ML EXAMS 18/01/2024

EX 1

1.1

confusion matrix report how many times a class C_i is classified in class C_j

	C_1	C_2	← predicted class
C_1			
C_2			

true classes ←

in the main diagonal we have the

accuracy = # element of diagonal / # num of all element

the number of element in a row must be equal number of samples of for class C_j

1.2

T \ P	A	B	C
A	140	50	10
B	20	160	20
C	5	15	180

T \ P	A	B	C
A	70%	25%	5%
B	10%	80%	10%
C	2.5%	7.5%	80%

1.3

accuracy = 480/600 = 0.8 = 80%

EX. 2

2.1

① ② ③ ④

$D = \{(FFF, F), (FTT, T), (TTF, T), (TFT, T)\}$

$h_1 = (x_1 \wedge \neg x_2 \wedge x_3) \vee x_2$

$h_2 = (\neg x_1 \wedge x_2 \wedge x_3) \vee x_1$

an hypothesis is consistent if and only if given a dataset D an hypothesis h and a target concept c $h(x) = c(x)$ for each training sample $(x, c(x))$

$$\text{consist}(h, D) = (\forall x \in D)\ h(x) = c(x)$$

- check with h_1 ~~2222~~ ③

x_1	x_2	x_3	h_1
F	F	F	F
F	T	T	T
T	T	F	T
T	F	T	T

- check with h_2

x_1	x_2	x_3	h_2
F	F	F	F
F	T	T	T
T	T	F	T
T	F	T	T

both h_1 and h_2 are consistent

2.2

$P(D|h_1) = 0.6 \quad P(D|h_2) = 0.8 \quad P(h_1) = 0.2 \quad P(h_2) = 0.1$

posterior hypothesis $P(h_1|D) = P(D|h_1)P(h_1) = 0.6 \cdot 0.2 = 0.12$

// $P(h_2|D) = P(D|h_2)P(h_1) = 0.8 \cdot 0.1 = 0.08$

$\arg\max \{(0.12)(0.08)\} = 0.12 \rightarrow h_1 = h_{MAP}$

EX. 3

3.1

using a non linear functions of input variables

$$y(x,w) = \sum_{n=1}^{M} w_n \phi_n(x) \quad \text{with} \quad w_n = [w_1, \dots, w_M]^T$$

and $\phi_n = [\phi_0(x), \dots, \phi_M(x)]^M$ still linear in w

number of trainable params $= M+1$
↳ BIAS

3.2

target value is effected by additive noise ε so $t = y(x,w) + \varepsilon$

assuming gaussian noise $p(\varepsilon|\beta) = \mathcal{N}(\varepsilon|0, \beta^{-1})$ we have

$$p(t|w,x,\beta) = \mathcal{N}(t|y(x,w), \beta^{-1})$$

assuming distribution iid

$$p(\{t_1, \dots, t_n\}|x_1, \dots, x_N, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|w^T\phi(x_n)\beta^{-1}) =$$

$$= -\beta \frac{1}{2} \sum_{n=1}^{N} [t_n - w^T\phi(x_n)]^2 - \frac{N}{2} \ln(2\pi\beta^{-1})$$

maximum likelihood correspond

$$\arg\max_w p(\{t_1, \dots, t_n\}|x_1, \dots, x_N, w, \beta)$$

correspond least squared error

$$\arg\min_w E_D(w) = \arg\min \frac{1}{2} \sum_{n=1}^{N} [t_n - w^T\phi(x_n)]^2$$

