Data Analytics and Machine Learning Group
Department of Informatics
Technical University of Munich

Ecorrection

Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance

ne Learning

Exam: IN2064 / Endterm **Date:** Thursday 13th February, 2020

Examiner: Prof. Dr. Stephan Günnemann **Time:** 17:00 – 19:00

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10
I										

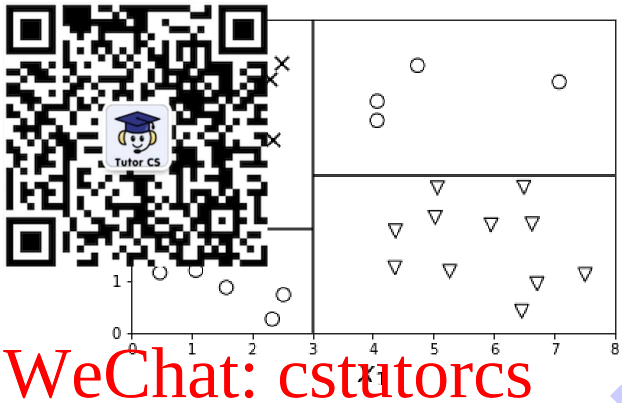
Working instructions

- This exam consists of **16 pages** with a total of **10 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 92 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - A4 sheet of handwritten notes (two sides)
  - **no other materials (e.g. books, cell phones, calculators) are allowed!**
- Only write on the sheets given to you by supervisors. If you need more paper, ask the supervisors.
- Last two pages can be used as scratch paper.
- All sheets (including scratch paper) have to be returned at the end.
- **Only use a black or a blue pen (no pencils, red or green pens)!**
- Write your answers only in the provided solution boxes or the scratch paper.
- **For problems that say "Justify your answer" you only get points if you provide a valid explanation.**
- **For problems that say "Prove" you only get points if you provide a valid mathematical proof.**
- If a problem does not say "Justify your answer" or "Prove" it's sufficient to only provide the correct answer.
- Exam duration - 120 minutes.

Left room from ___________ to ___________ / Early submission at ___________

Problem 1 Decision Trees (12 credits)

You are given a dataset with points from three different classes and want to classify them based on a decision tree. The plot below illustrates the data points (class labels are indicated by the symbols [$\times, \circ, \triangledown$]) and the decision boundaries of a decision tree.



0 1 2 3 4

a) Draw the corresponding decision tree. Make sure that you include the feature (x_1 or x_2) and threshold of the split as well as the number of samples of each class that pass the corresponding inner node or leaf node.

$x_1 < 3?$
[10, 9, 11]

$x_2 < 2?$ [10, 5, 0] | $x_2 < 3?$ [0, 4, 11]

[0, 5, 0] [10, 0, 0] [0, 0, 11] [0, 4, 0]

Each inner node of the tree must include: feature and threshold, number of samples of each class (value = [....]).
Each leaf node must include: number of samples of each class (value = [....]).
Point: root ✓, inner nodes ✓ ✓, leave node ✓

b) Compute the Gini index of each node of your decision tree.

Note: Your answer may contain improper fractions (e.g. $\frac{33}{117}$)

0 1 2 3 4

Gini index: $i_G(t) = \sum_{i \in C} \pi_i(1 - \pi_i) = 1 - \sum_{i \in C} \pi_i^2$

Root node: $i_G(t) = 1 - \left(\frac{10}{30}\right)^2 - \left(\frac{9}{30}\right)^2 - \left(\frac{11}{30}\right)^2 = \frac{598}{900} \approx 0.664$ ✓

Left child of root: $i_G(t)$ … $\left(\frac{0}{15}\right)^2 = \frac{100}{225} \approx 0.444$ ✓

Right child of root: $i_G(t$ … $\left(\frac{11}{15}\right)^2 = \frac{88}{225} \approx 0.391$ ✓

Left leaf node: $i_G(t) = 1$ … $= 0$

Left-middle leaf node: … $- \left(\frac{0}{5}\right)^2 = 0$

Right-middle leaf node … $)^2 - \left(\frac{0}{10}\right)^2 = 0$

Right leaf node: $i_G(t) =$ … $^2 = 0$ (all leaf nodes) ✓

c) Assume you have a dataset with two-dimensional points from two different classes C_1 and C_2. The points from class C_1 are given by $A = \{(i, i^2) \mid i \in \{1...100\}\} \subseteq \mathbb{R}^2$, while the points from class C_2 are $B = \{(i, \frac{125}{i}) \mid i \in \{1...100\}\} \subseteq \mathbb{R}^2$.

