WRITE FIRST NAME, LAST NAME, AND ID NUMBER ("MATRICOLA") BELOW AND READ ALL INSTRUCTIONS BEFORE STARTING WITH THE EXAM! TIME: 2.5 hours.

FIRST NAME: .....

LAST NAME: .....

ID NUMBER: .....

INSTRUCTIONS

- solutions to exercises must be in the appropriate spaces, that is:
  - Exercise 1: pag. 1, 2, 3
  - Exercise 2: pag. 4, 5
  - Exercise 3: pag. 6, 7, 8
  - Exercise 4: pag. 9, 10, 11, 12

  Solutions written outside the appropriate spaces (including other papersheets) will not be considered.
- the use of notes, books, or any other material is forbidden and will make your exam invalid;
- electronic devices (smartphones, calculators, etc.) must be turned off; their use will make your exam invalid;
- this booklet must be returned in its entirety.

Exercise 1 [8 points]

In the context of supervised learning:

1. provide the definition of the regression task (not "linear regression", "NN regression" no "squared loss", etc.)
2. consider the following model class that is linear in the parameter:

$$h(x) := \mathbf{w}^\top \Psi(x) \quad \Psi(x) = [\psi_1(x), .., \psi_L(x)]^\top \quad x \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^L$$

where $\Psi(x) = [\psi_1(x), .., \psi_L(x)]^\top$ can be a generic function, e.g., recall the polynomial regression case where $\Psi(x) = [1, x, x^2, ..., x^{L-1}]^\top$. Write the explicit expression of the least squares estimator of $\mathbf{w}$ given data (x_k, y_k), $k = 1, ..m$.

3. Recalling the answer to the previous question, consider the one-hidden-layer neural network

$$h(x) := \sum_{i=1}^{L} w_i \sigma(\alpha_i(x - \beta_i)) \quad x \in \mathbb{R}$$

where α_i, w_i, β_i, $i = 1, .., L$, are the network parameters. Show that for α_i and β_i fixed, the optimal w_i can be found in closed form under the square loss.

[Solution: Exercise 1]

1. Regression task is a supervised learning task with:
- domain set $\mathcal{X}$ ... [some more details needed]
- label set $\mathcal{Y} = \mathbb{R}$

Given training data $S = ((x_1, y_1), \dots, (x_m, y_m))$, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ $\forall\ i = 1, \dots, m$, we need to define an hypothesis class $\mathcal{H}$ and a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$, where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ that given an hypothesis $h \in \mathcal{H}$ provides a

---

[Solution: Exercise 1]

measure of how much we loose by predicting the value $h(x)$ for x instead of the (correct) value y. The goal is then to find an hypothesis $\hat{h} \in \mathcal{H}$ with low generalization error

$$L_{\mathcal{D}}(\hat{h}) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(\hat{h}, z)]$$

where $\mathcal{D}$ is the (unknown) probability distribution over $\mathcal{Z}$ from which $(x_i, y_i) \in S$, $i = 1, \dots, m$ have been drawn (as independent samples).

2. Since the model is linear in the parameter, the least square estimator is analogous to the least square estimator for linear models (i.e., $h(\vec{x}) = \vec{w}^T \vec{x}$), with $\vec{\psi}(x)$ playing the role of $\vec{x}$. In particular, let

$$X' = \begin{bmatrix} — & \vec{\psi}(x_1)^T & — \\ — & \vec{\psi}(x_2)^T & — \\ & \vdots & \\ — & \vec{\psi}(x_m)^T & — \end{bmatrix} \quad \text{and} \quad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

Then the least square estimator is:

$$\vec{w} = (X'^T X')^{-1} X'^T \vec{y} \quad (\star)$$

3. Since α_i and β_i are fixed, $\sigma(\alpha_i(x - \beta_i))$ is a generic function $\psi_i(x) = \sigma(\alpha_i(x - \beta_i))$. Therefore

$$h(x) = \sum_{i=1}^{} w_i \sigma(\alpha_i(x - \beta_i)) = \vec{w}^T \vec{\psi}(x)$$

[Solution: Exercise 1]

where $\vec{w} = [w_1, w_2, \cdots, w_L]^T$ and $\vec{\Psi}(x) = [\Psi_1(x), \Psi_2(x), \cdots, \Psi_L(x)]^T$

Since the optimal $\vec{w}$ for a linear model under the square loss is the least squares estimator, from 2. the optimal $\vec{w}$ (and therefore the optimal w_i's) can be found in closed form as (⋆).

