: There was a typo in this exercise: The index of $\zeta$ was always 1, but should have been incremented by 1 in every row.

b) *The learning rules for Perceptron, Adalines and MLP can be described by the generalized delta rule. Only the delta differs in the three rules.* This is true.The generalized delta rule is

$$\Delta w_{ij} = -\eta \xi_j \delta_i \ .$$

By comparing the specific learning rules, you can see that any differences that are specific to a particular learning rule can be part of $\delta_i$. However, you might argue that $\delta$ loses its semantic meaning.

c) *Adalines use a threshold, whereas Perceptrons do not.* This is false because exactly the opposite is the case. However, you might argue that a threshold can be applied to the Adaline output if used for classification.

d) *Perceptrons adjust their weights only in the case of an error, Adalines adjust them always.* This is true because the Perceptron update rule is skipped if there is no error. The Adaline update rule on the other hand is always applied. However, the weight change is 0 if there is no error (a condition that is hardly ever encountered in practice), so you might argue that adjusting the weights by 0 is not adjusting them at all.

e) *For the Perceptron there exists an explicit solution (without learning), if the classes are linearly independent.* This is false. There is no way to assign proper weights to a perceptron without a learning process of some sort.

f) *Perceptrons and Adalines can only learn linearly separable problems successfully.* This is true and is the motivator for MLPs.

g) *By adding more layers, Adalines can learn non-linear problems.* This is false: Adding linear units in hidden layers does not improve the situation. However, you might argue that the question allows for addition of nonlinear units in hidden layers, which would allow for learning non-linear problems.

h) *MLPs can learn non-linear problems successfully.* This is true.

10. The plots are shown in figure 3. The derivatives are

$$\frac{\mathrm{d}g(x)}{\mathrm{d}x} = \frac{\exp(-x)}{(1+\exp(-x))^2}, \ \frac{\mathrm{d}h(x)}{\mathrm{d}x} = 1 - \left(\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}\right)^2$$

and can be computed using the chain rule and the quotient rule. Also, the identity

$$\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$$

might be helpful.

The properties can be found e.g. as follows:

$$\begin{aligned} g(x)(1-g(x)) &= \frac{1}{1+\exp(-x)}\left(1-\frac{1}{1+\exp(-x)}\right) \\ &= \frac{1}{1+\exp(-x)} - \frac{1}{(1+\exp(-x))^2} \\ &= \frac{1+\exp(-x)}{(1+\exp(-x))^2} - \frac{1}{(1+\exp(-x))^2} \\ &= \frac{1+\exp(-x)-1}{(1+\exp(-x))^2} \\ &= \frac{\exp(-x)}{(1+\exp(-x))^2} = \frac{\mathrm{d}g(x)}{\mathrm{d}x} \end{aligned}$$

$\frac{\mathrm{d}h(x)}{\mathrm{d}x} = 1-h^2(x)$ follows immediately if the derivative is computed as shown above.

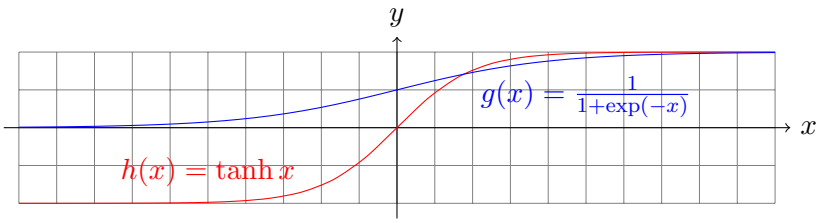


Figure 3: Sigmoid and hyperbolic tangent functions.

11. All you know is that the solution will be found in a finite number of iterations if the patterns are linearly separable. There is no guarantee that $10^6$ iterations are sufficient, though. Hence the answer to this question is no.

12. This is a general form of question 7. For two inputs, we classify the input as 1 if $w_0 + w_1\xi_1 + w_2\xi_2 \geq \Theta$ and as 0 otherwise. The decision boundary is therefore exactly at $w_0 + w_1\xi_1 + w_2\xi_2 = \Theta$. This equation is a straight line in the input space, as you can see by restructuring.