0 1 2 3 4

Construct a decision tree of minimal depth that assigns as many data points as possible to the correct class. Provide for each split the feature and corresponding thresholds. How many and which datapoints are missclassified?

- Split root node based on feature 1 ≤ 5 ✓
- Split both child nodes based on feature 2 ≤ 25 ✓

One point $(5, 25)$ is in both classes and missclassified. ✓
Decision three /Calculation of thresholds/ Picture. ✓

Problem 2 Probabilistic inference (4 credits)

0 1 2 3 4

We are interested in estimating a discrete parameter z that can take values in $\{1, 2, 3, 4\}$.

- We place a categorical prior on z, that is $p(z \mid \pi) = \text{Categorical}(z \mid \pi)$ with $\pi = (0.1, 0.05, 0.85, 0.0)$.
- We choose the following likelihood function: $p(x \mid z) = \text{Exponential}(x \mid 2^z) = 2^z \exp(-x2^z)$.
- We have observe[illegible]

What is the posterior p[illegible]al to 4, i.e. what is $p(z = 4 \mid x, \pi)$? Justify your answer.

Using the Bayes fo[illegible] (writing the Bayes formula)

$$[illegible]\pi) \propto p(x \mid z = 4)p(z = 4 \mid \pi)$$
$$\propto 2^4 \exp(-32 \cdot 2^4) \cdot 0$$
$$\propto 0$$

Since the prior probability $p(z = 4 \mid \pi)$ equals to zero, the posterior probability $p(z = 4 \mid x, \pi)$ equals to zero as well ✓ ✓ ✓.

minus ✓ for each mistake in the formulas, even if the final answer is correct

Problem 3 Probabilistic inference (8 credits)

0 1 2 3 4 5 6 7 8

We are interested in estimating the parameter $\theta \in \mathbb{R}$ of the following probabilistic model:

$$p(x \mid \theta) = \exp(\theta - x - \exp(\theta - x)).$$

We have observed a single sample $x \in \mathbb{R}$ drawn from the above model. Derive the maximum likelihood estimate (MLE) of the parameter θ. Justify your answer.

The maximum likelihood estimate of θ is defined as

$$\theta_{MLE} = \arg\max_\theta p(x \mid \theta) ✓ ✓$$
$$= \arg\max_\theta \log p(x \mid \theta) ✓$$
$$= \arg\min_\theta - \log p(x \mid \theta)$$
$$= \arg\min_\theta \left(-\theta + x + \frac{\exp(\theta)}{\exp(x)}\right).$$

Clearly, this is a convex function of θ. To minimize, compute the derivative ✓, set it to zero ✓ and solve for θ.

$$\frac{\partial}{\partial\theta}\left(-\theta + x + \frac{\exp(\theta)}{\exp(x)}\right) = -1 + \frac{\exp(\theta)}{\exp(x)} \stackrel{!}{=} 0$$
$$\iff \frac{\exp(\theta)}{\exp(x)} \stackrel{!}{=} 1 ✓ ✓$$ for correctly computing and simplifying
$$\iff \theta_{MLE} = x ✓$$ for the correct final answer

Therefore, $\theta_{MLE} = x$.

