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 12

### Exercise 1: Building Neural Networks (20 + 20 P)

We consider the problem of learning decision functions in $\mathbb{R}^2$ where $x = (x_1, x_2)$ denotes the two-dimensional input vector. For this exercise, you only have access to neurons of the type

$$a_j = \sigma\Big(b_j + \sum_i a_i w_{ij}\Big)$$

where $\sigma$ is the step function, i.e.

$$\sigma(t) = \begin{cases} 1 & \text{if } \ t > 0 \\ 0 & \text{if } \ t \leq 0 \end{cases}$$

(a) Construct a neural network that implements the decision boundary below

$$y = \begin{cases} 1 & \text{if } \ \max(x_1, x_2) > 2 \\ 0 & \text{if } \ \max(x_1, x_2) < 2 \end{cases}$$

Specifically, *draw* the neural network graph, and *indicate* for each neuron its weights and bias. (The exact behavior at the decision boundary does not need to be enforced.)

(b) Repeat the exercise for the decision function

$$y = \begin{cases} 1 & \text{if } \ \|x\|_1 > 2 \\ 0 & \text{if } \ \|x\|_1 < 2 \end{cases}$$

### Exercise 2: Condition Number (20 + 10 P)

Consider a supervised dataset composed of inputs $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$ and respective targets $t_1, \dots, t_N \in \mathbb{R}$. Assume that $\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i = \boldsymbol{0}$, i.e. the data is centered. Consider the homogeneous linear model $y = \boldsymbol{w}^\top \boldsymbol{x}$, with $\boldsymbol{w} \in \mathbb{R}^d$ a vector of parameters to be learned, and the regularized mean square objective:

$$\mathcal{E}(\boldsymbol{w}) = \alpha\|\boldsymbol{w}\|^2 + \frac{1}{N}\sum_{i=1}^N (\boldsymbol{w}^\top \boldsymbol{x}_i - t)^2$$

we would like to minimize.

(a) *Show* that the Hessian of the error function $\mathcal{E}(\boldsymbol{w})$ is given by the constant matrix:

$$H(\boldsymbol{w}) = 2(\Sigma + \alpha I)$$

where $\Sigma$ is the covariance of the data.

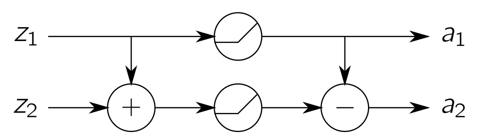
(b) *Show* that the condition number associated to this Hessian matrix is given by

$$c = \frac{\lambda_1 + \alpha}{\lambda_d + \alpha}$$

where $\lambda_1, \dots, \lambda_d$ are the eigenvalues of the matrix $\Sigma$ sorted in decreasing order.

### Exercise 3: Backpropagation (30 P)

Consider some portion of a neural network given by:




$$a_1 = \max(0, z_1)$$
$$a_2 = \max(0, z_1 + z_2) - a_1$$

(a) Assuming that we know the error gradient $(\partial E/\partial a_1\,, \partial E/\partial a_2)$, *compute* the error gradient $(\partial E/\partial z_1\,, \partial E/\partial z_2)$.

## Exercise 1: Building Neural Networks (20 + 20 P)

We consider the problem of learning decision functions in $\mathbb{R}^2$ where $x = (x_1, x_2)$ denotes the two-dimensional input vector. For this exercise, you only have access to neurons of the type

$$a_j = \sigma\Big(b_j + \sum_i a_i w_{ij}\Big)$$

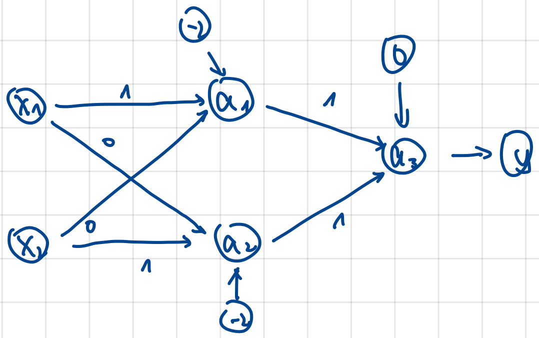
where $\sigma$ is the step function, i.e.

$$\sigma(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases}$$

(a) Construct a neural network that implements the decision boundary below

$$y = \begin{cases} 1 & \text{if } \max(x_1, x_2) > 2 \\ 0 & \text{if } \max(x_1, x_2) < 2 \end{cases}$$

Specifically, *draw* the neural network graph, and *indicate* for each neuron its weights and bias. (The exact behavior at the decision boundary does not need to be enforced.)