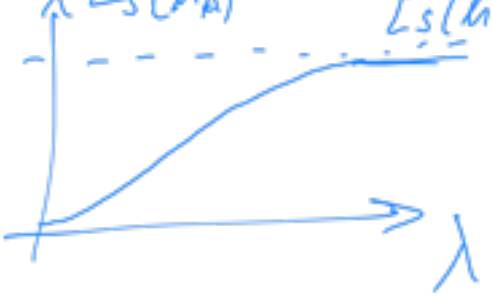
Exercise 2 [8 points]

Consider a generic machine learning problem and assume that a regularized loss function has been used by the selected algorithm A. In the loss function the relevance of the regularization term is controlled by a parameter λ. Let us denote with h_A the solution found by algorithm A and with $L_S(h_A)$ its empirical risk while the true risk (generalization error) of h_A is $L_D(h_A)$.

1. Which is the impact of the λ parameter on the empirical risk $L_S(h_A)$ of the solution found by A?
2. Which is the expected behavior of the true risk $L_D(h_A)$ of the found solution as a function of the λ parameter?
3. Describe how the behavior of the empirical risk and of the true risk in the answers to the previous questions are related to the bias-complexity trade-off.

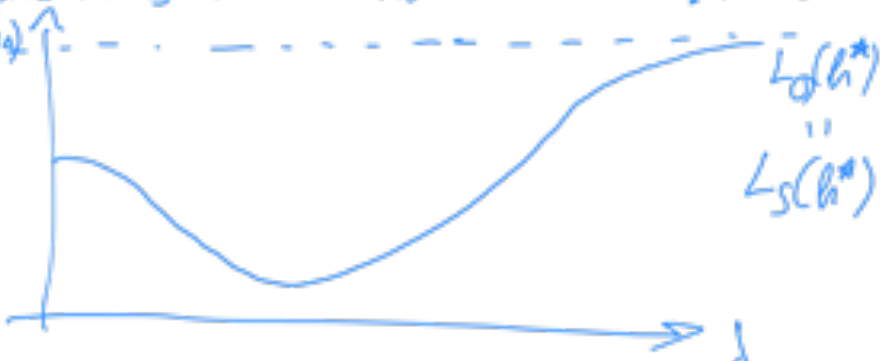
[Solution: Exercise 2]

1) λ controls the trade-off between the empirical risk $L_S(h_A)$ and the complexity of h_A. If $\lambda = 0$, h_A is the hypothesis of minimum empirical risk, but may suffer from overfitting. As $\lambda \to +\infty$, $L_S(h_A)$ increases till it reaches the empirical risk of the hypothesis h^* of minimum complexity (the complexity depends on the regularization function). The following plot shows the relation between λ and $L_S(h_A)$:



[Solution: Exercise 2]

2) For $\lambda = 0$, we have that h_A is complex and may be subject to overfitting, so $L_{\mathcal{D}}(h_A)$ is fairly high. Then as λ increases $L_{\mathcal{D}}(h_A)$ decreases until reaching its minimum value, and then increases again due to underfitting, reaching the generalization error of the hypothesis h^* of minimum complexity. For $\lambda \to +\infty$, $h^* = h_A$ is chosen independently of the data, therefore $L_S(h^*) = L_{\mathcal{D}}(h^*)$. The plot below describes the relation between λ and $L_{\mathcal{D}}(h_A)$:



3) λ is controlling the trade-off between the bias and the complexity. $L_{\mathcal{D}}(h_A) = \mathcal{E}_{app} + \mathcal{E}_{est}$, where $\mathcal{E}_{app} = \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h')$ and $\mathcal{E}_{est} = L_{\mathcal{D}}(h_A) - \mathcal{E}_{app}$. λ is essentially defining the complexity of $\mathcal{H}$, so for $\lambda = 0$ we have that $\mathcal{E}_{app}$ is small and $\mathcal{E}_{est}$ is large, due to having a complex $\mathcal{H}$, that results in a small $L_S(h_A)$ but still a large $L_{\mathcal{D}}(h_A)$. For $\lambda \to +\infty$, $\mathcal{H}$ has very low complexity (it consists of the hypothesis of smallest complexity) so $\mathcal{E}_{app}$ is large while $\mathcal{E}_{est}$ is small so $L_{\mathcal{D}}(h_A) = L_S(h_A)$ and $L_{\mathcal{D}}(h_A)$ is large. Starting from $\lambda = 0$, $\mathcal{E}_{est}$ decreases and $\mathcal{E}_{app}$ increases until the best choice of λ where $\mathcal{E}_{app}$ and $\mathcal{E}_{est}$ are somehow balanced (and $L_{\mathcal{D}}(h_A)$ is minimized), then by increasing λ more $\mathcal{E}_{est}$ decreases and $\mathcal{E}_{app}$ increases but $L_{\mathcal{D}}(h_A)$ increases.

Exercise 3 [8 points]

Consider a classification problem with 0-1 loss.

1. Provide the definition of VC dimension $VCdim(\mathcal{H})$ of a hypothesis set $\mathcal{H}$, and of empirical error and true risk (generalization error) for an arbitrary hypothesis $h \in \mathcal{H}$. What is the relation between the empirical error and the true risk in terms of the VC dimension of $\mathcal{H}$?

2. Consider the hypothesis set $\mathcal{H}$ defined as: $\mathcal{H} = \{h_{a,b} : a, b \in \mathbb{R}, a < b\}$ where $h_{a,b} : \mathbb{R} \mapsto \{0, 1\}$ is
$$h_{a,b}(x) = \begin{cases} 1 & \text{if } x \leq a \; OR \; x \geq b \\ 0 & \text{otherwise} \end{cases}$$
What's the value of $VCdim(\mathcal{H})$? Provide a proof of your claim.

3. Assume that you have many hypothesis sets, denoted by $\mathcal{H}_i, i = 1, 2, \ldots, n$. Describe one strategy to choose a good hypothesis set $\mathcal{H}_i$ and a good model $\hat{h}_i \in \mathcal{H}_i$.

[Solution: Exercise 3]

[Solution: Exercise 3]

1) Let $h \in \mathcal{H}$ be such that $h : \mathcal{X} \to \{0,1\}$. Let $C = \{c_1, \ldots, c_m\}$ with $C \subset \mathcal{X}$. The restriction $\mathcal{H}_C$ of $\mathcal{H}$ to C is $\mathcal{H}_C = \{[h(c_1), \ldots, h(c_m)] : h \in \mathcal{H}\}$. We say that $\mathcal{H}$ shatters C if $|\mathcal{H}_C| = 2^m$, that is, $\mathcal{H}_C$ contains all $2^m = 2^{|C|}$ functions from C to $\{0,1\}$.
The VC-dimension $VCdim(\mathcal{H})$ of $\mathcal{H}$ is the maximal size of a set $C \subset \mathcal{X}$ that can be shattered by $\mathcal{H}$; if $\mathcal{H}$ can shatter sets of arbitrary large size then $VCdim(\mathcal{H}) = +\infty$.
Let the training set S be $S = ((x_1, y_1), \ldots, (x_m, y_m)), x_i \in \mathcal{X}$, $y_i \in \{0,1\}$ $\forall i \in \{1, \ldots, m\}$. Let $\ell(h, (x,y))$ be the 0-1 loss: $\ell(h,(x,y)) = \begin{cases} 0 & \text{if } h(x) = y \\ 1 & \text{otherwise.} \end{cases}$ Let D be the unknown probab

