HUMBOLDT-UNIVERSITÄT ZU BERLIN
BERNSTEIN CENTER FOR COMPUTATIONAL NEUROSCIENCE

Humboldt-Universität zu Berlin
Philippstr. 13 House 6
Prof. Richard Kempter

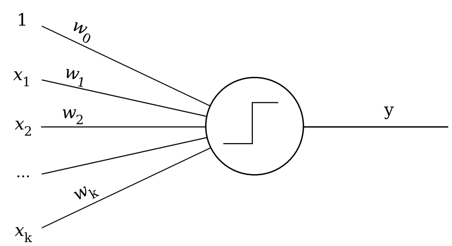
Phone: 030/2093-9110
Fax: 030/2093-6771
webpage: http://www.bccn-berlin.de/

**Models of Neural Systems, WS 2018/19**
**Computer Practical 1**
Solution due on November, 5th, 2018, at 10 am

## McCulloch-Pitts Neurons and Perceptron Learning

In this exercise we introduce a simple neuron model proposed by McCulloch and Pitts (1943). Historically, this model provided the first simplification of biological neurons as computational units. Here, we will show how such neurons can be trained with supervision to perform real-world computations such as classification. Finally, we will explore its severe limitations.

1. **McCulloch-Pitts neuron**




   (a) Implement a McCulloch-Pitts neuron (see the diagram):

   $$y(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}^T\mathbf{x}\right),$$

   where $\mathbf{x} = [-1, x_1, x_2, \dots, x_k]$ is a vector of inputs, $\mathbf{w} = [w_0, w_1, w_2, \dots, w_k]$ is a vector of weights and $y(\mathbf{x})$ is the output.

   (b) Take weights $\mathbf{w} = [3, 2, 2]$ and two binary inputs $x_1, x_2 \in \{-1, +1\}$. Show that the neuron performs a logical AND operation.

2. **Activation functions**

   Often, the step function (as used in the McCulloch-Pitts neuron) is too crude for estimating the activity of a neuron as a function of its input. Plot the following functions for three reasonable choices of the free parameter $a > 0$. What is the meaning of the parameter $a$?

   (a) Sigmoid function:

   $$f(x) = \frac{2}{1 + \exp(-ax)} - 1$$

   (b) Hyperbolic tangent function:

   $$g(x) = \tanh(ax).$$

   (c) Piecewise linear function:

   $$l(x) = \begin{cases} 1 & \text{if } x \geq \frac{1}{a} \\ ax & \text{if } -\frac{1}{a} < x < \frac{1}{a} \\ -1 & \text{if } x \leq -\frac{1}{a}. \end{cases}$$

How can you choose $a$ such that you obtain the step function from any of the three functions?

$$h(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0. \end{cases}$$

3. **Rosenblatt's perceptron**

   (a) Prepare a training set $\{\mathbf{x}_i, d(\mathbf{x}_i)\}$ for $i = 1, 2, \ldots, 1000$, where $\mathbf{x}_i = [-1, x_{i,1}, x_{i,2}]$ is an input vector with random values $x_{i,1}$ and $x_{i,2}$ drawn from a standard normal distribution. $d(\mathbf{x})$ is the desired response. Let the desired response be a comparison between the coordinates $x_1$ and $x_2$:

   $$d(\mathbf{x}) = \begin{cases} 1, & \text{if } x_2 \geq 0.5 - x_1, \\ -1, & \text{if } x_2 < 0.5 - x_1. \end{cases}$$

   (b) Train a McCulloch-Pitts neuron on the training set by presenting all examples iteratively and, after each example $\mathbf{x}_i$, updating the weights using an error-correction update rule:

   $$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \eta(d(\mathbf{x}_i) - y(\mathbf{x}_i))\mathbf{x}_i,$$

   where $\eta > 0$ is the learning rate and $y(\mathbf{x})$ is the response of a McCulloch-Pitts neuron, as defined in exercise 1:

   $$y(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}^T \mathbf{x}\right).$$

   One training "epoch" is completed, when all examples have been presented once. At the end of each epoch, check whether the weights have changed compared to the beginning of the epoch. Train the neuron over multiple epochs until no weight changes. What values for $\eta$ do you use, and why?

   (c) Test on a new dataset (validation set) that the neuron can indeed perform the trained comparison function.

   (d) Plot the training set and label each input vector according to its response class. Superimpose the weight vector on the same plot (think about the bias term $w_0$). Explain in what sense the weight vector is optimal.

4. **Linear separability**

   (a) Train a perceptron to perform the XOR function on binary inputs. As a reminder, the truth table of XOR is:

   | $p$ | $q$ | $p \oplus q$ |
   |---|---|---|
   | 0 | 0 | 0 |
   | 0 | 1 | 1 |
   | 1 | 0 | 1 |
   | 1 | 1 | 0 |

   (b) Show that the learning algorithm does not converge, i.e., the weights do not settle on fixed values.

   (c) Plot the XOR classification problem on an Euclidean plane. Explain why the problem is not linearly separable.

**Upload your solution to Moodle (see also submission checklist in the general information section). Please, comment your code and provide short answers to the questions in each task.**

CONTACT

| CONTACT | LOCATION | PHONE | EMAIL |
|---|---|---|---|
| GREGORY KNOLL | HAUS 2 | 2093 6247 | GREGORY@BCCN-BERLIN.DE |
| PAULA KUOKKANEN | ITB, HAUS 4, ROOM 107 | 2093 98407 | PAULA.KUOKKANEN@HU-BERLIN.DE |
| ERIC REIFENSTEIN | ITB, HAUS 4, ROOM 010 | 2093 98415 | E.REIFENSTEIN@GMAIL.COM |
| NATALIE SCHIEFERSTEIN | ITB, HAUS 4, ROOM 010 | 2093 98415 | NATALIE.SCHIEFERSTEIN@BCCN-BERLIN.DE |