ML EXAMS 20/12/2018

EXERCISE 1

Gram matrix is $K = (XX^T)$

$$K = \begin{bmatrix} x_1^T x_1 & & x_1^T x_M \\ \vdots & & \\ x_N^T x_1 & \cdots & x_N^T x_M \end{bmatrix}$$

represent the innerproduct of input vector with the transpose of input vector

if we have a generical kernell function $k(x,x') \rightarrow y(x) = \sum_{n=1}^{N} \alpha_n k(x,x')$

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

~~EXERCISE 2~~

2. is possible obtain a kernelited version of REGRESSION.

the error function become $J(w) = \sum_{n=1}^{N} E(y_n, t_n) + \lambda \|w\|^2$ with

$E(y_n, t_n) = (y_n - t_n)^2$

we can use the equations defined before and obtain

$y(x, w^*) = \sum_{n=1}^{N} \alpha_n x_n^T x$ appling kernell trick $y(x, w^*) = \sum_{n=1}^{N} \alpha_n k(x_n, x)$

EXERCISE 2

1. $\dim(W_1) = 128 \cdot 10 = 1280$ $\quad \dim(W_2) = 50 \cdot 10 = 500$

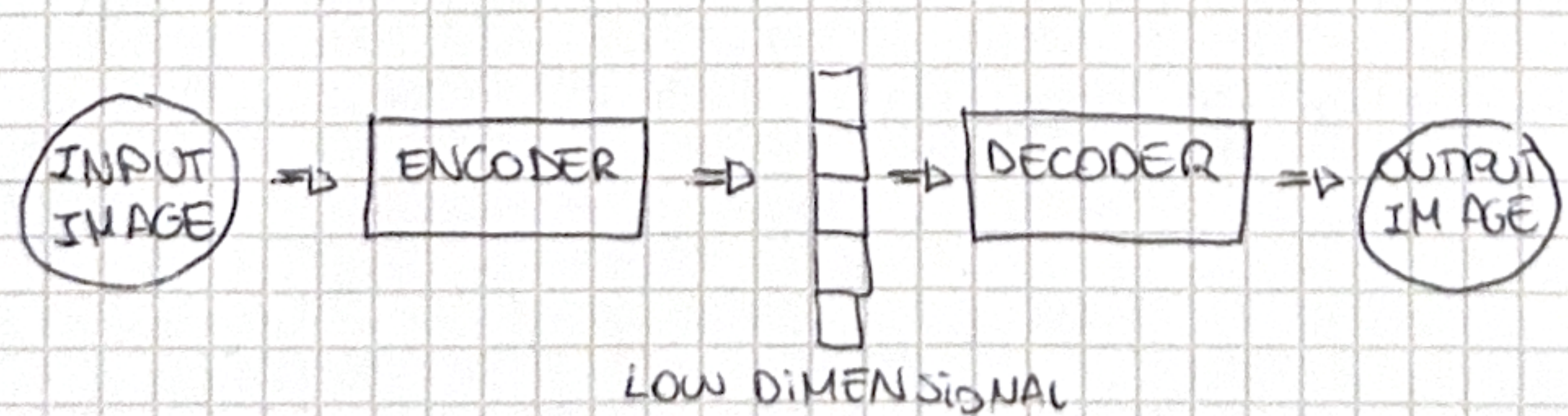
2. $h = g(W^T x + c)$ $\quad g(\alpha) = \max(0, \alpha) \Rightarrow$ ReLu function

