2. we can use least squares $ED(w) = \frac{1}{2}(t-\phi(w))^T(t-\phi(w))$ and $w^* = \phi^\dagger t$

we can also use sequential algorithm $\hat{w} \leftarrow \hat{w} + \eta[t_n - w^T\phi(w)]\phi(x_n)$

ML EXAMS 18/01/2018 A

EXERCISE 1

1.
if $C = c_1$ and $B = b_1$ then ~~YES~~ NO ①
if $C = c_1$ and $B = b_2$ then YES ②
if $C = c_2$ and $B = a_1$ then YES ③
if $C = c_2$ and $B = a_2$ and $B = b_1$ ④
if $C = c_2$ and $B = a_2$ and $B = b_1$ ⑤
if $C = c_2$ and $B = a_3$ then NO ⑥
if $C = c_3$ ~~and~~ then NO ⑦

2.
is consistent with S_1 because of ①
is consistent with S_2 because of ④
is NOT consistent with S_3 because of ⑦
is NOT consistent with S_4 because of ⑤

EXERCISE 2

1. the maximum a posterioni hypothesis is $h_{MAP} = \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)}$
if we assume $P(h_j) = P(h_i)$ we simplify and
we obtain $h_{ML} = \underset{h \in H}{\operatorname{argmax}} P(D|h)$

2 Boc is an optimal classifier that returns the most probable ~~classification~~ prediction:

$$v_{NB} = \underset{v \in V}{\operatorname{argmax}} \sum_{h_i \in H} P(v|x, D_{h_i}^{\dagger}) p(h_i|x, D) = \sum_{h_i \in H} P(v|x, h_i) P(h_i|D)$$

3. we can use it if we have analytical solutions or if hypothesis space is not to large otherwise is not practical

Question 3

1.

in linear regression we have a target function $f: x \to y$ with $x \subseteq \mathbb{R}^D$ and $y = \mathbb{R}$ and the model is :

$$y(x, w) = \sum_{j=2}^{m} w_j \phi_j(x) \text{ is linear in } w$$

we want find $w^* = \arg\min_w E_D(w)$ where $E_D(w)$ is error function

given a data $D = \{(x_n, y_n)\}_{n=1}^{N}$ target value t_n is given by $y(x, w)$ affected by additive noise ε

$$t = y(x, w) + \varepsilon$$

assume gaussian noise $p(\varepsilon|\beta) = N(\varepsilon|0, \beta^{-1})$

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

assume observations iid

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

so maximum likelihood (zero-mean Gaussian assumption

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

corresponds least mean square error

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

2.

BATCH MODE: we consider all dataset → require high computational resources

$$\Delta w_i = \eta \sum_{(x,t) \in D} (t - o(x)) x_i$$

MINI-BATCH MODE: we choose a small subset $S \subseteq D$

$$\Delta w_i = \eta \sum_{(x,t) \in S} (t - o(x)) x_i$$

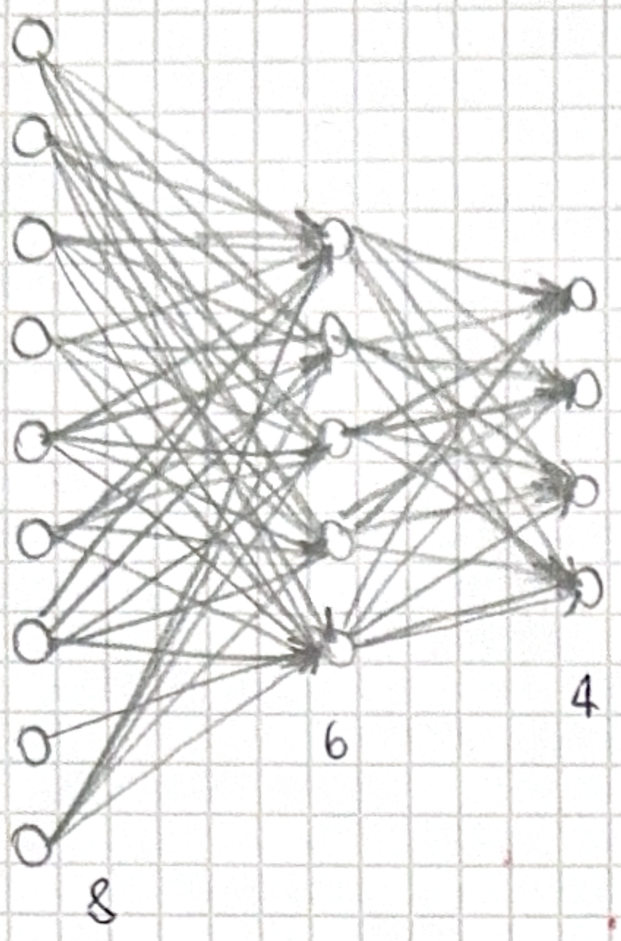
INCREMENTAL MODE: chose one sample $(x, t) \in D$ we have high variance

