HUMBOLDT-UNIVERSITÄT ZU BERLIN
BERNSTEIN CENTRE FOR COMPUTATIONAL NEUROSCIENCE

Humboldt-Universität zu Berlin
Philippstr. 13 House 6

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

Models of Neural Systems I, WS 2009/10
Computer Practical 3

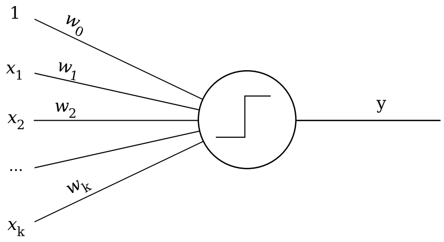
Solutions to hand in on: November, 9th, 2009

Supervised Learning

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

Exercises

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

(b) Take weights $w = [-3, 2, 2]$ and two binary inputs $x_1, x_2 \in \{0, 1\}$. Show that the neuron performs a logical AND operation.

2. **Rosenblatt's perceptron**

(a) Prepare a training set $\{\mathbf{x}, d(i)\}$ for $i = 1, 2, \ldots, n$, where $\mathbf{x}(i) = [1, x_1(i), x_2(i)]$ is the input vector and $d(i)$ is the desired response. Let the desired response be a comparison between two inputs:

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

(b) Train a McCulloch-Pitts neuron on the training set using an error-correction update rule:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta[d(k) - y(\mathbf{x}(k))]\mathbf{x}(k), \tag{1}$$

where $\eta > 0$ is the learning rate. Present the training set repeatedly until no weight changes.

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

(d) Plot the training set distinguishing between the response classes. Superimpose the weight vector on the same plot (without bias term w_0). Explain why the weight vector is optimal.

3. **Linear separability**

(a) Train a perceptron to perform the XOR function on binary inputs.

(b) Show that the learning algorithm does not converge i.e. the weights do not set to fixed values.

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

Contact

Richard Kempter	Phone: 2093-8925	Email: r.kempter(at)biologie.hu-berlin.de
Robert Schmidt	Phone: 2093-8926	Email: r.schmidt@biologie.hu-berlin.de
Bartosz Telenczuk	Phone: 2093-8838	Email: b.telenczuk@biologie.hu-berlin.de