SAPIENZA
UNIVERSITÀ DI ROMA

Nijat Mursali

Machine Learning

ALL EXAMS

Rome, Italy

EXAM - January 2018

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 × 378 × 3 compute the dimensions of the volume on the output of each layer and explain how it is computed.

$$w_{out} = \frac{(w_{in} - w_k + 2p)}{s} + 1 \qquad h_{out} = \frac{(h_{in} - h_k + 2p)}{s} + 1$$

In this context w_k is width of kernel and h_k is the height of the kernel. So, $w_{out} \times h_{out}$ will be dimensions of the output feature maps. In our example, $\times 3$ it came because of the RGB color. Thus, we keep it out of operations as constant.

Convolution 1

$$w_{out} = \frac{1242 - 5 + 2\times2}{1} + 1 = 1241 + 1 = 1242$$

$$h_{out} = \frac{378 - 5 + 2\times2}{1} + 1 = 377 + 1 = 378$$

64 feature maps means we use 64 kernels for one image. Thus, 64 different feature maps will be generated.

$$Output: 64 \times 1242 \times 378 \times 3 \rightarrow RGB$$

Relu 1 → the same result.

Max pooling → we will treat it as kernel, but there is no padding $\rightarrow \boldsymbol{p = 0}$

Input: 1242×378

$$w_{out} = \frac{1242 - 2}{2} = 620$$

$$h_{out} = \frac{378 - 2}{2} = 188$$

$$Output:\ 64\ \times 620\ \times 188\ \times 3\ \rightarrow RGB$$

Convolution 2

In this one, padding is 0 and stride is 2 and as we know from previous output the input dimensions for this will be $620\ \times 188$. Thus,

$$w_{out}\ =\ \frac{620-3}{2}+\ 1=\ 310$$

$$h_{out}\ =\ \frac{188-3}{2}+1=\ 94$$

$$Output:\ 128\ \times 64\ \times 310\ \times 94\ \times 3\ \rightarrow RGB$$

Relu 2

$128\ \times 64\ \times 310\ \times 94\ \times 3 \rightarrow ReLU \rightarrow 128\ \times 64 \times 310 \times 94 \times 3$

Pool 2

Stride in this case is 4, so *s = 4, and p = 0, thus*

$$w_{out}\ =\ \frac{310-2}{4}+\ 1=\ 78$$

$$h_{out}\ =\ \frac{94-2}{2}+1=\ 24$$

$$Output:\ 128\ \times 64\ \times 78\ \times 24\ \times 3\ \rightarrow RGB$$

Thus, it will be the result of CNN.

2. Describe what is the number of parameters of each layer.

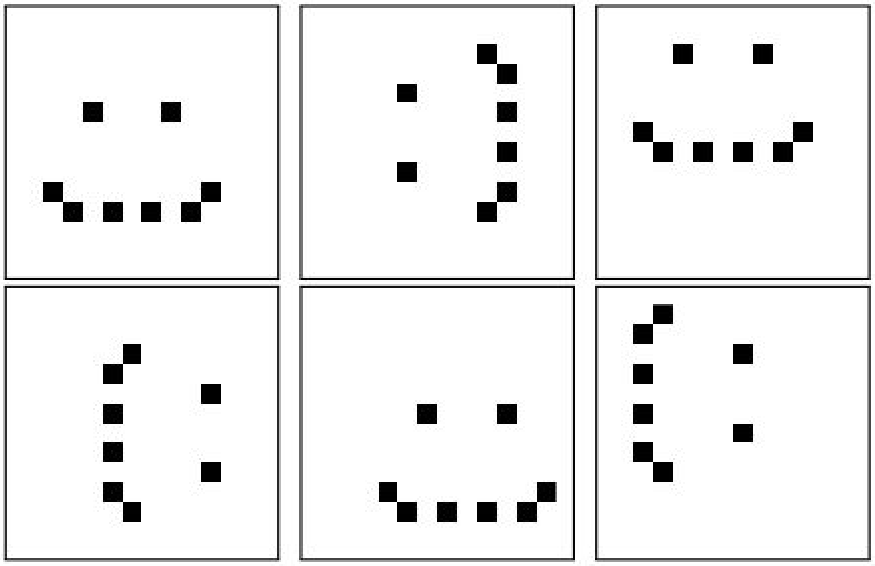
In order to find the number of parameters for layer we need to use formula:

$$parameters\ =\ w_k * w_k * last\ parameter\ of\ image + stride$$

$$final\ =\ parameters * filter$$

EXERCISE A2

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 $x_1, ..., x_6$ and find that the data can be fully described using M principal components, namely $u_1, ..., u_6$. 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.

EXERCISE A3

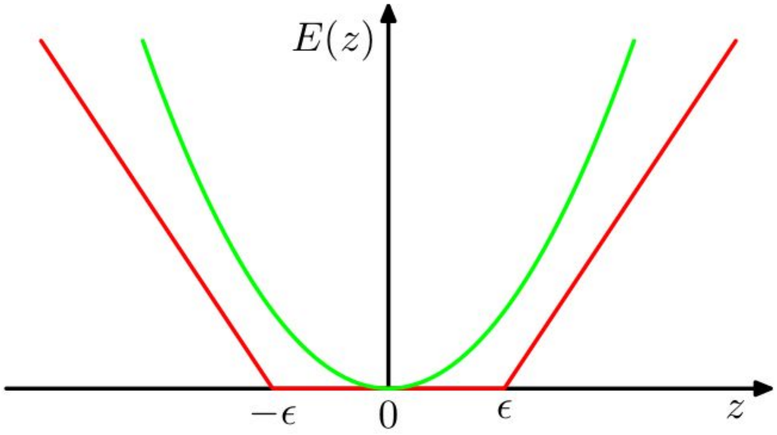
Consider the following energy-like function defining Support Vector Machine regression:

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

with y_i, t_i target and predicted values, respectively, and $L(t, y) = \{0 \; if \; |t-y| < \}$ the -intensive error function.

1. Plot the -insensitive error function and explain what is the difficulty in minimizing J.

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



2. To overcome this difficulty slack variables $\xi+$ and $\xi-$ are introduced. Explain (qualitatively) the role of the slack variables.

The fundamental idea is what if the data are almost linearly separable. We need to introduce *slack variables* $\xi_n^{\ +}, \ \xi_n^{\ -} \geq 0$.

$$t_n \leq y_n + \varepsilon + \xi_n^{\ +}$$
$$t_n \geq y_n - \varepsilon - \xi_n^{\ -}$$

Points inside the $\varepsilon - tube$ $y_n - \varepsilon \leq t_n \leq y_n + \varepsilon \Rightarrow \xi_n = 0$

$\xi_n^{\ +} > 0 \Rightarrow t_n > y_n + \varepsilon$
$\xi_n^{\ -} > 0 \Rightarrow t_n < y_n - \varepsilon$
with $y_n = y(x_n; w)$

EXERCISE B1

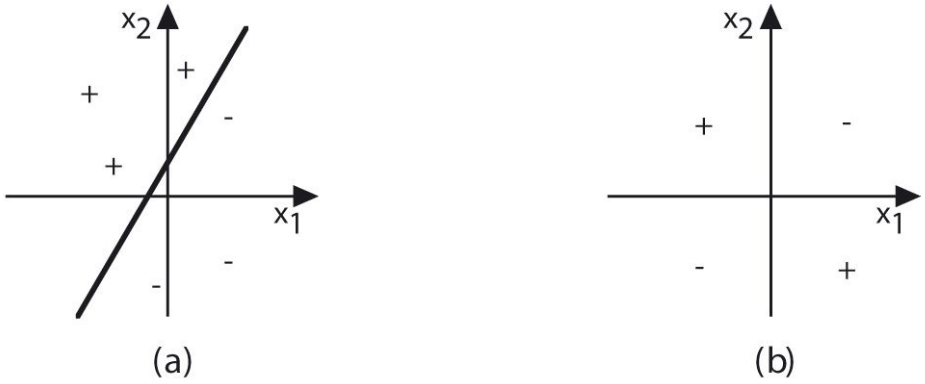
Briefly describe a linear classification method and discuss its performance in presence of outliers. Use a graphical example to illustrate the concept.

Learning a function $f : X \rightarrow Y$,with …

- $X \subseteq R^d$
- $Y = \{C_1, \ . \ . \ . \ , \ C_k\}$

assuming linearly separable data.

Instances in a data set are *linearly separable* if there exists a hyperplane that divides the instance space into two regions such that differently classified instances are separated.



There are several linear functions such as

1. **Least squares**

   Given $D = \{(x_n,\ t_n)^N{}_{n=1}\}$,find the linear discriminant

   $$y(x) = \overline{W}^T \overline{x}$$

   Minimize sum-of-squares error function

   $$E(W) = \tfrac{1}{2} Tr\{(XW - T)^T (XW - T)\}$$

   Closed-form solution:

   $$y(x) = W^T x = T^T (X^T)^T x$$

   However, it is not robust to outliers.

2. **Fisher's linear discriminant**

   Consider two classes case

   Determine $y = w^T x$ and classify $x \in C_1$ if $y \geq - w_0$, $x \in C_2$ otherwise.

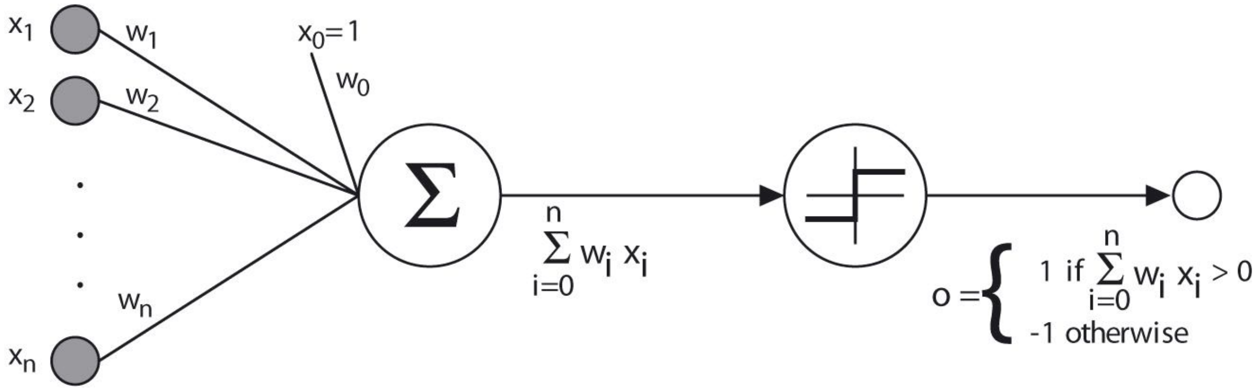
   Adjusting w to find a direction that maximizes class seperation.

   Consider a dataset with N_1 points in C_1 and N_2 point in C_2

   $$m_1 = \tfrac{1}{N_1} \sum_{n \in C_1} x_n \qquad m_2 = \tfrac{1}{N_2} \sum_{n \in C_2} x_n$$

   Choose w that maximizes $J(w) = w^T (m_2 - m_1)$,subject to ||w||=1.

3. Perceptron


x_1
w_1
x_2
w_2
$x_0=1$
w_0
w_n
x_n
$\sum_{i=0}^{n} w_i x_i$
$o = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$

$$o(x_1 , \ . \ . \ . \ , \ x_d) = \{ \frac{1 \ if \ w_0+w_1x_1+ \ . \ . \ . + w_dx_d > 0}{-1 \ otherwise}$$

Perceptron training rule

Consider the unthresholded linear unit, where

$$w_0 + w_1x_1 + \ . \ . \ . \ + \ w_dx_d \ = \ w^Tx$$

Let's learn w_i from training examples D that minimize the squared error (loss function)

$$E(w) \equiv \frac{1}{2} \sum_{n=1}^{N} (t_n - o_n)^2 = \frac{1}{2} \sum_{n=1}^{N} (t_n - w^Tx_n)^2$$

Unthresholded unit:

Update of weights ***w***

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

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

η is a small constant called learning rate.