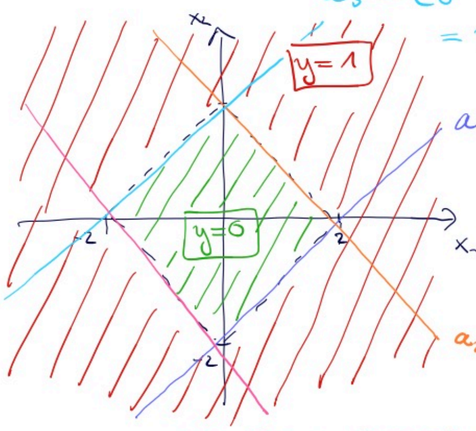



(b) Repeat the exercise for the decision function $\|x\|_1 = \sum |x_i|$

$$y = \begin{cases} 1 & \text{if } \|x\|_1 > 2 \\ 0 & \text{if } \|x\|_1 < 2 \end{cases}$$

b)

$$y = \begin{cases} 1 & \text{if } \|x\| > 2 \\ 0 & \text{if } \|x\| < 2 \end{cases}$$




$$a_3 = \begin{cases} 1 & \text{if } (-x_1) + x_2 > 2 \\ 0 & \text{else} \end{cases} = \sigma(-x_1 + x_2 - 2)$$

$$a_1 = \begin{cases} 1 & \text{if } x_1 + (-x_2) > 2 \\ 0 & \text{else} \end{cases} = \sigma(x_1 - x_2 - 2)$$

$$a_2 = \begin{cases} 1 & \text{if } x_1 + x_2 > 2 \\ 0 & \text{else} \end{cases} = \sigma(x_1 + x_2 - 2)$$

$$a_4 = \begin{cases} 1 & \text{if } (-x_1) + (-x_2) > 2 \\ 0 & \text{else} \end{cases} = \sigma(-x_1 - x_2 - 2)$$

$a_1 = \sigma(x_1 - x_2 - 2)$

$\Rightarrow w_{11} = 1, w_{21} = -1, b_1 = -2$

$a_3 = \sigma(-x_1 + x_2 - 2)$

$\Rightarrow w_{13} = -1, w_{23} = 1, b_3 = -2$

$a_2 = \sigma(x_1 + x_2 - 2)$

$\Rightarrow w_{12} = 1, w_{22} = 1, b_2 = -2$

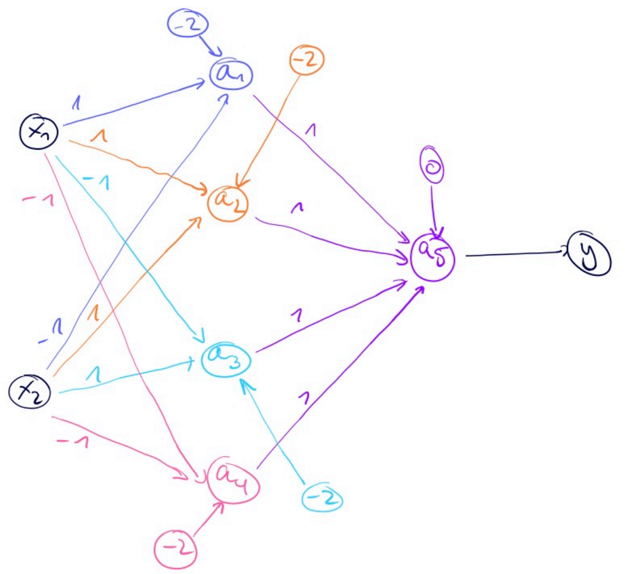
$a_4 = \sigma(-x_1 - x_2 - 2)$

$\Rightarrow w_{14} = -1, w_{24} = -1, b_4 = -2$

Again only one of $\{a_1, a_2, a_3, a_4\}$ has to be active for the condition to hold. Thus we construct $y - a_5$ similarly as above:

$$a_5 = \sigma(a_1 + a_2 + a_3 + a_4)$$

$$\Rightarrow w_{15} = w_{25} = w_{35} = w_{45} = 1, \quad b_5 = 0$$



**Exercise 2: Condition Number (20 + 10 P)**

Consider a supervised dataset composed of inputs $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$ and respective targets $t_1, \dots, t_N \in \mathbb{R}$. Assume that $\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i = \boldsymbol{0}$, i.e. the data is centered. Consider the homogeneous linear model $y = \boldsymbol{w}^\top \boldsymbol{x}$, with $\boldsymbol{w} \in \mathbb{R}^d$ a vector of parameters to be learned, and the regularized mean square objective:

$$\mathcal{E}(\boldsymbol{w}) = \alpha\|\boldsymbol{w}\|^2 + \frac{1}{N}\sum_{i=1}^N (\boldsymbol{w}^\top \boldsymbol{x}_i - t)^2$$

we would like to minimize.

$\Sigma^{-1} v_i = \lambda_i^{-1} v_i$

(a) *Show* that the Hessian of the error function $\mathcal{E}(\boldsymbol{w})$ is given by the constant matrix:

$$H(\boldsymbol{w}) = 2(\Sigma + \alpha I)$$

where $\Sigma$ is the covariance of the data.

