For Exam

NOTE: can be present some errors (probably id3 algorithm). We used github's solutions and slides.

SVM: maximal margin

SVM: kernel and epsilon tube, slack variable

Discuss why maxium margin solution is preferred for the classification problem

Definition of GMM, gaussian mixture model and describe parameters of the model.

Draw a 2D dataset generated by GMM with k=3 showing parameters of the model in the pictures

Describe KNN algorithm and determine answer of KNN given a dataset

Briefly describe a linear classification method and discuss its performance in presence of outliers. Use a graphical example to illustrate the concept.

Explain differences between regression and classification

Provide a mathematical formulation of linear regression

Provide and example of a linear regression model that overfits a dataset of your choice and discuss how it can be mitigated

define mathematically the problem solver by logistic regression

Explain kernel trick with necessary condition to apply it. Provide and application, in detail draw a suitable dataset for binary classifiction 2D, discuss with kernel would you use and show graphically a possible solution of such kernel-based model

formal definition of overfitting and illustrate with DT with a solution

Describe the Naive bayes classifer and highlight the approximation made with respect to the BOC

describe the goal of a linear regression model

Describe difference between generative and discriminative model. Draw a dataset and show the solution in both cases

Describe and application problem that can be modelled and solved with unsupervised learning method

Describe what is boosting

Comment the following statement: in a classification problem, the class returned by the ML hypothesis on a new instance x is always the most probable class.

PCA

Describe how PCA are identified based on principle of variance maximization

PCA - Consider the binary (black & white) images below defined on a 12 ⇥ 12 grid:

Formal definition of RL problem. Input and output of a RL problem

Describe main steps of a RL algorithm. Provide pseudo code of a generic RL algorithm

MDP: Markov decision process

Describe the concept of fully observability in models representing dynamic

Describe full observability property of MDP and it relation with non-determinist outcomes of actions

describe difference between a MDP, POMDP and draw and explain the graphical models

TODO: MDP vs HMM:

Descrive k-armed bandit problem (also know as One-state MDP

- Describe RL procedure to compute the optimal policy in k-armed bandit problem
- How solve a given problem with reinforcement learning and determine the trainig rule
- Discuss the strategy for balancing exploration and exploitation in this case.

Exercise

- function interpolation
- Descrive input and output of a CNN given a structure + number of parameters for each layer
- Simulate the execution od K-means in this 2-D dataset with k=2 and initial centroids circles. Use one diagram for each step of the algorithm. Describe explicitly how each step is obatained and what is the termination coniditon of the algorithm. Drawing only the steps is not sufficient
- Consider a data set D for scoring different schools with the following real-valued attribues: staff salaries per pupil x_1, teacher's test score x_2, parents' education x_3, school grade y
- Find rule for a DT
- Provide designb and implementation choices for solving the following problem through naive bayes classifier: categories = { ML, KR, PL }, D = { title, authors, abstract, name of journal)
- Design an ANN for learning a function $t = f(x, \theta)$
- Activation function of cnn parameters
- Consider a two-layer ANN which receives in input a vector x. Calculate dimension of weights matricies W_1, W_2

Questions

▼ ID3 algorithm execution

EXERCISE 1

The following data have been collected and we want to learn the general concept *Acceptable*, by using Decision Tree Learning.

House	Furniture	Nr rooms	New kitchen	Acceptable
1	No	3	Yes	Yes
2	Yes	3	No	No
3	No	4	No	Yes
4	No	3	No	No
5	Yes	4	No	Yes

1. Formalize the learning problem: decribe exactly the target function to learn and the dataset.
2. Describe qualitatively how attributes are chosen when building a Decision Tree.
3. Simulate the execution of ID3 algorithm on the data set above and generate the corresponding output tree.

Entropy formula:

$$entropy(S) = -\sum_i p_i \log(p_i)$$

With $p_i = \frac{samples\_label_i}{tot\_samples}$

1. Compute the entropy on dataset.
2. compute $Gain(S, A) = entropy(S) - p_+ entropy(S, A|+) - p_- entropy(S, A|-)$
3. do it for all attribute and take the one with highest gain
4. repeat until termination and everytime, if is it possible, reduce the dataset (check example)

3). $ent(S) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$

$IG(S,F) = ent(S) - \frac{2}{5}ent(F_R) - \frac{3}{5}ent(F_Y) = 0.0194$

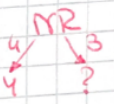
$ent(F_R) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.916$ $ent(F_Y) = 1$

$IG(S,NR) = ent(S) - \frac{2}{5}ent(NR_4) - \frac{3}{5}ent(NR_3) = 0.4214$ → HIGHEST IG!

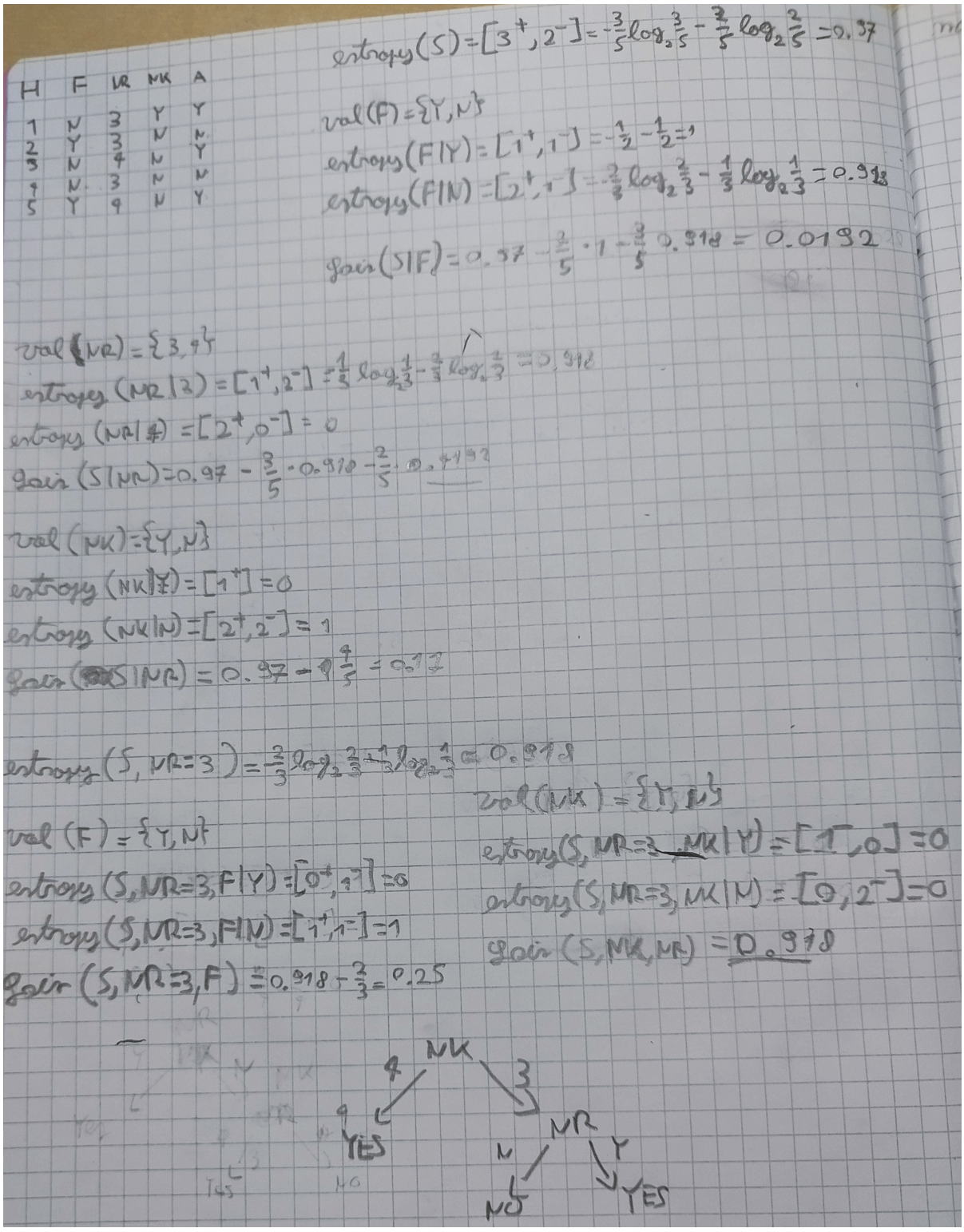
$ent(NR_3) = 0.916$ $ent(NR_4) = 0$

$IG(S,NK) = ent(S) - \frac{1}{5}ent(NK_4) - \frac{4}{5}ent(NK_R) = 0.171$

$ent(NK_4) = 0$ $ent(NK_R) = 1$



We continue using this technique.



▼ consistency of a DT

A DT is consistent when , given a sample (x,y) F(x)=y

▼ Features of boosting, error function and AdaBoost

EXERCISE B1

1. Provide the main features about boosting.
2. Write the error function whose minimization leads to a formulation equivalent to the AdaBoost algorithm.

In boosting we can consider the following key points:

- base classifier trained sequentially
- each classifier is trained on weighted data
- weighs depent on performance of previous classifier
- point misclassified by previous classifier are given greater weights
- predictions are based on weighted majority votes

Take for example AdaBoost, given $D = \{(x_n, t_n)\},\ t_n \in \{-1, 1\}$, it minimaze the loss function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)$$

With I be 1 if the argument is True, else 0.

Ad the end the weight are updated

$$w_n^{(m+1)} = w_n^{(m)} \exp[\alpha_m I(y_m(x_n) \neq t_n]$$

The final classifier will be

$$Y_M(x) = sign(\sum_{m=1}^{M} \alpha_m y_m(x))$$

▼ Definition and how avoid overfitting

EXERCISE 1

Consider the notion of *overfitting*:

1. Provide a formal and general definition, without referring to any particular model.
2. Show two examples of overfitting in two distinct models.
3. For one of the models above, explain how the problem can be mitigated.

- Definition:

Consider error of hypothesis h over
- training data: $error_S(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

$$error_S(h) < error_S(h')$$

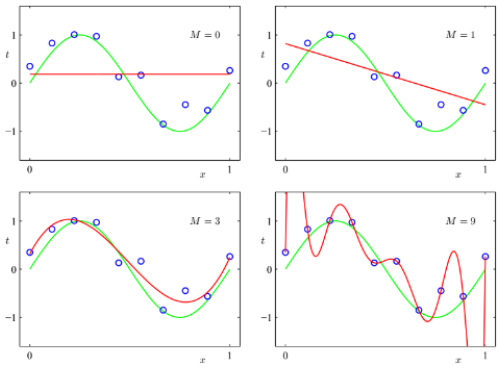
and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

- Examples

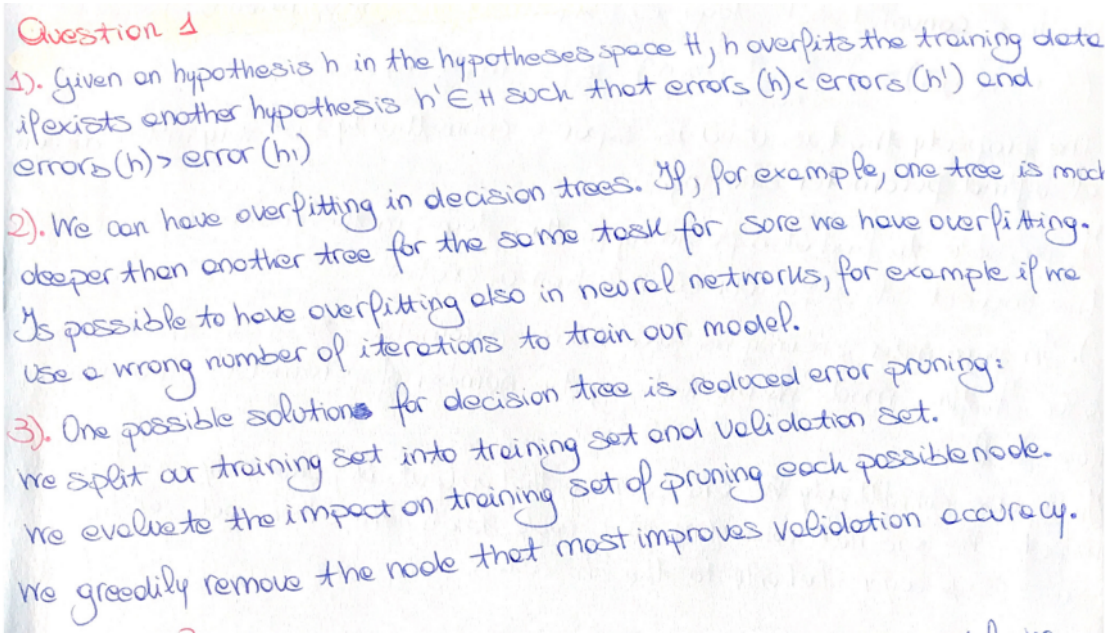
Example: Polynomial curve fitting

$$y = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$



Warning: overfitting!!!

- In decision trees
  - solutions: tree pruning
- in neural networks
  - solutions: regularization, early stopping, dropout, data-augmentation, parameter sharing.


Question 1
1). Given an hypothesis h in the hypotheses space H, h overfits the training data if exists another hypothesis $h' \in H$ such that errors(h) < errors(h') and errors(h) > error(h1)
2). We can have overfitting in decision trees. If, for example, one tree is much deeper than another tree for the same task for sure we have overfitting. Is possible to have overfitting also in neural networks, for example if we use a wrong number of iterations to train our model.
3). One possible solutions for decision tree is reduced error pruning: we split our training set into training set and validation set. We evaluate the impact on training set of pruning each possible node. We greedily remove the node that most improves validation accuracy.