Perceptron algorithm

Given perceptron model $o(x) = sign(w^Tx)$ and data set D, determine weights **w**.

- Initialize **w**
- Repeat until termination condition
  - $w_i \leftarrow w_i + \Delta w_i$
- Output **w**

EXERCISE B2

In Bayesian Learning, given a data set D and a hypothesis h, we can express the following relationship between the probability distributions (Bayes theorem):

$$P(h \mid D) = \frac{P(D \mid h)\, P(h)}{P(D)}$$

In this context:

1. define *Maximum a posteriori* (MAP) hypotheses and *Maximum likelihood* (ML) hypotheses.

Maximum a posteriori hypothesis h_{MAP} :

$$h_{MAP} = argmax\, P(h \mid D) = argmax \frac{P(D \mid h)\, P(h)}{P(D)} = argmax\, P(D \mid h)\, P(h)$$

Maximum likelihood hypothesis

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

$$h_{ML} = argmax\, P(D \mid h)$$

2. define the concept of *Bayes Optimal Classifier*

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

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

Class of new instance *x:*

$$v_{OB} = argmax \sum_{h_i \in H} P(v_j \mid x,\ h_i)\, P(h_i \mid D)$$

3. discuss about practical applicability of the *Bayes Optimal Classifier.*

Optimal learner: no other classification method using the same hypothesis space and same prior knowledge can outperform this method on average. This algorithm is very powerful to label the new instances *x* with $argmax\, P(v_j \mid x,\ D)$ can correspond to none of hypotheses in *H.*

EXERCISE B3

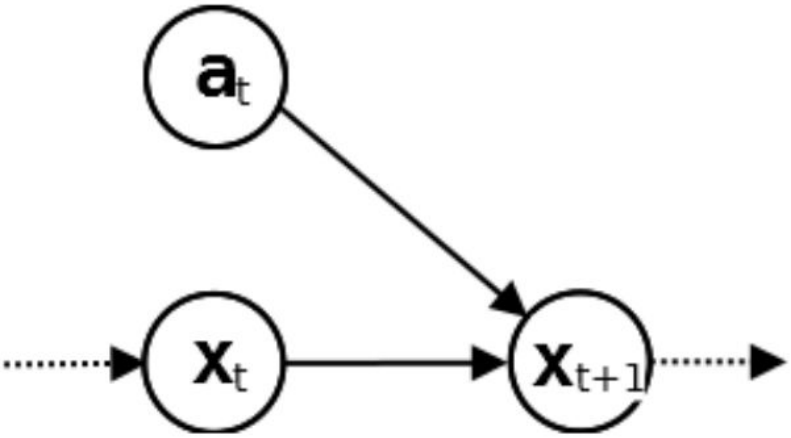
Describe the Markov property of Markovian models representing dynamic systems. Describe the difference between a Markov Decision Process (MDP) and a Hidden Markov Model (HMM). Draw and explain the graphical models of MDP and HMM.

Markov property says that:

1. Once the current state is known, the evolution of the dynamic system does not depend on the history of states, actions and observations.
2. The current state contains all the information needed to predict the future.
3. Future states are conditionally independent of past states and past observations given the current state.
4. The knowledge about the current state makes past, present and future observations statistically independent.

Markov process is a process that has the Markov property.

Markov processes for decision making. In Markov processes states are fully observable and there is no need for observations.



$MDP = < X, A, \delta, r >$

where

- **X** is a finite set of states
- **A** is a finite set of actions
- $\delta : X \times A \rightarrow X$ is a transition function
- $r : X \times A \rightarrow R$ is a reward function

$$HMM = < X, Z, \pi_0 >$$

- transition model: $P(x_t \mid x_{t-1})$
- observation model: $P(z_t \mid x_t)$
- initial distribution: π_0

State transition matrix $A = \{A_{ij}\}$

$$A_{ij} = P(x_t = j \mid x_{t-1} = i)$$

Observation model(discrete or continuous):

$$b_k(z_t) \equiv P(z_t \mid x_t = k)$$

The Markov model is a *state machine* with the state changes being probabilities. In a hidden Markov model, you don't know the probabilities, but you know the outcomes.

For example, when you flip a coin, you can get the probabilities, but, if you couldn't see the flips and someone moves one of five fingers with each coin flip, you could take the finger movements and use a hidden Markov model to get the best guess of coin flips.

EXAM - February 2018

EXERCISE A1

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 a supervised learning model, the algorithm learns on a labeled dataset, providing an answer key that the algorithm can use to evaluate its accuracy on training data. An unsupervised model, in contrast, provides unlabeled data that the algorithm tries to make sense of by extracting features and patterns on its own.

In supervised learning:

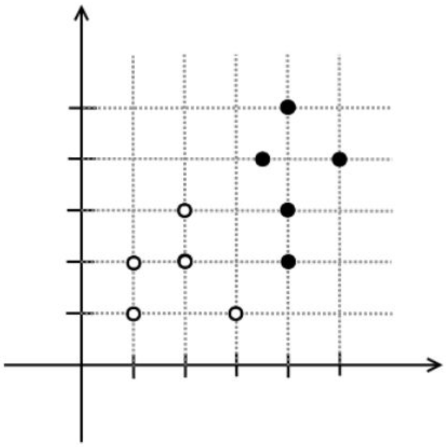
Learning a function $f : X \rightarrow Y$, given $D = \{< x_i, y_i >\}$

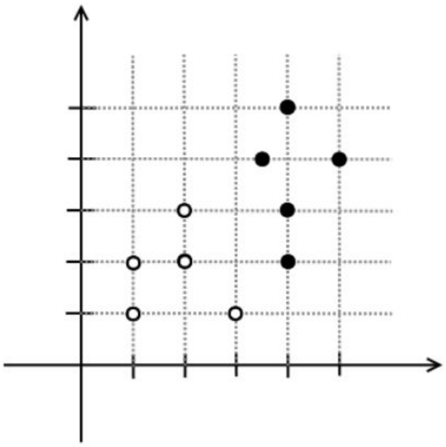
In unsupervised learning:

Learning a function $f : X \rightarrow Y$ with input data $D = \{ x_n \}$*, but without target values.*

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.





Before going into the figures, first let's explain what Perceptron and SVM are. For both cases, we need to find a line which separates the two classes. The parameters for Perceptron are

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

η = Learning rate

Higher you make the learning rate, the faster it is going to take steps, but overshooting a lot. If you make learning rate lower, slower it will make progress to optimal line,

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

$$w_0 + w_1x_1 + w_2x_2$$

Update step is one of the important concepts in Perceptron where it gets the formula

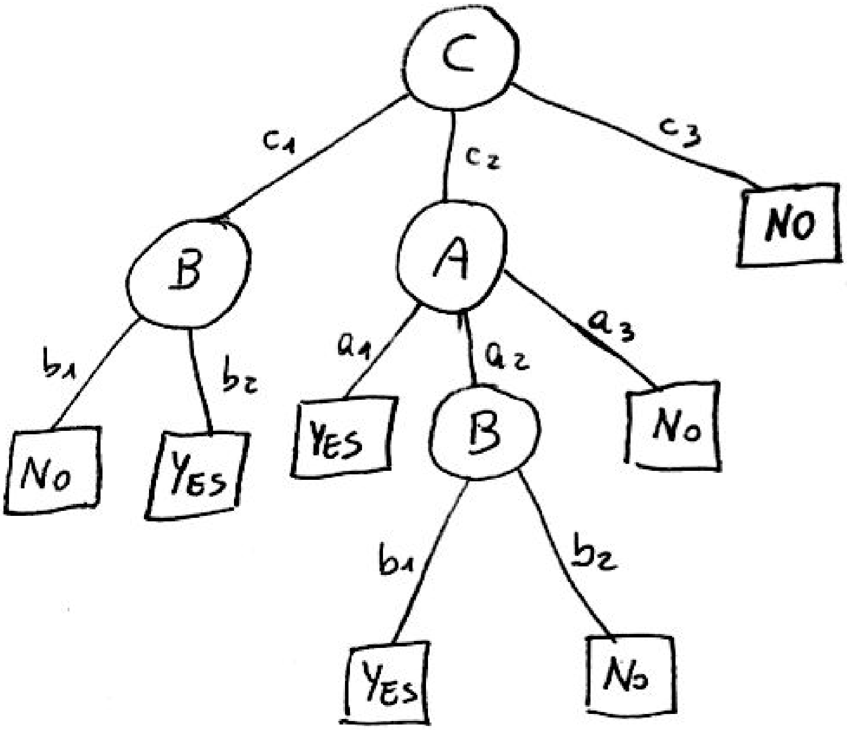
$$w_i' = w_i + \eta d x_i$$

d here is 1 if the class should be upper, and -1 if the class should be lower.

The idea of SVM is also similar to Perceptron, but there are differences as well. In SVMs you need to take the support vectors for two classes where you need to put the support vectors in the boundary of those classes. Then, we can check the hyperplane that perfectly divides those two classes.

EXERCISE A3

Given a classification problem for the function *f : A × B × C → {YES, NO}*, 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 training on a given data set:



1. Provide a rule based representation of the tree T .

In order to make a rule based decision tree we need to remember how it should be done from slides. A rule is generated for each path to the leaf node. Thus, we need to write like

$IF\ (C\ =\ c_4)\ \wedge\ (B\ =\ b_1)$

THEN NO

$IF\ (C\ =\ c_4)\ \wedge\ (B\ =\ b_2)$

$THEN\ YES$

$IF\ (C\ =\ c_2)\ \wedge\ (A\ =\ a_1)$

$THEN\ YES$

So, it goes like that.

2. Determine if the tree T is consistent with the following set of samples

$S\ \equiv \{s_1 =< a_1, b_1, c_1,\ NO >,\ s_2 =< a_2, b_1, c_2,\ YES >,\ s_3 =< a_1, b_2,\ c_3,\ NO >$

$s_4 =\ \ < a_3, b_2, c_1, YES >\}$.

Motivate your answer.

In order to solve this problem, we need to look at the tree itself and see if the samples are going in the way our tree goes. As we see from the tree itself, only s_2 *is consistent.*

EXERCISE B1

1. Provide the main features about boosting.

Main points of boosting are:

- Base classifiers trained sequentially.
- Each classifier trainer on weighted data.
- Weights depend on performance of previous classifiers.
- Points misclassified by previous classifiers are given greater weight.
- Predictions based on weighted majority of votes.

Base classifiers are trained in sequence using a weighted data set where weights are based on performance of previous classifiers.

2. Write the error function whose minimization leads to a formulation equivalent to the **AdaBoost algorithm**.

Given $D = \{(x_1,\ t_1)\ , ...,\ (x_N,\ t_N)\ \}$, where $x_n\ \in X,\ t_n\ \in\ \{-1,\ +1\}$

1. Initialize $w_n{}^{(1)} = \frac{1}{N},\ n\ =\ 1, ...,\ N.$
2. For m = *1, . . . , M*:
    a. Train a weak learner $y_m(x)$ by minimizing the weighted error function:

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) \quad \text{, with } I(e) = 1 \textit{ if e is true, } 0 \textit{ otherwise}$$

b. Evaluate: $\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$ and $\alpha_m = ln[\frac{1-\varepsilon_m}{\varepsilon_m}]$

c. Update the data weighting coefficients:

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

3. Output the final classifier

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

AdaBoost can be explained as the sequential minimization of an exponential error function. Consider the error function

$$E = \sum_{n=1}^{N} exp[-t_n f_M(x_n)],$$

where

$$f_M(x) = \frac{1}{2} \sum_{m=1}^{M} \alpha_m y_m(x), \qquad t_n \in \{-1, +1\}$$

Goal:

$$minimize\ E\ w.r.t.\ \alpha_m,\ y_m(x),\ m = 1, ..., M$$

EXERCISE B2

Consider the problem of finding a function which describes how the salary of a person (in hundreds of euros) depends on his/her age (in years), the months in higher education and average grades in higher education. A dataset in the form $D = \{(x_1^T, t_1), \dots, (x_N^T, t_N)\}$ is provided, with $x \in R^3$ 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 θ.

