3. take each sample in sequence and compute its distance from centroid.
   if sample is not in the closest centroid switch and update the centroid of two cluster

4. repeat all steps until convergence

II.

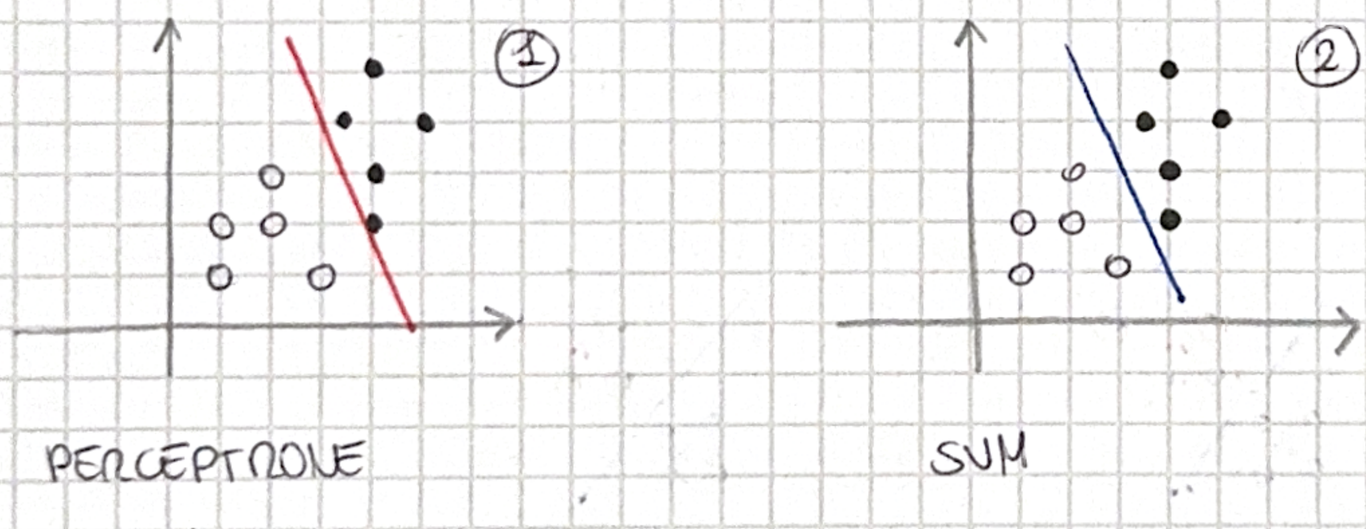
terminations condition
1. for each switch the sum of distances each training samples decrease to centroid decrease
2. the are only many finitely partitions of the training examples into k cluster

ML EXAMS 12/02/2018

EXERCISE 1

supervised learning are machine problem where the model that will solve it, are trained with a dataset composed by x and y
$D = \{(x_i, y_i)_{i=1}^N\}$
different is the case of unsupervised learning where the model used for solve this task are trained with dataset $D = \{(x_i)_{i=1}^N\}$
the label are NOT used in this case

EXERCISE 2



the difference between perceptrone and SVM is perceptron is sequential algorithm that depend on learning rate η while SVM try to maximite the margine no in this case due the fact data is linear separable I will use SVM

EXERCISE 3

1. IF $C = C_1$ and $B = b_1$ then NO ①

IF $C = C_1$ and $B = b_2$ then YES ②

IF $C = C_2$ and $A = a_1$ then YES ③

IF $C = C_2$ and $A = a_2$ and $B = b_1$ YES ④

IF $C = C_2$ and $A = a_2$ and $B = b_2$ NO ⑤

IF $C = C_2$ and $A = a_3$ then NO ⑥

IF $C = C_3$ then NO ⑦

2.