$(I + \alpha \Sigma^{-1}) v_i$

(b) *Show* that the condition number associated to this Hessian matrix is given by

$$c = \frac{\lambda_1 + \alpha}{\lambda_d + \alpha}$$

$v_i + \alpha \lambda_i^{-1} v_i$

where $\lambda_1, \dots, \lambda_d$ are the eigenvalues of the matrix $\Sigma$ sorted in decreasing order.

$1 + \alpha \lambda_i^{-1}$

a)

$$\frac{\partial \mathcal{E}(w)}{\partial w} = 2\alpha w + \frac{1}{N} \cdot \sum_{i=1}^{N} 2 \cdot (w^T x_i - t)\, x_i$$

$$H(w) = \frac{\partial^2 \mathcal{E}(w)}{\partial w^2} = 2\alpha + \underbrace{\frac{1}{N} \sum_{i=1}^{N} 2 x_i x_i^T}_{= \Sigma}$$

$$= 2(\alpha I + \Sigma)$$

b)

$$\lambda = v^T H v = 2 v^T(\alpha I + \Sigma) v \qquad \text{with } \|v\| = 1$$

$\uparrow$ for $H$

$$= 2\alpha \cdot v^T v + 2 v^T \Sigma v$$

$$= 2\alpha \|v\|^2 + 2 \underbrace{v^T \Sigma v}_{= \lambda_i \text{ for } \Sigma}$$

$$= 2(\alpha + \lambda_i)$$

$$c = \frac{\lambda_{max}}{\lambda_{min}} = \frac{2(\alpha + \lambda_1)}{2(\alpha + \lambda_n)}$$

b) Let $\hat{\lambda}_1, \dots, \hat{\lambda}_d$ be the eigenvalues of $2(\Sigma + \alpha I)$ sorted in decreasing order. Then we have

$$c = \frac{\hat{\lambda}_1}{\hat{\lambda}_d}$$

We will now show that $\hat{\lambda}_i = k \cdot (\lambda_i + \alpha)$ for some constant $k$.

Let $v_i$ be an eigenvector of $\Sigma$ with corresponding eigenvalue $\lambda_i$, i.e. $\Sigma v_i = \lambda_i v_i$. Then, $v_i$ is also an eigenvector of $2(\Sigma + \alpha I)$:

$$2(\Sigma + \alpha I) v_i = 2\underbrace{\Sigma v_i}_{=\lambda_i v_i} + 2\alpha \underbrace{I v_i}_{= v_i}$$

$$= 2\lambda_i v_i + 2\alpha v_i$$

$$= \underbrace{2(\lambda_i + \alpha)}_{= \hat{\lambda}_i} v_i$$

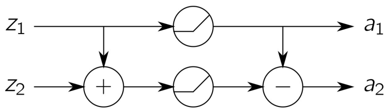
with eigenvalue $\hat{\lambda}_i = 2(\lambda_i + \alpha)$

Thus, we have

$$c = \frac{\hat{\lambda}_1}{\hat{\lambda}_d} = \frac{2(\lambda_1 + \alpha)}{2(\lambda_d + \alpha)} = \frac{\lambda_1 + \alpha}{\lambda_d + \alpha}$$

**Exercise 3: Backpropagation (30 P)**

Consider some portion of a neural network given by:




$$a_1 = \max(0, z_1)$$
$$a_2 = \max(0, z_1 + z_2) - a_1$$

(a) Assuming that we know the error gradient $(\partial E/\partial a_1\,, \partial E/\partial a_2)$, *compute* the error gradient $(\partial E/\partial z_1\,, \partial E/\partial z_2)$.

$$\frac{\partial E}{\partial z_1} = \frac{\partial E}{\partial a_1}\frac{\partial a_1}{\partial z_1} + \frac{\partial E}{\partial a_2}\frac{\partial(\max(0, z_1 + z_2) - a_1)}{\partial z_1}$$

$$= \frac{\partial E}{\partial a_1}\frac{\partial a_1}{\partial z_1} + \frac{\partial E}{\partial a_2}\left(\frac{\partial \max(0, z_1 + z_2)}{\partial z_1} - \frac{\partial a_1}{\partial z_1}\right)$$

$$= \frac{\partial E}{\partial a_1} 1_{z_1 > 0} + \frac{\partial E}{\partial a_2}\left(1_{z_1 + z_2 > 0} - 1_{z_1 > 0}\right)$$

$$\frac{\partial E}{\partial z_2} = \frac{\partial E}{\partial a_2}\frac{\partial a_2}{\partial z_2} = \frac{\partial E}{\partial a_2}\left(1_{z_1 + z_2 > 0} - 1_{z_1 > 0}\right)$$