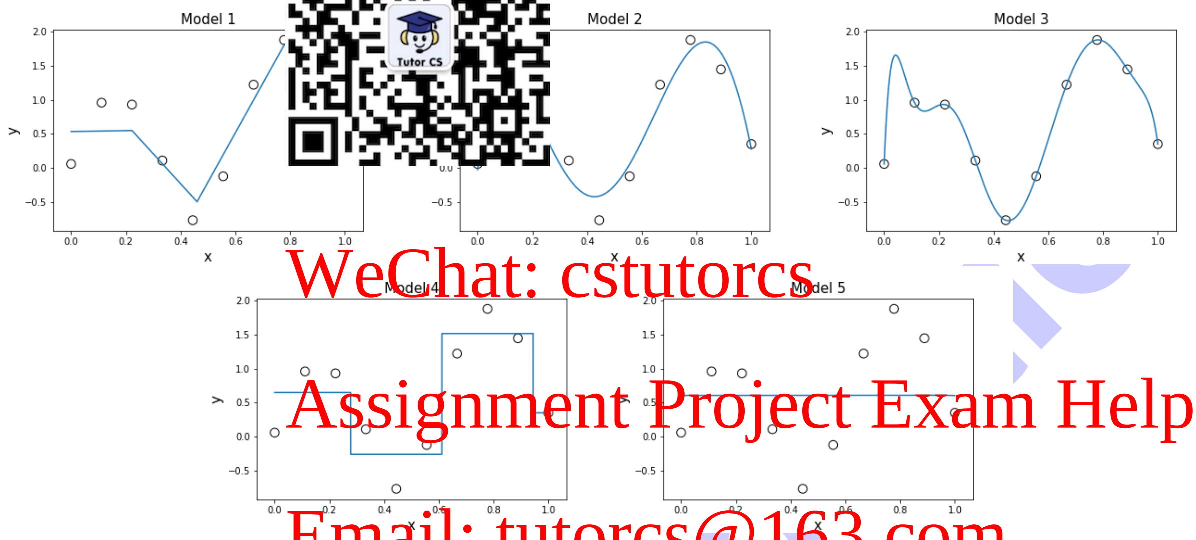
Mark correct answers with a cross ☒

To undo a cross, completely fill out the answer option ■

To re-mark an option, use a human-readable marking ☒■

Problem 4 **Regression (10 credits)**

The following five plots sh ion models fitted to the same dataset. Your task is to assign each of the plots to the c



For each subtask ✓ ✓ if only the correct option is chosen, otherwise 0.

a) Polynomial regression (degree = 5), no regularization

☐ Model 1 ☒ Model 2 ☐ Model 3 ☐ Model 4 ☐ Model 5

b) Polynomial regression (degree = 10), no regularization

☐ Model 1 ☐ Model 2 ☒ Model 3 ☐ Model 4 ☐ Model 5

c) Polynomial regression (degree = 50), L_2 regularization with $\lambda = 10^3$

☐ Model 1 ☐ Model 2 ☐ Model 3 ☐ Model 4 ☒ Model 5

d) Feed-forward neural network with ReLU activation functions, no regularization

☒ Model 1 ☐ Model 2 ☐ Model 3 ☐ Model 4 ☐ Model 5

e) Decision tree of depth 2

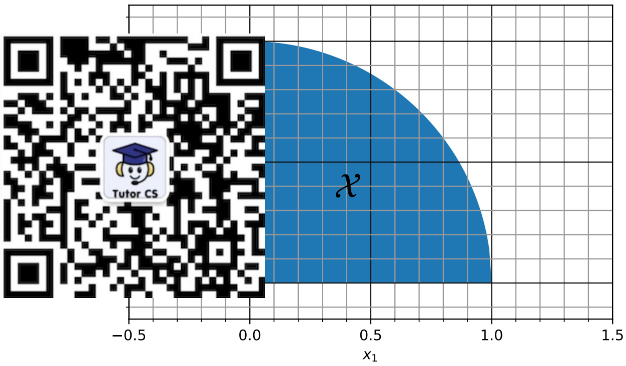
☐ Model 1 ☐ Model 2 ☐ Model 3 ☒ Model 4 ☐ Model 5

Problem 5 Convex optimization (18 credits)

Consider the set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^D : \|\mathbf{x}\|_2 \leq 1 \text{ and } x_i \geq 0 \text{ for all } i = 1, \dots, D\}$ with $1 < D \in \mathbb{N}$.

0 1

a) Draw $\mathcal{X}$ on the provided axes below for $D = 2$.



0 1 2 3 4

b) Write down the function $\pi_{\mathcal{X}}$ projecting an arbitrary point $\mathbf{p} \in \mathbb{R}^2$ on $\mathcal{X}$ for the case $D = 2$.

