EXAM 15.10.19

Question 1

1). Given a classification problem with many classes, is possible to compute, using a confusion matrix, how many times an instance of the class C_i is classified in the class C_j.

2).

	LOW	MID	HIGH
LOW	80	15	5
MID	5	90	5
HIGH	5	25	70

3). accuracy = Sum of the elements in the main diagonal / sum of all the elements

$$\text{accuracy} = \frac{240}{300} = 80\%$$

Question 2

1). We can use a model based on linear regression. We know that $y(x,w) = \sum_{j=1}^{M} w_j \phi_j(x)$ that is a linear model in the parameters w. We know that $t = y(x,w) + \varepsilon$ with ε additive Gaussian noise $\Rightarrow p(\varepsilon|\beta) = N(\varepsilon|0,\beta^{-1}) \Rightarrow p(y|x_1, x_N, w, \beta) = N(y|y(x,w), \beta^{-1})$.

If we use the iid hypothesis: $p(\{y_1 - y_n\}|x_1 \dots x_N, w, \beta) = \prod_{n=1}^{N} N(y_n|w^T\phi(x_n), \beta^{-1}) =$

$$= \sum_{n=1}^{N} \ln N(y_n|w^T\phi(x_n), \beta^{-1}) = -\beta \cdot \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T\phi(x_n))^2 - \frac{N}{2}\ln(2\pi\beta^{-1})$$

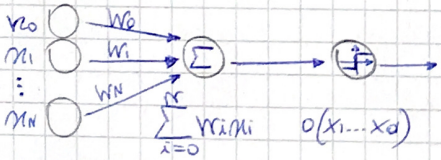
The error function is $E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T\phi(x_n))$ and we want to find $w^* = \arg\min_w E_D(w)$

2). We can apply regularization: $w^* = \arg\min_w E_D(w) + \lambda E_W(w)$

with $\lambda > 0$ regularization factor and $E_W(w) = \frac{1}{2}w^T w$

Question 3

1). The structure of perceptron model is the following:



$$o(x_1 \dots x_d) = \begin{cases} +1 \text{ if } w_0 + w_1 x_1 + \dots + w_d x_d > 0 \\ -1 \text{ otherwise} \end{cases}$$

For the moment we consider $o(x) = w^T x$.

We want to minimize an error function $E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - o_n)^2 = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T x_n)^2$

$$\frac{\partial E(w)}{\partial w_i} = \sum_{n=1}^{N}(t_n - w^T x_n)(-x_{i,n}) = -\sum_{n=1}^{N}(t_n - \text{sign}(w^T x_n))(x_{in})$$

Now we consider another time the sign function.

2). We want to find w^* in a sequential way:

$\hat{w} \leftarrow \hat{w} + \Delta w_i$ with $\Delta w_i = -\eta \sum_{n=1}^{N} (t_n - \text{sign}(w^T x_n))(x_{i,n})$. Δw_i can be update in batch, mini-batch or sequential mode.

3). η is a very important hyperparameter of our network. If η is too large is possible that we diverge and we don't find a solution. If η is too small is possible that we have a solution but can take a lot of time.

Question 4

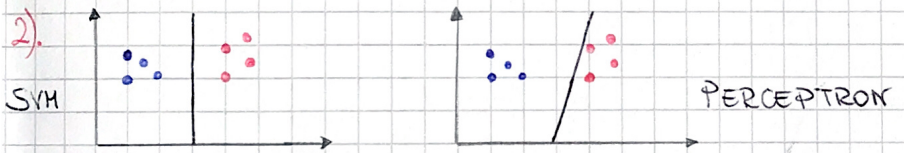
1). SVM aims at maximum margin with the better accuracy. Let x_n be the closest point to the seperation surface $h = w_0 + w^T x = 0$, we define the margin as $\frac{|y(x_n)|}{\|w\|}$. The problem of find the margin becomes to find $\min_{n=1...N} t_n[w_0 + w^T x_n] \cdot \frac{1}{\|w\|}$. But we want to find the maximum margin $w_0^*, w^* = \underset{w_0, w}{\text{argmax}} \frac{1}{\|w\|} \min_{n=1...N} t_n[w^T x_n + w_0]$. If we scale we are not affecting our solution $\Rightarrow t_n y(x_n) = 1$ and $t_n(y(x_n)) \geq 1 \ \forall n = 1...N$

Now we move to a Lagrangian domain: we know that if $t_n y(x_n) \geq 1 \Rightarrow a_n^* = 0$

$w^* = \sum_{n=1}^{N} a_n^* t_n x_n$ and we define the support vector $SV = \{x_n \in D : t_n y(x_n) = 1\}$

$w^* = \sum_{x_j \in SV} a_j^* t_j x^T x_j + w_0^* = 1$

2).



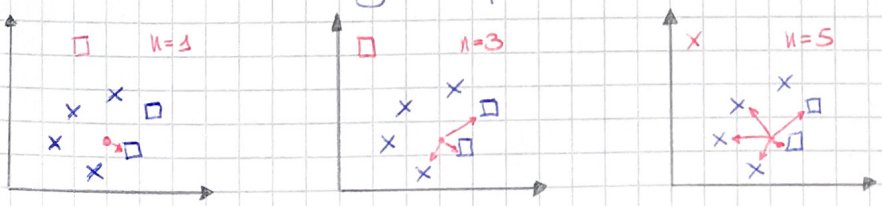
Question 5

1). We compute the k nearest neighbors of a new instance $x \notin D$ and then we assign to x the most common label among the majority of neighbors.

$p(c|x, D, k) = \frac{1}{k} \sum_{x_n \in N_k(x_n, D)} \mathbb{I}(c = t_n)$ with $\mathbb{I}(e) = \begin{cases} 1 \text{ if } e \text{ is true} \\ 0 \text{ otherwise} \end{cases}$

With $N_k(x_n, D)$ k nearest neighbors of x_n.

2).



Question 6

1). Boosting is an ensemble learning method. The main idea of an ensemble is that instead of training one simple complex classifier we train different classifiers and we combine the result. In boosting we are training models in sequence and each model is based on the error of the previous model. We can do this using weights, in fact we use very high weights for errors and very low weights for correct predictions.

2). $E = \exp[-t_n f_M(x_n)]$ with $f_M(x) = \frac{1}{2}\sum_{m=1}^{M} \alpha_m y_m(x)$

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

$$y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right) \text{ with } \alpha_M = \ln\left(\frac{1-\varepsilon_M}{\varepsilon_M}\right)$$