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.

EXERCISE B3

Given input values x i and the corresponding target values t_i with i = 1, . . . , N , the solution of regularized linear regression can be written as:

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

with $\alpha = (XX^T + \lambda I)^{-1} t,\ X = [x_1, ..., x_N]^T$ and λ the regularization weight.

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

1. Provide a definition of the Gram matrix.

Linear model with any kernel k

$$y(x;\ \alpha) = \sum_{n=1}^{N} \alpha_n {x_n}^T x$$

Solution

$$\alpha = (K + \lambda I_N)^{-1} t$$

Gram matrix

$$K = \begin{bmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \cdots & \mathbf{x}_1^T \mathbf{x}_N \\ \vdots & \ddots & \vdots \\ \mathbf{x}_N^T \mathbf{x}_1 & \cdots & \mathbf{x}_N^T \mathbf{x}_N \end{bmatrix}$$

2. Explain how a kernelized version for regression can be obtained based on the equations provided above.

Linear model for regression $y = w^T x$ and data set $D = \{(x_n,\ t_n)_{n=1}{}^N\}$

Minimize the regularized loss function

$$J(w) \; = \; \sum_{n=1}^{N} E(y_n,\; t_n) + \lambda\|w\|^2$$

where $y_n = w^T x_n$.

Consider $E(y_n,\; t_n) = (y_n - t_n)^2$: i.e, regularized linear regression.

Solution

$$\widehat{w} = (X^T X + \lambda l_M)^{-1} X^T t = X^T \alpha$$

EXAM - April 2018

EXERCISE A1

1. Provide a formal definition of overfitting.

Consider error of hypothesis h over

- Training data: $error_S(h)$
- Entire distribution D of data: $error_D(h)$

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

$$error_S(h) \; < \; error_S(h')$$

and

$$error_D(h) \; > \; error_D(h')$$

2. Discuss the problem of overfitting in learning with *Decision Trees* and illustrate possible solutions to it.

In decision trees, overfitting occurs ***when the tree is designed so as to perfectly fit all samples in the training data set***. Thus it **ends up with branches with strict rules of sparse data**. Thus this affects the accuracy when predicting samples that are not part of the training set.

How can we avoid overfitting?

- Stop growing when data split not statistically significant
- Grow a full tree, then post-prune

EXERCISE A2

1. Describe the *Naive Bayes Classifier* and highlight the approximation made with respect to the Bayes Optimal Classifier.

Bayes Optimal Classifier:

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

Bayes optimal classifier provides the best result, but it is not a practical method when hypothesis space is large. *Naive Bayes Classifier* uses conditional independence to approximate the solution. X is *conditionally independent* of Y given Z

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

Assume target function $f: X \to V$, where each instance x is described by attributes $< a_1, a_2 \ldots a_n >$.

Compute

$$argmax\, P(v_j | \ x,\ D)\ =\ argmax\, P(v_j | \ a_1,\ a_2 \ldots a_n,\ D)$$

without explicit representation of hypothesis.

2. Provide design and implementation choices for solving the following problem through *Naive Bayes Classifier* :

Classification of scientific papers in categories according to their main subject. The categories to be considered are: ML (Machine Learning), KR (Knowledge Representation), PL (Planning). Data available for each scientific paper are: title, authors, abstract and publication site (name of the journal and/or of the conference).

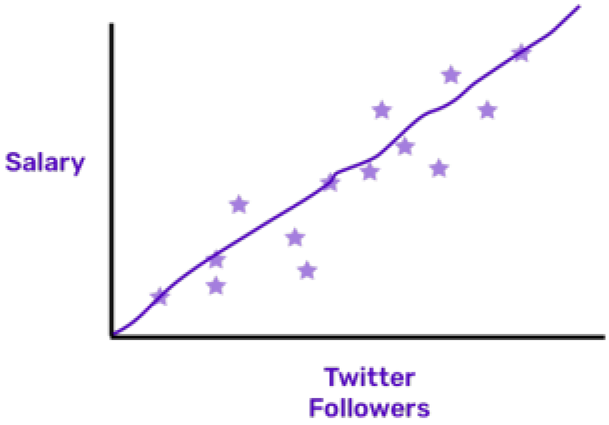
EXERCISE A3

1. Define with a precise formal definition the unsupervised learning problem.

Unsupervised machine learning cannot be directly applied to a regression because it is unknown what the output values could be, therefore making it impossible to train the algorithm how you normally would. In unsupervised learning input data is available $D = \{x_n\}$, but target values not available. Unsupervised data clustering: finding multiple classes from data.

2. Provide a full example of unsupervised learning problem (i.e., a specific invented data set), possibly in a graphical form.

Now apply this framework to machine learning. Traditional datasets in ML have labels (think: the answer key), and follow the logic of “X leads to Y.” For example: we might want to figure out if people with more Twitter followers typically make higher salaries. We think that our input (Twitter followers) might lead to our output (salary), and we try to approximate what that relationship is.



3. Describe a solution to the defined problem based on K-Means, providing examples of execution of some steps of the algorithm and a reasonable solution.

Computing K means of data generated from K Gaussian distributions.

$$Input: \ D = \{x_n\}, \ value \ K \qquad Output: \ \mu_1, ..., \mu_k$$

1. Begin with a decision on the value of k = *number of clusters*
2. Put any initial partition that classifies the data into *k* clusters. You may assign the training samples randomly, or systematically as follows
    a. Take the first k training samples as single-element clusters
    b. Assign each of the remaining training samples to the cluster with the nearest centroid. After each assignment, recompute the centroid of the new cluster.
3. Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the two clusters involved in the switch
4. Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

EXERCISE B1

1. Provide the main steps of classification based on K-nearest neighbors (K-NN).

K-nearest neighbors is a simple non-parametric model which we can call ***instance-based learning.***

Classification with K-NN (target $f : X \to C$, data set $D = \{(x_i, t_i)\}$:

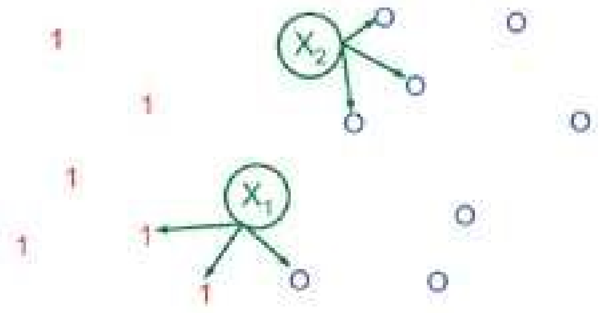
- Find K nearest neighbors of new instance **x**
- Assign to **x** the most common label among the majority of neighbors

Likelihood of class *c* for new instance **x**:

$$p(c \mid x, D, K) = \frac{1}{K} \sum_{i \in N_K(x, D)} I(t_i = c),$$

with $N_K(x, D)$ the K nearest points to x and $I(e)$ *is 1 if e is true and 0 if e is false.*

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.



EXERCISE B2

1. Describe the role of the following notions related to parameter estimation of an artificial neural network:

- backpropagation

  The back-propagation or backprop algorithm is used to propagate gradient computation from the cost through the whole network.
- forward and backward pass
- Stochastic Gradient Descent

  SGD is an iterative method for optimizing an objective function with suitable smoothness properties.

2. Provide the main steps of the backpropagation algorithm.

Forward step

```
Require: Network depth l
Require: $W^{(i)}$ , $i \in \{1, . . . , l\}$ weight matrices
Require: $b^{(i)}$ , $i \in \{1, . . . , l\}$ bias parameters
Require: x input value
Require: t target value
$h^{(0)} = x$
for $k = 1, . . . , l$ do
        $\alpha^{(k)} = b^{(k)} + W^{(k)}h^{(k-1)}$
        $h^{(k)} = f(\alpha^{(k)})$
end for
$y = h^{(l)}$
$J = L(t, y)$
```

Backward step

```
$g \leftarrow \square_y J = \square_y L(t, y)$
For $k = l, l - 1, ... 1,$ do
        Propagate gradients to the pre-nonlinear activations:
        $g \leftarrow \square_\alpha J = g \odot f'(\alpha^{(k)})$ {$\odot$ denotes element wise product }
        ...
end for
```

EXERCISE B3

1. Briefly describe the goal of linear regression and define the corresponding model.

Linear regression is a common Statistical Data Analysis technique. It is used to determine the extent to which there is a linear relationship between a dependent variable and one or more independent variables. There are two types of linear regression, simple linear regression and multiple linear regression. In simple linear regression a single independent variable is used to

predict the value of a dependent variable. In multiple linear regression two or more independent variables are **used to predict the value of a dependent variable**. The difference between the two is the number of independent variables. In both cases there is only a single dependent variable.

Define a model $y(x;\ w)$ with parameters **w** to approximate the target function f.

Linear model for linear function

$$y(x;\ w) = w_0 + w_1 x_1 + \ ... \ + w_d x_d = w^T x$$

Linear Basis Function Models

Using nonlinear function of input variables:

$$y(x;\ w) = \sum_{j=0}^{M} w_j \varphi_j(x) = w^T \varphi(x),$$

2. Given a dataset $D = \{(x_1{}^T,\ t_1),\ ...\ ,(x_N{}^T,\ t_N)\}$ with x_n n the input values and t n the corresponding target values, explain how the parameters of the model can be estimated either in a batch or in a sequential mode.

Stochastic gradient descent algorithm:

$$\widehat{w} \leftarrow \widehat{w} - \eta E_n,$$

with η the learning rate parameter.

Therefore:

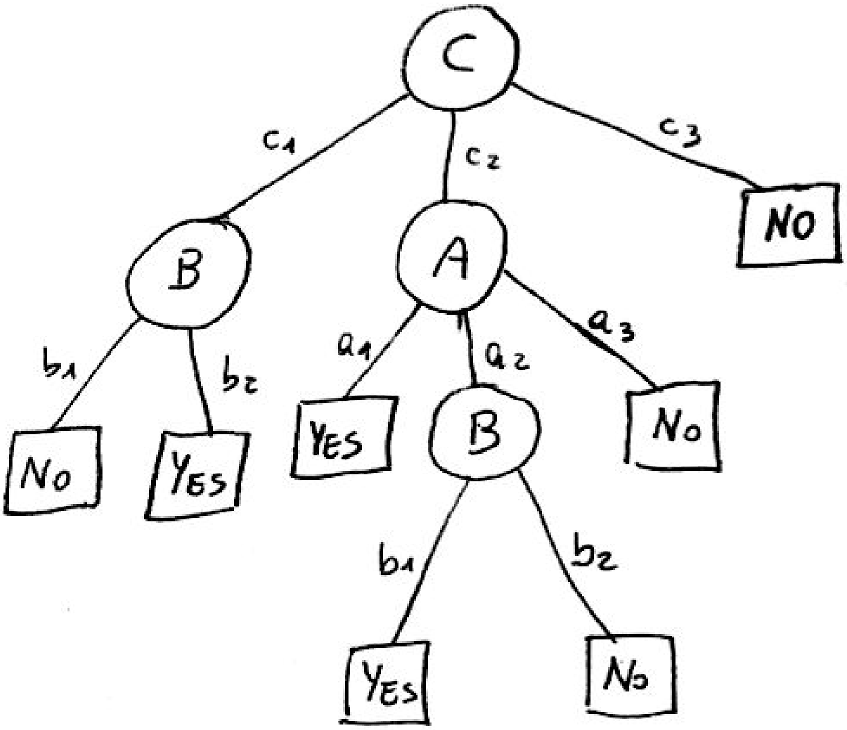
$$\widehat{w} \leftarrow \widehat{w} + \eta[t_n - \widehat{w}^T \varphi(x_n)]\ \varphi(x_n)$$

Algorithm converges for suitable small values of η.

