Benjamin Grewe, Matthew Cook, Giacomo Indiveri, Daniel Kiper, Wolfger von der Behrens, Valerio Mante
Lecture 10
Asst: Lucas Pompe, Eduarda Streit Morsch, Reinhard Loidl{`lucas, eduarda, rloidl`}`@ini.uzh.ch`

Exercise 10.1: The McCulloch-Pitts Neuron

The first computational model for an artificial neuron was proposed by McCulloch and Pitts in 1943. The inputs $\mathbf{x} = (x_1, \ldots, x_n)$ and the output y have binary values (0 or 1).

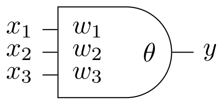


Figure 1: Example McCulloch-Pitts neuron with three inputs.

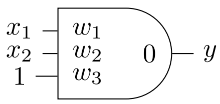
The output of such a neuron is determined by multiplying each input x_i with the corresponding weight w_i and summing these values over all inputs. The resulting *activation* is compared to the neuron's threshold θ. If the activation exceeds the threshold (including equality), the neuron becomes *active* ($y = 1$), otherwise it remains *inactive* ($y = 0$).

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

1. Find similarities and differences to biological neurons.
2. Characterize the functions that can be realized by a McCulloch-Pitts neuron. For that, consider a neuron with two inputs. From all possible mappings of two binary inputs to one binary output, find two mappings that can be realized and two that can not be realized. Explain why some mappings can not be realized.
3. Figure out what the effect of any of the following modifications is. Does it change the number of realizable functions of the neuron?
   (a) Represent the input inactivity by -1 instead of 0. In this way, $x_i = -1$ for inactive units.
   (b) Allow only integer values for the weights (i.e., $w_i \in \mathbb{Z}$ for all i).
   (c) Set the threshold θ to 0.

Exercise 10.2: The Perceptron Learning Algorithm

Consider the following McCulloch-Pitts neuron with two inputs x_1 and x_2 and no threshold ($\theta = 0$), but an "input" x_0 whose value is fixed to 1.



Perform the perceptron learning with a learning rate of $\alpha = 0.3$ on this neuron by completing the Table **??**.
The learning rate expresses (as an absolute value) by how much you update the weight of an input that was active, when the perceptron made an error (the sign depends on the sign of the error):

$$\Delta w_i = \pm \alpha \cdot x_i$$

Table 1: The steps of a perceptron learning algorithm.

x_1	x_2	desired output	current output	w_1	w_2	w_3
1	0	1	0	–0.4	0.5	–0.4
0	1	1				
1	1	1				
0	0	0				
1	0	1				
0	1	1				
1	1	1				
0	0	0				
1	0	1				
0	1	1				
1	1	1				
0	0	0				