OUTPUT $\rightarrow$ $y(x) = w^T \max(0, W^T x + c) + \underset{\text{bias}}{b}$

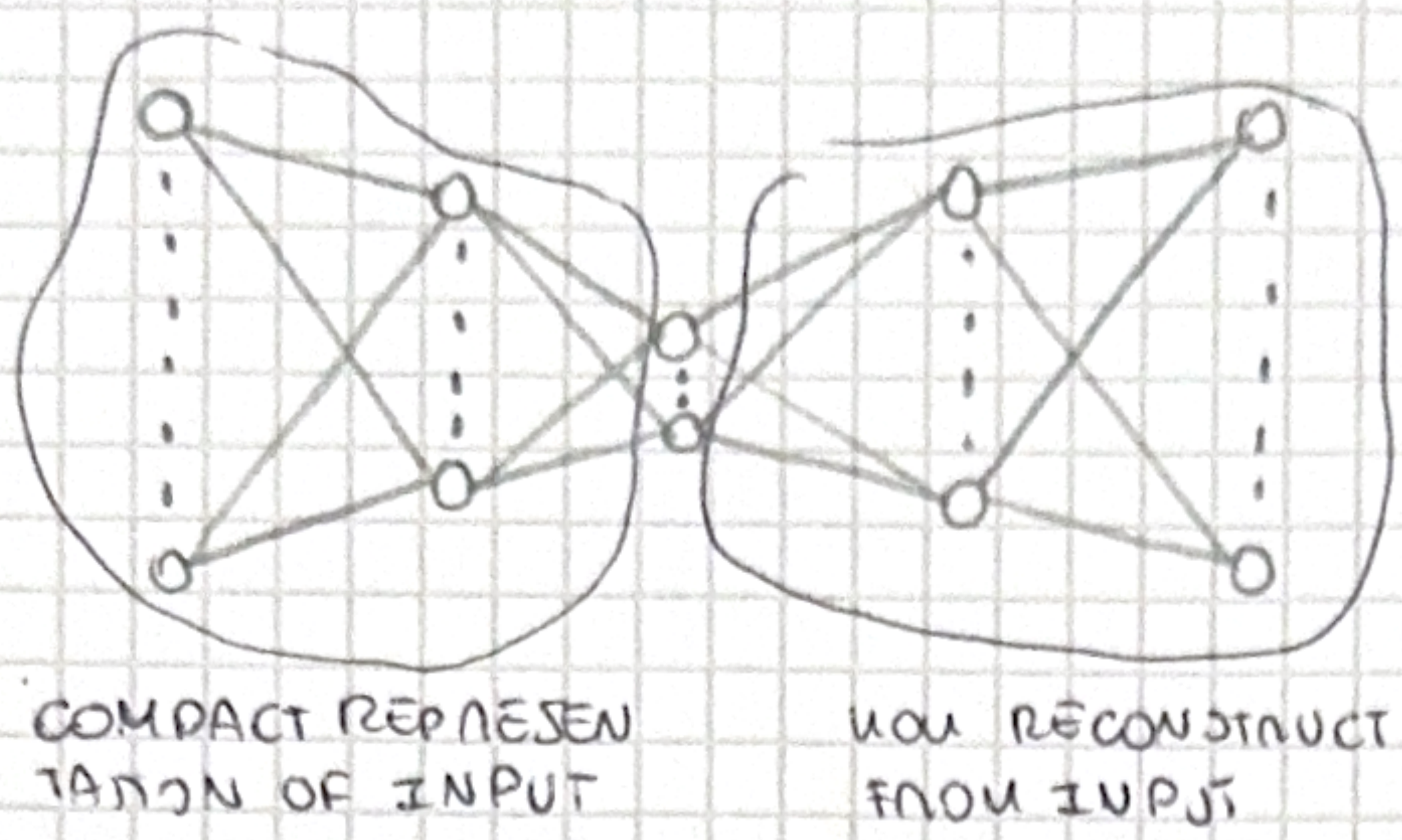
EXERCISE 3

the goal of autoencoders is dimensionality reduction an autoencoder is composed by 2 neural networks an encoders and decoders