▼ MAP and ML: maxium posterior hypotheses and maxium likelihood

h_{map} is the most probable hypotheses given the dataset D

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

h_{ml} is the hypotheses which maxiumize the probability on the dataset

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

Generally we want the most probable hypothesis h given D

Maximum a posteriori hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &\equiv \underset{h \in H}{\operatorname{argmax}} P(h|D) = \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)} \\ &= \underset{h \in H}{\operatorname{argmax}} P(D|h)P(h) \end{aligned}$$

If assume $P(h_i) = P(h_j)$, we can further simplify, and choose the *Maximum likelihood* (ML) hypothesis

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} P(D|h)$$

▼ Applicability of BOC, bayes optimal classifier

BOC (Bayes Optimal Classifier) is a probabilistic model which calculate the most probable classification for a new instance.

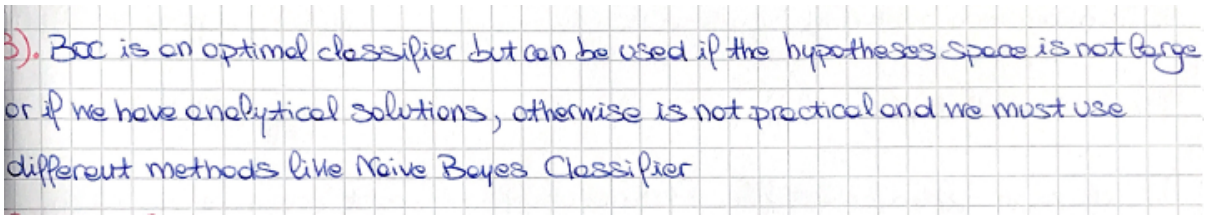
Consider a target function $f : X \to V,\ V = v_1, ..., v_k$, dataset D and a new isntance $x \notin D$

$$P(v_j|x, D) = \sum_{h_i \in H} P(v_j|x, h_i)P(h_i|D)$$

BOC is defined as

$$v_{OB} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|x, h_i)P(h_i|D)$$

Considering the BOC work maximizing the argument v_j and use the whole hypotheses space, it will be unfeasible when these spaces are big



Example:

$$\begin{aligned} P(h_1|D) = 0.4, \quad P(\ominus|x, h_1) = 0, \quad P(\oplus|x, h_1) = 1 \\ P(h_2|D) = 0.3, \quad P(\ominus|x, h_2) = 1, \quad P(\oplus|x, h_2) = 0 \\ P(h_3|D) = 0.3, \quad P(\ominus|x, h_3) = 1, \quad P(\oplus|x, h_3) = 0 \end{aligned}$$

therefore

$$\begin{aligned} \sum_{h_i \in H} P(\oplus|x, h_i)P(h_i|D) &= 0.4 \\ \sum_{h_i \in H} P(\ominus|x, h_i)P(h_i|D) &= 0.6 \end{aligned}$$

and

$$v_{OB} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|x, h_i)P(h_i|D) = \ominus$$

▼ Bayes optimal classifier(slide):

Consider target function $f : X \mapsto V$, $V = \{v_1, ..., v_k\}$, data set D and a new instance $x \notin D$:

$$P(v_j|x, D) = \sum_{h_i \in H} P(v_j|x, h_i)P(h_i|D)$$

total probability over H
$P(v_j|x, h_i)$: probability that $h_i(x) = v_j$ is independent from D given h_i
$\Rightarrow P(v_j|x, h_i) = P(v_j|x, h_i, D)$
h_i does not depend on $x \notin D \Rightarrow P(h_i|x, D) = P(h_i|D)$

Bayes Optimal Classifier
Class of a new instance x:

$$v_{OB} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|x, h_i)P(h_i|D)$$

▼ describe perceptron model

The perceptron model is one of the possible approaches to linear discriminant for linear classification problems. The perceptron model performs a linear combination of samples in the dataset X and a weight vector and then apply an activation function.

So its formula is : $y(x) = \alpha(w^T x + b)$.

The weights are learned using an iterative algorithm.

The goal is to minimize the error function $E(X) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T x_n)^2$

Then you update the weights for each iteration using this formula:

$w_i \leftarrow w_i - \eta\frac{\partial E}{\partial w_i}$

with η the learning rate.

The iteration stops when is reached a predefined number of iteration or then is reached a threshold of minimization of E.

A drawback of this method is that the perceptron doesn't generate a good separator, ad the hyperplane is generated near one of the classes. For this reason SVM could be a better approach.

▼ ANN/FNN: general formula, activation function and their formulas

EXERCISE B2

Consider the problem of finding a function which describes how the salary of a person (in hundreds of euros) depens on his/her age (in years), the months in higher education and average grades in higher education. A dataset in the form $\mathcal{D} = \{(\mathbf{x}_1^T, t_1), \ldots, (\mathbf{x}_N^T, t_N)\}$ is provided, with $\mathbf{x} \in \mathbb{R}^3$ denoting the input values and t the target values (salary).

Assuming that one tries to identify this function with a deep feed-forward network:

1. Explain how the problem is formalized by writing the parametric form of the function to be learned highlighting the parameters $\boldsymbol{\theta}$.
2. Explain what are suitable choices for the activation functions of the hidden and output units of the network.
3. Explain what is a suitable choice for the loss function used for training the network and write the corresponding mathematical expression.

An FNN (feed forward network) is defined as

$$f(x) = f^{(n)}(f^{(n-1)}(f^{(n-2)}(...;\theta_{n-2});\theta_{n-1});\theta_n)$$

With $f^{(i)}$ which define the i-th layer and θ_i its parameter. If we consider Deep neural network the n-value is big, so it has lots of hidden layers.

In general in hidden layer common used activation function are:

- $ReLU(x) = \begin{cases} 0 \ \ if x \leq 0 \\ x \ \ else \end{cases}$
- $Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

While for output layer is used

- $Softmax(x) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$

used to compute the probability distribution on the output

▼ Given an ANN, describe backpropagation, forward and backward pass, stochastic gradient descent. Provide main steps of backpropagation algorithms

EXERCISE B2

1. Describe the role of the following notions related to parameter estimation of an artificial neural network:
   - backpropagation
   - forward and backward pass
   - Stochastic Gradient Descent
2. Provide the main steps of the backpropagation algorithm.

The backpropagation is an algorithm to compute the gradient and calculate parameters to minimize the error function following the gradient.

The process is split in two parts:

1. Forward:
   The input data x is fed forward through the neural network to obtain the predicted output y. Then is done a comparation through the predicted output and the actual target values, computing the loss or error.
2. Backward:
   The goal of backpropagation is to update the weights of the network in a way that minimizes the loss. It involves computing the gradient of the loss

with respect to the weights of the network. The computed gradients are used to update the weights using an optimization algorithm, often the SGD. The weights are adjusted in the opposite direction of the gradient to minimize the loss.

These steps iterate til convergence.

▼ K-means: definition and execution

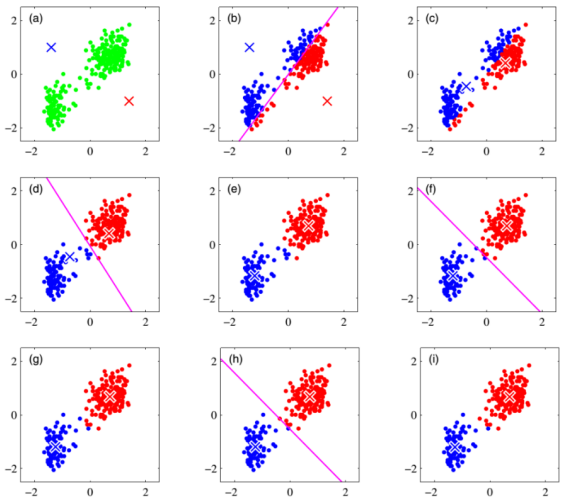
EXERCISE 5

1. Describe the K-means algorithm in a formal way (i.e., with precise mathematical formulas and equations), including: input and output of the algorithm, its main steps, and the termination condition.
2. Describe common drawbacks of this algorithm (i.e., situations in which solutions are not optimal)
3. Draw a 2-D data set and the corresponding (qualitative) K-means solution highlighting some of the drawbacks illustrated in the previous point.

1. It is an algorithm for unsupervised classification and consist to estimate K-cluster generated respectively from K gaussian distribution where a distribution is defined as $\mathcal{N}(x; \mu_k, \Sigma_k)$ with μ_k, Σ_k mean and variance of k-distribution. Given in input $D = \{x_n\}$ the algorithm will output $\mu_1, ..., \mu_n$ means.

algorithm's steps:

1. Initialize k-cluster selecting k-samples as centroids
2. Assign N-k remaining example to the nearest cluster
3. Take each sample and find the nearest centroid. If it has the already the label of the nearest centroid, do nothing, else re-assing the label as the same of the nearest centroid
4. repeat step 3 until convergence



2. Drawbacks:
    a. number of cluster are choosen before hand
    b. sensitive to initial condition when few data are avaible
    c. not robust to outliers. Very far data may from centroid may pull centroid away from the real one
    d. The result is a circular cluster shape because it is based on distance.
    e. Some solution can be: use k-mean only many data, use median instead means, define better distance function

Question 5

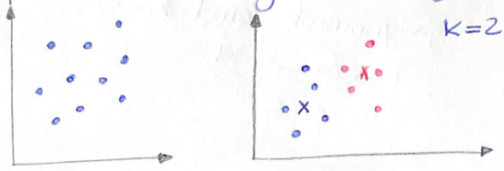
1). K-means is an unsupervised learning technique. The input is $D = \{x_n\}$ the dataset and the output is $\mu_1, \mu_2 \dots \mu_k$.

- We select the value of k = #clusters.
- We have to split our samples: we can do this randomly or sistematically We take the first k samples and we assign each of them to a 1 elem nt cluster. For the remaining $N-k$ samples we assign each sample to the cluster with the nearest centroid and we recompute the centroid.
- For all the samples we compute the distance from all the clusters and if the sample is not correctly classified we move it to the correct cluster and we recompute the centroid of the modified clusters.
- We repeat the previous step until convergence is achieved.

2). The algorithm will fail if:

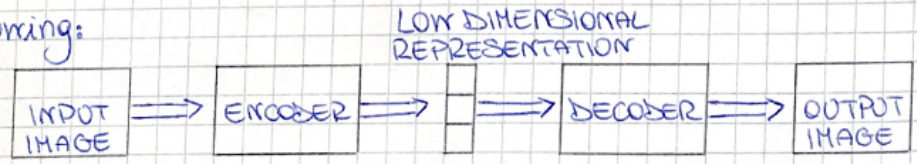
- There are ∞ partitions of training samples into k clusters
- For each ~~step~~ switch in step 2, the sum of distances from each training sample to that training sample's group centroid increases.

3).



▼ describe architecture of autoencoder

1). The goal of an autoencoder is dimensionality reduction. An autoencoder is composed by 2 neural networks, an encoder and a decoder. The structure is the following:



In autoencoder usually we have hidden layers with reduced size that are also called bottlenecks. The training is based on reconstruction loss.

Given a dataset $\{x_n\}$, autoencoders are trained with the sample x_n in input and in output.

An autoencode is a type of ANN used to learn efficient codings of unlabeld data.

It learn two function:

- an encoding function with trasform the input from input space to a latent
- a decoder function which recreate the input data from the encoded representation

This type of architecture is used for dimensionality reduction

Usually the loss function compare the difference between the input data and the reconstructed data after have compressed it.

▼ definition of gram matrix

EXERCISE 1

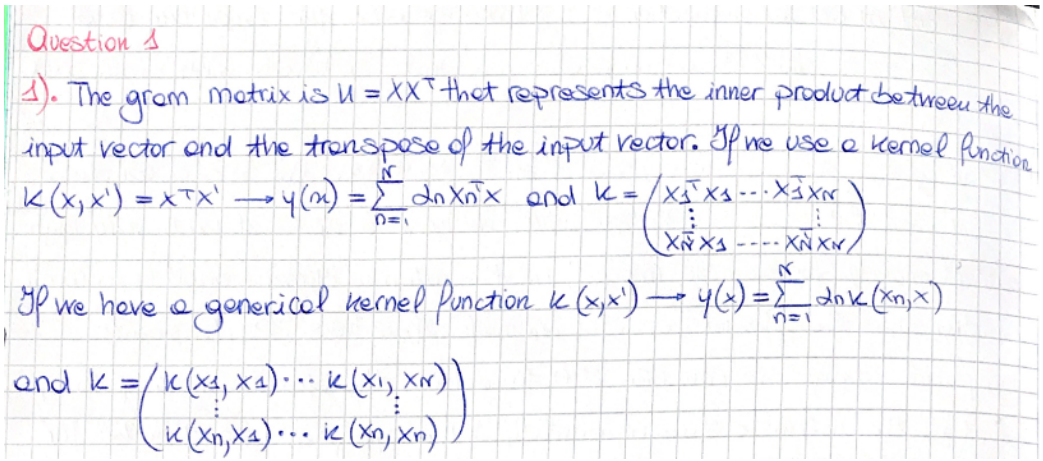
Given input values $\mathbf{x}_i$ and the corresponding target values t_i with $i = 1, \ldots, N$, the solution of regularized linear regression can be written as:

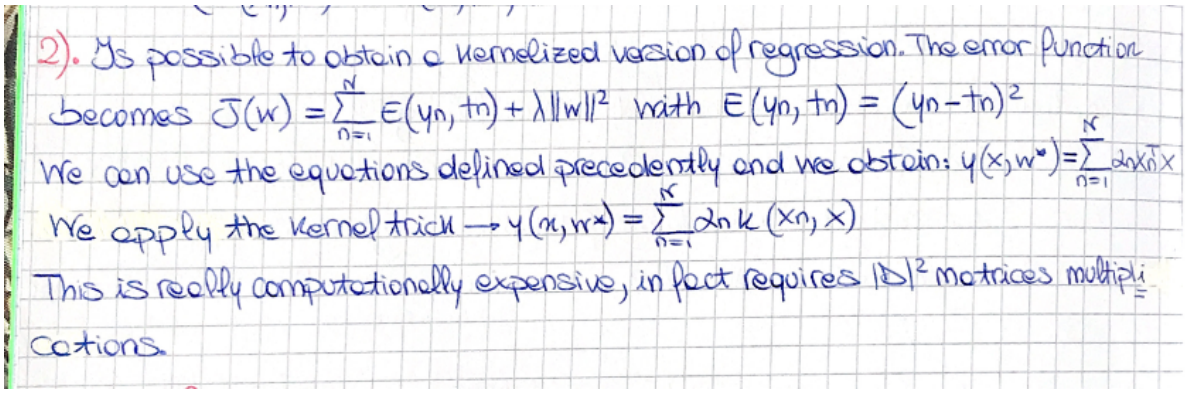
$$y(\mathbf{x}) = \sum_{i}^{N} \alpha_i \mathbf{x}_i^T \mathbf{x},$$

with $\boldsymbol{\alpha} = (XX^T + \lambda I)^{-1}\mathbf{t}$, $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ and λ the regularization weight.