Note: if you decide to split $\mathbb{R}^2$ into regions and consider them separately then you have to describe the regions analytically (just a reference to your plot from a) will not be sufficient).

We consider the following disjoint decomposition of $\mathbb{R}^2 = \mathcal{X} \cup \mathcal{X}_0 \cup \mathcal{X}_1$ with

- $\mathcal{X}_0 = \{\mathbf{x} \in \mathbb{R}^2 : x_1 < 0 \text{ or } x_2 < 0\}$, here $\pi_{\mathcal{X}}$ is the same as the projection on $[0, 1]^2$, that is

$$\pi_{\mathcal{X}}(\mathbf{x}) = \begin{pmatrix} \min(1, \max(0, x_1)) \\ \min(1, \max(0, x_2)) \end{pmatrix} \text{ for } \mathbf{x} \in \mathcal{X}_0\,,$$

- $\mathcal{X}_1 = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| > 1, x_1 \geq 0 \text{ and } x_2 \geq 0\}$, here $\pi_{\mathcal{X}}$ is the the projection on $\mathcal{B}_1(\mathbf{0})$, therefore

$$\pi_{\mathcal{X}}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \text{ for } x \in \mathcal{X}_1\,,$$

✓✓✓ if the projection $\pi_{\mathcal{X}}$ is correct for the points $\mathbf{x} \notin \mathcal{X}$.

- and finally $\pi_{\mathcal{X}}(\mathbf{x}) = \mathbf{x}$ for $\mathbf{x} \in \mathcal{X}$.

✓ for the case $\mathbf{x} \in \mathcal{X}$.

From now on we consider the setting with an arbitrary $1 < D \in \mathbb{N}$.

c) Prove that $\mathcal{X}$ is convex.

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^D : \|\boldsymbol{x}\|_2 \leq 1, x_i \geq 0 \text{ for all } i = 1, \dots, D\}$$

$$= \{\text{[illegible]} \underbrace{\{\boldsymbol{x} \in \mathbb{R}^D : 0 \leq x_i \leq 1 \text{ for all } i = 1, \dots, D\}}_{\text{cube } [0,1]^D} ✓$$

Therefore, $\mathcal{X}$ is an inte[illegible] sets and hence convex. ✓

If proven by definition of convexity:
✓ ✓ for proving that an intermediate point $\boldsymbol{x}_\lambda$ satisfies $\|\boldsymbol{x}_\lambda\|_2 \leq 1$,
✓ for proving that $(\boldsymbol{x}_\lambda)_i \geq 0$ for all i.

d) Fill in the space in the box below using mathematical notation with a description of the vertices of $\mathcal{X}$. Note that just writing down the definition of vert($\mathcal{X}$) will not be sufficient.

vert($\mathcal{X}$) =

$\{\boldsymbol{x} \in \mathbb{R}^D : \|\boldsymbol{x}\|_2 = 1 \text{ and } x_i \geq 0 \text{ for all } i = 1, \dots, D\}$ ✓ $\cup \{\boldsymbol{0}\}$ ✓

0 1 2 3 4 5 6 7 8

e) Find the maximum of the following constrained optimization problem. Justify your answer, all properties of the objective function and $\mathcal{X}$ that you use should be clearly stated and derived from the previous tasks or results considered in the course.

Hint: results from c) and d) might help you.
Hint: for arbitrary $\boldsymbol{c} \in \mathbb{R}^D$ the maximum of the constrained problem $\max_{\boldsymbol{x}} \boldsymbol{c}^T \boldsymbol{x}$ subject to $\|\boldsymbol{x}\|_2 = 1$ is $\|\boldsymbol{c}\|_2$.

$$\begin{aligned} \text{maximize}_{\boldsymbol{x}} \quad & \sum_{i=1}^{D} x_i + e^{\|\boldsymbol{x}\|_2^2} \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{X} \end{aligned}$$

We will utilize the following:

- $\mathcal{X}$ is convex (from c).
- Function $f : \mathbb{R}^D \to \mathbb{R}$ with $f(\boldsymbol{x}) = \sum_{i=1}^{D} x_i + e^{\|\boldsymbol{x}\|_2^2}$ is convex since the function $h : \mathbb{R}^D \to [0, \infty)$ with $h(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$ is convex and the function $g : [0, \infty) \to \mathbb{R}$ with $g(x) = e^x$ is convex and non-decreasing. Therefore, using the convexity preserving operations we see that $g(h(\boldsymbol{x})) = e^{\|\boldsymbol{x}\|_2^2}$ and linear $l(\boldsymbol{x}) = \sum_{i=1}^{D} x_i$ are both convex and hence $f(\boldsymbol{x}) = l(\boldsymbol{x}) + g(h(\boldsymbol{x}))$ is convex as well.
- Maximum of a convex function over a convex domain is attained at one of its vertices (from the lecture).

✓ ✓ ✓ ✓ for a correct argumentation that it is enough to consider the optimization problem on vert($\mathcal{X}$) only.

Now it is sufficient to look for the maximum value of f over vert($\mathcal{X}$) that consists of $\{\boldsymbol{0}\}$ and a set where for each point it holds that $\|\boldsymbol{x}\|_2 = 1$ and $x_i \geq 0$ for all i (from d).
Case 0: $\boldsymbol{x} = \boldsymbol{0}$, then $f(\boldsymbol{x}) = f(\boldsymbol{0}) = 0 + e^0 = 1$.
Case 1: $\boldsymbol{x} \in B := \{\boldsymbol{x} \in \mathbb{R}^D : \|\boldsymbol{x}\|_2 = 1 \text{ and } x_i \geq 0 \text{ for all } i = 1, \dots, D\}$ then the largest value that f attains at these points is

$$\max\{f(\boldsymbol{x}) : \|\boldsymbol{x}\|_2 = 1, x_i \geq 0\} = e + \max\{\boldsymbol{c}^T\boldsymbol{x} : \|\boldsymbol{x}\|_2 = 1\} \text{ where } \boldsymbol{c} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Using the provided hint we get that the maximum of f over B is $e + \|\boldsymbol{c}\|_2 = e + \sqrt{D}$. Since $\sqrt{D} + e > 1$ we get that the maximum of f over $\mathcal{X}$ is $\sqrt{D} + e$.
✓ ✓ ✓ ✓ if all verteces were considered and the correct results were obtained using valid argumentation.

Problem 6 Kernels (10 credits)

Let $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ be a positive semi-definite matrix and consider for $p \in \mathbb{N}$ the following function

$$k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}, \quad k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_2 + 1)^p.$$

a) Prove that k is a valid kernel using kernel preserving operations known from the course.

0 1 2 3 4

Solution 1:

- ✓ $k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}$ [illegible] $\boldsymbol{A}$ is positive semi-definite (kernel is known from the course).
- ✓ $k_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = 1$ i[illegible]$\boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)^T \phi(\boldsymbol{x}_2)$ for $\phi(\boldsymbol{x}) = 1$.
- ✓ $k_3(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_2 + 1$ is a kernel since it is a sum of two kenels (kernel preserving operation known from the lecture)
- ✓ Finally, subsequently applying the rule that a product of two kernels is a kernel for $k_3 * k_3$ we get that k is a kernel as well.

Solution 2:

- ✓ $k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_2$ is a kernel since $\boldsymbol{A}$ is positive semi-definite. Therefore there exists a map ϕ_A such that $k_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi_A(\boldsymbol{x}_1)^T \phi_A(\boldsymbol{x}_2)$.
- Polynomial kernel $k_p(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \boldsymbol{x}_2 + 1)^p$ is a kernel. ✓ if this fact is used later in the solution
- ✓ ✓ Using the composition rule (kernel preserving operation known from the lecture) we know that $k_p(\phi_A(\boldsymbol{x}_1), \phi_A(\boldsymbol{x}_2)) = (\boldsymbol{x}_1^T \boldsymbol{A} \boldsymbol{x}_2 + 1)^p = k(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is again a kernel.

b) For the special case of $p = 2$ and $\boldsymbol{A} = \boldsymbol{I}$ (identity matrix) write down the corresponding feature map $\phi : \mathbb{R}^D \to \mathbb{R}^M$ such that

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)^T \phi(\boldsymbol{x}_2).$$

What is the dimension M of the feature space in this case?