[Solution: Exercise 3]

lity distribution from which (x_i, y_i), $i \in \{1, \dots, m\}$ is drawn independently from the other samples.
Given an hypothesis $h \in \mathcal{H}$, the empirical risk is:

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, (x_i, y_i))$$

and the true risk is: $L_{\mathcal{D}}(h) = \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}}[\ell(h, (x_i, y_i))]$.

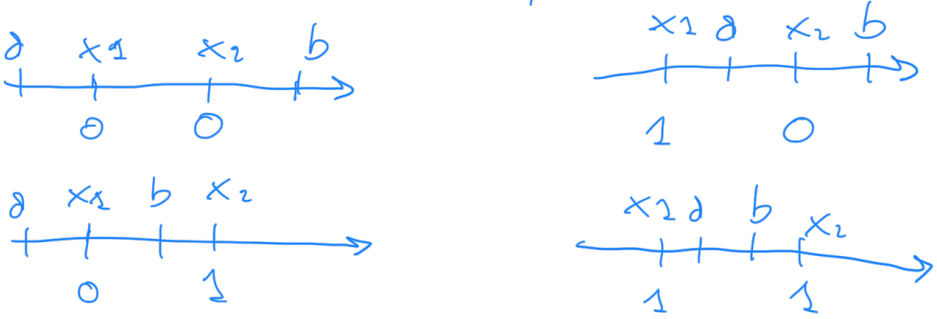
For any $h \in \mathcal{H}$ we have:

$$L_{\mathcal{D}}(h) \leq L_S(h) + C\sqrt{\frac{VCdim(\mathcal{H}) + \log(1/\delta)}{2m}}$$

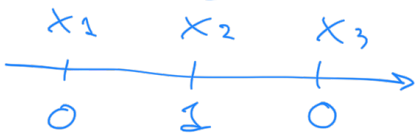
where C is a universal constant.

2) $VCdim(\mathcal{H}) = 2$. Let's prove it:

- $VCdim(\mathcal{H}) \geq 2$: let's take two arbitrary points $x_1, x_2 \in \mathbb{R}$ with $x_1 < x_2$. The the following shows that $\{x_1, x_2\}$ can be shattered by $\mathcal{H}$:



- $VCdim(\mathcal{H}) < 3$: let's take 3 arbitrary points $x_1, x_2, x_3 \in \mathbb{R}$ with $x_1 < x_2 < x_3$. The following assignment cannot be achieved by any hypothesis $h \in \mathcal{H}$:



[Solution: Exercise 3]

since to have that $h_{a,b}(x_2) = 1$ we have that $x_2 < a$, which implies $x_1 < x_2 < a$ and therefore $h_{a,b}(x_1) = 1$, or $x_2 > b$, which implies $x_3 > x_2 > b$, therefore $h_{a,b}(x_3) = 1$.

3) One strategy is to use cross-validation to choose $\mathcal{H}_i$ and then, once $\mathcal{H}_i$ is chosen, to use all the data to learn $\hat{h}_i \in \mathcal{H}_i$.

[Short explanation of how cross-validation works.]

[Alternative answer: given enough data, we can split it in training & validation, learn best model $h_i^* \in \mathcal{H}_i, \forall i$ using the training set, and then choose $i = \arg\min_{j \in \{1,\dots,n\}} L_V(h_j^*)$, where $L_V(h_j^*)$ is the error of h_j^* on the validation set. Once i (and therefore $\mathcal{H}_i$) is chosen, all the data is used to learn the model $\hat{h}_i \in \mathcal{H}_i$.]