EXAM - January 2019

EXERCISE 1

Given a classification problem for the function *f : A × B × C → {+, -}*, 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 training on a given data set:



1. Provide a rule based representation of the tree T .

In order to make a rule based decision tree we need to remember how it should be done from slides. A rule is generated for each path to the leaf node. Thus, we need to write like

$IF\ (C = c_4) \wedge (B = b_1)$

$THEN\ NO$

$IF\ (C = c_4) \wedge (B = b_2)$

$THEN\ YES$

$IF\ (C = c_2) \wedge (A = a_1)$

$THEN\ YES$

So, it goes like that.

2. Determine if the tree T is consistent with the following set of samples

$S \equiv \{s_1 =< a_1, b_1, c_1,\ NO >,\ s_2 =< a_2, b_1, c_2,\ YES >,\ s_3 =< a_1, b_2,\ c_3,\ NO >$

$s_4 = \ < a_3, b_2, c_1, YES >\}$.

Motivate your answer.

In order to solve this problem, we need to look at the tree itself and see if the samples are going in the way our tree goes. As we see from the tree itself, only s_2 *is consistent.*

EXERCISE 2

In Bayesian Learning, given a data set *D* and a hypothesis *h*, we can express the following relationship between the probability distributions (Bayes theorem):

$$P(h \mid D) \ = \ \frac{P(D \mid h)\, P(h)}{P(D)}$$

In this context:

1. define *Maximum a posteriori* (MAP) hypotheses and *Maximum likelihood* (ML) hypotheses.

Maximum a posteriori hypothesis h_{MAP} :

$$h_{MAP} = argmax\ P(h \mid D) \ = \ argmax \frac{P(D \mid h)\, P(h)}{P(D)} = argmax\ P(D \mid h)\, P(h)$$

Maximum likelihood hypothesis

If assume $P(h_i) \ = \ P(h_j)$, we can further simplify and choose the ML hypothesis.

$$h_{ML} = argmax\ P(D \mid h)$$

2. define the concept of *Bayes Optimal Classifier*

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

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

Class of new instance *x:*

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

3. discuss about practical applicability of the *Bayes Optimal Classifier.*

Optimal learner: no other classification method using the same hypothesis space and same prior knowledge can outperform this method on average. This algorithm is very powerful to label the new instances *x* with $argmax\, P(v_j \mid x,\ D)$ can correspond to none of hypotheses in *H.*

EXERCISE 3

1. Briefly describe the goal of linear regression and define the corresponding model.

Linear regression is a common statistical data analysis technique. It is used to determine the extent to which there is a linear relationship between a dependent variable and one or more independent variables. There are two types of linear regression, simple linear regression and multiple linear regression. In simple linear regression a single independent variable is used to **predict the value of a dependent variable**. In multiple linear regression two or more independent variables are **used to predict the value of a dependent variable**. The difference between the two is the number of independent variables. In both cases there is only a single dependent variable.

Define a model $y(x;\ w)$ with parameters **w** to approximate the target function f.

Linear model for linear function

$$y(x;\ w)\ =\ w_0 + w_1 x_1 +\ ...\ +\ w_d x_d\ =\ w^T x$$

Linear Basis Function Models

Using nonlinear function of input variables:

$$y(x;\ w)\ =\ \sum_{j=0}^{M} w_j \varphi_j(x)\ =\ w^T \varphi(x)\,,$$

2. Given a dataset $D\ =\ \{(x_1{}^T,\ t_1),\ ...\ ,(x_N{}^T,\ t_N)\}$ with x_n n the input values and t n the corresponding target values, explain how the parameters of the model can be estimated either in a batch or in a sequential mode.

Stochastic gradient descent algorithm:

$$\widehat{w} \leftarrow \widehat{w} - \eta E_n\,,$$

with η the learning rate parameter.

Therefore:

$$\widehat{w} \leftarrow \widehat{w} + \eta[t_n - \widehat{w}^T \varphi(x_n)]\ \varphi(x_n)$$

Algorithm converges for suitable small values of η.

EXERCISE 4

Consider a regression problem for the target function $f:\ R^8 \rightarrow R^4$. Design a solution based on Artificial Neural Network for this problem: draw a layout of a suitable ANN for this problem and discuss the choices.

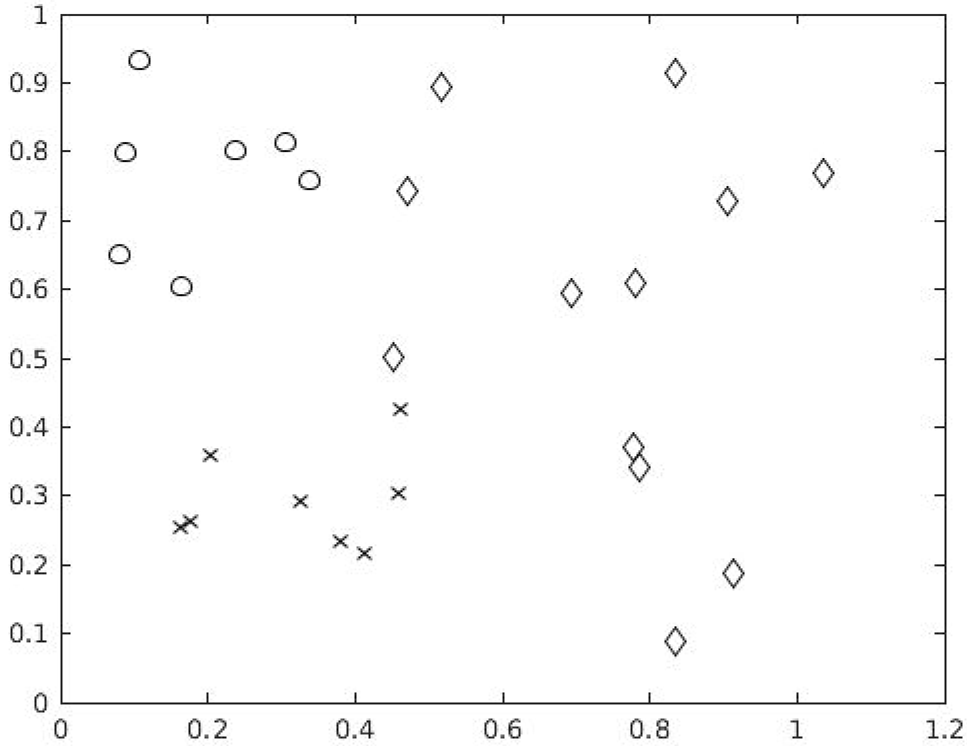
1. Determine the size of the ANN model (i.e., the number of unknown parameters).

2. Is Backpropagation algorithm affected by local minima? If so, how can we avoid or attenuate it?

3. Is Backpropagation algorithm affected by overfitting? If so, how can we avoid or attenuate it?

EXERCISE 5

Consider the data shown in the figure below:



Considering classification based on support vector machines (SVMs):

1. Explain if the data are separable and motivate your answer (only ‘yes’ or ‘no’ are not acceptable answers).

2. Explain what type of kernel function you would use in this case.

3. Describe what are the possible solutions for applying SVMs for classification of multiple classes.

EXERCISE 6

1. Describe the perceptron model for classification and its training rule.

2. Draw a graphical representation of a 2D data set for binary classification and provide a qualitative graphical example of a possible evolution of perceptron training (4 images showing a possible temporal evolution of the solution of the algorithm on the sketched data set).

EXAM - FEBRUARY 2019

EXERCISE 1

1. Provide a formal definition of the Reinforcement Learning (RL) problem. Describe formally what are the inputs and the outputs of a RL algorithm.
2. Describe the main steps of a RL algorithm. Provide an abstract pseudo-code of a generic algorithm for RL (e.g., Q-learning).

EXERCISE 2

Describe two different methods to overcome overfitting in Convolutional Neural Networks (CNN).

There are several ways to overcome overfitting in CNN such as implementing dropout, batch normalization and regularization.

EXERCISE 3

1. Describe the principle of maximal margin used by SVM classifiers. Illustrate the concept with a geometric example.
2. Draw a linearly separable dataset for binary classification of 2D samples. Draw two solutions (i.e., two separation lines): one corresponding to the maximum margin, the other one can be any other solution.
3. Discuss why the maximum margin solution is preferred for the classification problem.

EXERCISE 4

1. Provide the definition of Confusion matrix for a multi-class classification problem.
2. Provide a numerical example of a confusion matrix for a 3-classes classification problem with a balanced data set including 100 samples for each class. Show the confusion matrix in two formats: with absolute values and with the corresponding percentage values.
3. Compute the accuracy of the classifier for the numerical example provided above.

Hint: use simple numerical values, so that you do not need to make complex calculations.

EXERCISE 5

Given an unsupervised dataset $D = \{x_n\}$

1. Define the Gaussian Mixture Model (GMM) and describe the parameters of the model.

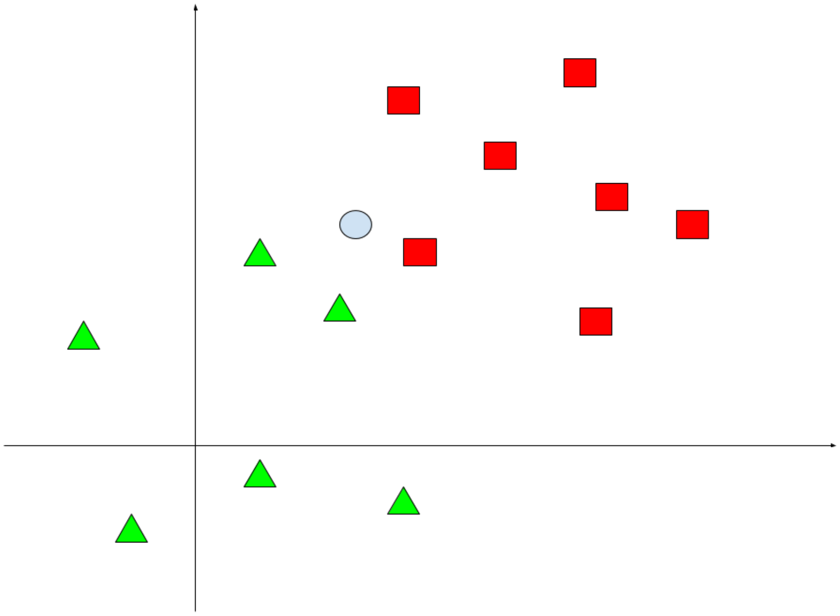
2. Draw an example of a 2D data set (i.e., $D \subset R^2$) generated by a GMM with K = 3, qualitatively showing in the picture also the parameters of the model.

3. Determine the size of the model (i.e., number of independent parameters) for the dataset illustrated above.

EXERCISE 6

1. Describe the K-nearest neighbors (K-NN) algorithm for classification.

2. Given the dataset below for the two classes *{square, triangle}*, determine the answers of K-NN for the query point indicated with symbol o for K=1, K=3, and K=5. Motivate your answer, showing (with a graphical drawing) which instances contribute to the solution.



EXAM - JUNE 2019

EXERCISE 1

1. Describe with pseudo-code the K-Fold Cross Validation method to estimate the accuracy of a learning algorithm L on a dataset D.

```
- Partition data set D into k disjoint sets $S_1, S_2, \dots, S_k$ ($|S_i| > 30$)
- For $i = 1, \dots, k$ do
    - use $S_i$ as test set, and the remaining data as training set $T_i$
    - $T_i \leftarrow \{D - S_i\}$
    - $h_i \leftarrow L(T_i)$
    - $\delta_i \leftarrow errors_{S_i}(h_i)$
- Return
```

$$error_{L,D} = \frac{1}{k} \sum_{i=1}^{k} \delta_i$$

Note: $accuracy_{L,D} = 1 - error_{L,D}$

2.Describe how the method can be extended to comparing two different learning algorithms L_A, L_B.