0 1 2 3 4 5 6

$$\phi(x) = \begin{pmatrix} 1 \\ \sqrt{2}x_1 \\ \vdots \\ \sqrt{2}x_D \\ x_1x_1 \\ x_1x_2 \\ \vdots \\ x_1x_D \\ \vdots \\ x_2x_1 \\ \vdots \\ x_Dx_D \end{pmatrix} \quad \text{and} \quad M = 1 + D + D^2.$$

Comment: in this case we get the quadratic kernel and the feature map includes constant, linear and all quadratic terms we can produce from the initial features.
✓ for the constant term, ✓ ✓ for the linear terms, ✓ ✓ ✓ for the quadratic terms, -✓ if M is wrong, -✓ if only the case of $D = 2$ is considered.

Problem 7 Deep learning (8 credits)

The code snippet below shows an implementation of two functions $f : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$. Given two input vectors $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ we perform the following computations:

$$z = f(\mathbf{x}, \mathbf{y})$$
$$t = g(z)$$

The code below uses [illegible] compute $\frac{\partial t}{\partial \mathbf{x}}$ and $\frac{\partial t}{\partial \mathbf{y}}$ (similarly to how we did it in Tutorial 9: Deep Learning I). How[illegible]ments are missing. Your task is to complete the missing code fragments.

Note: It's also fine to w[illegible] pseudocode (we won't deduct points for small Python syntax errors, etc.).

```
class F:
    def forward(self, x, y):
        self.cache = (x, y)
        ##################################################
        # MISSING CODE FRAGMENT #1
        ##################################################
        return out

    def backward(self, d_out):
        # x, y are np.arrays of shape [N]
        x, y = self.cache
        N = len(x)
        # np.ones(N) returns a vector of ones of shape [N]
        d_y = (x + np.ones(N)) * d_out
        d_x = (y + np.ones(N)) * d_out
        return d_x, d_y

def sigmoid(a):
    return 1 / (1 + np.exp(-a))

class G:
    def forward(self, z):
        self.cache = z
        return sigmoid(z)

    def backward(self, d_out):
        z = self.cache
        ##################################################
        # MISSING CODE FRAGMENT #2
        ##################################################
        return d_z

# Example usage
f = F()
g = G()
x = np.array([1., 2., 3])
y = np.array([-2., 3., -1.])

z = f.forward(x, y)
t = g.forward(z)
d_z = g.backward(1.0)
d_x, d_y = f.backward(d_z)
```

a) Complete the `MISSING CODE FRAGMENT #1`

0 1 2 3 4

Option 1:

```
out = np.dot(x, y) + np.sum(y) - np.sum(x)
```

Option 2:

```
N = len(x
ones = np
out = np                    ot(x, ones) + np.dot(y, ones)
```

Option 3:

```
N = len(x
ones = np
out = np                    - ones)
```

Two points for $x^T y$ ✓ ✓, one point for each of the sums ✓ ✓

b) Complete the `MISSING CODE FRAGMENT #2`

0 1 2 3 4

Option 1:

```
d_z = sigmoid(z) * sigmoid(-z) * d_out
```


Option 2:

```
d_z = sigmoid(z) * (1 - sigmoid(z)) * d_out
```

Three points for the derivative of sigmoid ✓ ✓ ✓, one point for not forgetting `d_out` ✓

Problem 8 SVD and linear regression (8 credits)

0 1 2 3 4 5 6 7 8

You want to perform linear regression on a data set with features $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and targets $\boldsymbol{y} \in \mathbb{R}^N$. Assume that you have already computed the SVD of the feature matrix $\boldsymbol{X} = \boldsymbol{U\Sigma V}^T$. Additionally, assume that $\boldsymbol{X}$ has full rank.

Show how we can compute the optimal linear regression weights $\boldsymbol{w}^\star$ in $\mathcal{O}(ND^2)$ operations by using the result of the SVD.