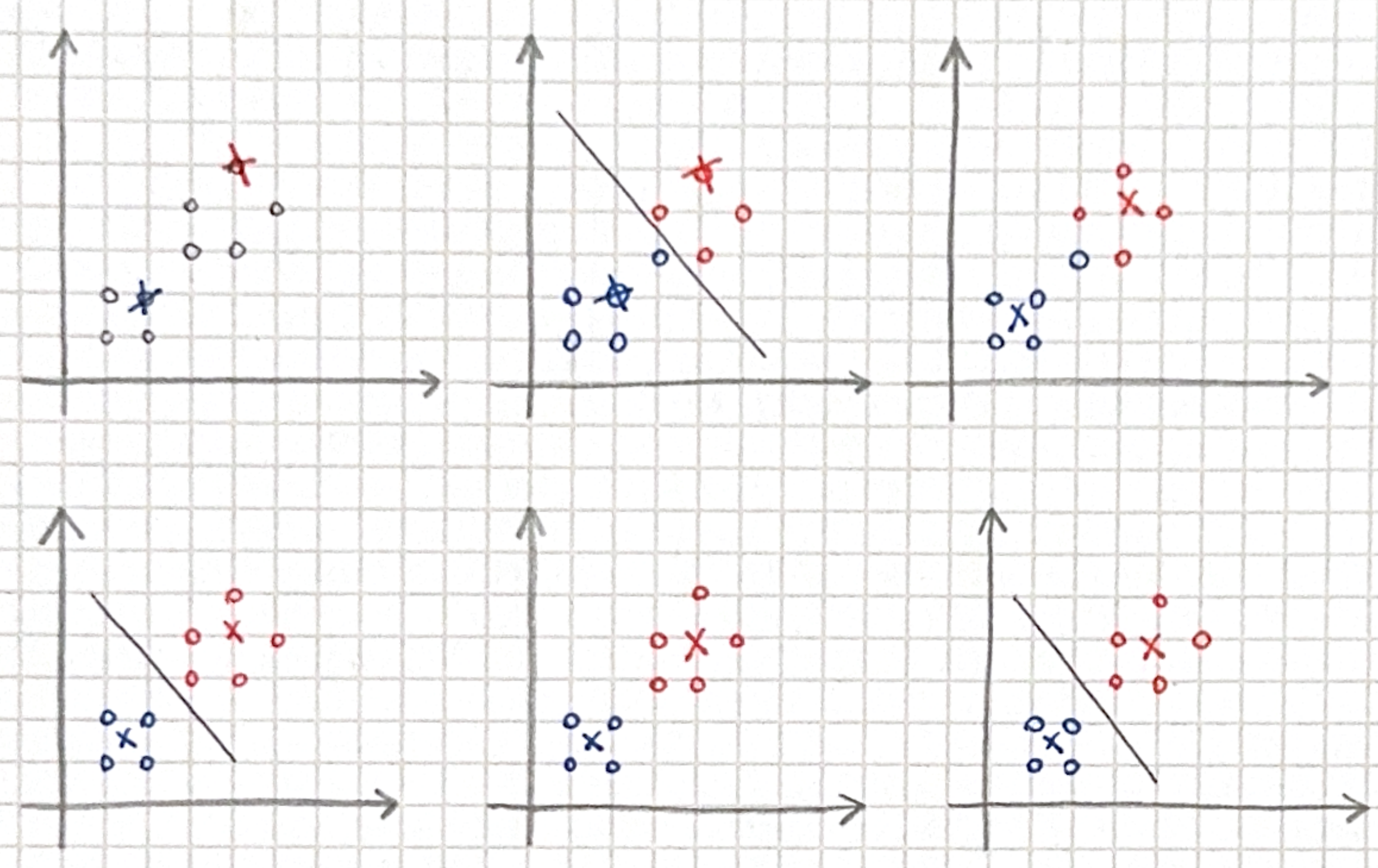
the structure is the following



in autoencoders we have usually hidden layers with reduced size until the bottleneck. the training is based on reconstruction loss



EXERCISE 4



1. begin with initial values of K = numbers of clusters
2. put an initial partition that classify data into K clusters can be done randomly or systematically as follow:
   I. take the first K training sample as single-element cluster
   II. Assign each of the remaining (N-K) training samples to the cluster with nearest centroid after each assignment recompute the centroid
3. take the sample in sequence and compute its distance from the centroid of each of the clusters
   if the a sample is not currently in the cluster with the closest centroid switch that sample to that cluster and update the centroids involved in that switch

4. repeat step 3 until termination

TERMINATION: for each iteration in step 2 the sum of distances from each training samples to that training sample's group centroid decrease

there are only many (finitely) partitions of training samples into k clusters

EXERCISE 5

1

classification with KNN

1. find k nearest neighbours of new instance x
2. assign to x the most common label among the majority of neighbors