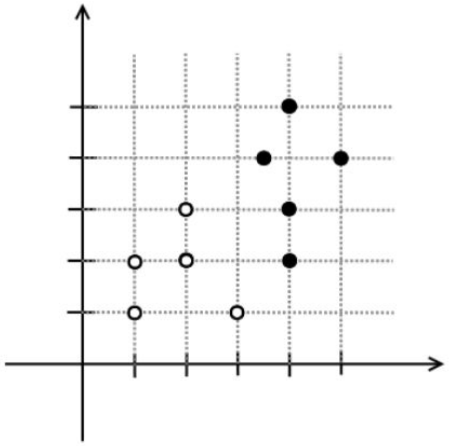
```
- Partition data set D into k disjoint sets $S_1, S_2, \dots, S_k$ ($|S_i| > 30$)
- For $i = 1, \dots, k$ do
    - use $S_i$ as test set, and the remaining data as training set $T_i$
    - $T_i \leftarrow \{D - S_i\}$
    - $h_A \leftarrow L_A(T_i)$
    - $h_B \leftarrow L_B(T_i)$
    - $\delta_i \leftarrow errors_{S_i}(h_A) - error_{S_i}(h_B)$
- Return
```

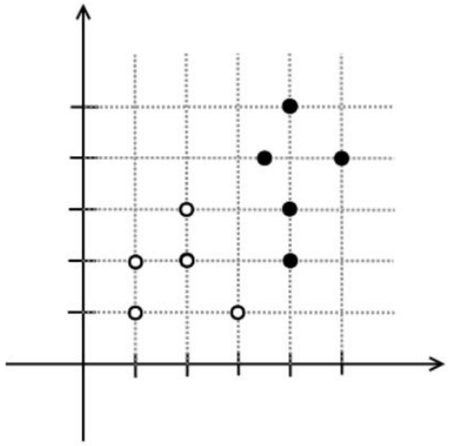
$$\delta = \frac{1}{k} \sum_{i=1}^{k} \delta_i$$

Note: if $\bar{\delta} < 0$ we can estimate that L_A is better than L_B,

EXERCISE 2

Consider the following data set for binary classification, where the two classes are represented with white and black circles.





1. Draw in each of the diagrams below a possible solution for a method based on Perception with very small learning rate and a possible solution for a method based on SVM.
2. Describe the difference between the two solutions and briefly explain how these are obtained with the two methods.
3. Discuss which solution would you prefer and why.

EXERCISE 3

1. Describe the k-armed bandit problem (also known as One-state MDP).
2. Describe the Reinforcement Learning to compute the optimal policy in the k-armed bandit problem.

EXERCISE 4

Given a dataset D for classification problem with classes $\{C_1, ..., C_n\}$.

1. Describe the difference between generative and discriminative probabilistic models for classification.

   Generative: estimate $P(C_i \mid x)$ through $P(x \mid C_i)$ and Bayes theorem

Consider first the case of two classes

Find the conditional probability

$$P(C_1| x) = \frac{P(x \mid C_1)P(C_1)}{P(x \mid C_1)P(C_1)+P(x \mid C_2)P(C_2)} = \frac{1}{1+exp(-a)} = \sigma(a)$$

with

$$a = ln\frac{p(x|C_1)P(C_1)}{p(x|C_2)P(C_2)}$$

and

$$\sigma(a) = \frac{1}{1+exp(-a)} \text{ the sigmoid function}$$

Discriminative: estimate $P(C_i \mid x)$ directly from model

Logistic regression is a classification method based on maximum likelihood.

Given data set D, consider $\{x_n, t_n\}$, with $t_n \in \{0,\ 1\},\ n = 1,\ \ldots,\ N$

Likelihood function

$$p(t \mid w) \ = \prod_{n=1}^{N} y_n^{\,t_n}(1 - y_n)^{1-t_n}$$

with $y_n = p(C_1|x_n) = \sigma(w^T x_n)$

2. Draw a 2D dataset for a binary classification problem and show (also in a graphical form) a possible solution using a probabilistic generative model.

EXERCISE 5

1. Describe the convolution stage of a Convolutional Neural Network (CNN).
2. Describe the properties of sparse connectivity and parameter sharing for CNN.

EXERCISE 6

Machine learning problems can be categorized in supervised and unsupervised.

1. Explain the difference between them providing a precise formal definition (only showing an explanatory text will not be enough) 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.

EXAM - OCTOBER 2019

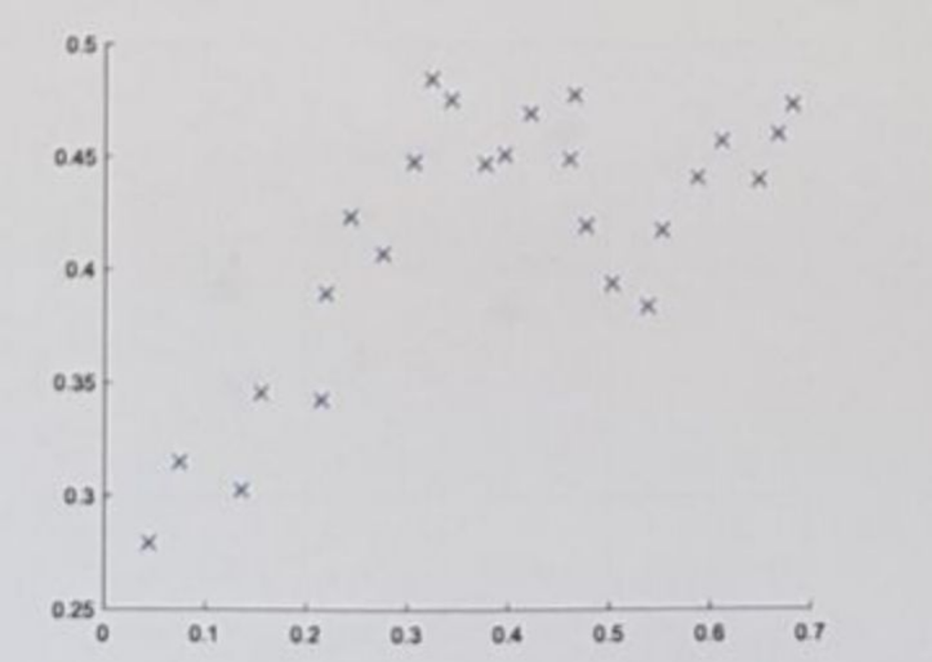
EXERCISE 1

1. Provide the definition of *Confusion matrix* for a multi-class classification problem.
2. Provide a numerical example of a confusion matrix for a 3-classes classification problem with a balanced data set including 100 samples for each class (300 samples in total). Show the confusion matrix in two formats: with absolute values and with the corresponding percentage values.
3. Compute the accuracy of the classifier for the numerical example provided above.

Hint: use simple numerical values, so that you don't need to make complex calculations.

EXERCISE 2

Consider the learning problem of estimating the function $f : \Re \rightarrow \Re$ with dataset D plotted in the figure below:



1. Describe how to perform regression based on these data using a method of your choice. Specifically, provide a mathematical formulation of the model, highlighting the model parameters.
2. Consider the method you have chosen to describe a way to reduce overfitting.
3. Draw a plausible plot of the learned model based on your choices.

EXERCISE 3

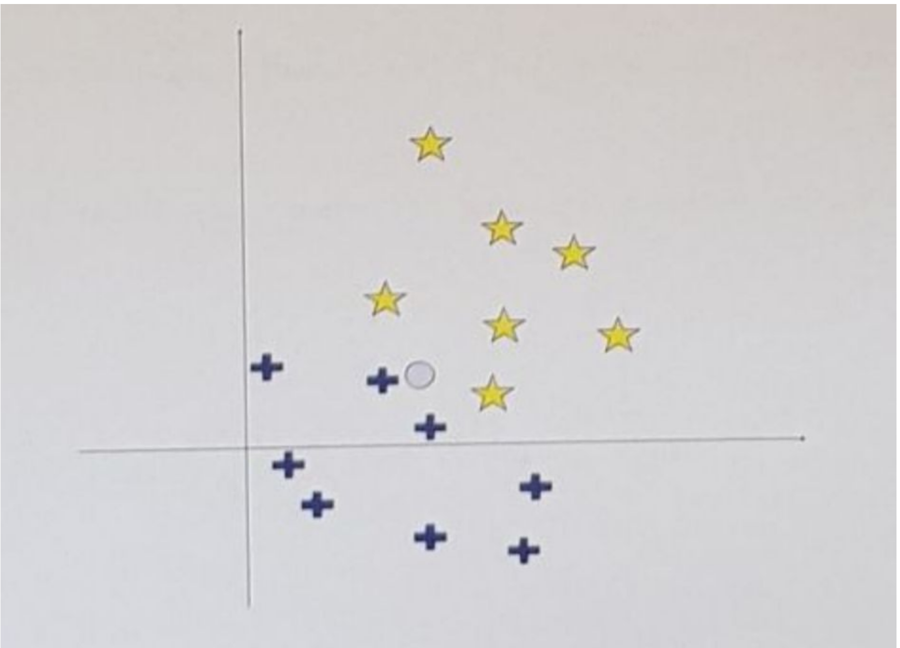
1. Describe the perceptron model for classification.
2. Describe the perceptron training algorithm.
3. Discuss convergence properties of perceptron training algorithm.

EXERCISE 4

1. Qualitatively explain the maximum margin principle on which Linear SVM classification is based.
2. Draw an example of 2-D binary classification problem solving:
   a. A generic margin separating points belonging to two different classes,
   b. Margin identified by Linear SVM.

EXERCISE 5

1. Describe the K-nearest neighbors (K-NN) algorithm for classification.
2. Given the dataset below for the two classes *{star, plus},* determine the answer of K-NN for the query point indicated with symbol o for K=1, K=3, and K=5. Motivate your answer, showing which instances contribute the solution.



EXERCISE 6

1. Describe the general approach of boosting.
2. Assume you have an image classifier with low classification accuracy. Provide the main steps for achieving higher classification accuracy by combining multiple instances of the classifier.

EXAM - JANUARY 2020

EXERCISE A1

Assume the following data about an online shop have been collected:

- Customers are: 25% young men (class Y M); 45% young women (Y W); 30% neither of the above (O).
- Young men buy: Shoes 30%; Trousers 50%; Shirts 20%.
- Young women buy: Shoes 50%; Trousers 30%; Shirts 20%.
- Other customers buy: Shoes 30%; Trousers 30%; Shirts 40%.

1. If you receive an order for trousers, which is the most probable class the customer who issued the order belongs to? Why?

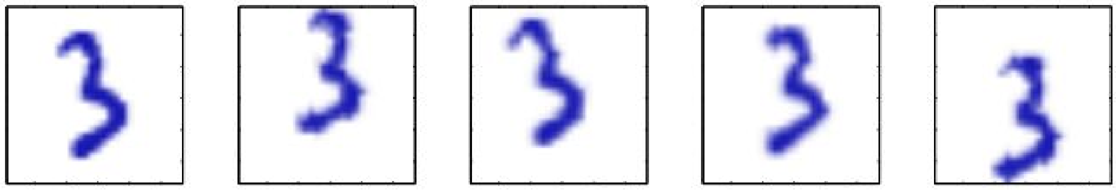
2. Which is, and how do you compute, the likelihood that an order is for trousers?