Considering a kernel function $k(\mathbf{x}, \mathbf{x}')$:

1. Provide a definition of the Gram matrix.
2. Explain how a kernelized version for regression can be obtained based on the equations provided above.





Gram matrix is a matrix $N \times N$ where the entries $i, j = 1, ..., N$ contain the scalar product between $x_i x_j$.

This matrix is a simmetric and $x_i x_j = x_j x_i$

$$G = \begin{bmatrix} x_1^T x_1 & ... & x_1^T x_N \\ ... & ... & ... \\ x_N^T x_1 & ... & x_N^T x_N \end{bmatrix}$$

▼ Describe an ensemble method for achieving higher classification accuracy by combining such classifiers.

EXERCISE 6

Assume you have 4 image classifiers with medium-good classification accuracy.

1. Describe an ensemble method for achieving higher classification accuracy by combining such classifiers.
2. Are there any specific properties that each classifier has to have to achieve higher accuracy? If the answer is positive, explain which these properties are.

1. We can use multiple learnes to have as output a combination of each single model (for example majority voting, mean, ...). One can be “boosting” which train this model in sequence where the previous model influence the next one. Each sample has a weight. If a samples is misclassified by a model, the weight will be increase for the next classier and in this way this model will give more importance to samples which the previous model was not able to classify correcly resulting in a more accuracy. At the end we combine the performance of each single model.

$$Y_M(x) = sign(\sum_{m=1}^{M} \alpha_m y_m(x))$$

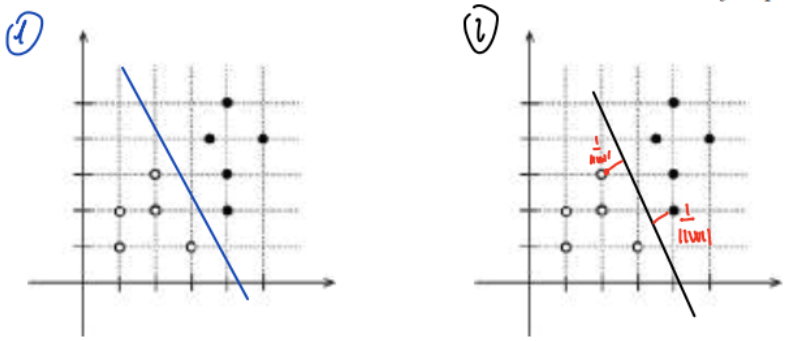
▼ Are there any specific properties that each classifier has to have to achieve higher accuracy? If the answer is positive, explain which these properties are.

- Change method(use a binary tree, a SVM, or a NN), you can do dataprocessing (like data augmentation), or yu can do an ensemble model.

▼ Consider a binary classification problem

EXERCISE A2

Consider the following data set for binary classification, where the two classes are represented with white and black circles. Draw in each of the diagrams a possible solution for a method based on Perceptron with very small learning rate and a method based on SVM. Describe the difference between the two solutions and explain how these are obtained with the two methods. Discuss which solution would you prefer and why.



The main difference between SVM and perceptron is that SVM always aims at maximum margin with better accuracy, while perceptron is based on a sequential algorithm that depends on the learning rate η.

In the perceptron you learn weights that minimize the squared error (loss function)

$$E(x) = \frac{1}{2}\sum_{n=1}^{N}(t_n - o_n)^2 = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T x_n)^2$$

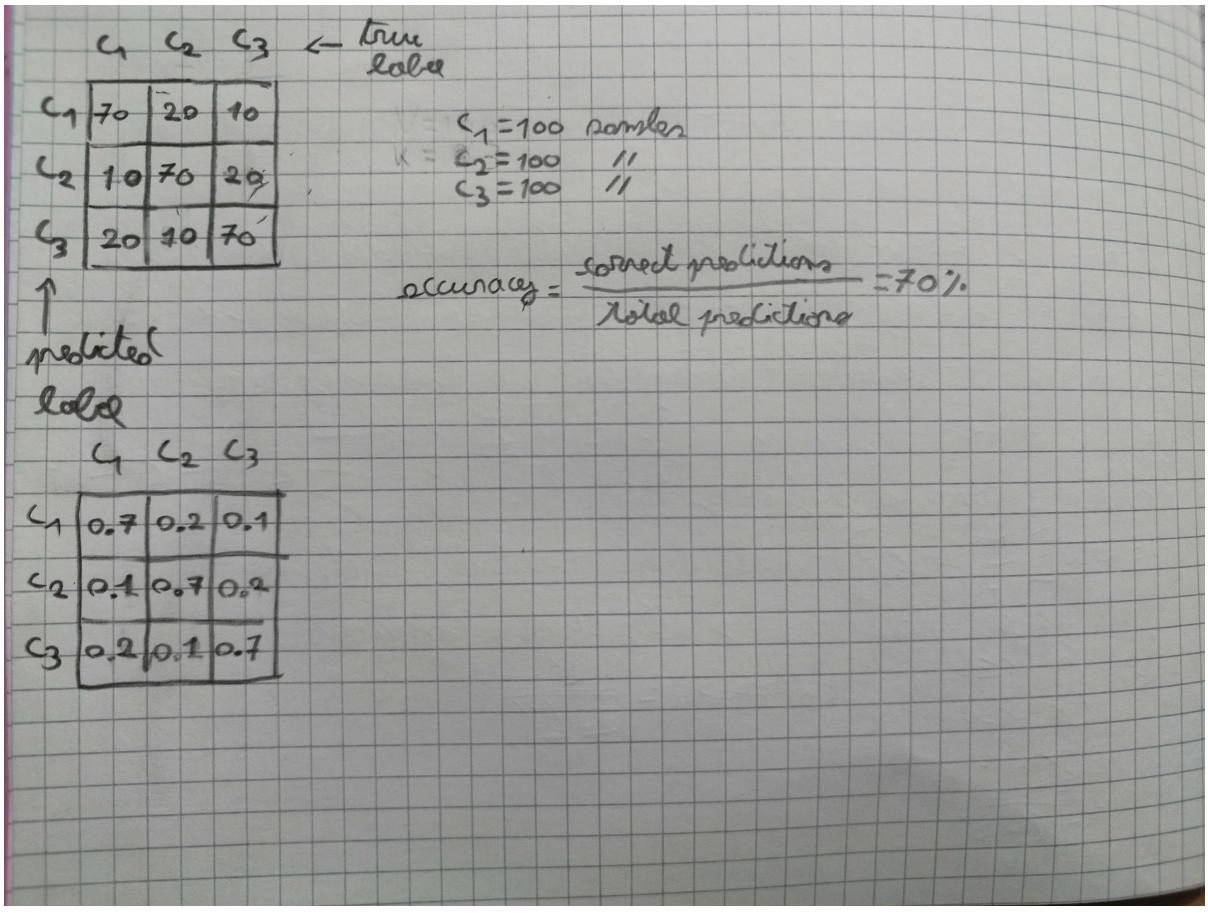
then you have to compute $\frac{\partial E}{\partial w_i}$ and in an iterative way you have to update the weights.

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} = \eta \sum_{n=1}^{N} (t_n - \mathbf{w}^T \mathbf{x}_n)\, x_{i,n}$$

▼ Definition of confusion matrix + example + how compute accuracy with it

The confusion matrix is a metrics which report in a matrix the number (or %) of samples classified as c_i with true label c_j (where j can be equal or not equal to i)



▼ Given a model of a problem, define an error function, discuss if a linear model for regression can

be good and describe an tierative approach to solve the problem

Given

$$y(x;\theta) = \theta^T x$$

1. $E = \frac{1}{2}\sum_{i=0}^{N}(y(x_i;\theta) - t_i)^2$
2. Yes, this model can be used to determine the score value of each school
3. We can find the best parameter in an iterarive way using the following rule

$$\theta_{i+1} \leftarrow \theta_i - \eta\frac{\partial E}{\partial \theta_i}$$

▼ Formal definition of supervised and unsupervised ML problem giving a formal definition (not only explanatory text)

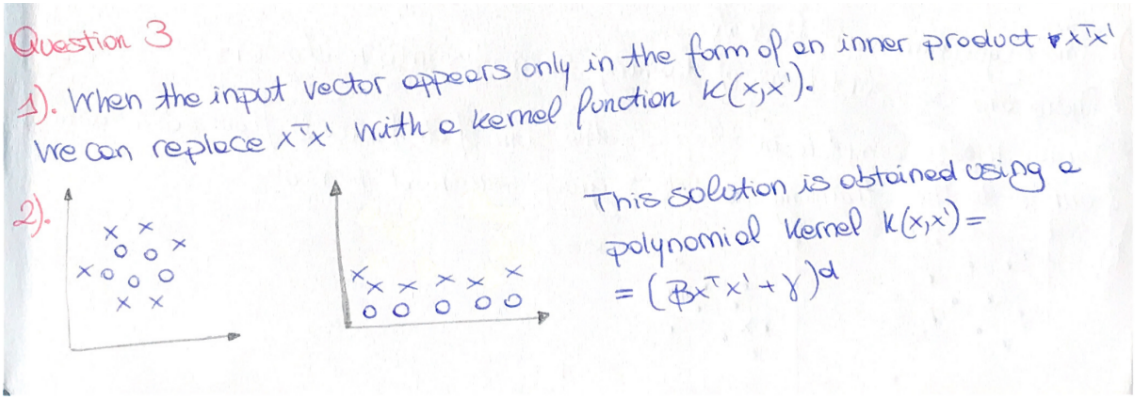
Machine learning problems can be categorized in supervised and unsupervised. Explain the difference between them providing a precise formal definition (not only explanatory text) in terms of input and output of the two categories of problems.

In supervised learning the dataset is composed in this way: $D = \{(x_i, y_i)_{i=1}^N\}$. So each element of the dataset is composed by the input and its associated label.

In the unsupervised learning the dataset is composed in a different way: $D = \{(x_n)_{i=1}^N\}$. So in this case each element is composed by only the single output without labels.

▼ kernel function

1. Give a short explanation of the *kernel trick/kernel substitution*. What is the necessary condition for applying the kernel trick?
2. Provide an example of its application. In detail:
   - draw a suitable dataset for binary classification in 2D;
   - discuss which kernel you would use for this dataset;
   - show graphically a possible solution of such a kernel-based model.



Kernel trick or *kernel substitution*

If input vector $\mathbf{x}$ appears in an algorithm only in the form of an inner product $\mathbf{x}^T\mathbf{x}'$, replace the inner product with some kernel $k(\mathbf{x}, \mathbf{x}')$.

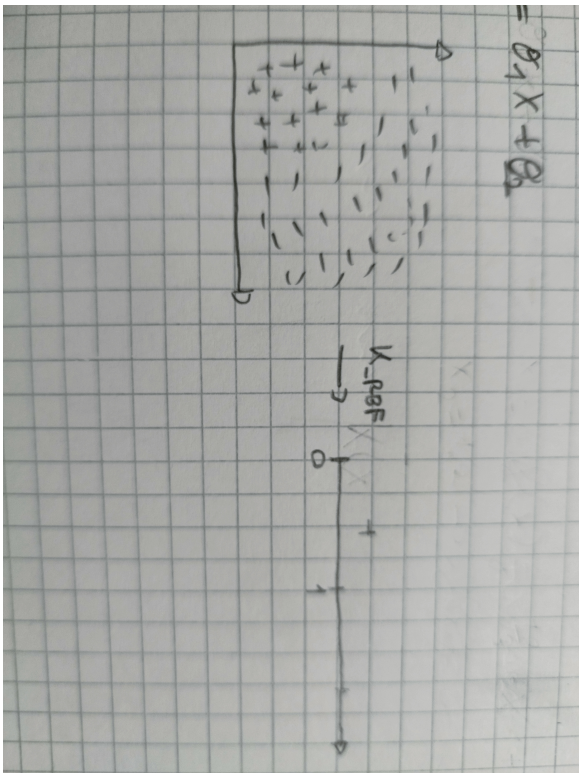
- Can be applied to any $\mathbf{x}$ (even infinite size)
- No need to know $\phi(\mathbf{x})$
- Directly extend many well-known algorithms

Definition
Kernel function: a real-valued function $k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$, for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, where $\mathcal{X}$ is some abstract space.