$$\Delta w_i = \eta (t - o(x)) x_i$$

$o(x) = w^T x$ for unthresholded $o(x) = \text{sign}(w^T x)$ for thresholded

Exercice 4

1.



params = (8+1)·5 + (5+1)·4 = 69

2. and 3.

Backprop is an algorithm used to compute the gradient and the gradient will be propagate through all the network is not affect by OVERFITING and local minima

Exercice 5

1. the dataset is separable because exist an hyperplane that divided our dataset into two regions such that different ~~classes~~ classified instances are separated

2. in this case I use a polynomial kernel function $k(x,x') = (\beta x^T x' + \gamma)^d$

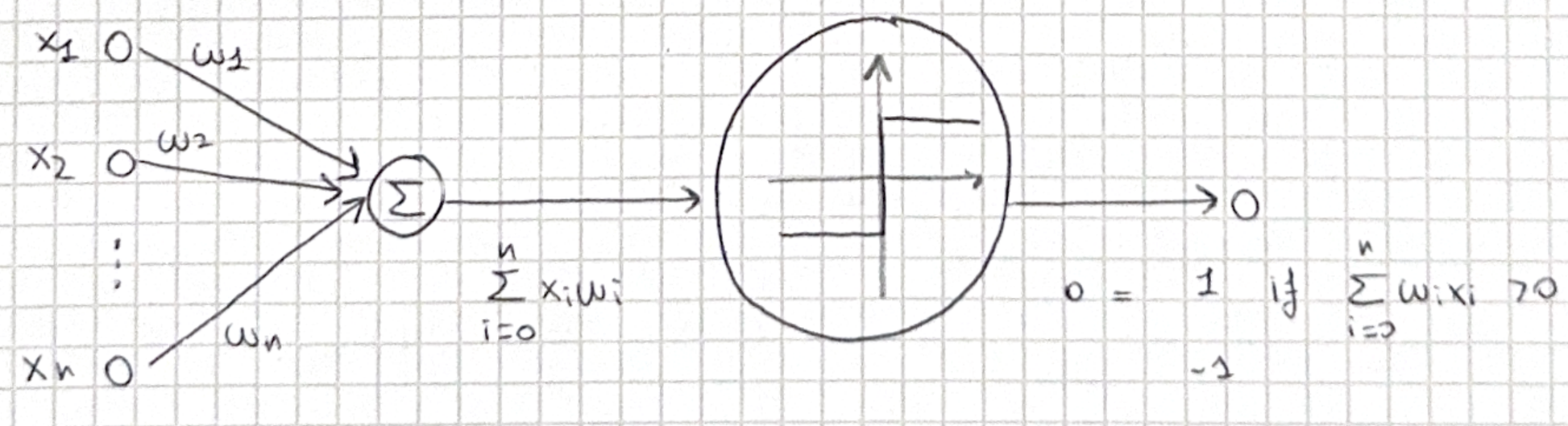
3. $y(x,\alpha) = sign\left(w_0 + \sum_{n=2}^{N} \alpha_n x_n^T x\right)$ I apply kernel trick

$$y(x,\alpha) = sign\left(w_0 + \sum_{n=1}^{N} \alpha_n k(x_n, x)\right)$$

$$w_0 = \frac{1}{|SV|} \sum_{x_i \in SV} \left(t_i - \sum_{x_j \in SV} a_j t_j k(x, x_j)\right)$$

Exercice 6

1.



we want find w_i that minimize loss function (squared error)