Hint: Matrix operations ... ymptotic complexity

- *Matrix multiplica... $\in \mathbb{R}^{P \times Q}$ and $\boldsymbol{B} \in \mathbb{R}^{Q \times R}$ takes $\mathcal{O}(PQR)$*
- *Matrix multiplica... $\boldsymbol{A} \in \mathbb{R}^{P \times Q}$ and a diagonal $\boldsymbol{D} \in \mathbb{R}^{Q \times Q}$ takes $\mathcal{O}(PQ)$*
- *Matrix inversion ... atrix $\boldsymbol{C} \in \mathbb{R}^{M \times M}$ takes $\mathcal{O}(M^3)$*
- *Matrix inversion ... trix $\boldsymbol{D} \in \mathbb{R}^{M \times M}$ takes $\mathcal{O}(M)$*

$$
\begin{aligned}
\boldsymbol{w}^* &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \\
&= ((\boldsymbol{U\Sigma V}^T)^T(\boldsymbol{U\Sigma V}^T))^{-1}(\boldsymbol{U\Sigma V}^T)^T\boldsymbol{y} = \\
&= (\boldsymbol{V\Sigma U}^T\boldsymbol{U\Sigma V}^T)^{-1}\boldsymbol{V\Sigma U}^T\boldsymbol{y} = \\
&= (\boldsymbol{V\Sigma}^2\boldsymbol{V}^T)^{-1}\boldsymbol{V\Sigma U}^T\boldsymbol{y} = \\
&= (\boldsymbol{V}^T)^{-1}\boldsymbol{\Sigma}^{-2}\boldsymbol{V}^T\boldsymbol{V\Sigma U}^T\boldsymbol{y} = \\
&= \boldsymbol{V\Sigma}^{-2}\boldsymbol{V}^T\boldsymbol{V\Sigma U}^T\boldsymbol{y} = \\
&= \boldsymbol{V\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{y}
\end{aligned}
$$

Multiplication $\boldsymbol{a} = \boldsymbol{U}^T\boldsymbol{y}$ takes $\mathcal{O}(N \cdot D \cdot 1)$
Multiplication $\boldsymbol{b} = \boldsymbol{\Sigma}^{-1}\boldsymbol{a}$ takes $\mathcal{O}(D)$
Multiplication $\boldsymbol{w} = \boldsymbol{Vb}$ takes $\mathcal{O}(D^2)$
In total, $\mathcal{O}(ND + D + D^2) = \mathcal{O}(ND)$ if $N > D$.

✓ for $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$,
✓ for $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$
✓ for $(\boldsymbol{V\Sigma}^2\boldsymbol{V}^T)^{-1} = (\boldsymbol{V}^T)^{-1}\boldsymbol{\Sigma}^{-2}\boldsymbol{V}^{-1}$
✓ for $(\boldsymbol{V}^T)^{-1}\boldsymbol{\Sigma}^{-2}\boldsymbol{V}^{-1} = \boldsymbol{V\Sigma}^{-2}\boldsymbol{V}^T$
✓ for $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$
✓ for the final answer $\boldsymbol{w}^* = \boldsymbol{V\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{y}$
minus ✓ for any mistake
✓ ✓ for correct big-O analysis

There is a mistake in the task, saying "$\mathcal{O}(ND^2)$" instead of "$\mathcal{O}(ND)$". The task was graded taking this into account.

Problem 9 K-Means (10 credits)

Let $\gamma_i \in \mathbb{R}^D$ for $i = 1, \ldots, K$ be a set of K points more than 4 apart, i.e. $\|\gamma_i - \gamma_j\|_2 > 4$ for all $i \neq j$. Consider K non-empty datasets $\mathcal{X}_i$ each contained within a unit ball around γ_i, i.e. $\|\boldsymbol{x} - \gamma_i\|_2 \leq 1$ for all $\boldsymbol{x} \in \mathcal{X}_i$. Let $\mathcal{X} = \bigcup_{i=1}^{K} \mathcal{X}_i$ be the combined dataset.

Now consider a centroid initialization procedure similar to k-means++, though it deterministically chooses the data point farthest away from all previous centroids. That means it initializes the cluster centers $\boldsymbol{\mu}_i$ as

[illegible] le from $\mathcal{X}$ if $i = 1$

[illegible] $_{-1\}} \|\boldsymbol{x} - \boldsymbol{\mu}_j\|_2$ if $i \in \{2, \ldots, K\}$

a) Explain why this determ[illegible]alization of K clusters assigns each $\boldsymbol{\mu}_i$ to a different ball, i.e. for $i \neq j$ such that $\boldsymbol{\mu}_i \in \mathcal{X}_{i'}$ [illegible]hat $i' \neq j'$.