this is consistent with S_1 become of ①

// // // // S_2 // of ④

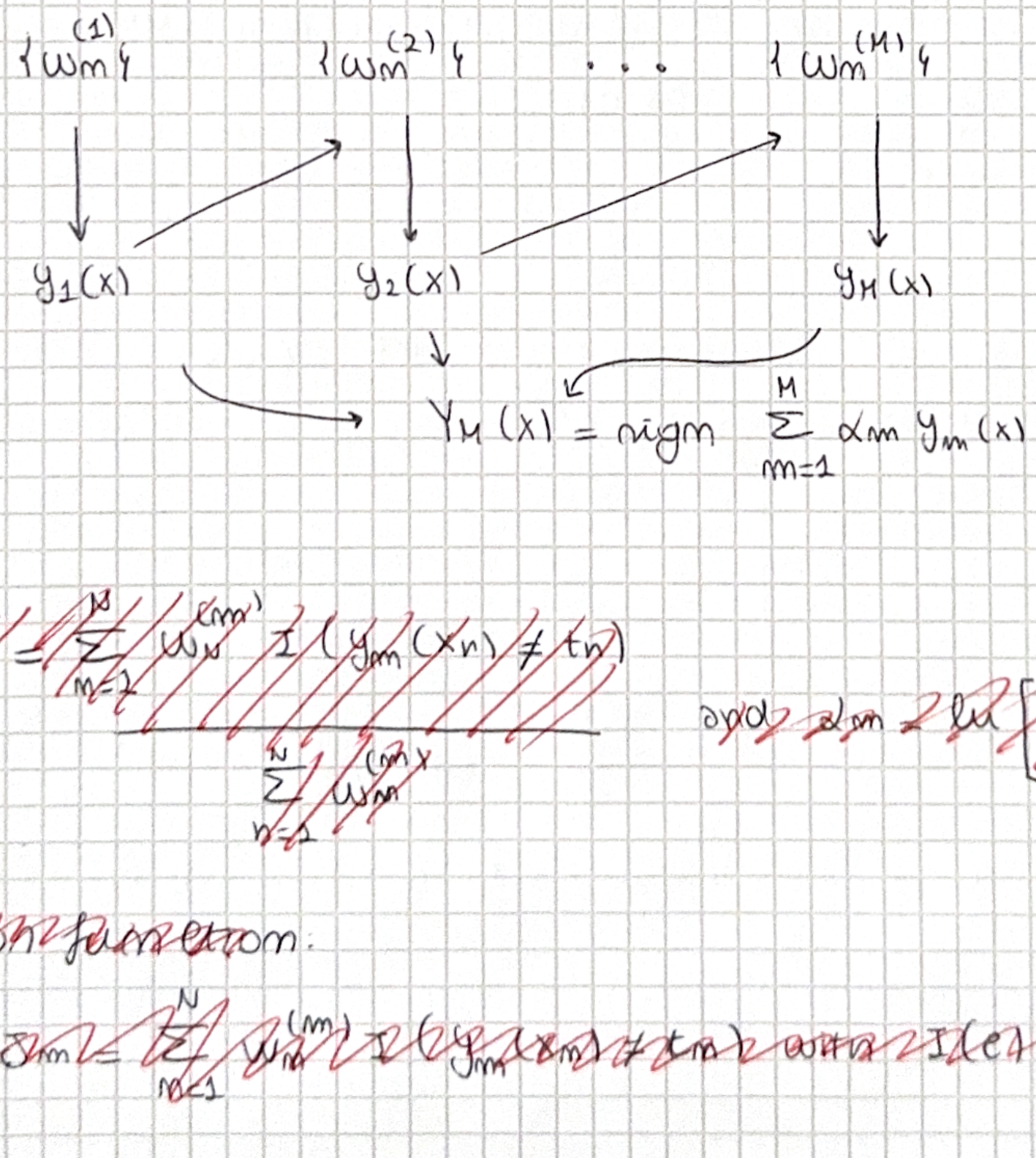
// // // // S_3 // // ~~③~~ ②

// // // // S_4 // // ②

EXERCISE 4

Boosting is an ensable methods where instead of training one single complex classifier we train different classifiers and we combine their results.

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



2.

$J_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$ and $\alpha_m = \ln\left[\frac{1-\epsilon_m}{\epsilon_m}\right]$

error function:

$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)$ with $I(e) = \begin{cases} 1 & \text{if } e = \text{true} \\ 0 & \text{otherwise} \end{cases}$

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

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

instead of minimite E ~~repeatedly~~ globally we can minimite il repeatedly:

$$E = \sum_{n=1}^{N} e\left[-t_n f_{M-1}(x_n) - \frac{1}{2} t_n \alpha_M y_M(x_n)\right] =$$

$$= \sum_{n=1}^{N} w_n^{(M)} e\left(-\frac{1}{2} t_n \alpha_M y_M(x_n)\right) \text{ where } w_n^{(M)} = e^{-t_n f_{M-1}(x_n)}$$

~~Yee~~ $Y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)$

EXERCISE 5

1. this is regression task we can use a feed forward network

$$f(x, \theta) = f^{(3)}(f^{(2)}(f^{(1)}(x, \theta^{(1)}); \theta^{(2)}); \theta^{(3)})$$

2. for hiddens units we can use ReLu function for the output we can use identity activation function $y = w^T h + b$

3. the error function that we can use is Mean Squared Error (MSE)

$$E(\theta) = \frac{1}{N} \sum_{n=1}^{N} (t_n - f(x, \theta))^2$$

EXERCISE 6

to

1. the GRAM MATRIX is $K = XX^T$. if we have a model $y(x, \alpha) = \sum_{n=2}^{N} \alpha_n x_n^T x_n$ with a kernell function $k(x, x') = x^T x'$ the gram will be

$$K = \begin{bmatrix} k(x_1 x_1) & & k(x_1, x_M) \\ \vdots & & \\ k(x_N, x_1) & \cdots & k(x_N x_M) \end{bmatrix}$$

2. the error function beomes $E(w) = \sum_{n=1}^{N} E(y_m, t_n) + \lambda \|w\|^2$

applying kernell trick $y(x, w^*) = \sum_{n=1}^{N} \alpha_n k(x_n, x)$ with $\alpha = (K + \lambda I_N)^{-1} t$