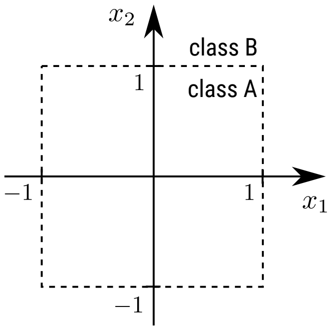
Machine Learning Group
Faculty IV – Electrical Engineering and Computer Science
Technische Universität Berlin

Exercise Sheet 1–2

Exercise 1: Building a Neural Network (10 P)

We would like to implement a neural network that classifies data points in $\mathbb{R}^2$ according to decision boundary given in the figure below.

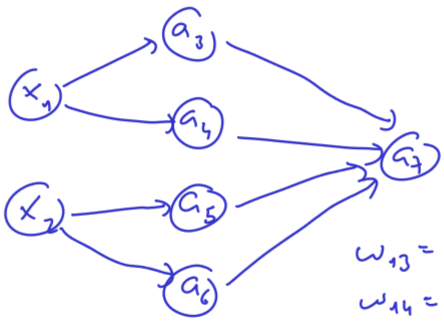


We consider as an elementary computation the *threshold neuron* whose relation between inputs $(a_i)_i$ and output a_j is given by

$$z_j = \sum_i a_i w_{ij} + b_j \qquad a_j = I(z_j > 0).$$

where I is an indicator function that outputs 1 when the statement given as input is true, and 0 otherwise.

(a) *Design* at hand a neural network composed of two layers of parameters, 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* the weights w_{ij} and bias b_j of each neuron.



$w_{13} = 1 \quad b_3 = 1$

$w_{14} = -1 \quad b_4 = 1$

$w_{25} = 1 \quad b_5 = 1$

$w_{26} = -1 \quad b_6 = 1$

$w_{37} = w_{47} = w_{57} = w_{67} = 1 \quad b_7 = -3.5$

Exercise 2: Backpropagation in the Error Function (5 + 5 + 5 + 5 P)

Let y be the prediction of a neural network for some data point $\boldsymbol{x}$. The true target value that the network should predict is given by t. We define the error function to be

$$E = \log\cosh(y - t).$$

This error function can be used as an alternative to the square error and is more robust to outliers.

(a) *Compute* the gradient of the error with respect to the output y of the neural network.

Observe that error can be decomposed as

$$z = y - t \tag{1}$$
$$E = \log\cosh(z) \tag{2}$$

Application of the chain rule gives

$$\frac{\partial E}{\partial y} = \frac{\partial E}{\partial z}\frac{\partial z}{\partial y} \tag{3}$$
$$= \tanh(z) \cdot 1 \tag{4}$$
$$= \tanh(y - t) \tag{5}$$

(b) Assume we have a dataset composed of neural network inputs $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ and associated targets $t_1, \dots, t_N$. We denote by $y_1, \dots, y_N$ the predictions of the neural network for these points. We define the error

$$E = \frac{1}{N}\sum_{k=1}^{N} E_k \qquad \text{with} \qquad E_k = \log\cosh(y_k(\boldsymbol{x}_k, \boldsymbol{w}) - t_k)$$

State the chain rule for transmitting the gradient from the output of the neural network to the model parameters.

$$\frac{\partial E}{\partial \boldsymbol{w}} = \sum_k \frac{\partial E}{\partial E_k}\frac{\partial E_k}{\partial y_k}\frac{\partial y_k}{\partial \boldsymbol{w}}$$

(c) Assume that $y_k(\boldsymbol{x}_k, \boldsymbol{w}) = \sum_{i=1}^{d} w_i x_i^{(k)}$ where $x_i^{(k)}$ denotes the ith element of the vector $\boldsymbol{x}_k$. Compute the gradient of the error function w.r.t. the parameter w_i, i.e. compute $\partial E / \partial w_i$.

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \sum_{k=1}^{N} \frac{\partial E}{\partial E_k}\frac{\partial E_k}{\partial y_k}\frac{\partial y_k}{\partial w_i} \\
&= \sum_{k=1}^{N} \frac{1}{N} \cdot \tanh(y_k - t_k) \cdot \frac{\partial}{\partial w_i}\Big(\sum_{i'=1}^{d} w_{i'} x_{i'}^{(k)}\Big) \\
&= \frac{1}{N}\sum_{k=1}^{N} \tanh(y_k - t_k) x_i^{(k)}
\end{aligned}$$

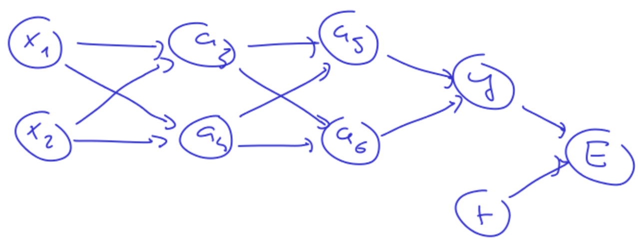
Exercise 3: Backpropagation in a Multilayer Network (10 + 10 P)

We consider a neural network that takes two inputs x_1 and x_2 and produces an output y based on the following set of computations:

$$z_3 = x_1 \cdot w_{13} + x_2 \cdot w_{23} \qquad z_5 = a_3 \cdot w_{35} + a_4 \cdot w_{45} \qquad y = a_5 + a_6$$
$$a_3 = \tanh(z_3) \qquad a_5 = \tanh(z_5) \qquad E = (y - t)^2$$

$$z_4 = x_1 \cdot w_{14} + x_2 \cdot w_{24} \qquad z_6 = a_3 \cdot w_{36} + a_4 \cdot w_{46}$$
$$a_4 = \tanh(z_4) \qquad a_6 = \tanh(z_6)$$

(a) *Draw* the neural network graph associated to this set of computations.



(b) *Write* the set of backward computations that leads to the evaluation of the partial derivative $\partial E/\partial w_{13}$. Your answer should avoid redundant computations. Hint: $\tanh'(t) = 1 - (\tanh(t))^2$.

$$\begin{aligned}
\delta_y &= 2(y-t)\\
\delta_6 &= \delta_y\\
\delta_5 &= \delta_y\\
\delta_3 &= w_{36}(1-a_6^2)\delta_6 + w_{35}(1-a_5^2)\delta_5\\
\frac{\partial y}{\partial w_{13}} &= x_1(1-a_3^2)\delta_3
\end{aligned}$$

Exercise 4: Backpropagation with Shared Parameters (5 + 5 + 5 + 5 P)

Let x_1, x_2 be two observed variables. Consider the two-layer neural network that takes these two variables as input and builds the prediction y by computing iteratively:

$$z_3 = x_1 w_{13}, \quad z_4 = x_2 w_{24}, \qquad a_3 = 0.5\, z_3^2, \quad a_4 = 0.5\, z_4^2, \qquad y = a_3 + a_4.$$

(a) *Draw* the neural network graph associated to these computations.

(b) We now consider the error $E = (y-t)^2$ where t is a target variable that the neural network learns to approximate. Using the rules for backpropagation, compute the derivatives $\partial E/\partial w_{13}$ and $\partial E/\partial w_{24}$.

$$\frac{\partial E}{\partial w_{13}} = \underbrace{\frac{\partial z_3}{\partial w_{13}}}_{x_1} \cdot \underbrace{\frac{\partial a_3}{\partial z_3}}_{z_3} \cdot \underbrace{\frac{\partial y}{\partial a_3}}_{1} \cdot \underbrace{\frac{\partial E}{\partial y}}_{2\cdot(y-t)} \qquad \frac{\partial \ell}{\partial w_{24}} = \dots$$

(c) Let us now assume that w_{13} and w_{24} cannot be adapted freely, but are a function of the same shared parameter v:

$$w_{13} = \log(1+\exp(v)) \qquad \text{and} \qquad w_{24} = -\log(1+\exp(-v))$$

State the multivariate chain rule that links the derivative $\partial E/\partial v$ to the partial derivatives you have computed above.

(d) Using the computed $\partial E/\partial w_{13}$ and $\partial E/\partial w_{24}$, write an analytic expression of $\partial E/\partial v$.

$$\frac{\partial E}{\partial v} = \frac{\partial w_{13}}{\partial v} \cdot \frac{\partial \ell}{\partial w_{13}} + \frac{\partial w_{24}}{\partial v} \cdot \frac{\partial E}{\partial w_{24}} \qquad \text{where} \quad \frac{\partial w_{13}}{\partial v} = \frac{e^v}{1+e^v}\,, \quad \frac{\partial w_{24}}{\partial v} = \frac{e^{-v}}{1+e^{-v}}$$

Exercise 5: Programming (30 P)

Download the programming files on ISIS and follow the instructions.