Exercises for the course
Machine Learning 1
Winter semester 2018/19

Abteilung Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 11

Exercise 1: Designing a Neural Network (30 P)

We would like to implement a neural network that classifies data points in $\mathbb{R}^2$ according to decision boundary given in Figure 1 (see below). We consider as an elementary computation the *threshold neuron* whose relation between inputs $(a_i)_i$ and output a_j is given by

$$z_j = \textstyle\sum_i a_i w_{ij} + b_j \qquad a_j = 1_{z_j > 0}.$$

(a) *Design* at hand a neural network that takes x_1 and x_2 as input and produces the output "1" if the input belongs to class A, and "0" if the input belongs to class B. *Draw* the neural network model and *write down* its corresponding weight and bias parameters.

Exercise 2: Backward Propagation (40 P)

We consider a neural network with the structure given in Figure 2 (see below). For simplicity, we consider biases to be zero. The elementary computation of this network is the *sigmoid neuron* defined as:

$$z_j = \textstyle\sum_i a_i w_{ij} \qquad\qquad a_j = \frac{e^{z_j}}{1 + e^{z_j}}$$

For example, when applied to the neuron 7, it can be written explicitely as:

$$z_7 = x_3 \cdot w_{37} + a_5 \cdot w_{57} \qquad\qquad a_7 = \frac{e^{z_7}}{1 + e^{z_7}}$$

The error function is the squared norm between a target vector $\boldsymbol{t} = (t_8, t_9)$ and the vector of output activations $\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}) = (a_8, a_9)$. That is, the error for a given labeled data point $(\boldsymbol{x}, \boldsymbol{t})$ can be expressed as a function of the parameter vector $\boldsymbol{w} = (w_{14}, w_{16}, w_{26}, \dots)$ as:

$$E(\boldsymbol{w}) = \|\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}) - \boldsymbol{t}\|^2.$$

(a) *Write* the sequence of computations that leads to the evaluation of the error partial derivatives $\partial E/\partial w_{14}$ and $\partial E/\partial w_{16}$. Your answer should outline which computations are common to these two evaluations. Also note that the derivative of the sigmoid activation function can be expressed in terms of neuron activations themselves as $\partial a_j/\partial z_j = a_j \cdot (1 - a_j)$.

Exercise 3: Optimization (30 P)

Consider a simple neural network composed of two neurons with linear activation functions. It is described by the equations

$$z_2 = x_1 w_{12} \qquad a_2 = z_2$$
$$z_3 = a_2 w_{23} \qquad a_3 = z_3$$

with one-dimensional input $x = x_1$, and one-dimensional output $f(x) = a_3$, and where w_{12} and w_{23} are the parameters to learn. We consider the labeled dataset consisting of three data points: $(x^{(1)}, t^{(1)}) = (-1, -1)$, $(x^{(2)}, t^{(2)}) = (0, 0)$, and $(x^{(3)}, t^{(3)}) = (1, 1)$. We would like to minimize the error $E(w_{12}, w_{23}) = \sum_{n=1}^{3} (f(x^{(n)}) - t^{(n)})^2$.

(a) *Analyze* the error function at $(w_{12}, w_{23}) = (0, 0)$, and explain what would be the outcome of starting gradient descent at this point in the parameter space.

(b) *Show* that there are an infinite number of solutions with error $E(w_{12}, w_{23}) = 0$.

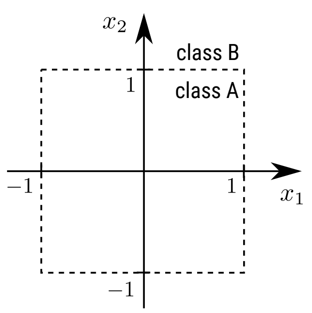

Figure 1

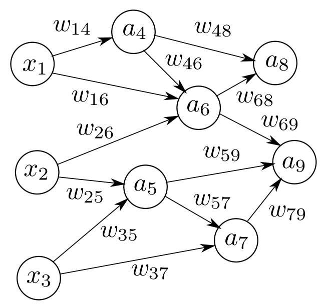

Figure 2