a way to solve it is using a sequential algorithm

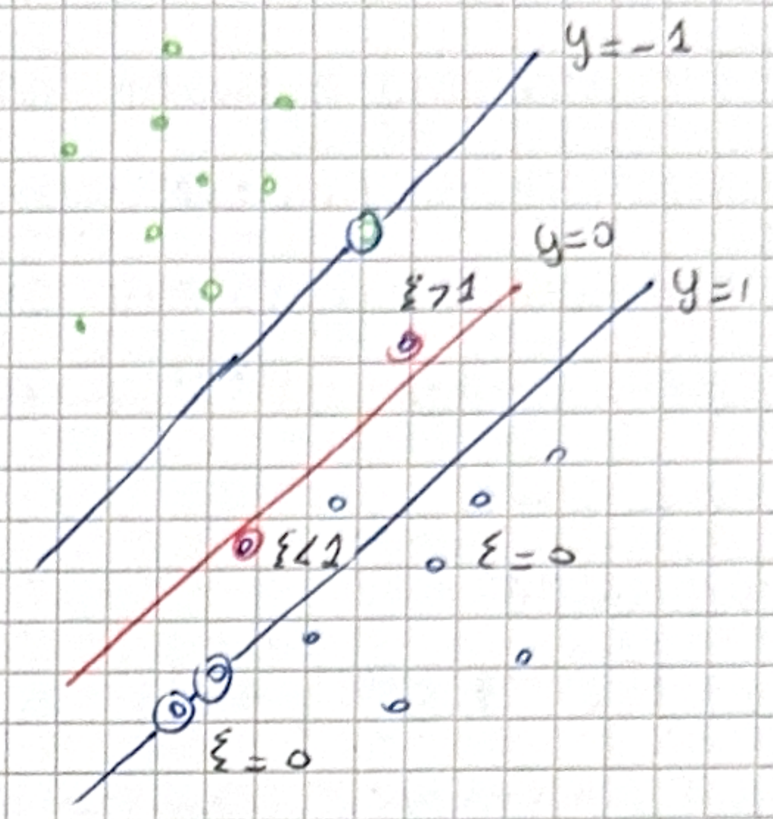
$$\hat{w} \leftarrow \hat{w} - \eta \nabla E_n \quad \Rightarrow \quad \hat{w} \leftarrow \hat{w} - \eta [t_n - w^T\phi(x_n)]\phi(x_n)$$

EX. 4

4.1

for hard SVM with soft margin constraints we introduce slack variable $\xi_n \geq 0$ with $n = 1, \dots, N$

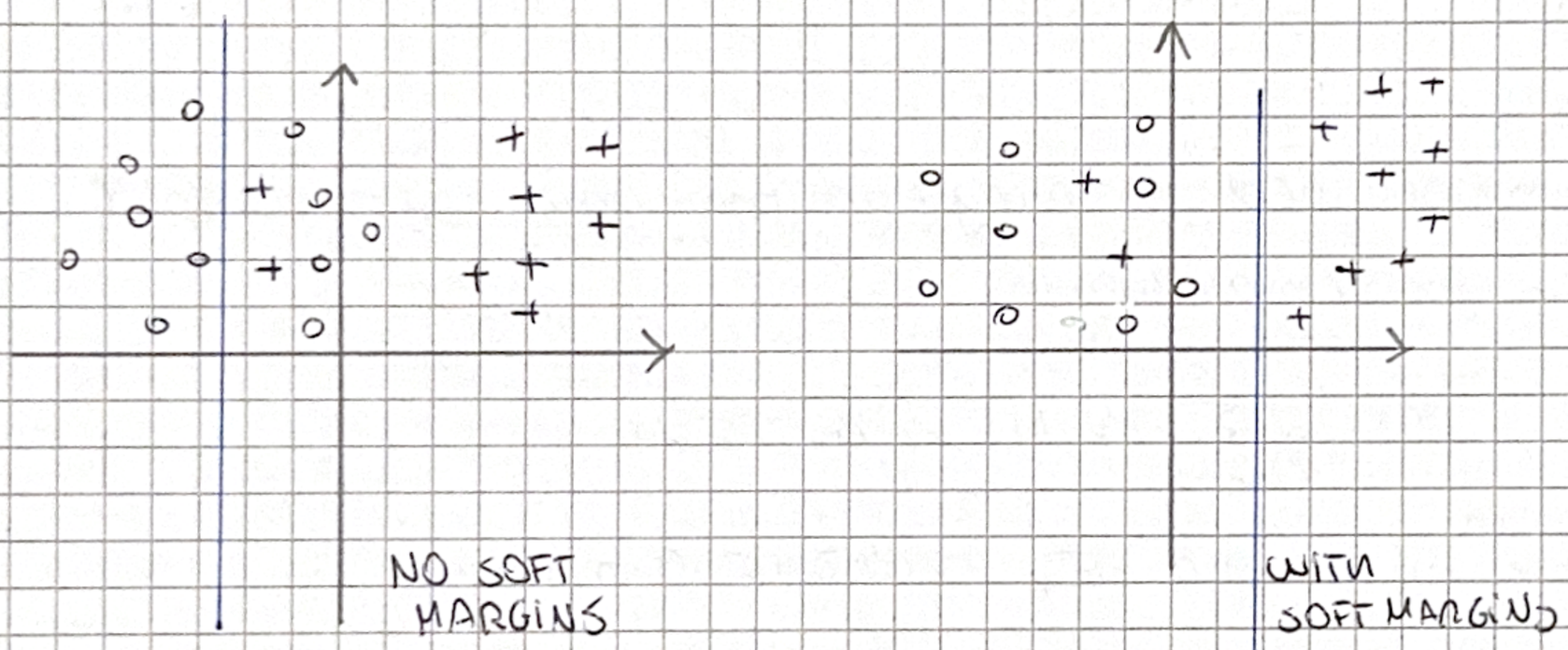
- $\xi = 0$ if point is ON or INSIDE the correct margin boundary
- $0 < \xi_n \leq 1$ if point is inside the margin but correct side
- $\xi_n > 1$ point in wrong side of boundary



optimization with soft constraints

$$w^*, w^0 = \arg\min \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N} \xi_n$$