EXERCISE A2

1. Explain when a dataset is linearly separable
2. Draw an example of a linearly separable dataset in a 2D setting, with two classes $C = \{+, -\}$
3. Draw an example of a non linearly separable dataset in a 2D setting, with two classes $C = \{+, -\}$
4. For each dataset shown above, draw a possible solution based on SVM and explain how it can be obtained.

EXERCISE B1

Consider the following data $x_1, ..., x_N$ where the intrinsic dimensions are described in terms of a 2D translation and rotation (3 parameters) and the set of principal components $u_1, ..., u_M$ recovered from this data.

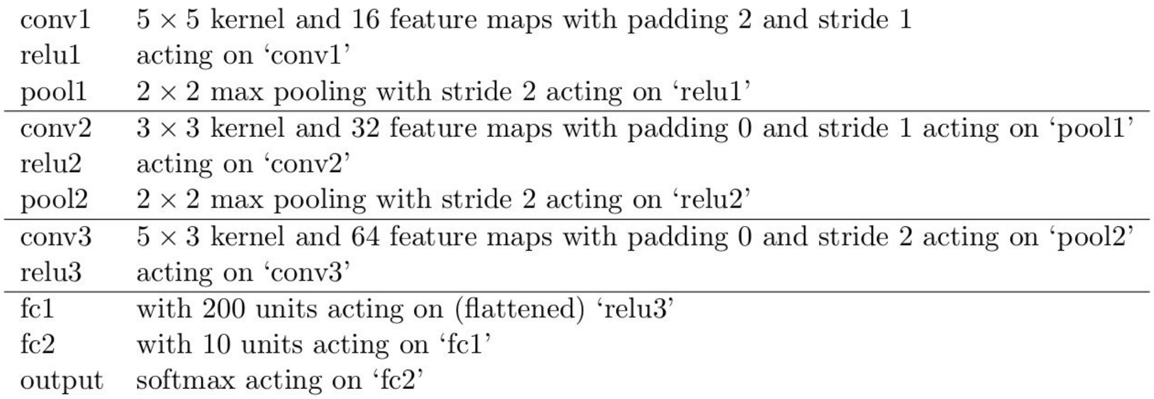


- How can these points be expressed in the basis defined by the principal components? Provide the relative formula.
- Is PCA able to recover a 3 dimensional space that fully describes the data (apart from noise)? Explain your answer.

  Yes, because even if in theory there are as many dimension as there are pixels in the image (assuming black/white), in practice you can fully recover every image using 3 dimensions: x, y of the center and an angle of rotation because images are all the same, so you need no extra info about the image itself, only about its position

EXERCISE B2

Consider the following Convolutional Neural Network acting on images of dimension 32 × 32 × 3:



conv1	5×5 kernel and 16 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 32 feature maps with padding 0 and stride 1 acting on 'pool1'
relu2	acting on 'conv2'
pool2	2×2 max pooling with stride 2 acting on 'relu2'
conv3	5×3 kernel and 64 feature maps with padding 0 and stride 2 acting on 'pool2'
relu3	acting on 'conv3'
fc1	with 200 units acting on (flattened) 'relu3'
fc2	with 10 units acting on 'fc1'
output	softmax acting on 'fc2'

1. Compute the number of parameters for each layer of the network.

2. What is a suitable loss function to train the network defined above?

EXERCISE C1

Consider the dataset $D = \{(x_1{}^T, t_1), \dots, (x_N{}^T, t_N)\}$ where each tuple (x_n, t_n) corresponds to an input value $x_I \in R^3$ and the corresponding target value $t_I \in R^3$.

1. Provide the definition of a linear regression model (in its most general form) with parameters ***w*** that can be used for estimating a nonlinear function y such that $t \approx y(x, w)$.

2. Provide a suitable loss function and sketch an algorithm for estimating the parameters of the model.

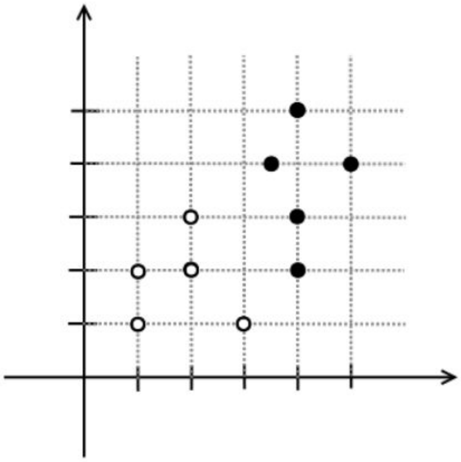
EXERCISE C2

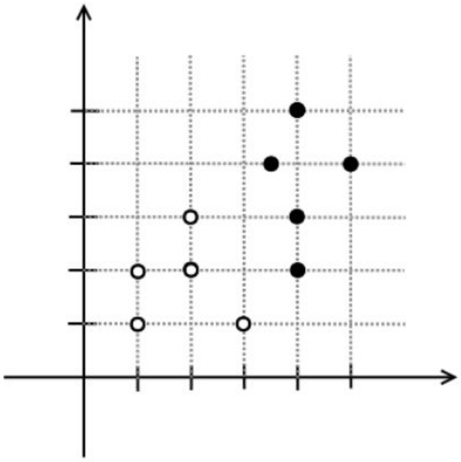
Consider the following data set for binary classification (white vs black circles).

1. Draw in each of the diagrams below a possible solution for a method based on Perceptron with very small learning rate and a possible solution for a method based on SVM.

2. Describe the difference between the two solutions and briefly explain how these are obtained with the two methods.

3. Discuss which solution would you prefer and why.





EXAM - FEBRUARY 2020 A

EXERCISE A1

1. Explain the difference between regression and classification.

The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete).

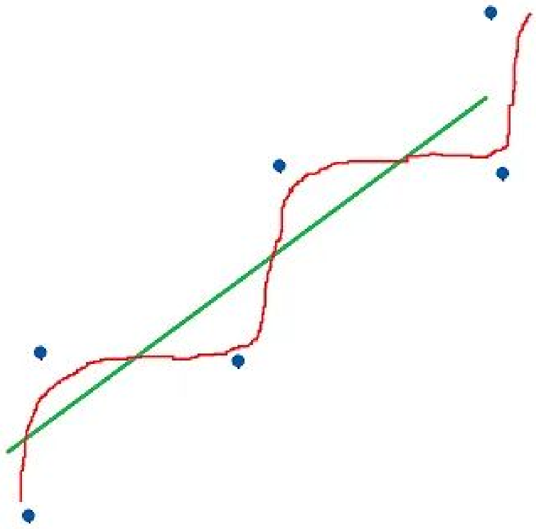
2. Provide a mathematical formulation of linear regression.

Linear model for linear function

$$y(x;\ w)\ =\ w_0 + w_1x_1 + w_2x_2 +\ ...\ w_Nx_N = w^Tx$$

3. Provide an example of a linear regression model that overfits a dataset of your choice, and discuss how this can be mitigated.

The image below illustrates an overfit model. The green line represents the true relationship between the variables. The random error inherent in the data causes the data points to fall randomly around the green fit line. The red line represents an overfit model. This model is too complex, and it attempts to explain the random error present in the data.



EXERCISE A2

1. Define mathematically the problem solved by logistic regression

First of all, let’s mention that logistic regression is a classification method based on maximum likelihood. Given data set *D,* consider $\{x_n,\ t_n\}$, with $t_n \in \{0,\ 1\},\ n\ =\ 1,\ ...,\ N$

Likelihood function:

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

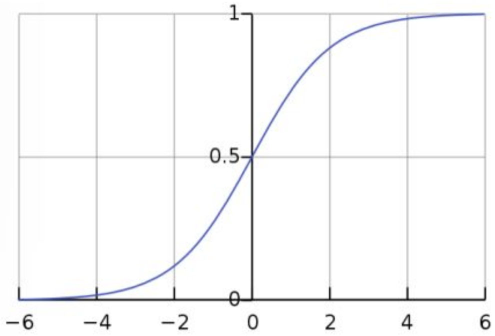
with $y_n = p(C_1|\,x_n) = \ \sigma(w^T x_n)$

Solve the optimization problem

$$w^* = argmax\ E(w)$$

2. Consider the following dataset and the sigmoid function:

x_1	x_2	x_3	t
0	0	1	1
1	2	3	1
4	4	1	0



Which one among the following solutions fits the data better? Why?

$$\overline{w_1{}^T = (1,\ 0,\ -1)} \qquad\qquad \overline{w_2{}^T = (-1,\ -1,\ 2)}$$

A plot of the sigmoid function is reported above. You do not need to compute explicit values of the model.

In order to solve this problem, we need to check the formula for linear regression which is

$$y(x;\ w)\ \ =\ \ w_0 + w_1x_1 + w_2x_2 +\ \ ...\ w_Nx_N = w^T x$$

So, we can find the result which would be

$$r\ \ =\ \ w_1x_1 + w_2x_2 + w_3x_3 = 0 * 1\ \ +\ \ 0\ \ * 0\ \ +\ \ 1\ \ *\ \ (-1) =- 1$$

As you see, it is not in the figure, so it is not fitting the data better. For the second case, we can find

$$r = w_1x_1 + w_2x_2 + w_3x_3 = 0 * (-1) + 0 * (-1) + 1 * 2 = 2$$

As you see, this one fits the data better than the first one.

EXERCISE B1

1. Give a short explanation of the kernel *trick/kernel substitution*. What is the necessary condition for applying the kernel trick?

If input vector x appears in an algorithm only in the form of an inner product x T x 0 , replace the inner product with some kernel k(x T , x 0).

- Can be applied to any x (even infinite size)
- No need to know φ(x)
- Directly extend many well-known algorithms

2. Provide an example of its application. In detail:

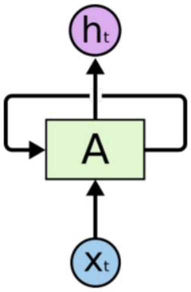
- draw a suitable dataset for binary classification in 2D;
- discuss which kernel you would use for this dataset;
- show graphically a possible solution of such a kernel-based model.

EXERCISE B2

Consider the structure of a recurrent neural network (RNN):

1. Design a generic RNN model (or give the relative formula).

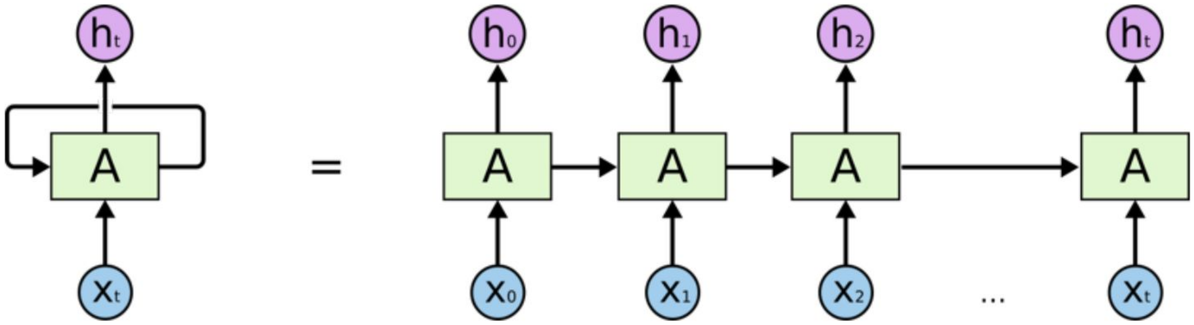
In RNNs we are dealing with sequences where the data points are related and order is important. However, CNNs disregard this information.



2. Explain the concept of ‘unfolding’ (or ‘unrolling’) an RNN.

Another way to see RNN is to unfold the loop

- Each cell dedicated to different data sample
- Output h_t computed based also on h_{t-1}



3. For what type of input would you use an RNN? Describe a specific use case of your choice providing details both for the input and output of the RNN.