Typically k is:

- symmetric: $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$
- non-negative: $k(\mathbf{x}, \mathbf{x}') \geq 0$.

Note: Not strictly required!



▼ explain how kernelized version for regression can be obtained based of some equations

EXERCISE 1

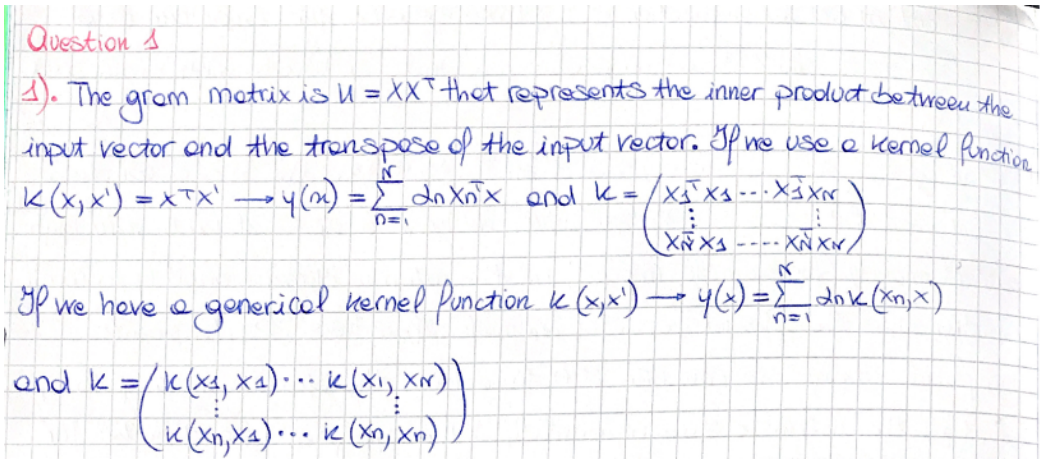
Given input values $\mathbf{x}_i$ and the corresponding target values t_i with $i = 1, \ldots, N$, the solution of regularized linear regression can be written as:

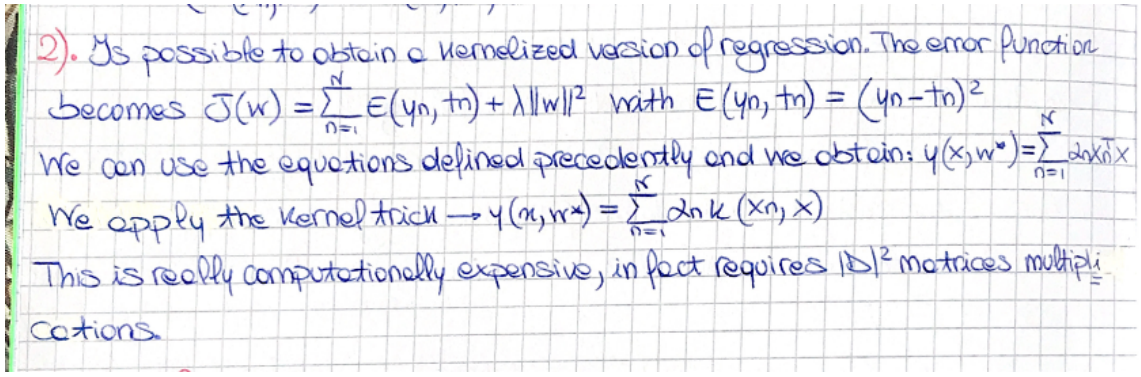
$$y(\mathbf{x}) = \sum_{i}^{N} \alpha_i \mathbf{x}_i^T \mathbf{x},$$

with $\boldsymbol{\alpha} = (XX^T + \lambda I)^{-1}\mathbf{t}$, $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ and λ the regularization weight.

Considering a kernel function $k(\mathbf{x}, \mathbf{x}')$:

1. Provide a definition of the Gram matrix.
2. Explain how a kernelized version for regression can be obtained based on the equations provided above.





A kernelized version for regression can be obtain applied a kernel function to every pair $x_i^T x_i$, so my equation became

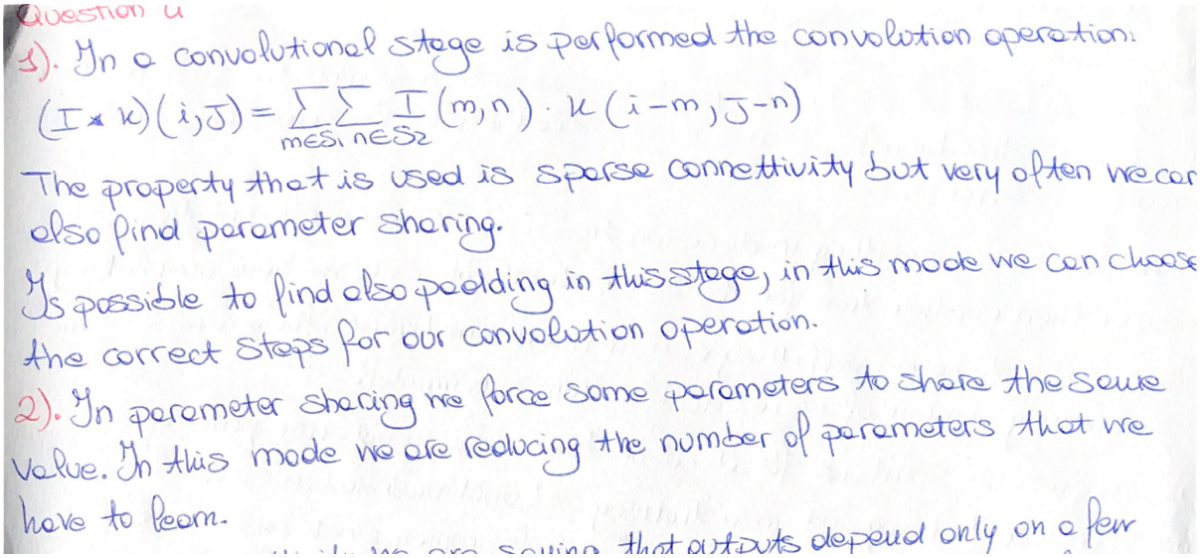
$$y(x) = \sum_{i}^{N} \alpha k(x_i^T x_i)$$

$$\alpha = (G + \lambda I)^{-1} t$$

$$G = \begin{bmatrix} k(x_1^T x_1) & ... & k(x_1^T x_N) \\ ... & ... & ... \\ k(x_N^T x_1) & ... & k(x_N^T x_N) \end{bmatrix}$$

▼ Describe main parts of a single convolutional layer and their function

Question 2. (5 points) Briefly describe the main parts that form a single convolutional layer and their function.



Convolutional stage in discrete case

$$I * K(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

Given a kernel $m \times n$ the number of trainable paramters are the size of the kernel

A convolutional layer is typically used to extract feature from images and it can reduce the imput size (so computaional cost) depend on used parameters like kernel size, functions, stride, padding etc.

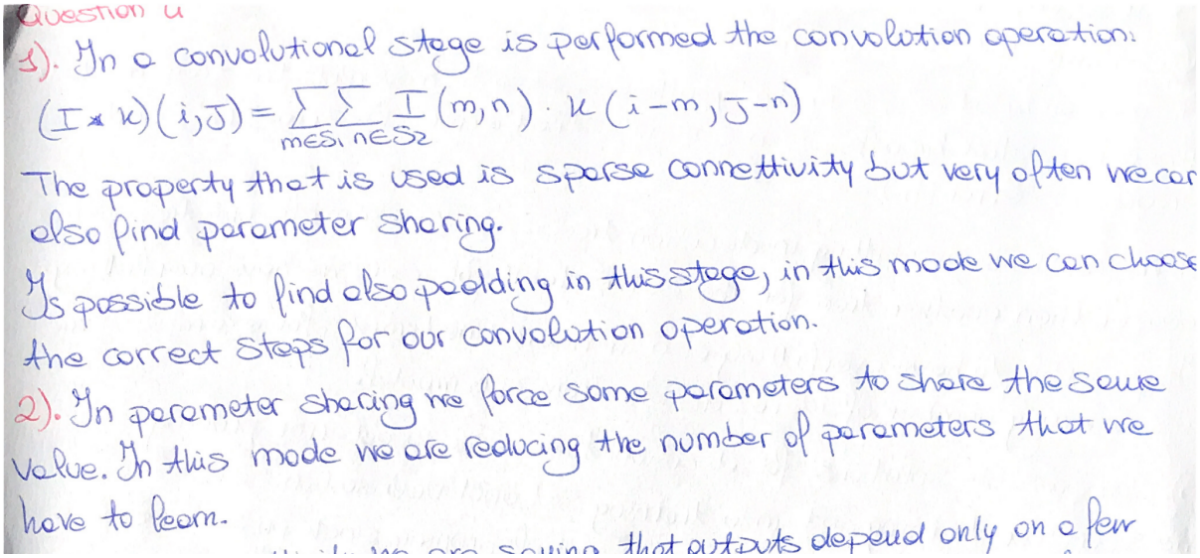
▼ Discuss the properties of sparse connectivity and parameter sharing for CNN

- parameter sharing: it is technique to reduce to overfitting and it consist in sharing the same weights between one or more weights and one I update one, I update all the other one “linked”
- sparse connectivity: it is a property that each neuron is connected to only a limited number of other neurons

▼ Describe convolutional stage in CNN

EXERCISE 4

1. Describe the convolution stage of a Convolutional Neural Network (CNN), illustrating all the elements involved and highlighting the trainable parameters.
2. Discuss the properties of sparse connectivity and parameter sharing for CNN.


Question 4
1). In a convolutional stage is performed the convolution operation:
$(I * K)(i,j) = \sum_{m \in S_1} \sum_{n \in S_2} I(m,n) \cdot K(i-m, j-n)$
The property that is used is sparse connettivity but very often we can also find parameter sharing.
Is possible to find also pooling in this stage, in this mode we can choose the correct steps for our convolution operation.
2). In parameter sharing we force some parameters to share the same value. In this mode we are reducing the number of parameters that we have to learn.
we are saying that outputs depend only on a few

Convolutional stage in discrete case

$$I * K(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

Given a kernel $m \times n$ the number of trainable paramters are the size of the kernel

- parameter sharing: it is technique to reduce to overfitting and it consist in sharing the same weights between one or more weights and one I update one, I update all the other one “linked”
- sparse connectivity: it is a property that each neuron is connected to only a limited number of other neurons

▼ Describe two different methods to overcome overfitting in CNN

Two methods to avoid overfitting in CNN are:

- dropout: during the training some neurons are selected randomly according to a certain probability and they set their output to 0 creating a “new network” with missing connection. In that way if we have some neurons which act badly, we can fix them
- early stopping: it is a technique which consist to stop in advance the training. Reason to stop can be that the model does meaningles improvement on validation set or maybe when the training loss start decrease and validation loss increase.

▼ SVM: maximal margin

SVM (vector support machine) is a model and its aim is the find an hyperplane which maximally separated the different classes.

Given $h(x, w) = wx + w_0$ (w_0 is the learnable bias) to find the best hyperplane we need to solve the following optimization problem

$$w^*, w_0^* = \arg\max_{w, w_0} \frac{1}{\|w\|} = \arg\min_{w, w_0} \frac{1}{2}\|w\|^2$$

draw a dataset 2D for binary classification with SVM and sign the distance hyperplane-svm as $\frac{1}{\|w\|}$

▼ SVM: kernel and epsilon tube, slack variable

Consider the following energy-like function defining Support Vector Machine regression:

$$J(\mathbf{w}, C) = C \sum_{i=1}^{N} L_\epsilon(t_i, y_i) + \frac{1}{2}\|\mathbf{w}\|^2,$$

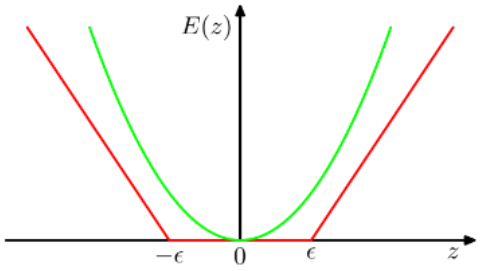
with y_i, t_i target and predicted values, respectively, and $L_\epsilon(t, y) = \begin{cases} 0 & \text{if } |t-y| < \epsilon \\ |t-y| - \epsilon & otherwise \end{cases}$ the ϵ-insensitive error function.

1. Plot the ϵ-insensitive error function and explain what is the difficulty in minimizing J.
2. To overcome this difficulty slack variables ξ^+ and ξ^- are introduced. Explain (qualitatively) the role of the slack variables.

Consider the following

$$J(w) = C\sum_{n=1}^{N} E_\epsilon(y_n, t_n) + \frac{1}{2}\|w\|^2$$

$$E_\epsilon(y,t) = \begin{cases} 0 & \text{if } |y-t| < \epsilon \\ |y-t| - \epsilon & \textit{otherwise} \end{cases}.$$



ϵ indicative the sensibility to make error in regression but this is not differentiable so difficult to solve.