4.2



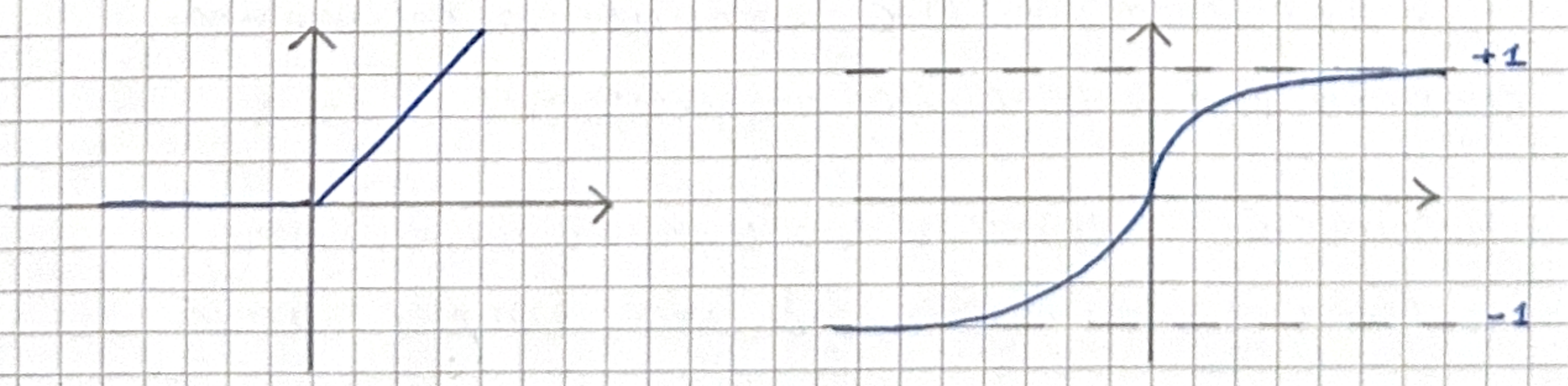
EX. 5

5.1

Is a regression task. for the activation function of the output layer I will use ~~IDENTY~~ IDENTITY ACTIVATION FUNCTION $y = W^T h + b_0$ and as loss function MSE mean square error $MSE = \frac{1}{N}\sum_{n=1}^{N}(t_n - x_n)^2$

5.2

we can use as activation function Relu ($g(\alpha) = \max(0, \alpha)$) or sigmoid and hyperbolic tangent $g(\alpha) = \sigma(\alpha)$ or $g(\alpha) = \tanh(\alpha)$



5.3

Require learning rate $\epsilon > 0$

require initial values of $\theta^{(1)}$

$k \leftarrow 1$

while stopping criterion not met DO:

sample ~~another~~ subset (minibatch) $\{x^{(1)}, \dots, x^{(m)}\}$

compute the gradient estimate $g = \frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)}, \theta^{(k)}), t^{(i)})$

apply update: $\theta^{(k+1)} \leftarrow \theta^{(k)} - \epsilon g$

$k \leftarrow k + 1$

end while

$\nabla_\theta L(f(x^{(i)}, \theta^{(k)}), t^{(i)})$ is obtain with BACKPROPAGATION

hyperparameter (m) → number of batch

ϵ → learning rate

if ϵ too small could not converge ϵ too large could diverge

EX. 6

6.1

K-MEANS is an unsupervised learning algorithm that try to compute

π_k, μ_k and Σ_k for compute

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

where π_k is prior probability μ_k the mean and Σ_k is covariance matrix

each instance x_n is generating by

1. choosing k gaussian according to prior $\pi_1, \dots \pi_k$
2. generating an instance at random according to that gaussian (μ_k, Σ_k)

PSEUDOCODE of KMEANS

1. begin a decision on the value k = number of cluster
2. put any partition of data into k clusters that can be done

systematically or randomly

I. select k sample as centroid of cluster

II. assign the remain N-k samples to the nearest centroid and recompute the centroide

3. take each sample in sequence and compute its distance from centroid of each clusters. if ~~sample~~ sample is not in