EXERCISE C1

1. Describe the difference between supervised learning and reinforcement learning with a formal definition of the two problems.

In supervised learning you are learning a function $f : X \rightarrow Y$, *given*

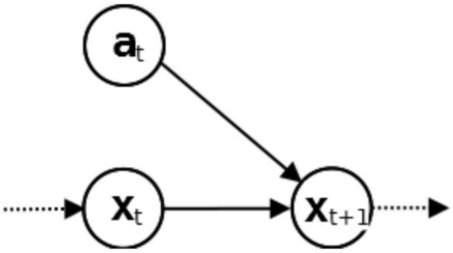
$$D = \{(x_n,\ t_n)\}$$

However, in reinforcement learning you are learning a behavior function $\pi :\ X \rightarrow A$, given

$$D = \{< x_1, a_1,\ r_1, ...,\ x_n,\ a_n,\ r_n >^{(i)}\}$$

2. Describe the full observability property of Markov Decision Processes and its relation with non-deterministic outcomes of actions.

In Markov Decision Processes(MDPs) states are fully observable and we don't need observations. Graphical notation/representation is



In non-deterministic transitions

$$MDP =< X, A,\ \delta,\ r >$$

where

- **X** is a finite set of states
- **A** is a finite set of actions
- $\delta : X \times A \rightarrow 2^X$ is a transition function
- $r \times X \times A \times X \rightarrow R$ is a reward function

EXERCISE C2

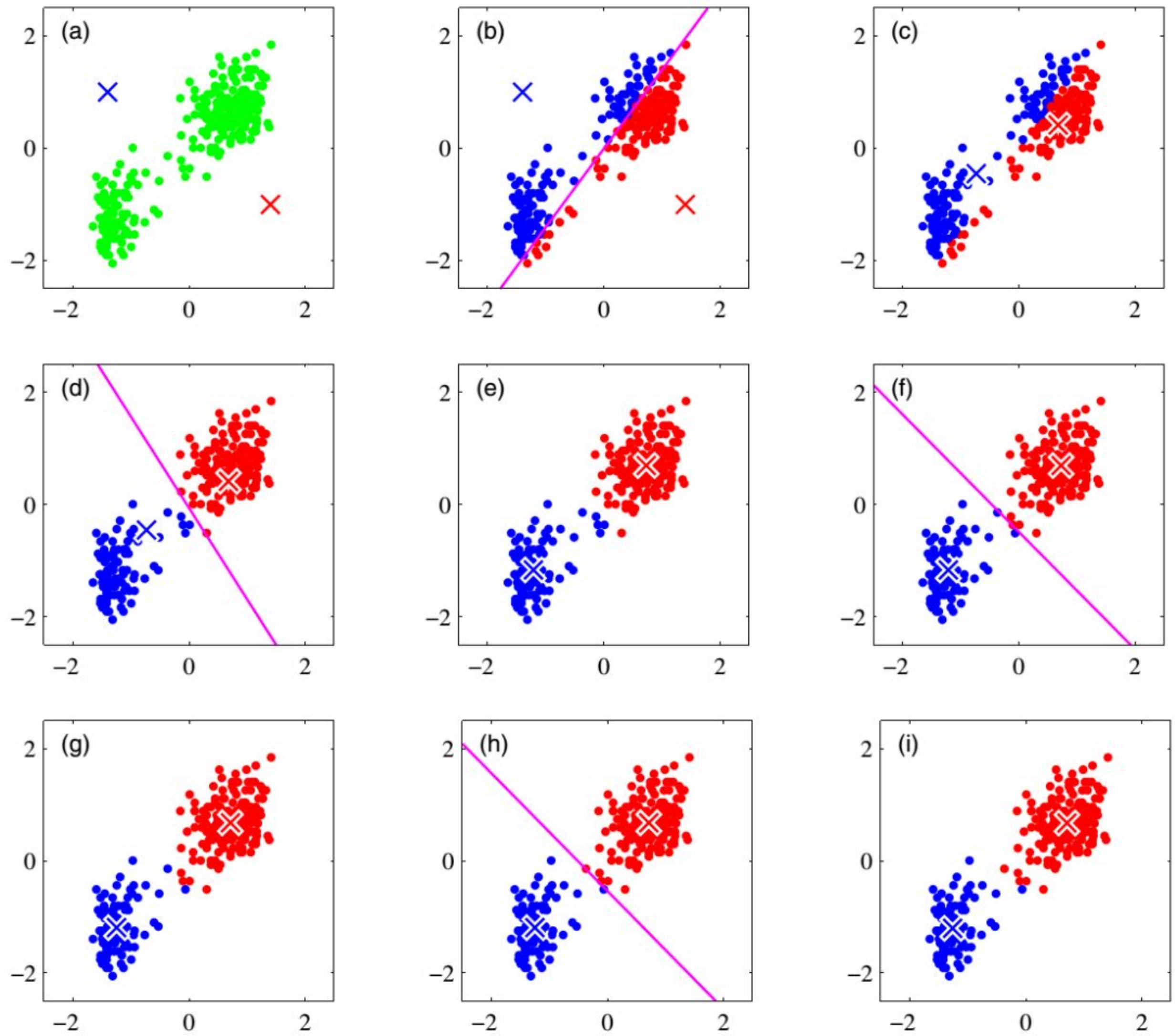
1. Describe the K-means algorithm in a formal way (i.e., with precise mathematical formulas and equations), including: input and output of the algorithm, its main steps, and the termination condition.

Computing K means of data generated from K Gaussian distributions.

$$Input:\ D\ =\ \{x_n\},\ value\ K \qquad Output:\ \mu_1, ..., \mu_k$$

5. Begin with a decision on the value of k = *number of clusters*
6. Put any initial partition that classifies the data into *k* clusters. You may assign the training samples randomly, or systematically as follows
    a. Take the first k training samples as single-element clusters
    b. Assign each of the remaining training samples to the cluster with the nearest centroid. After each assignment, recompute the centroid of the new cluster.
7. Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the two clusters involved in the switch
8. Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

2. Draw a suitable 2-D data set for K-means.

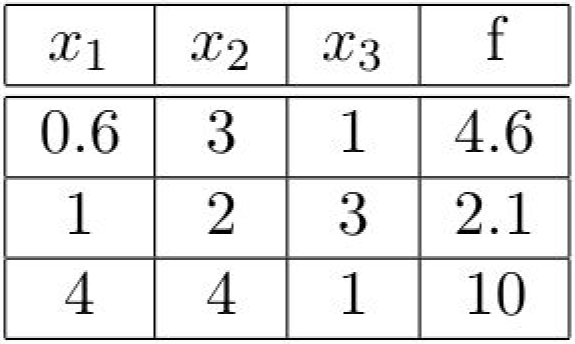


3. Simulate the execution of K-means in such 2-D data, showing at least three steps of the algorithm and the final output.

EXAM - FEBRUARY 2020 B

EXERCISE A1

Assume you are given the following dataset, representing the samples of a function f :



x_1	x_2	x_3	f
0.6	3	1	4.6
1	2	3	2.1
4	4	1	10

1. Which technique would you use to estimate f ?

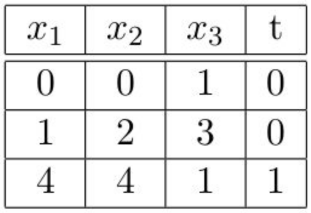
2. Provide a mathematical formulation of the problem solved by the chosen technique.

3. Provide an example of a solution using a simple dataset of your choice. (Show the solution only, you don’t have to illustrate the steps followed to obtain it).

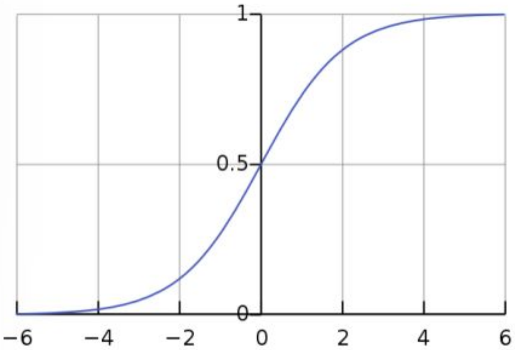
EXERCISE A2

1. Define mathematically the problem solved by logistic regression

2. Consider the following dataset and the sigmoid function:



x_1	x_2	x_3	t
0	0	1	0
1	2	3	0
4	4	1	1



Which one among the following solutions fits the data better? Why?

$$\overline{w_1{}^T = (2,\ 0,\ -2)} \qquad\qquad \overline{w_2{}^T = (-2,\ -2,\ 4)}$$

A plot of the sigmoid function is reported above. You do not need to compute explicit values of the model.

In order to solve this problem, we need to check the formula for linear regression which is

$$y(x;\ w)\ =\ w_0 + w_1x_1 + w_2x_2 +\ ...\ w_Nx_N = w^Tx$$

So, we can find the result which would be

$$r\ =\ w_1x_1 + w_2x_2 + w_3x_3 = 0 * 2\ +\ 0\ * 0\ +\ 1\ *\ (-2) =- 2$$

As you see, it is not in the figure, so it is not fitting the data better. For the second case, we can find

$$r = w_1x_1 + w_2x_2 + w_3x_3 = 0 * (-2)\ +\ 0 *\ (-2)\ +\ 1\ *\ 4\ =\ 4$$

As you see, this one fits the data better than the first one.

EXERCISE B1

1. Explain what properties a kernel function should typically satisfy.

Definition

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

Typically ***k*** is:

1. **symmetric**: $k(x,\ x') = k(x',\ x)$
2. **non-negative**: $k(x,\ x') \geq 0$.

2. Indicate which of the following kernel functions are not valid explaining why:

(a) $k(x,\ x')\ =\ 1;$

(b) $k(x,\ x')\ =\ (x^Tx'\ +\ \gamma)^4;$

(c) $k(x,\ x')\ =\ \sum_i\ [sin(x_i)\ -\ sin(x_i')];$

(d) $k(x,\ x')\ =\ \sum_i - log(x_i)\ log(\frac{x_i'}{x_i}),\ with\ x_i,\ x_i'\ >\ 0\, for\ all\ i;$

(e) $k(x,\ x')\ =\ 1\ -\ \frac{|x^Tx'|}{||x||\ ||x'||};$

EXERCISE B2

Consider the problem of finding a function which describes how the salary of a person (in hundreds of euros) depends on his/her age (in years), the months in higher education and average grades in higher education. A dataset in the form $D = \{(x_1{}^T,\ t_1),\ \dots,(x_N{}^T,\ t_N)\}$ is provided, with x $\in$ R^3 denoting the input values and $t \in R$ the target values (salary). Assuming that one tries to estimate 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 θ.
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.
4. Assuming that the gradients of the loss with respect to the parameters are available, describe an algorithm for training the parameters of the network. What are the hyper-parameters of the training algorithm (if any)?

EXERCISE C1

1. Describe the Markov Decision Process (MDP) model used in reinforcement learning, provide its mathematical formulation, and explain the elements of the model.

MDPs are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The agent and the environment interact continually, the agent selecting actions and the environment responding to these actions and presenting new situations to the agent. Formally, an MDP is used to describe an environment for reinforcement learning, where the environment is fully observable. Almost all RL problems can be formalized as MDPs.

In deterministic transitions

$$MDP =< X, A,\ \delta,\ r >$$

where

- **X** is a finite set of states

- **A** is a finite set of actions
- $\delta : X \times A \rightarrow X$ is a transition function
- $r \times X \times A \rightarrow R$ is a reward function

Given an MDP, we want to find an optimal policy. Policy is a function

$$\pi : X \rightarrow A$$

Value function can be found as:

$$V^{\pi}(x) = r_1 + \gamma r_2 + \gamma r_3 + ..$$

2. Describe the Q-learning algorithm, referring to the mathematical formulation of the MDP given above.

$Q^{\pi}(x,\ a)$: expected value when executing a in the state **x** and then act according to π.

$$Q^{\pi}(x,\ a) \equiv r(x,\ a)\ +\ \gamma V^{\pi}(\delta(x, a))$$

$$Q^{\pi}(x,\ a) \equiv \sum_{x'} P(x' \mid x,\ a)\ (r(x,\ a,\ x')\ +\ \gamma V^{\pi}(x))$$

If the agent learns *Q,* then it can determine the optimal policy without knowing δ *and* r.

$$\pi^*(x) = argmax\ Q(x,\ a)$$

EXERCISE C2

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.