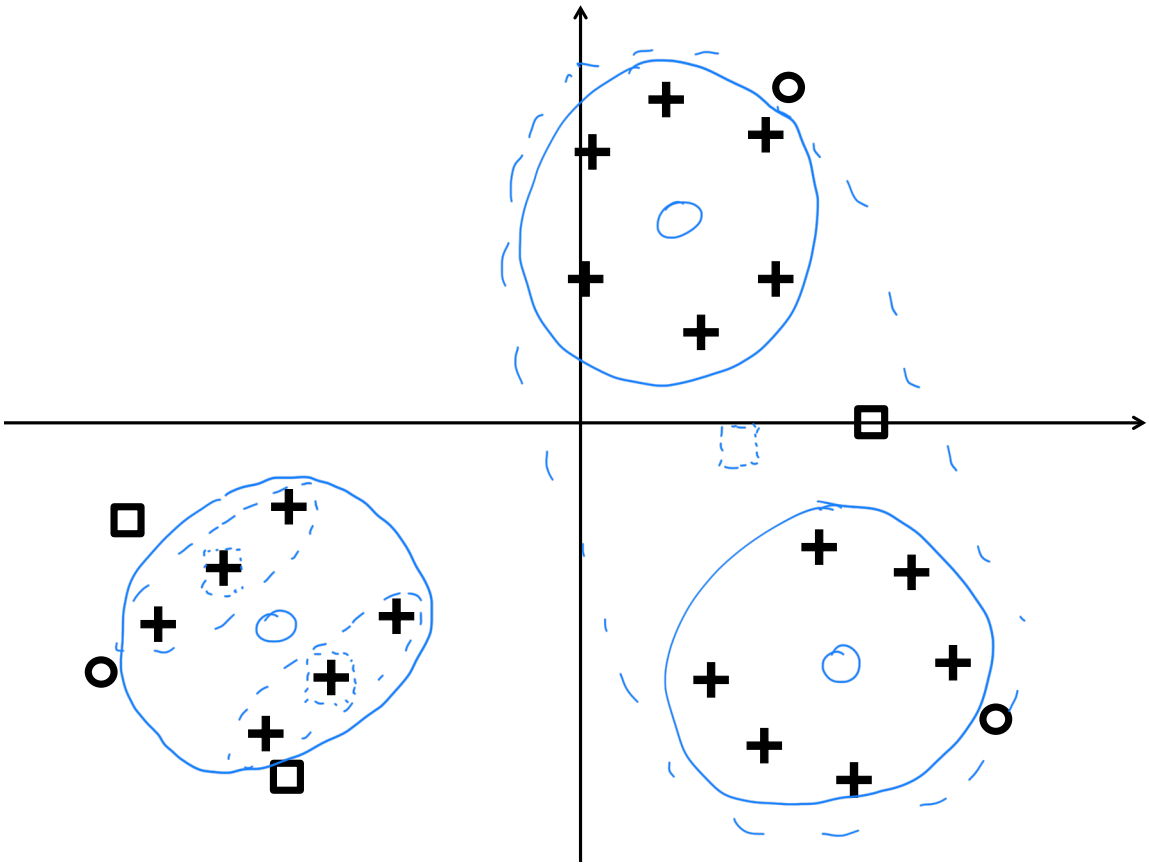
Exercise 4 [8 points]

Consider the problem of clustering.

1. Introduce the k-means clustering problem, rigorously defining its cost function.
2. Consider Lloyd's algorithm. What is the rule that is used to update the cluster centers after the points are assigned to clusters? Prove that such rule minimizes the k-means cost for the given assignment of points to clusters (i.e., once the assignment of points to clusters is fixed).
3. Consider the data in the figure below where each point $\mathbf{x} \in \mathbb{R}^2$ is represented by a cross. Draw (approximately) the output of Lloyd's algorithm for $k = 3$ when
   (a) the initial centers for the algorithm are the circles;
   (b) the initial centers for the algorithm are the squares.

   Which one of the two resulting clusterings has a lower cost?