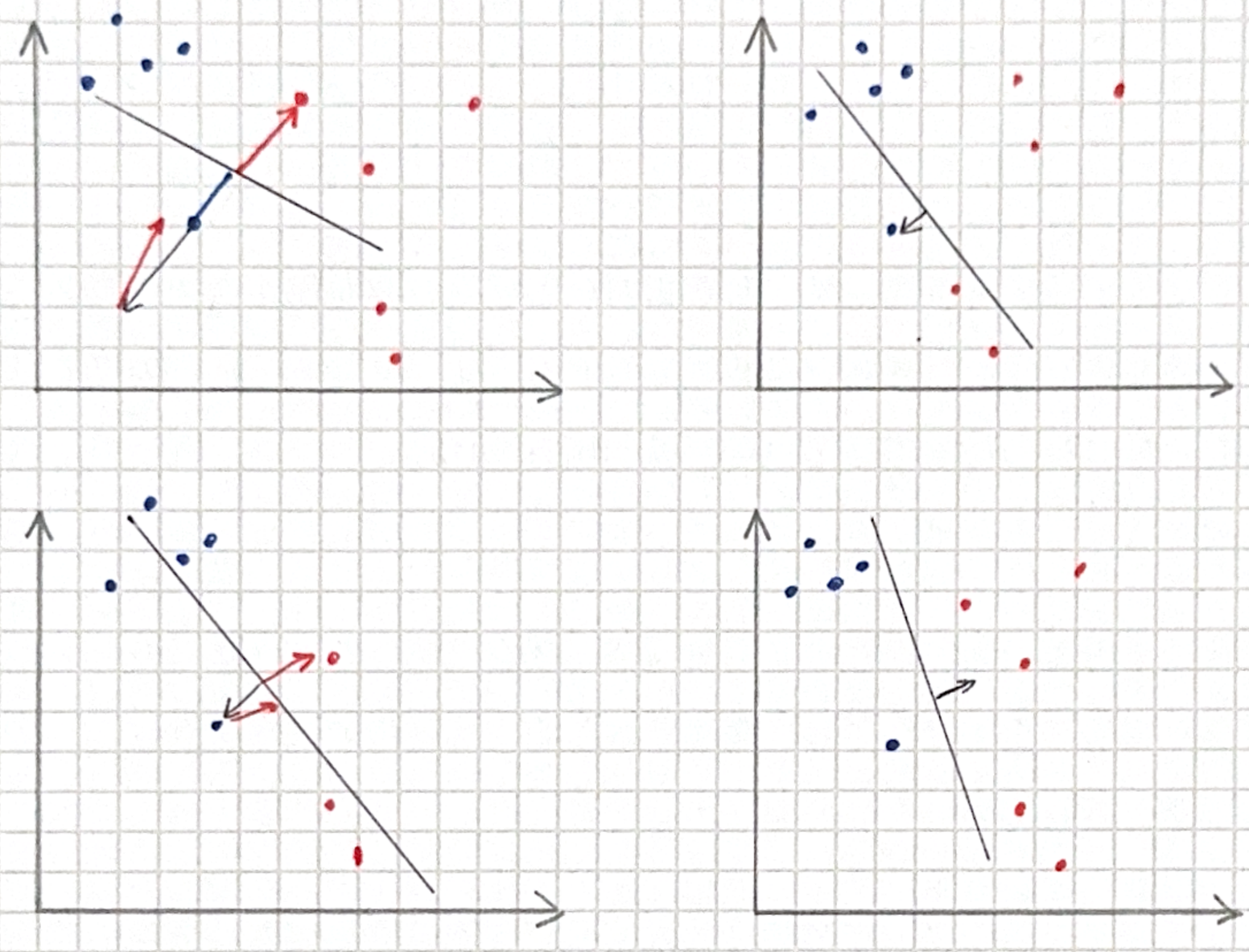
$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - o_n)^2 = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T x_n)^2$$

$$\frac{\partial E_D(w)}{\partial w} = \sum_{n=1}^{N}(t_n - w^T x_n)(-x_{i,n})$$

we find w^* in a sequential way by apply

$$w_i \leftarrow w_i + \Delta w_i \quad \text{where} \quad \Delta w_i = -\eta \frac{\partial E}{\partial w_i} = \eta \sum_{n=1}^{N}(t_n - w^T x_n)x_{i,n}$$

2.



HL EXAMS 12/02/2018 A

EX. 1

1. In a Reinforcement Learning problem we have a dataset $D = \{(\langle x_0, a_1, r_1, x_1, \dots, a_n, r_1 x_n \rangle)_{m=1}^{N}\}$ and we want learning behavior $\pi: S \to A$, meaning we want find the optimal policy function that maximize the reward
   in RL problemes we don't have output and input in tuple
   action, reward, state

2. for each x, a initialize table entry $\hat{Q}_{(0)}(x, a) \leftarrow 0$
   observe current state x
   for each time $t = 1, \dots, T$ (until termination condition)
   - choose an action a
   - execute action a
   - observe the new state x'