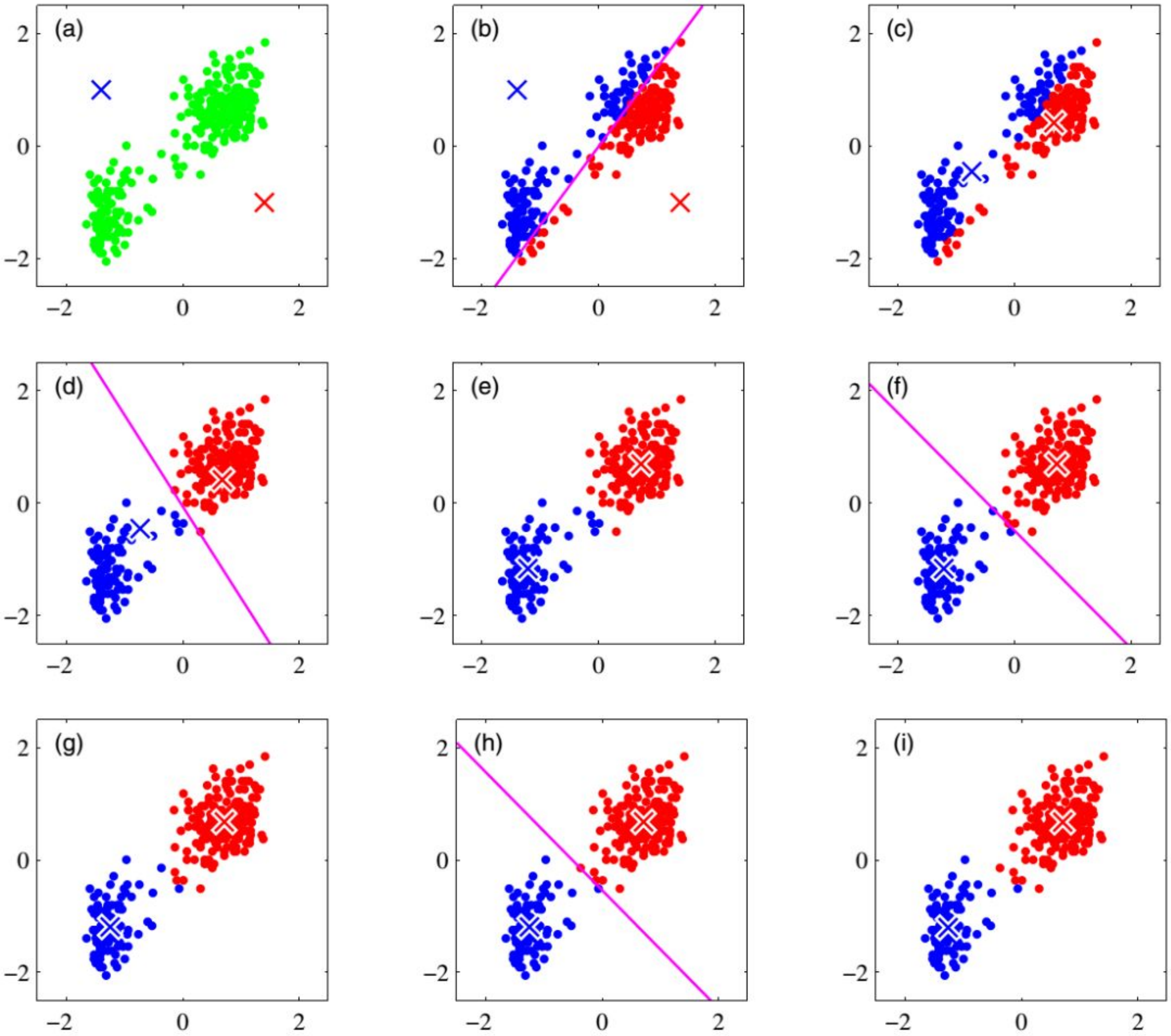
Computing K means of data generated from K Gaussian distributions.

$$Input:\ D\ =\ \{x_n\},\ value\ K \qquad Output:\ \mu_1, ..., \mu_k$$

9. Begin with a decision on the value of k = *number of clusters*
10. Put any initial partition that classifies the data into *k* clusters. You may assign the training samples randomly, or systematically as follows
    a. Take the first k training samples as single-element clusters
    b. Assign each of the remaining training samples to the cluster with the nearest centroid. After each assignment, recompute the centroid of the new cluster.

11. Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the two clusters involved in the switch
12. Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

2. Draw a suitable 2-D data set for K-means.



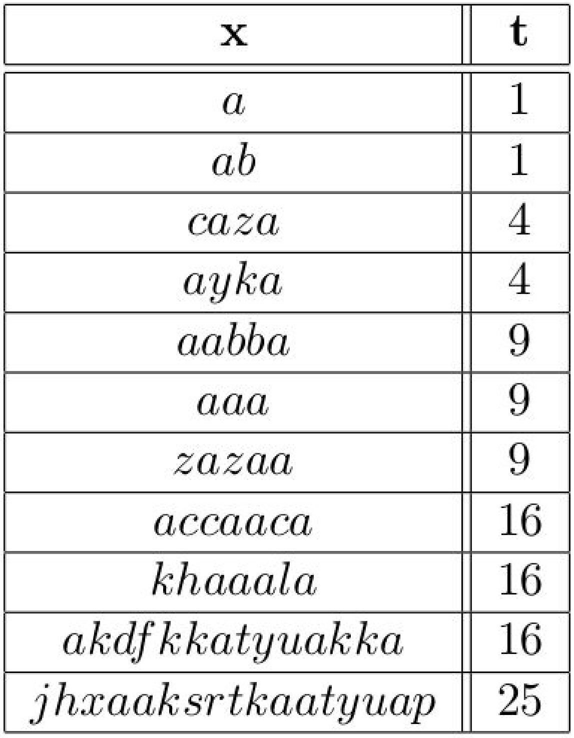
3. Simulate the execution of K-means in such 2-D data, showing at least three steps of the algorithm and the final output.

EXAM - June 2020

EXERCISE 1

Consider a setting where the input space I is the set of finite strings over the characters a, b, c, · · · , z. Notice that input strings can be of different length.

Given the following dataset D:



$\mathbf{x}$	$\mathbf{t}$
a	1
ab	1
$caza$	4
$ayka$	4
$aabba$	9
aaa	9
$zazaa$	9
$accaaca$	16
$khaaala$	16
$akdfkkatyuakka$	16
$jhxaaksrtkaatyuap$	25

1. Identify the learning problem at hand, in particular the form of the target function, and define a suitable linear model for it.

2. Apply the kernel trick to the model defined above and provide the analytical form of the corresponding error function.

3. Define the solution obtained with your choices for the dataset D.

EXERCISE 2

A secret string $s = b_0b_1b_2b_3$ of 4 bits fulfills the following constraints:

- if $b_0 = 0$, s contains an even number of 0's and 1's

- if $b_0 = 1$, s contains at least three 1's.

No other prior information about s is available.

1. Define the prior probability distribution P (s) of the hypothesis string s.

7 possible values $\rightarrow P(s) =< \frac{1}{7};\ \frac{1}{7};\ \frac{1}{7};\ \frac{1}{7};\ \frac{1}{7};\ \frac{1}{7};\ \frac{1}{7};>$

1) 0 0 0 1 prior probability distribution

2) 0 0 1 0

so it goes like this

2. Assuming $b_0 = 0$, define the conditional probability distribution $P(s \mid b_0 = 0)$ and indicate all maximum a-posteriori hypothesis.

$P(b_0 = 0 \mid s) * P(s) =< 1; 1; 1; 0; 0; 0; 0 >$

$P(b_0 = 0 \mid s) * P(s) =< \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; 0; 0; 0; 0 >$

$$P(s|b_0 = 0) = \frac{P(b_0=0|s)P(s)}{P(b_0=0} = \frac{7}{3} *< \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; 0; 0; 0; 0 >=<< \frac{1}{3}; \frac{1}{3}; \frac{1}{3}; 0; 0; 0; 0 >$$

$P(b_0 = 0 \mid s) = \sum P(b_0 = 0 \mid s)P(s) = \frac{1}{7} + \frac{1}{7} + \frac{1}{7} \ = \frac{3}{7}$

$h_{MAP} = argmax\ P(s|b_0 = 0) =< 0001, 0010, 0100 >$

3. Assuming $b_0 = 1$ and $b_1 = 1$, indicate all maximum likelihood hypotheses and compute the likelihood that $b_2 = 1$.

$P(b_0 = 1 \mid s) =< 0; 0; 0; 1; 1; 1; 1 >$

$P(b_0 = 1 \mid s) * P(s) =< 0; 0; 0; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7} >$

EXERCISE 3

Consider a dataset D for the binary classification problem $f:\ R^3 \rightarrow \{A,\ B\}$.

1. Describe a probabilistic generative model for such a classification problem, assuming Gaussian distributions.

3. Describe the difference between generative and discriminative probabilistic models for classification.

   Generative: estimate $P(C_i \mid x)$ through $P(x \mid C_i)$ and Bayes theorem

   Consider first the case of two classes

   Find the conditional probability

$$P(C_1|\, x) = \frac{P(x \mid C_1)P(C_1)}{P(x \mid C_1)P(C_1)+P(x \mid C_2)P(C_2)} = \frac{1}{1+exp(-a)} = \sigma(a)$$

with

$$a = ln\frac{p(x|C_1)P(C_1)}{p(x|C_2)P(C_2)}$$

and

$$\sigma(a) = \frac{1}{1+exp(-a)} \text{ the sigmoid function}$$

Discriminative: estimate $P(C_i \mid x)$ directly from model

Logistic regression is a classification method based on maximum likelihood.

Given data set D, consider $\{x_n, t_n\}$, with $t_n \in \{0,\ 1\},\ n = 1,\ \ldots,\ N$

Likelihood function

$$p(t \mid w)\ = \prod_{n=1}^{N} y_n^{\,t_n}(1-y_n)^{1-t_n}$$

with $y_n = p(C_1|x_n) = \sigma(w^T x_n)$

2. Identify the parameters of the model and determine the size of the model (i.e., the number of independent parameters).

EXERCISE 4

1. Describe the k-armed bandit problem (also known as One-state MDP).

2. Describe the Reinforcement Learning procedure to compute the optimal policy in the k-armed bandit problem with stochastic behavior and unknown functions.

EXERCISE 5

Consider that the output of layer l of a CNN is the set of feature maps M with size 256 × 256 × 64

1. What is the size of the feature maps N when max-pooling with a 2 × 2 kernel and stride 2 is applied on M ?

2. Design a convolutional layer which, when applied on M , produces feature maps with the same size as N . Describe all the relevant parameters of the layer you have designed.

3. What happens if the non-linear activation functions of the hidden layers of the CNN are replaced with linear functions? Is the effective depth of the network affected and how?

EXERCISE 6

Consider N convolutional neural networks trained to classify images of cats and dogs with a corresponding confidence value (output of sigmoid activation function)

- Describe a way to combine the predictions of the CNNs in order to get a single more accurate prediction.
- Assume N = 3, class '0' represents dogs, class '1' cats and for a given image the three CNN outputs are: (0.912, 0.432, 0.444). Apply the method described above to classify the image using the predictions of the three CNNs.