We can avoid this problem introducing slack variables

$$\begin{cases} \xi_n^+, \xi_n^- \geq 0 \\ t_n \leq y_n + \epsilon + \xi_n^+ \\ t_n \geq y_n - \epsilon - \xi_n^- \end{cases}$$

Points inside the ϵ-tube $y_n - \epsilon \leq t_n \leq y_n + \epsilon \Rightarrow \xi_n = 0$ else

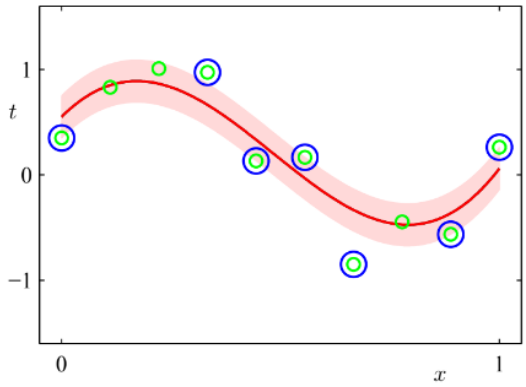
$$\begin{cases} y_n - \epsilon \leq t_n \leq y_n + \epsilon \Rightarrow \xi_n = 0 \quad inside\ tube \\ \xi_n^+ > 0 \quad t_n > y_n + \epsilon \\ \xi_n^- > 0 \quad t_n < y_n - \epsilon \end{cases}$$

Only one slack variable per samples will be activated and one indicate it is “above” the tube (too high classification), the other one “under” (too low classification)

At this poitn we can rewrite our loss function as

$$J(w) = C\sum_{n=1}^{N} (\xi_n^+ + \xi_n^-) + \frac{1}{2}\|w\|^2$$

Example: support vectors and ϵ insensitive tube



▼ Discuss why maxium margin solution is preferred for the classification problem

The SVM approach is the best respect other solutions to classification problem because it generates the best linear discriminant that separates better the classes. Using perceptron we generate a hyperplane that is near to one of the classes: this makes the model sensible to outliers and can misclassify new samples. With SVM the margin calculated respect to support vectors, is maximized, solving this problem.

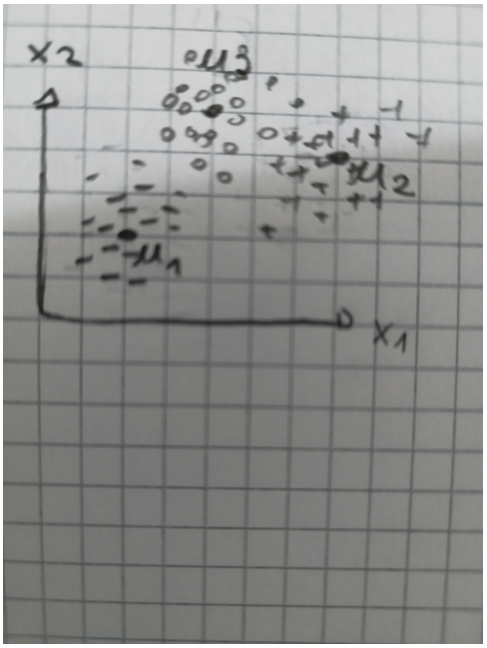
▼ Definition of GMM, gaussian mixture model and describe parameters of the model.

GMM (gaussian mixture model) is an unsupervised learning algorithm which determine the mixed probability distribution from data and it is formed by K different gaussian distribution

$$P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

- π_k prior probability
- μ_k mean
- Σ_k covariance matrix

▼ Draw a 2D dataset generated by GMM with k=3 showing parameters of the model in the pictures



- $\mu_{1,2,3}$ are tthe respective mean of the gaussian distribution
- $\pi_{1,2,3}$ are the prior probability, here they have an uniform probability
- $\Sigma_{1,2,3}$ are the covariance matrix which are the same for the model

Note the π, Σ are equale because what I draw are 3 identical distributions.

Given the GMM $P(x)$

$$P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

The toal number of parameter with $K = 3$ and $D = 2$ is:

- $K - 1 = 2$ for prior probability $\pi_{1,2,3}$ (The last one can be calcualted using 1-sum_i)
- $K * D = 6$ for $\mu_{1,2,3}$. I have sample in 2D and 3 means

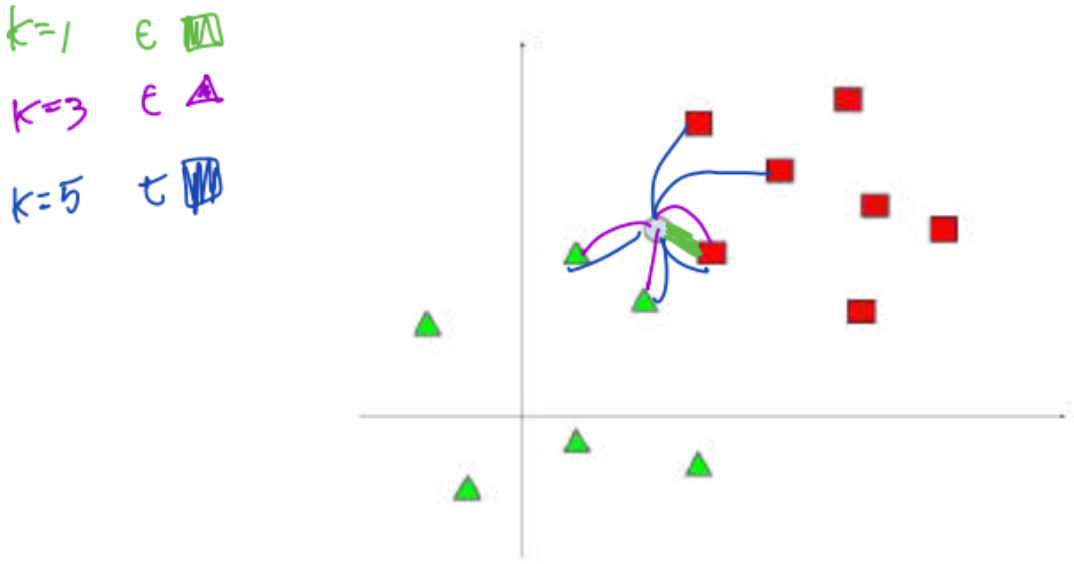
- $K * \frac{D*(D+1)}{2} = 3 * 6 = 18$ the parameter of covariance matrix (since is is symmetric we have less than $D * D$ parameters)
- **Total number: 26 parameters**

▼ Describe KNN algorithm and determine answer of KNN given a dataset

KNN (k nearest neighbours) is a supervised learning algorithm which can be used for classification and regression problems.

KNN-algorithm

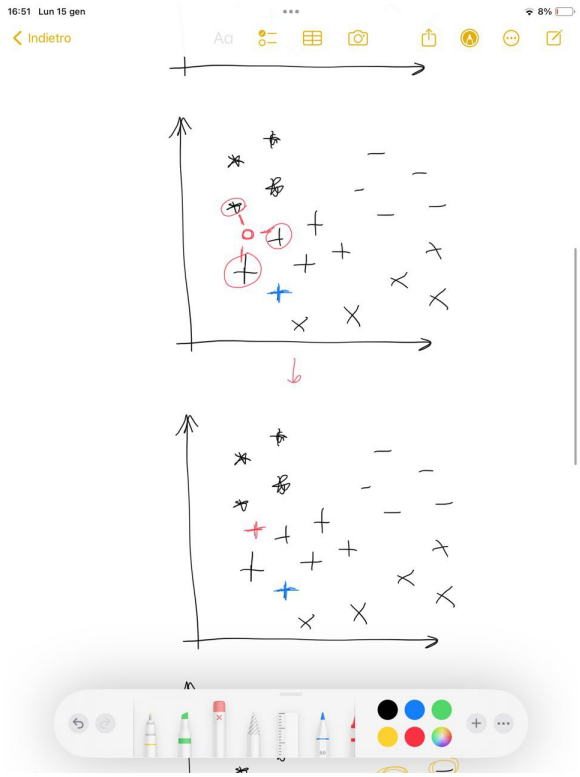
1. $D = \{(x_i, y_i)^N\},\ k = \text{nearest neighbour}, d(x, y) = \text{distance metrics},\ x = \text{test sample}$
2. For each training samples $(x_i, y_i) \in D$
   a. compute $d(x, x_i)$
3. Consider the label of the first k samples which have the minium distance from x
4. assign the label $y_k = \text{maxcount}_{k-samples}(y_i)$ so the label will be assigner for majority between the nearest k samples

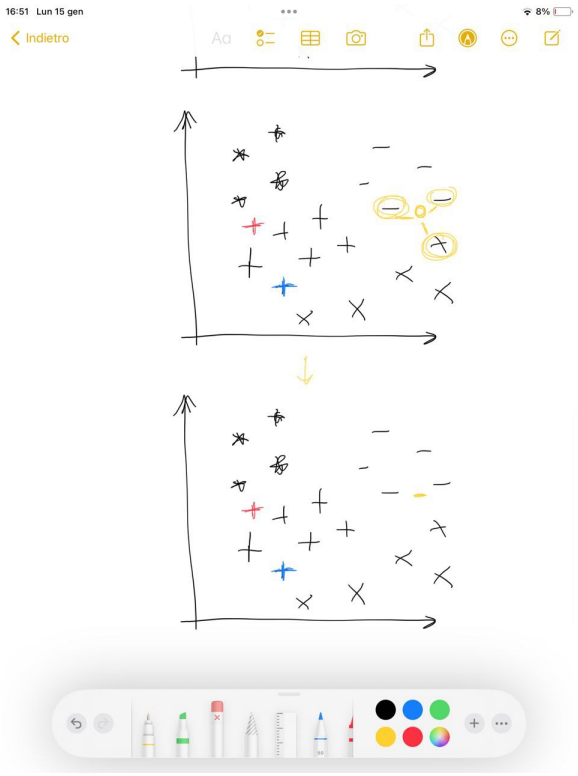


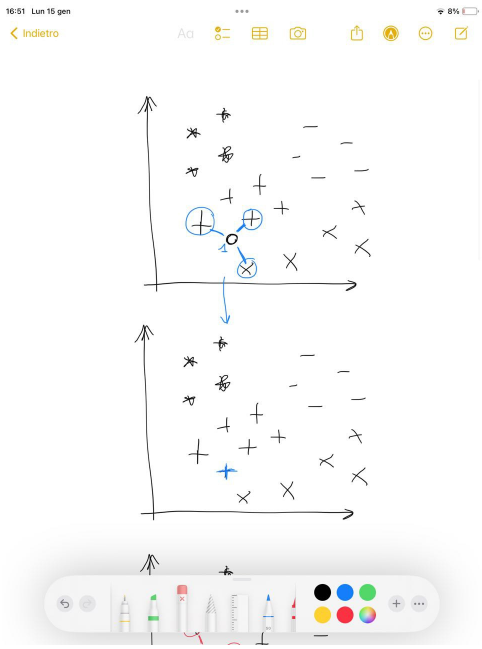
1. Provide the main steps of classification based on K-nearest neighbors (K-NN).
2. Draw an example in 2D demonstrating the application of the 3-NN algorithm for the classification of 3 points given a dataset consisting of points from 4 different classes.

Notes: You can choose how the points of the 4 classes are distributed. Use a different symbol for each class (e.g. use (*,x,+,-) for the classes and (o) for the points to be classified).

k=3, points=3, 4 classi







▼ Briefly describe a linear classification method and discuss its performance in presence of outliers. Use a graphical example to illustrate the concept.

A linear classification method is the perceptron model which is describe by the following function

$$f(x;\theta) = a(\theta_1 x + \theta_{bias})$$

With a the appropriate activation function, and example is the “sign” function.

Given the following loss function

$$MSE = \frac{1}{2}\sum_{i=1}^{N}(t_i - f(x_i;\theta))^2$$

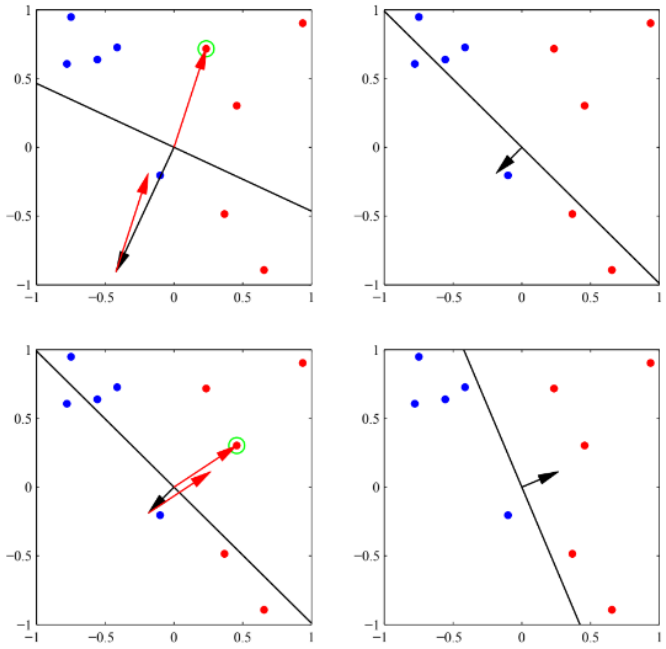
and consider the update

$$\theta_i \leftarrow \theta_i - \eta\frac{\partial MSE}{\partial \theta_i}$$

- η is a small constant learning learning rate

We can find parameters of our function which minimize the MSE using an iterative approach.

Regarding to outlier, this method is not too much robust to outlier because the model find a hyperplane which it can be very close to a certain class and outliers can move badly this separator leading to bad performance for the model.



The linear classification method that i have chosed is least squares. The target function is $f: X \to Y$ with $X \subseteq \mathbb{R}^d$ and the dataset is $\mathcal{D} = \{(x_n, t_n)_{n=1}^N\}$. We want to find $\tilde{w}$ such that $y(x) = \tilde{w}^T \tilde{x}$.