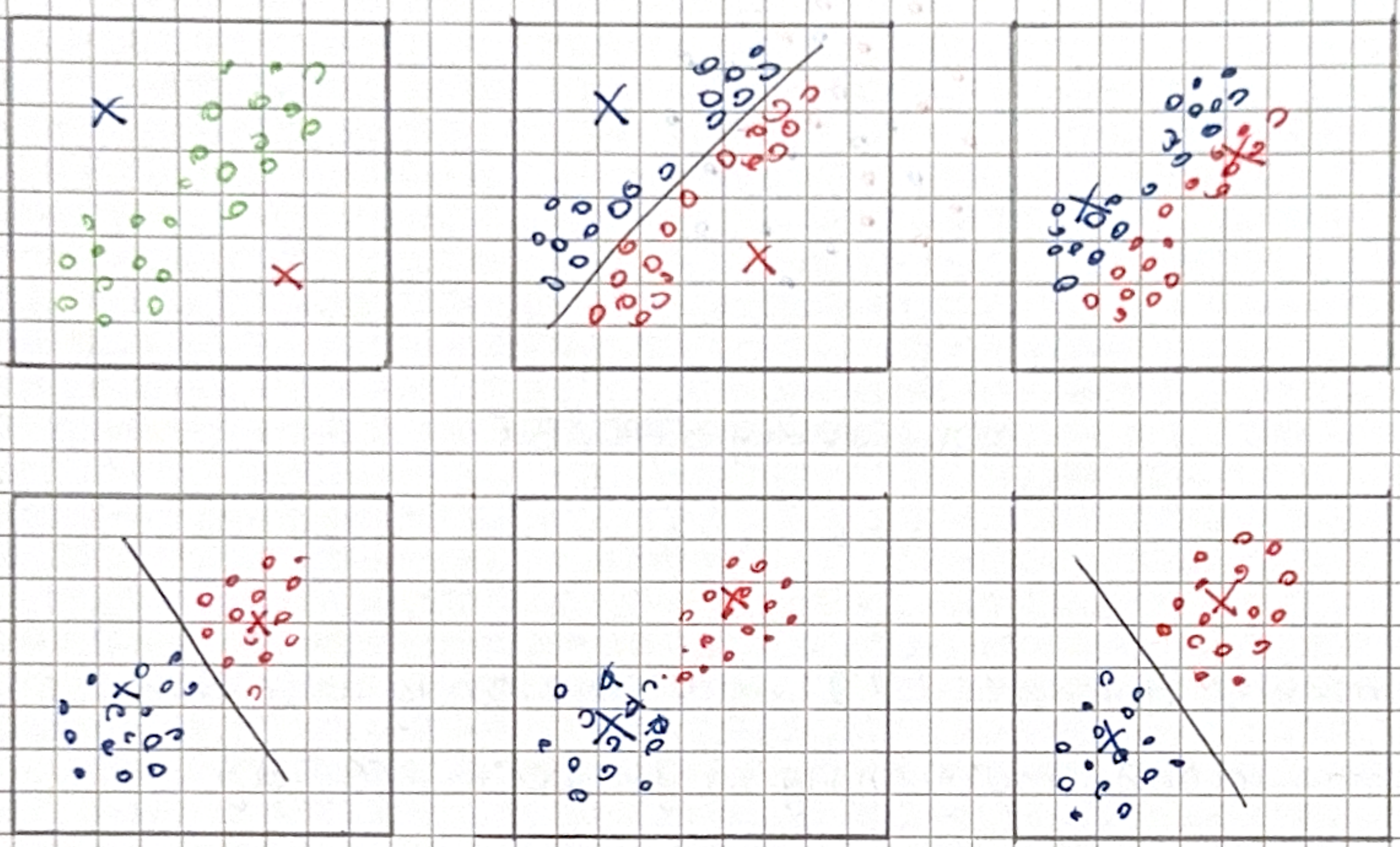
the nearest centroid cluster and recompute the centroids

4. repeat step 3 until convergence

termination: ~~[illegible]~~ the sum of distance from each training sample is decrease (from centroid)

there are only finitely partitions of training example into k cluster

6.2



ML EXAMS 22/06/2022

EX. 1

1.1

we have overfitting when given a dataset D exist an hypothesis h' that $error_S(h) < error(h')$ and $error_D(h) > error_D(h')$

where $S \subseteq D$ and $h, h' \in$ Hypothesis space

1.2

we are sure that we have overfitting with decision tree when for same solution we have two different trees one more deeper than other

~~so consider two decision tree T and T' obtain with ID3 algorithm~~

there are two methods for reduce overfitting in decision tree

- REDUCE ERROR PRUNING: we must split training dataset in validation and we evaluate impact on validation set of pruning each possible node. greedly remove the one that most improves validation set accuracy