EXERCISE 1

1. Provide the definition of *Confusion matrix* for a multi-class classification problem, formally explain the content of component $C_{i,j}$ of the matrix.
2. Provide a numerical example of a confusion matrix for a 3-classes classification problem with a balanced data set including 200 samples for each class (600 samples in total) and an average accuracy around 70% for class 1, 80% for class 2 and 90% for class 3. The matrix **must not be simmetric**. Show the confusion matrix in two formats: with absolute values and with the corresponding percentage values. (Hint: use simple numerical values, so that you do not need to make complex calculations.)
3. Compute the accuracy of the classifier for the numerical example provided above.

the confusion matrix is the matrix C where C_{ij} = number of objects of class i classified as class j. The trace of C is the number of correct classification.

Example:

class \ class	1	2	3
1	140	20	15
2	30	160	5
3	30	20	180

in perc. ⟹

class \ class	1	2	3
1	93.3	3.3	2.5
2	5	26.6	0.8
3	5	3.3	30

the accuracy is $\frac{\text{trace}(C)}{\sum C_{ij}} = 0.8 = 80\%$

EXERCISE 2

Consider a binary classification problem $X \to \{T,F\}$, with $X = \{T,F\}^3$, i.e. $(x_1,x_2,x_3) \in X$ and $x_i \in \{T,F\}$, and the dataset $D = \{\langle(F,F,F),F\rangle, \langle(F,T,T),T\rangle, \langle(T,T,F),T\rangle, \langle(T,F,T),T\rangle\}$. Consider the two hypothesis $h_1 = (x_1 \wedge \neg x_2 \wedge x_3) \vee x_2$ and $h_2 = (\neg x_1 \wedge x_2 \wedge x_3) \vee x_1$.

1. Determine whether h_1 and h_2 are consistent with D, showing all the passages needed to answer.
2. Assuming the likelihood probabilities $P(D|h_1) = 0.6$ and $P(D|h_2) = 0.8$ and the prior probabilities $P(h_1) = 0.2$ and $P(h_2) = 0.1$, determine the higher a poteriori hypothesis between h_1 and h_2.

check for h_1):

$h_1(x_1) = F = t_1$ ✓ $\quad h_1(x_3) = T = t_3$ ✓

$h_1(x_2) = T = t_2$ ✓ $\quad h_1(x_4) = T = t_4$ ✓

$\Rightarrow h_1$ is cons.

chear for h_2):

$h_2(x_1) = F = t_1$ ✓ $\quad h_2(x_3) = T = t_3$

$h_2(x_2) = T = t_2$ ✓ $\quad h_2(x_4) = T = t_4$

$\Rightarrow h$ is cons.

$$h_{MAP} = \arg\max_h \mathbb{P}(h|D) = \arg\max_h \mathbb{P}(D|h)\mathbb{P}(h) \begin{cases} \mathbb{P}(D|h_1)\mathbb{P}(h_1) = 3/25 \\ \mathbb{P}(D|h_2)\mathbb{P}(h_2) = 2/25 \end{cases} \Rightarrow h_{MAP} = h_1.$$

EXERCISE 3

Consider a regression problem $f : \mathbb{R}^d \to \mathbb{R}$ with a dataset $D = \{(x_n, t_n)_{n=1}^N\}$, where f is known to be non-linear in x.

1. Describe a linear model for this problem and determine the trainable parameters and the size of the model (i.e., number of trainable parameters).
2. Describe a solution of the problem in terms of least square error minimization. Define the error function corresponeding to the model given above and illustrate a method to find a solution of the optimization problem.

We consider $y(x) = w^T \phi(x) + w_0$ where $w \in \mathbb{R}^d$, $w_0 \in \mathbb{R}$ are the parameters and $\phi : \mathbb{R}^d \to \mathbb{R}^d$ is a non linear function.

We consider the error function $E(w) = \frac{1}{2}\sum_{i=1}^N (t_i - y_i)^2$ where $y_i = y(x_i)$. We have:

$$\frac{\partial E}{\partial w_j} = \frac{\partial}{\partial w_j}\left(\frac{1}{2}\sum_{i=1}^N (-w^T\phi(x_i) - w_0 + t_i)^2\right) = -\sum_{i=1}^N (t_i - y_i)\phi(x)_{ij} \quad \text{we consider } \nabla E = \left(\frac{\partial E}{\partial w_j}\right)_{j=1}^d$$

and we apply the gradient descent

Initialize w
until some termination condition do{
  $w \leftarrow w - \gamma \nabla E$
}

where γ is a small positive real number.

EXERCISE 4

1. Describe the principle of soft margins used by SVM classifiers. Illustrate the concept also with a geometric example.
2. