Exercises for the course
Deep Learning 1
Winter Semester 2024/25

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

Exercise Sheet 2

Exercise 1: Drawing and Simulating a Neural Network $(5 + 15 = 20$ P)

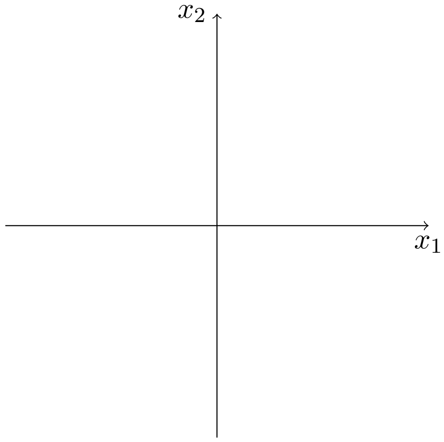
You are given a neural network with two input variables x_1 and x_2 and that produces the output:

$$y = 1 - \rho(x_1) - \rho(x_2) + \rho(x_1 + x_2) \tag{1}$$

where $\rho(z) = \max(0, z)$ is the ReLU nonlinear activation function. The network predicts that the instance fed as input is of class 1 if $y < 0$ and of class 2 if $y > 0$.

(a) *Draw* the neural network associated to Eq. (1), i.e. draw the input variables, the neurons, and the way they are connected. Indicate for each connection its associated weight, and for each neuron its bias and the type of nonlinearity.

(b) *Draw* the decision boundary implemented by this neural network in a 2D coordinate system.
Hint: You can break down the problem into multiple sub-problems, specifically (i) observing that the neural network function is piecewise linear, and identifying its linear pieces, and (ii) for each linear piece, solving the equations $y < 0$ and $y > 0$. You can also observe that $y(x_1, x_2) = y(x_2, x_1)$ for this neural network in order to reduce the number of calculations.



Exercise 2: Backpropagation in the Error Function $(10 + 10 + 10 = 30$ P)

Let $y \in \mathbb{R}$ be the output of a neural network for some data point $\boldsymbol{x} \in \mathbb{R}^d$. 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 is commonly used when training neural network on real-valued prediction tasks (e.g. predicting the energy of a physical system, the price of an object, etc.). We would like to extract the gradient of this error function so that a neural network using it can be learned. Unless stated otherwise we use log to denote the natural logarithm.

(a) Using the chain rule for derivatives, *compute* the gradient of E with respect to the output y of the neural network. Show each step of your derivation.
Hint: The derivative of $\cosh(z)$ *is* $\sinh(z)$. *You can further use the identity* $\tanh(z) = \sinh(z)/\cosh(z)$.

(b) Assume we have a dataset composed of neural network inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ and associated targets $t_1, \ldots, t_N \in \mathbb{R}$. We denote by $y_1, \ldots, y_N \in \mathbb{R}$ 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.

(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 $\partial E / \partial w_i$.

Exercise 3: Backpropagation in a Multilayer Network ($5 + 15 = 20$ 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:

$$z_3 = x_1 \cdot w_{13} + x_2 \cdot w_{23} \qquad z_5 = a_3 \cdot w_{35} + a_4 \cdot w_{45} \qquad y = a_5 \cdot w_{57} + a_6 w_{67}$$
$$a_3 = \tanh(z_3) \qquad a_5 = \tanh(z_5) \qquad E = \log\cosh(y - t)$$

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

(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: $\tanh'(t) = 1 - (\tanh(t))^2$.

Exercise 4: Backpropagation with Shared Parameters ($5 + 10 + 5 + 10 = 30$ 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 $\partial E / \partial w_{24}$.

(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 $\partial E / \partial v$.

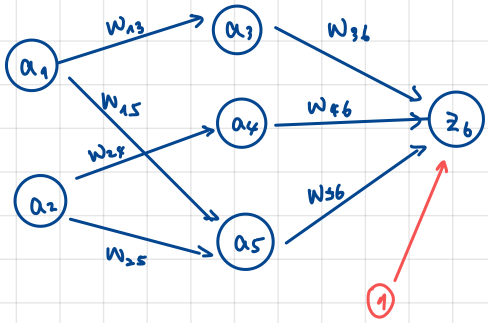
Exercise 1: Drawing and Simulating a Neural Network (5 + 15 = 20 P)

You are given a neural network with two input variables x_1 and x_2 and that produces the output:

$$y = 1 - \rho(x_1) - \rho(x_2) + \rho(x_1 + x_2) \tag{1}$$

where $\rho(z) = \max(0, z)$ is the ReLU nonlinear activation function. The network predicts that the instance fed as input is of class 1 if $y < 0$ and of class 2 if $y > 0$.

(a) *Draw* the neural network associated to Eq. (1), i.e. draw the input variables, the neurons, and the way they are connected. Indicate for each connection its associated weight, and for each neuron its bias and the type of nonlinearity.



$a_0 = 1$

$a_1 = x_1$

$a_2 = x_2$

$a_3 = w_{13} a_1 + b_3$

$a_4 = w_{24} a_2 + b_4$

$a_5 = w_{15} a_1 + w_{25} a_2 + b_5$

$z_6 = w_{36} z_3 + w_{46} z_4 + w_{56} z_5 + b_6$

$b_3 = b_4 = b_5 = 0$

$z_3 = \rho(a_3)$ $\quad w_{13} = 1$

$z_4 = \rho(a_4)$ $\quad w_{24} = 1$

$z_5 = \rho(a_5)$ $\quad w_{15} = w_{25} = 1$

$w_{36} = w_{46} = -1 \quad w_{56} = 1 \quad b_6 = 1$

(b) *Draw* the decision boundary implemented by this neural network in a 2D coordinate system.
Hint: You can break down the problem into multiple sub-problems, specifically (i) observing that the neural network function is piecewise linear, and identifying its linear pieces, and (ii) for each linear piece, solving the equations $y < 0$ and $y > 0$. You can also observe that $y(x_1, x_2) = y(x_2, x_1)$ for this neural network in order to reduce the number of calculations.

$x_1 < 0, x_2 > 0$:

① $x_1 + x_2 > 0$

$y = 1 - 0 - x_2 + x_1 + x_2 = x_1 + 1$

1.1: $x_1 < -1$: $y < 0$ class 1

1.2: $x_1 \geq -1$: $y \geq 0$ class 2

② $x_1 + x_2 < 0$

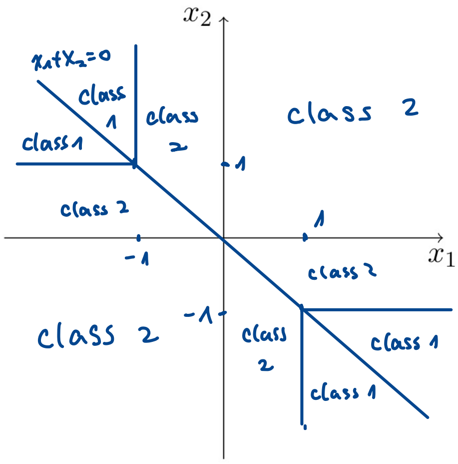
$y = 1 - 0 - x_2 + 0 = 1 - x_2$

2.1: $0 < x_2 \leq 1$, $y > 0$ class 2

2.2: $x_2 > 1$, $y < 0$ class 1

$x_1, x_2 < 0$

$y = 1 > 0$



$x_1, x_2 > 0$:

$y = 1 - x_1 - x_2 + x_1 + x_2 = 1 > 0$

$x_1 > 0, x_2 < 0$

① $x_1 + x_2 > 0$

$y = 1 - x_1 + x_1 + x_2 = 1 + x_2$

1.1: $x_2 < -1$: $y < 0$ class 1

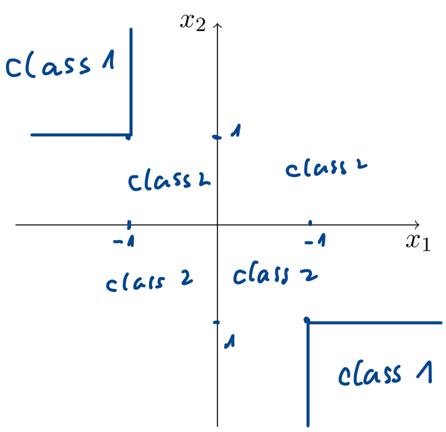
1.2: $-1 \leq x_2 < 0$; $y > 0$ class 2

② $x_1 + x_2 < 0$

$y = 1 - x_1$

2.1: $0 < x_1 < 1$: $y > 0$ class 2

2.2: $x_1 > 1$: $y < 0$ class 1



Exercise 2: Backpropagation in the Error Function ($10 + 10 + 10 = 30$ P)

Let $y \in \mathbb{R}$ be the output of a neural network for some data point $\boldsymbol{x} \in \mathbb{R}^d$. 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 is commonly used when training neural network on real-valued prediction tasks (e.g. predicting the energy of a physical system, the price of an object, etc.). We would like to extract the gradient of this error function so that a neural network using it can be learned. Unless stated otherwise we use log to denote the natural logarithm.

(a) Using the chain rule for derivatives, *compute* the gradient of E with respect to the output y of the neural network. Show each step of your derivation.
Hint: The derivative of $\cosh(z)$ *is* $\sinh(z)$. *You can further use the identity* $\tanh(z) = \sinh(z)/\cosh(z)$.

$$\frac{\partial E}{\partial y} = \frac{\partial E}{\partial(\cosh(y-t))} \cdot \frac{\partial(\cosh(y-t))}{\partial(y-t)} \cdot \frac{\partial(y-t)}{\partial y}$$

$$= \frac{1}{\cosh(y-t)} \cdot \sinh(y-t) \cdot 1$$

$$= \tanh(y-t)$$

-2 (b) Assume we have a dataset composed of neural network inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ and associated targets $t_1, \ldots, t_N \in \mathbb{R}$. We denote by $y_1, \ldots, y_N \in \mathbb{R}$ 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.

$$\frac{\partial E_k}{\partial w} = \frac{\partial E_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial w}$$

$$= \frac{\sinh(y_k(x_k,w) - t_k)}{\cosh(y_k(x_k,w) - t_k)} \cdot \frac{\partial y_k}{\partial w}$$

$$= \tanh(y_k - t_k) \cdot \frac{\partial y_k}{\partial w}$$

$$\frac{\partial E}{\partial w} = \sum_k \frac{\partial E}{\partial E_k} \cdot \frac{\partial E_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial w} = \frac{1}{N}\sum_{k=1}^{N} \frac{\partial E_k}{\partial w} = \frac{1}{N}\sum_{k=1}^{N}\left(\tan(y_k - t_k) \cdot \frac{\partial y_k}{\partial w}\right)$$

-1 (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 $\partial E/\partial w_i$.

$$\frac{\partial y_k}{\partial w_i} = x_i^{(k)}$$

$$\frac{\partial E}{\partial w_i} = \sum_k \frac{\partial E}{\partial E_k} \cdot \frac{\partial E_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial w_i}$$

$$= \frac{1}{N} \cdot \sum_{k=1}^{N}\left(\tanh(y_k - t_k) \cdot x_i^{(k)}\right)$$

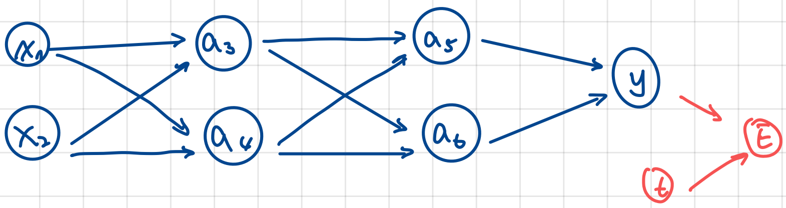
Exercise 3: Backpropagation in a Multilayer Network (5 + 15 = 20 P)

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:

$$z_3 = x_1 \cdot w_{13} + x_2 \cdot w_{23} \qquad z_5 = a_3 \cdot w_{35} + a_4 \cdot w_{45} \qquad y = a_5 \cdot w_{57} + a_6 w_{67}$$
$$a_3 = \tanh(z_3) \qquad a_5 = \tanh(z_5) \qquad E = \log\cosh(y - t)$$

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

(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: $\tanh'(t) = 1 - (\tanh(t))^2$.

$$\delta_7 = \frac{\partial E}{\partial y} = \tanh(y-t)$$

$$\delta_6 = \frac{\partial E}{\partial a_6} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial a_6} = \tanh(y-t) \cdot W_{67}$$

$$\delta_5 = \frac{\partial E}{\partial a_5} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial a_5} = \tanh(y-t) \cdot W_{57}$$

$$\delta_3 = \frac{\partial E}{\partial a_3} = \frac{\partial E}{\partial a_5} \cdot \frac{\partial a_5}{\partial z_5} \cdot \frac{\partial z_5}{a_3} + \frac{\partial E}{\partial a_6} \cdot \frac{\partial a_6}{\partial z_6} \cdot \frac{\partial z_6}{a_3}$$

$$= \tanh(y-t) \cdot W_{57} \cdot \left[1 - \tanh^2(z_5)\right] \cdot W_{35} + \tanh(y-t) \cdot W_{67} \cdot \left[1 - \tanh^2(z_6)\right] W_{36}$$

$$\frac{\partial E}{\partial W_{13}} = \frac{\partial E}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial W_{13}}$$

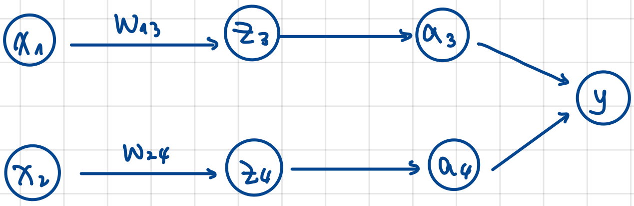
$$= \left(\tanh(y-t) \cdot W_{57} \cdot \left[1 - \tanh^2(z_5)\right] \cdot W_{35} + \tanh(y-t) \cdot W_{67} \cdot \left[1 - \tanh^2(z_6)\right] W_{36}\right) \left[1 - \tanh^2(z_3)\right] \cdot x_1$$

Exercise 4: Backpropagation with Shared Parameters ($5 + 10 + 5 + 10 = 30$ 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.



-2

(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 $\partial E / \partial w_{24}$.

$$\frac{\partial E}{\partial W_{13}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial W_{13}}$$

$$= 2(y-t) \cdot 1 \cdot z_3 \cdot x_1$$

$$= 2(y-t) \cdot x_1^2 W_{13} \qquad (\text{Because of } z_3 = x_1 W_{13})$$

$$\frac{\partial E}{\partial W_{24}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial a_4} \cdot \frac{\partial a_4}{\partial z_4} \cdot \frac{\partial z_4}{\partial W_{24}}$$

$$= 2(y-t) \cdot 1 \cdot z_4 \cdot x_2$$

$$= 2(y-t)\, x_2^2 W_{24}$$

(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.

$$\frac{\partial E}{\partial v} = \frac{\partial E}{\partial W_{13}} \cdot \frac{\partial W_{13}}{\partial v} + \frac{\partial E}{\partial W_{24}} \cdot \frac{\partial W_{24}}{\partial v}$$

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

-2

$$\frac{\partial E}{\partial v} = \frac{\partial E}{\partial W_{13}} \cdot \frac{\partial W_{13}}{\partial v} + \frac{\partial E}{\partial W_{24}} \cdot \frac{\partial W_{24}}{\partial v}$$

$$= 2(y-t) \cdot x_1^2 W_{13} \cdot \frac{e^v}{1+e^v} + 2(y-t)\, x_2^2 W_{24} \cdot \frac{e^{-v}}{1+e^{-v}}$$