We minimize the error function that is called sum of squares.

$E(\tilde{w}) = \frac{1}{2} Tr\{(T - \tilde{X}\tilde{w})^T (T - \tilde{X}\tilde{w})\}$. Is called sum of squares because the trace operator is simply a sum and because the product between one matrix and the transpose is the square. We want to find $\tilde{w}^* = \arg\min_{\tilde{w}} E(\tilde{w})$

$\tilde{w} = \tilde{X}^\dagger T$ and $y(x) = T^T (\tilde{X}^\dagger)^T \tilde{X}$. This method is not robust to outliers because is simply based on a distance and outliers are sample derived from a different probability so they are very far away from all the other samples and the solution will be affected.

▼ Explain differences between regression and classification

- Classification: it a function of type $f : X \to C$ with $C = \{c_1, c_2, ..., c_n\}$ that are the label / classes of each input
- Regression: is a function of type $f : X \to R$

▼ Provide a mathematical formulation of linear regression

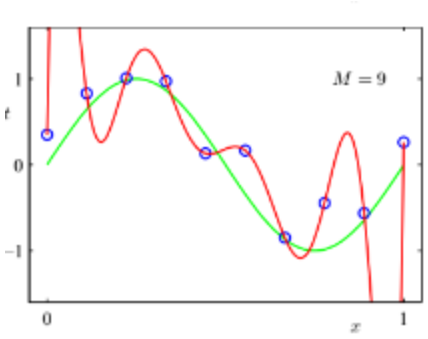
$$y(x; w) = \sum_{j=0}^{M} w_j \phi_j(x) = w^T \phi(x)$$

With:

- $w = \begin{bmatrix} w_0 & ... & w_M \end{bmatrix}^T$
- $\phi(x) = \begin{bmatrix} \phi_0(x) & ... & \phi_M(x) \end{bmatrix}^T$
- $\phi_0(x) = 0$

ϕ remap the input in a linear space

▼ Provide and example of a linear regression model that overfits a dataset of your choice and discuss how it can be mitigated



We can avoid overfitting is regularization.

Given a generic error funciton E and a model $y(x; w)$, the new error function is the following

$$E_{reg}(y) = E(y) + \lambda \|w\|^2$$

▼ define mathematically the problem solver by logistic regression

Logics regression is a probabilistic discriminative model for binary classification.

Given a datasaet $D = \{(x_i, t_i)^n\}$ with $t_i \in \{0, 1\}$, thefunction is

$$p(t|\tilde{w}) = \prod_{n=0}^{N} y_n^{t_n} (1 - y_n)^{(1-t_n)}$$

With error function

$$E(\tilde{w}) = -\ln p(t|\tilde{w}) = -\sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$$

The aim is to solve the optimization problem

$$\tilde{w}^* = \arg\min_{\tilde{w}} E(\tilde{w})$$

▼ Explain kernel trick with necessary condition to apply it. Provide and application, in detail draw a suitable dataset for binary classifiction 2D, discuss with kernel would you use and show graphically a possible solution of such kernel-based model

▼ formal definition of overfitting and illustrate with DT with a solution

Overfitting is a phenomenon in machine learning where a model learns the training data too well, capturing noise and random fluctuations rather than the underlying pattern or trend. As a result, an overfit model performs well on the training data but fails to generalize effectively to new, unseen data.

Consider a scenario where you have a dataset with various features and a binary target variable (0 or 1). The Decision Tree algorithm recursively splits the data based on the features to create a tree structure that predicts the target variable.

Construct a complex Decision Tree that perfectly fits the training data.This tree may have many levels and nodes, resulting in a structure that accounts for every data point.

It achieves high accuracy during training but very poor results of validation or test accuracy.

Solution: Pruning the Decision Tree:

To address overfitting in Decision Trees, a common solution is to prune the tree. Pruning involves removing branches that add little predictive power to the model. This simplifies the tree, making it less complex and more likely to generalize well to new data.

▼ Describe the Naive bayes classifer and highlight the approximation made with respect to the BOC

Naive Bayes Classifier is define as

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j, |D) \prod_{a_i \in x} \hat{P}(a_i|v_j, D)$$

Note that if $a_i \in x$ are i.i.d $\arg\max_{v_j \in V} \prod_{a_i \in x} \hat{P}(a_i|v_j, D) = v_{map}$

In general

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n) P(v_j|D)$$

Naive Bayes Classifier uses conditional indipendece to approximate the solution, infact given X, Y, Z, X is conditionally indipendet of Y given Z if

$$P(X, Y|Z) = P(X|Y, Z)P(Y|Z) = P(X|Z)P(Y|Z)$$

Bayes Optimal Classifier instead consider the total probability of the hypotheses space to compute the label given a sample and it is defien as

$$v_{OB} = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|x, h_i)P(h_i|D)$$

▼ describe the goal of a linear regression model

The goal of a linear regression model is to establish a linear relationship between input variables and targets in a dataset.
In mathematical terms, the goal of linear regression is to find the coefficients (weights) of the linear equation that minimizes the difference between the predicted values and the actual values, doing the minimization of a least squared error function. The linear regression model is represented as:

$y(x, w) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$

and to solve the problem avoiding overfitting we have to find the optimal weights w^* minimizing the LS function and using regularitazion:

$w^* = argmin\, E(w) + \lambda E_w(w)$

▼ Describe difference between generative and discriminative model. Draw a dataset and show the solution in both cases

In the context of probabilistic models for classification, we can use two approaches, for the goal of calculating $P(c_i, |x)$: a generative or a discriminative model.

For the generative model, we estimate $P(x|c_i)$ and then we compute $P(c_i|x)$ using Bayes.

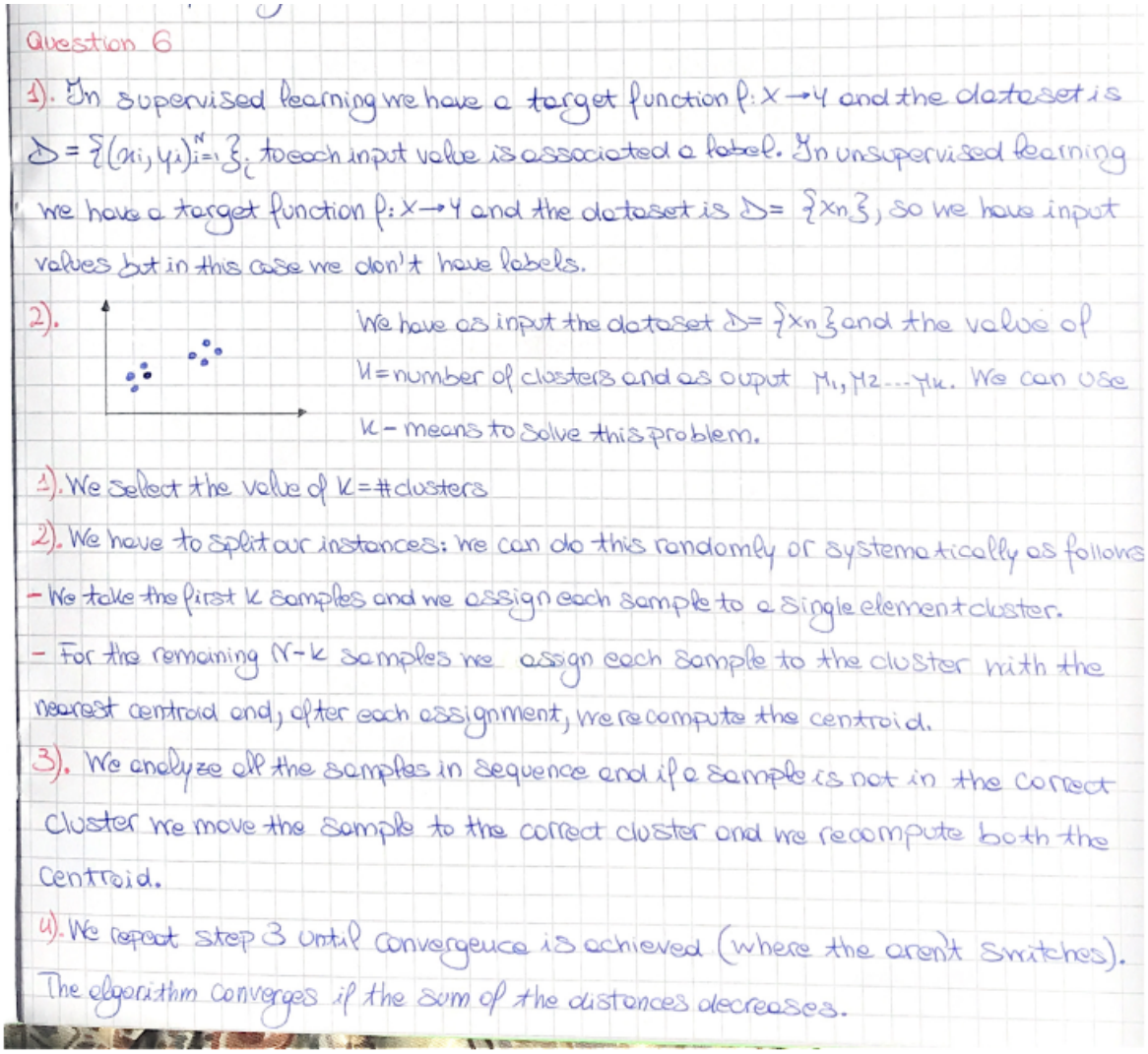
For the discriminative model, we estimate directly $P(c_i|x)$.

▼ Describe and application problem that can be modelled and solved with unsupervised learning method

EXERCISE 6

Machine learning problems can be categorized in supervised and unsupervised.

1. Explain the difference between them providing a precise formal definition (not only explanatory text) in terms of input and output of the two categories of problems.
2. Describe an application problem that can be modelled and solved with an unsupervised learning method.



Question 6

1). In supervised learning we have a target function $f: X \to Y$ and the dataset is $D = \{(x_i, y_i)_{i=1}^{N}\}$, to each input value is associated a label. In unsupervised learning we have a target function $f: X \to Y$ and the dataset is $D = \{x_n\}$, so we have input values but in this case we don't have labels.

2). We have as input the dataset $D = \{x_n\}$ and the value of K = number of clusters and as ouput $\mu_1, \mu_2 \dots \mu_K$. We can use K-means to solve this problem.

1). We select the value of K = #clusters

2). We have to split our instances: we can do this randomly or systematically as follows

- We take the first K samples and we assign each sample to a single element cluster.
- For the remaining $N-K$ samples we assign each sample to the cluster with the nearest centroid and, after each assignment, we recompute the centroid.

3). We analyze all the samples in sequence and if a sample is not in the correct cluster we move the sample to the correct cluster and we recompute both the centroid.

4). We repeat step 3 until convergence is achieved (where the aren't switches). The algorithm converges if the sum of the distances decreases.

Given a dataset $D = \{x_n\}$ (draw a simple datasaet with two clusters) and a certain number of cluster N we can use K-means to solve the problem because it will find the best centroid to classified all the N classes

▼ Describe what is boosting

Boosting is an ensemble learning technique that aims to improve the performance of weak learners. The idea is to split the model into simple different models and train them sequentially so that the previous model influences the next one.

Every sample in the dataset is associated with a weight, that says how much is important in the dataset. When we iterate the first model then we compare the results to the sample target function. If the sample is misclassified the corresponding weight increases. In this way, the next models put more attention on the sample misclassified. Finally, all outputs are combined together $Y_M(x) = sign(\sum_{m=1}^{M} \alpha_m y_m(x))$.

▼ Comment the following statement: in a classification problem, the class returned by the ML hypothesis on a new instance x is always the most probable class.

▼ Provide main steps for achieving higher classification accuraacy by combining multiple instances of the classifier

▼ PCA

EXERCISE 6

1. Explain the goal of dimensionality reduction. Give an example of dimensionality reduction using a problem of your choice.
2. Explain the difference between Principal Component Analysis and Autoencoders in dimensionality reduction.

▼ PCA's algorithm:

given data $X = \begin{bmatrix} x_1^T \\ ... \\ x_n^T \end{bmatrix}$ I want to find a project u on X which maximze the variance on the dataset. Firstly I calculate the mean $\bar{x} = \frac{1}{N} \sum x$, Set $X =$

$\begin{bmatrix}(x_1 - \bar{x})^T \\ ... \\ (x_n - \bar{x})^T\end{bmatrix}$ and the matrix $S = \frac{1}{N} X X^T$ the goal is $\max_{u_1} u_1^T S u_1 + \lambda_1(1 - u_1 u_1^T)$.

Solution is obtained calculating the derivate with respect to u_1 in zero and we get $Su_1 = \lambda_1 u_1$ where u_1 must he and eigenvector of S so $u_1^T S u_1 = \lambda_1$ that correspond to the largest eigenvector.

STEPS:

1. compute mean $\bar{x}$
2. compute S, the covariance matrix of the dataset
3. find m eigenvector of S corresponding to the m largest eigenvalues
4. take the one which maxize the objective function

1. Goal of PCA is to reduce the number of feature in the dataset i.e. find the projection operator u that maxize the variance on the dataset. If out dataset is composed by images that represent hand-written number, we will have a lot of white pixel and only few of them black and we don't need whole images to determine the type of number.
2. In PCA the goal is to reduce the dimensionality of data for speed-up computation (for example) while in autoencoder the aim is to learn a representation on the embedded space and this can be useful in tasks like compression.

▼ How PCA work. Suppose you apply PCA on the data x1,..., x6 and find that the data can be fully described using M principal components, namely u1,..., uM. Describe how the original data can be written in the space defined by these M principal components. Is M going to be equal to the number of intrinsic dimensions? Explain.