0 1 2 3 4 5

The first centroid is placed into a random ball. Now assume that a subsequent centroid $\boldsymbol{\mu}_j$ would be placed into the same ball as some previous $\boldsymbol{\mu}_i$. That means that $\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_i\|_2 \leq 2$ because each $\mathcal{X}_{i'}$ is contained within a ball of radius 1. However, because we are placing K centroids into K balls and one ball has been chosen twice, there is at least one ball $\mathcal{X}_{i'}$ that does not have a centroid in it. $\mathcal{X}_{i'}$ is non-empty, so it has an element $\boldsymbol{x} \in \mathcal{X}_{i'}$ which is more than 2 away from any previously chosen centroid because the ball centers are more than 4 apart and the data points can deviate at most 1 from their closest ball center. But that contradicts the construction of $\boldsymbol{\mu}_j$ as the data point that is the furthest away from any previously chosen centroid, in particular $\boldsymbol{\mu}_i$. Therefore, every centroid is placed in a different ball.

✓ ✓ ✓ if the student shows an understanding that the statement is true because the balls are far apart in some way

✓ ✓ if the student works out the crux of the argument clearly, i.e. ≤ 2 vs. > 2 or in words "at most 2" vs. "more than 2"

-✓ for mistakes, erroneous statements or arguments that do not make explicit use of ≤ 2 vs. > 2 and thus would work in the setting $\|\gamma_i - \gamma_j\|_2 > 3.9$ just the same

b) Assuming a), explain why k-means clustering of $\mathcal{X}$ with K clusters and our deterministic k-means++ initialization recovers the underlying structure of the data, i.e. all data points $\boldsymbol{x} \in \mathcal{X}_i$ will be assigned to the same centroid for all i.

0 1 2 3 4 5

Without loss of generality, let the centroids be assigned such that $\boldsymbol{\mu}_i \in \mathcal{X}_i$ for all i. Then each data point $\boldsymbol{x} \in \mathcal{X}_i$ will be assigned to $\boldsymbol{\mu}_i$ because $\|\boldsymbol{\mu}_i - \boldsymbol{x}\|_2 \leq 2$ and $\|\boldsymbol{\mu}_j - \boldsymbol{x}\|_2 > 2$ for all i and $j \neq i$, see a). Because updating centroid $\boldsymbol{\mu}_i$ is a convex combination of data points $\boldsymbol{x} \in \mathcal{X}_i$, $\boldsymbol{\mu}_i$ stays within the bounding ball of $\mathcal{X}_i$ for any i. So the assignments stay the same and k-means terminates in the next iteration.

✓ ✓ ✓ ✓ for arguing convincingly that each $\boldsymbol{x}$ is assigned to the centroid in its ball because again ≤ 2 vs. > 2

✓ for stating that and why the algorithm converges/terminates

-✓ for mistakes, erroneous statements or arguments that do not make explicit use of ≤ 2 vs. > 2 and thus would work in the setting $\|\gamma_i - \gamma_j\|_2 > 3.9$ just the same

Problem 10 Differential Privacy & Fairness (4 credits)

0 1 2

a) What is the *robustness to post-processing* property of Differential Privacy?

You can apply any function on the output from a ϵ-DP mechanism and the new output remains ϵ-DP ✓ as long as you do not touch again the data. ✓

Only one point if "do not touch again the data" is missing.

0 1 2

b) Suppose that we require that the same *percentage* of applicants from two different groups must receive a loan. What fairness criterion are we implementing? What is the biggest downside/con of this criterion?

Demographic Parity, also called Independence and Statistical Parity. ✓

Any of the following downsides are acceptable:

- Rules out the perfect predictor when base rates are different across groups. ✓
- Laziness: We can trivially satisfy the criterion if we give loan to qualified people from one group and random people from the other. ✓
- Too strong, if one group is much more likely to repay compared to the other group. ✓

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.