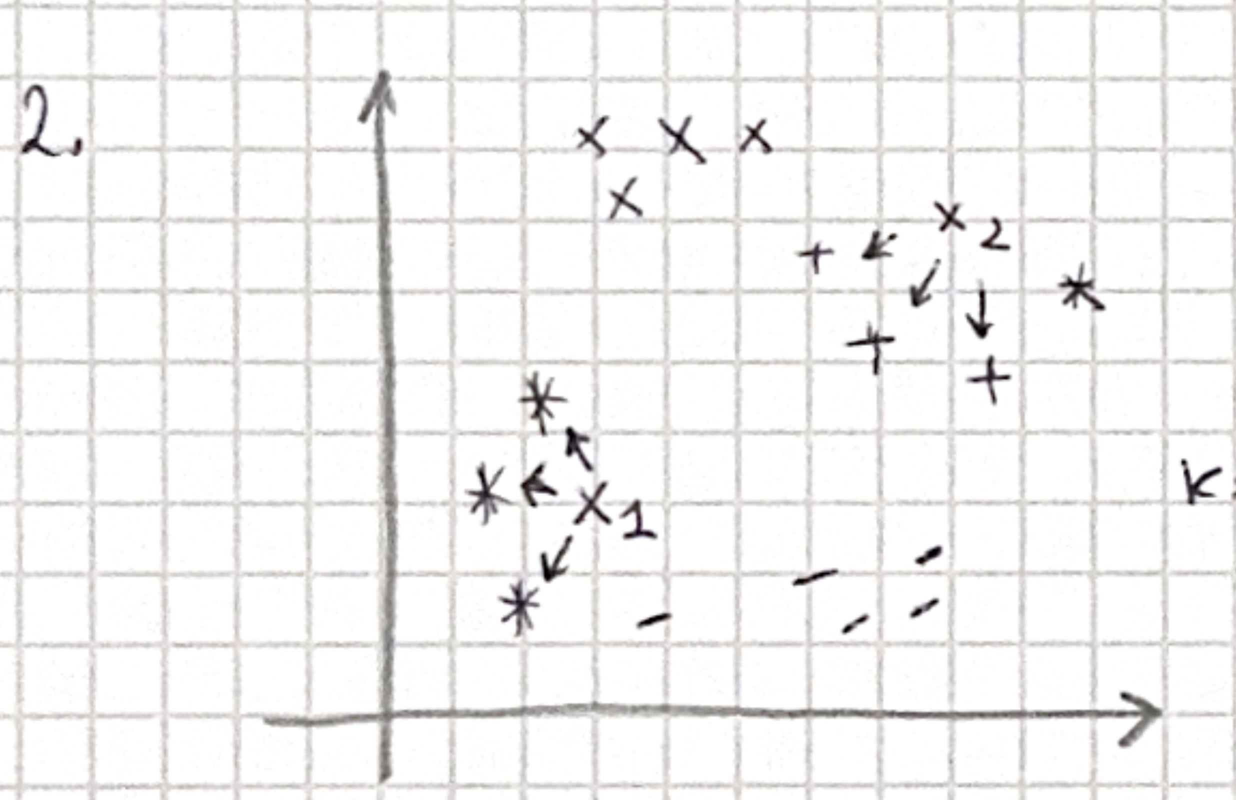
likelihood of class c for new instance x

$$p(c \mid x, D, k) = \frac{1}{K} \sum_{x_n \in N_k(x_n, D)} \mathbb{I}(t_n = c)$$

where:

$$\mathbb{I}(t_n = c) = \begin{cases} 1 & \text{if } t_n = c \\ 0 & \text{otherwise} \end{cases}$$

2.



EXERCISE 6

1. we can use boosting we ~~divide~~ generate dataset D into M bootsampled $D_1, \dots, D_M$ $D_i \subset D$
2. we use each of them to train a model $y_m(x)$ for $m = 1, \dots, M$
3. we make a prediction with voting scheme

$$y(x) = \frac{1}{M} \sum_{m=1}^{M} y_m(x)$$

$$\begin{array}{ccccc} \{w_n^{(1)}\} & & \{w_n^{(2)}\} & \dots & \{w_n^{(M)}\} \\ \downarrow & \nearrow & \downarrow & \nearrow & \downarrow \\ y_1(x) & & y_2(x) & & y_M(x) \end{array}$$

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