▼ Describe how PCA are identified based on principle of variance maximization

Question 3. (5 points) Describe how Principle Components are identified based on the principle of variance maximization.

The aim of PCA is to reduce the number of feature of the input and in order to do so it want to find the projector operator u_1 which maximize the variance of the input data, so it solve the optimization problem

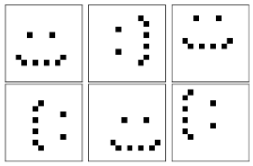
$$\max_{u_1} u_1^T S u_1$$

With S the covariance matrix.

To find the value of u_1 we can set the derivate of this expression to zero and we find that $Su_1 = \lambda_1 u_1$, then you have to left-multiply for u_1^T, so : $u_1^T S u_1 = \lambda_1$ which is the highest eigenvalue, the first principal component.

▼ PCA - Consider the binary (black & white) images below defined on a 12 ⇥ 12 grid:

Consider the binary (black & white) images below defined on a 12×12 grid:



1. Explain what is the dimensionality of the data space and what is the intrinsic dimensionality of the given data.
2. Suppose you apply PCA on the data $\mathbf{x}_1, \ldots, \mathbf{x}_6$ and find that the data can be fully described using M principal components, namely $\mathbf{u}_1, \ldots, \mathbf{u}_M$. Describe how the original data can be written in the space defined by these M principal components.
3. Is M going to be equal to the number of intrinsic dimensions? Explain.

1. Dimensionlity of dataspace is the dimension of all possible comination of images which is $2^{12\times 12}$ while the intrisic dimension is 3: 2 for translation and one for rotation.
2. The aim is to maximize the variance $\max_{u_1} u_1^T S u_1$ with u_1 the project operator and constraint $u_1^T u_1$. In our case the variance is

$$u_1^T S u_1 = \frac{1}{6}\sum_{n=1}^{6}(u_1 x_n - u_1^T \bar{x})^2$$

With $\bar{x} = \frac{1}{6}\sum_{i=1}^{6} x_i$ the mean

We can solve our optimization problem calculating its derivate and imposing it equal to 0.

Now we find that

$$Su_1 = \lambda_1 u_1 \rightarrow u_1^T S u_1 = \lambda_1$$

λ is the largest eigenvector and S the eigenvector associated to λ

3. In general M is usually greater than the number of intrisic dimension because the principale component of PCA are not latent variables
4. version 2. In general M is usually greater than the number of intrisic dimension because the intrisic dimension represetn the minium dimension to represent data and if we go below of that we lose important information

▼ Formal definition of RL problem. Input and output of a RL problem

Reinforcemente learning is field of machine learning which consist to learn a behavior π given a certain task so learn a function $\pi : X \rightarrow A$ with X the state space of the problem and A represent the action available for the agent. Usually dataset of RL have the form $D = \{< x_0, a_0, r_0, x_1 >\}$

▼ Describe main steps of a RL algorithm. Provide pseudo code of a generic RL algorithm

Example with Q-learning

1. Problem initialization (as initializing Q-table in discrete case)
2. Until termination contiion
    a. observe actual state x and take and action a
    b. observe new state x' and collect reward r

c. Update Q-function as: $Q(\hat{x}, a) = r + \gamma \max_{a'} Q(\hat{x'}, a')$

d. $x = x'$

3. Optimal policy: $\pi^*(x) = \arg\max_a \hat{Q}(x, a)$

▼ MDP: Markov decision process

MDP, Markov Decision Process ia a discrete-time stochastic control process and it provide a mathematical framework for modelling decision making.

An enviroment is called Markovian if the agetn have enough information to estimate the future.

For example, if we want estimate velocities through images, we need at least two otherwise we are not able to understand were the objects are mouving

▼ Describe the concept of fully observability in models representing dynamic

When an agent is able to determine the state of system all times is is called “fully observability”, for example in chess game each player can know the actual state of the system i.e. it is fully observable.

In the other hand when the agent is not able to extract all information about the state it is called “partially observable”. And example is a a card game like poker which some card hard face-down and some of them handled by the other player.

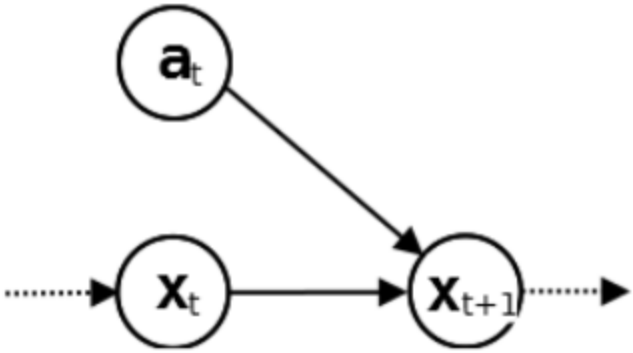
▼ Describe full observability property of MDP and it relation with non-determinist outcomes of actions

When an agent is able to determine the state of system all times is is called “fully observability”, for example in chess game each player can know the actual state of the system i.e. it is fully observable.

When we have non-determinstc outcome it mean that our transition function is stocastic, so we pass from $\delta(s, a) = s'$ to $\delta(s, a) = P(s'|s, a)$ (probability distribution over transitions) so whenver the agetn take an action it has a probability to go on a state s', it may terminate or go on a another state s''

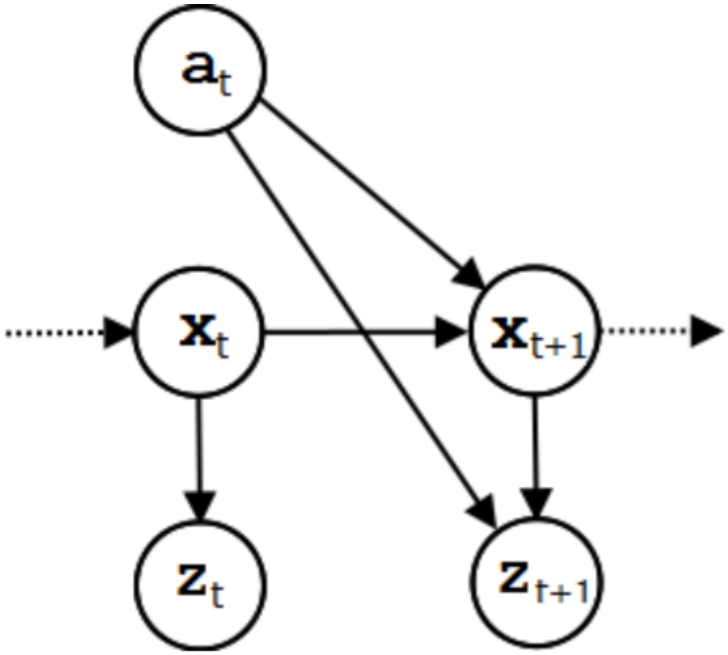
▼ describe difference between a MDP, POMDP and draw and explain the graphical models

- MDP is defined as $\text{MDP} =< X, A, \delta, r >$
  - X a set of finite state,
  - A a set of finite actions
  - δ is the transition function
  - r is the reward function
  - This definition apply when the state are fully observable so I can determine everything about it (an example of fully observable world is chess game)



- POMPD is defined as $\text{POMDP} =< X, A, Z, \delta, r, o >$
  - Xa set of finite state,
  - A a set of finite actions
  - Z a set of observation (are the information captured by the agent)
  - δ is the transition function
  - r is the reward function
  - o is a probability distribution over observations

- This definition apply when the state are not fully observable (an example is a card game like poker where some of that are face-down and players can't see them, so they can't determine everything about the state)
- It has 2 more variable to represent the problem



▼ TODO: MDP vs HMM:

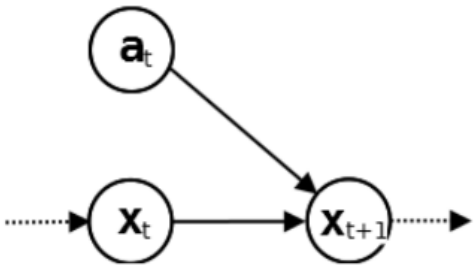
▼ esercizio:

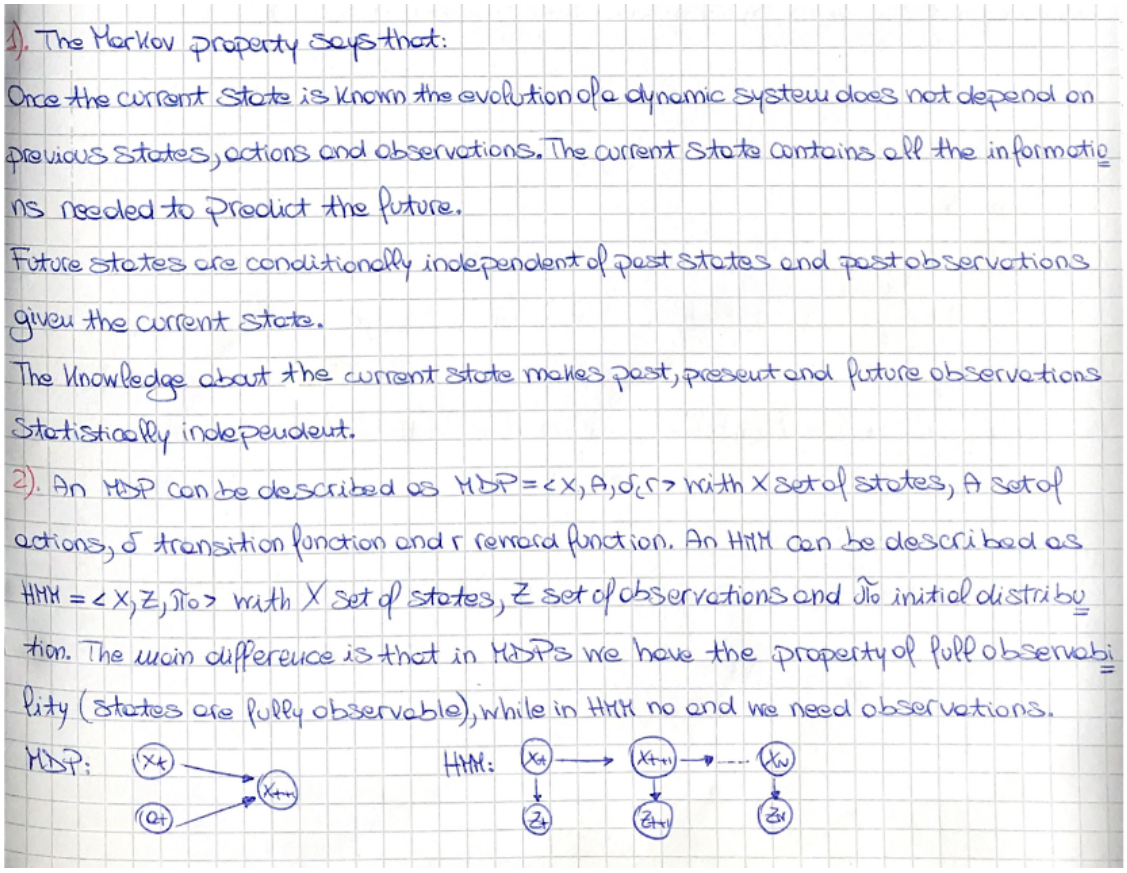
EXERCISE B3

Describe the Markov property of Markovian models representing dynamic systems. Describe the difference between a Markov Decision Process (MDP) and a Hidden Markov Model (HMM). Draw and explain the graphical models of MDP and HMM.

- MDP is defined as $\text{MDP} =< X, A, \delta, r >$
  - X a set of finite state,
  - A a set of finite actions
  - δ is the transition function
  - r is the reward function

- This definition apply when the state are fully observable so I can determine everything about it (an example of fully observable world is chess game)




1). The Markov property says that:
Once the current state is known the evolution of a dynamic system does not depend on previous states, actions and observations. The current state contains all the informations needed to predict the future.
Future states are conditionally independent of past states and past observations given the current state.
The knowledge about the current state makes past, present and future observations statistically independent.
2). An MDP can be described as MDP = <X, A, δ, r> with X set of states, A set of actions, δ transition function and r reward function. An HMM can be described as HMM = <X, Z, π₀> with X set of states, Z set of observations and π₀ initial distribution. The main difference is that in MDPs we have the property of full observability (states are fully observable), while in HMM no and we need observations.
MDP:
HMM:

▼ Descrive k-armed bandit problem (also know as One-state MDP

K-armed bandit problem is a type of problem which has only one state, for describe it think to a world where there is an octopus with k arm which play k

slot machine and it want to maximize the reward when playing.

To solve problem the agent must understand what is the best arm to pull to maxize the reward and we can try to learn the optimal behavior according to the following algorithm

1. Set $N = T/K$ where $T >> K$ and k is the number of arm
2. for $k = 1, ..., K$
    a. pull arm k for N times
    b. observe the reward
    c. compute the empirical mean $\hat{\mu}_k = \sum_{i=1}^{N} \frac{r_i}{N}$
3. for $t = N, ..., T$
    a. pull the empirical best arm: $I_t = \arg\max_{i \in k} \hat{\mu}_i$

▼ Describe RL procedure to compute the optimal policy in k-armed bandit problem

1. Set $N = T/K$ where $T >> K$ and k is the number of arm
2. for $k = 1, ..., K$
    a. pull arm k for N times
    b. observe the reward
    c. compute the empirical mean $\hat{\mu}_k = \sum_{i=1}^{N} \frac{r_i}{N}$
3. for $t = N, ..., T$
    a. pull the empirical best arm: $I_t = \arg\max_{i \in k} \hat{\mu}_i$

▼ How solve a given problem with reinforcement learning and determine the trainig rule

Algorithm:

1. Initialize a data structure Θ
2. For each time $t = 1, \ldots, T$ (until termination condition)
   - **choose** an action $a_{(t)} \in \mathbf{A}$
   - **execute** $a_{(t)}$ and **collect** reward $r_{(t)}$
   - Update the data structure Θ
3. Optimal policy: $\pi^*(\mathbf{x}_0) = \ldots$, according to the data structure Θ

▼ Discuss the strategy for balancing exploration and exploitation in this case.