(a) —
(b) - - -



[Solution: Exercise 4]

1) k-means is a cost minimization clustering problem. Let $\mathcal{X} \subseteq \mathcal{X}'$ be the set of points to be clustered, with $\mathcal{X} = \{\vec{x}_1, \dots \vec{x}_m\}$ while $\mathcal{X}'$ is the space of possible points, that we assume to be $\mathbb{R}^d$ (i.e., $\mathcal{X}' = \mathbb{R}^d$). Let $k \in \mathbb{N}^+$ be the number of clusters, that is, with $\mathcal{X}$ the input of the problem. Let $d(\cdot)$ be the distance function: $d(\vec{x}, \vec{x}') = \|\vec{x} - \vec{x}'\|$. The goal of the k-means clustering problem is to find:

- a partition $C = (C_1, C_2, \dots, C_k)$ of $\mathcal{X}$
- centroids $\vec{\mu}_1, \vec{\mu}_2, \dots, \vec{\mu}_k$ of $C_1, C_2, \dots, C_k$ respectively

that minimizes the k-means cost function:

$$\sum_{i=1}^{k} \sum_{\vec{x} \in C_i} d(\vec{x}, \vec{\mu}_i)^2.$$

2) The rule that Lloyd's algorithm uses to update clusters' centers after the points assigned to clusters is:

$$\vec{\mu}_i \leftarrow \frac{1}{|C_i|} \sum_{\vec{x} \in C_i} \vec{x}$$

[Solution: Exercise 4]

Proof: Let consider the cost function as a function of $\vec{\mu}_1, \vec{\mu}_2, \cdots, \vec{\mu}_k$: $f(\vec{\mu}_1, \vec{\mu}_2, \cdots, \vec{\mu}_k) = \sum_{i=1}^{k} \sum_{\vec{x} \in C_i} d(\vec{x}, \vec{\mu}_i)^2$. At the optimum, the gradient is equal to $\vec{0}$. Let's compute a part of the gradient, in particular $\frac{\partial f}{\partial \vec{\mu}_j}$:

$$\frac{\partial}{\partial \vec{\mu}_j} \left(\sum_{i=1}^{k} \sum_{\vec{x} \in C_i} d(\vec{x}, \vec{\mu}_i)^2 \right) = \sum_{i=1}^{k} \left(\frac{\partial}{\partial \vec{\mu}_j} \left(\sum_{\vec{x} \in C_i} d(\vec{x}, \vec{\mu}_i)^2 \right) \right)$$

$$= \frac{\partial}{\partial \vec{\mu}_j} \left(\sum_{\vec{x} \in C_j} d(\vec{x}, \vec{\mu}_j)^2 \right) = \sum_{\vec{x} \in C_j} \frac{\partial}{\partial \vec{\mu}_j} \left(d(\vec{x}, \vec{\mu}_j)^2 \right)$$

$$= \sum_{\vec{x} \in C_j} \frac{\partial}{\partial \vec{\mu}_j} (\vec{x} - \vec{\mu}_j)^T (\vec{x} - \vec{\mu}_j) = \sum_{\vec{x} \in C_j} (-2\vec{x} + 2\vec{\mu}_j)$$

$$= \left(-2 \sum_{\vec{x} \in C_j} \vec{x} \right) + \left(2 \sum_{\vec{x} \in C_j} \vec{\mu}_j \right) = -2 \sum_{\vec{x} \in C_j} \vec{x} + 2|C_j| \vec{\mu}_j$$

At the optimum : $2|C_j| \vec{\mu}_j - 2 \sum_{\vec{x} \in C_j} \vec{x} = \vec{0}$

$$\iff |C_j| \vec{\mu}_j = \sum_{\vec{x} \in C_j} \vec{x} \iff \vec{\mu}_j = \frac{1}{|C_j|} \sum_{\vec{x} \in C_j} \vec{x}$$

□

[Solution: Exercise 4]

3) Since the cost depends on the square of the distance of points to the centroids, clustering (a) has lower cost.