When training a neural network with reinforcement learning, we have to balance exploration and exploitation, it mean that during train we must evaluate what the agent learn through this action but at the same time we should find a way to see if there exists better action in a certain state.

At this scope the ϵ-greedy policy can help as in this and it is define as

$$\epsilon\text{-greedy}: \begin{cases} \arg\max Q(s,a) & \text{if } \epsilon > \text{random.rand}(0,1) \\ \text{random}(\text{a} \in A) & \text{otherwise} \end{cases}$$

Time to time, during training, instead choose the greedy action, we choose a random one to see if better options exists.

Exercise

▼ function interpolation

EXERCISE 2

Consider the following dataset, containing the samples of a function f:

x_1	f
0.6	2
1.2	4
1.5	5

1. Based on the available data, select a reasonable model for learning f, explicitly indicating its parameters.
2. Show an optimal and a non-optimal solution, explicitly indicating, for each of them, the corresponding value of the loss function.

1. $f(x) = \frac{10}{3}x \rightarrow f(x) : R \rightarrow R, \quad D((x_n, t_n)_{1..N})$
2. optimal is least square ($w^* = (X^t X)^{-1} X^T f$), non optimal is sequential learning (gradient descent update rule)

▼ Descrive input and output of a CNN given a structure + number of parameters for each layer

EXERCISE A1

Consider a CNN with the following structure for its first two layers:

conv1 5×5 kernel and 64 feature maps with padding 2 and stride 1

relu1 acting on 'conv1'

pool1 2×2 max pooling with stride 2 acting on 'relu1'

conv2 3×3 kernel and 128 feature maps with padding 0 and stride 2

relu2 acting on 'conv2'

pool2 2×2 max pooling with stride 4 acting on 'relu2'

1. For input images of dimension $1242 \times 378 \times 3$ compute the dimensions of the volume on the output of each layer and explain how it is computed.
2. Describe what is the number of parameters of each layer.

$$w_{out} = \frac{w_{in} - w_k + 2p}{s} + 1\ , \quad h_{out} = \frac{h_{in} - h_k + 2p}{s} + 1,$$

$$|\theta| = \underbrace{w_k \cdot h_k \cdot d_{in} \cdot d_{out}}_{\text{kernel weights}} + \underbrace{d_{out}}_{\text{bias}}$$

- initial conf: $w_{in} = 1242, h_{in} = 378, d_{in} = 3, d_{out} = 64, w_k = 5, h_k = 5$

conv1: $\begin{cases} w_{in} = \frac{1242-5+2*2}{1} + 1 = 1242 \\ h_{in} = \frac{378-5+2*2}{1} + 1 = 378 \end{cases}$

out = 1242×378×64 , params =(5×5×3+1)x64 = 4864

- pool1: $\begin{cases} w_{in} = \frac{1242-2}{2} + 1 = 621 \\ h_{in} = \frac{378-2}{2} + 1 = 189 \end{cases}$

  out = 621×189×64, params=0

- initial conf: $w_{in} = 621, h_{in} = 189, d_{in} = 64, d_{out} = 128, w_k = 3, h_k = 3$

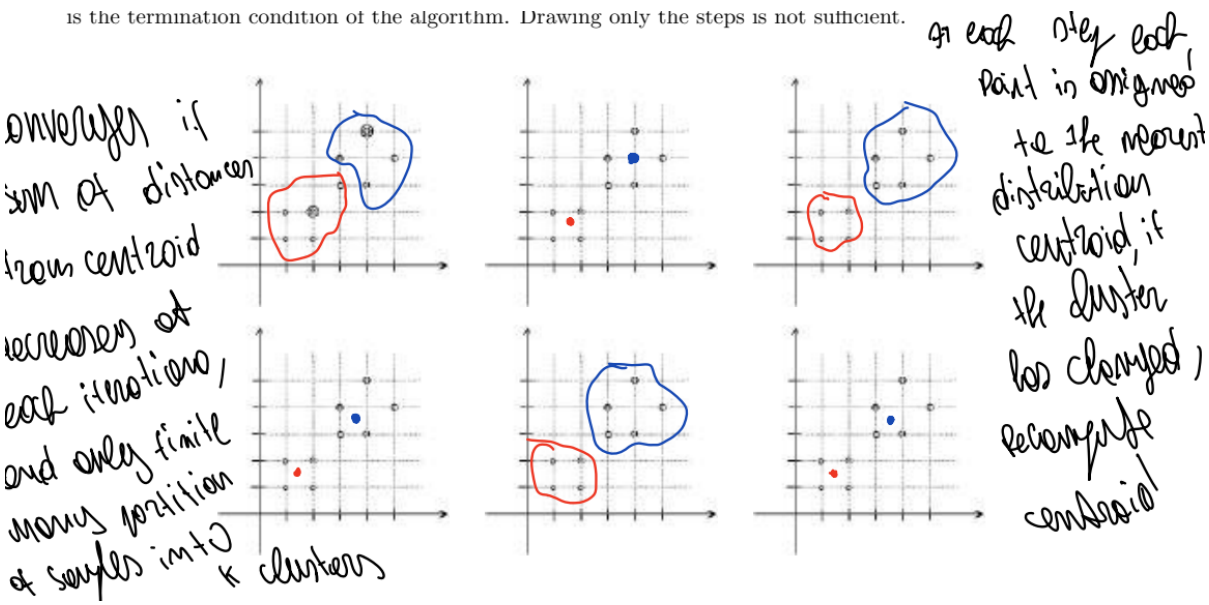
  conv2: $\begin{cases} w_{in} = \frac{621-3}{2} + 1 = 310 \\ h_{in} = \frac{189-3}{2} + 1 = 94 \end{cases}$

  out = 310×94×128 , params =(3×3×64+1)x128 =73856

Total number of paramters: $4864 + 73856 = 78720$

▼ Simulate the execution od K-means in this 2-D dataset with k=2 and initial centroids circles. Use one diagram for each step of the algorithm. Describe explicitly how each step is obatained and what is the termination coniditon of the algorithm. Drawing only the steps is not sufficient

is the termination condition of the algorithm. Drawing only the steps is not sufficient.



We associated label to data point to the nearest centroids. Later we we will re-calculate the new cetroid of the class and we continue to iterate and we stop when there are no more changes of label / centroid

1. assign label
2. calculate new centroid
3. assign label
4. calculate new centroid
5. assign label
6. calculate new centroid and stop because label didn't change

▼ Consider a data set D for scoring different schools with the following real-valued attribues: staff salaries per pupil x_1, teacher's test score x_2, parents' education x_3, school grade y

Question 6. (5 points)

Consider a data set D for scoring different schools with the following real-valued attributes: staff salaries per pupil (x_1), teacher's test score (x_2), parents' education (x_3), school grade (y).

For this problem, an expert of the domain proposes to use the following model.

$$y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1 x_2 + \theta_5 x_3^2$$

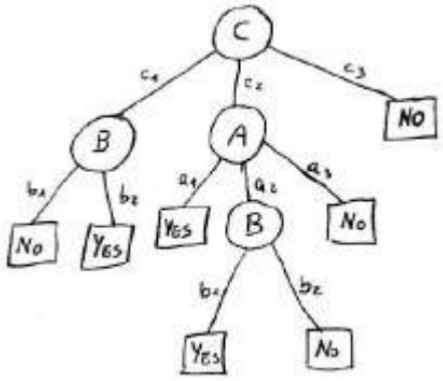
1. Define an error function for this model.
2. Discuss if a linear model for regression can be used in this case.
3. Describe an iterative approach to solve the problem.

1. $E = \frac{1}{2} \sum_{i=0}^{N} (y(x_i; \theta) - t_i)^2$
2. Yes, this model can be used to determine the score value of each school
3. We can find the best parameter in an iterarive way using the following rule

$$\theta_{i+1} \leftarrow \theta_i - \eta \frac{\partial E}{\partial \theta_i}$$

▼ Find rule for a DT

Given a classification problem for the function $f : A \times B \times C \rightarrow \{+, -\}$, with $A = \{a_1, a_2, a_3\}, B = \{b_1, b_2\}, C = \{c_1, c_2, c_3\}$ and the following decision tree T that is the result of a learning algorithm on a given data set:



1. Provide a rule based representation of the tree T.
2. Determine if the tree T is consistent with the following set of samples $S \equiv \{s_1 = \langle a_1, b_1, c_1, No \rangle, s_2 = \langle a_2, b_1, c_2, Yes \rangle, s_3 = \langle a_1, b_2, c_3, Yes \rangle, s_4 = \langle a_2, b_2, c_2, Yes \rangle\}$. Show all the passages needed to get to the answer.

Given the dataset, we can find a rule in this way

- each branch from the initial node caraterize a LOGIC OR
- a path is linked with LOGIC AND
- At the and I will have a rule which contain all possible path to YES linked by LOGIC OR

So

Rule for yes: (C=c1 and B=b1) or (C=c2 and A=a1) or (C=c2 and A=a2 and B=b1)

s1 yes, s2 yes, s3 no, s4 no → its not consistent.

- pool2: $\begin{cases} w_{in} = \frac{310-2}{4} + 1 = 78 \\ h_{in} = \frac{94-2}{4} + 1 = 24 \end{cases}$

  out = 78×24×128, params=

▼ Provide designb and implementation choices for solving the following problem through naive bayes classifier: categories = { ML, KR, PL }, D = { title, authors, abstract, name of journal)

Naive Bayes Classifier is based on the property of condiitonal indipendence so given X, Y, Z if X is indipendet from Y given Z is

$$P(X, Y|Z) = P(X|Y, Z)P(Y|Z) = P(X|Z)P(Y|Z)$$

Naive Bayes Classifier is define as following

$$v_{NB} = \arg\max_{v_j \in V} P(v_j|D) \prod_{a_i \in x} P(a_i|v_j, D)$$

And for our problem we get:

- $V = \{ML, KR, PL\}$
- $x = \{\text{title, authors, abstract, name of journal}\}$

▼ Design an ANN for learning a function $t = f(x, \theta)$

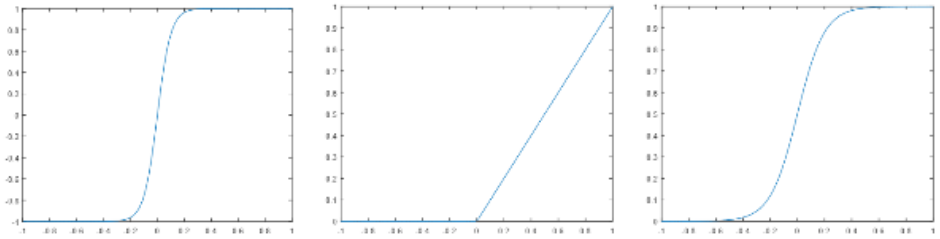
Question 4. (5 points) Let $x \in \mathcal{X} = \{(1.8, 2.1)^T, (3.2, 1.4)^T, (2.5, 4.1)^T, (1.6, 7.7)^T, (3.1, 9.1)^T\}$ and $t \in T = \{0, 1, 1, 0, 1\}$. Consider designing an artificial neural network for learning the function $t = f(x, \theta)$.

1. Explain what is a suitable activation function for the output layer of the network.
2. For the selected activation function explain if the output units saturate and how learning the parameters of the network is affected if this is the case.

▼ Activation function of cnn parameters

Question 5. (5 points)

1. Consider an image of $w_{im} \times h_{im}$ elements (pixels) and a convolution kernel of dimension $3 \times 3 \times 16$. What are the possible values of the stride and padding in order to convolve without skipping any pixels, while obtaining a feature map with the same dimensions with the input image.
2. Associate the correct name of the activation function to the plots above and provide the corresponding mathematical definition. The list of names is {*ReLU, Sigmoid, Hyperbolic Tangent*}.



▼ Consider a two-layer ANN which receives in input a vector x. Calculate dimension of weights matricies W_1, W_2

EXERCISE 2

Consider a two-layers ANN which receives in input vectors $\mathbf{x}$ of dimension 128 and produces output vectors $\mathbf{y}$ of dimension 10. The hidden layer of the ANN is composed of 50 units which use the ReLU activation function. The output units use a linear activation function.

- The weight matrices of the hidden and output layers are denoted W_1 and W_2. Provide the dimensions of the weight matrices W_1 and W_2
- Provide the formula explicitly stating how the values of $\mathbf{y}$ are computed given an input vector $\mathbf{x}$ in terms of the weight matrices and the activation functions (you can ignore the bias terms).

- W_1: rows=50, columns=128
- W_2: rows=10, columns=50

$$f(x) = f^{(n)}(f^{(n-1)}(f^{(n-2)}(...;\theta_{n-2});\theta_{n-1});\theta_n)$$

With:

- $f^{(n)}(x) = \theta_N f^{(n-1)}(x;\theta_{n-1})$
- $f^{(i)}(x) = \max(0, f^{(i')}(x;\theta_i))$, with $\forall i < N, i \geq 0$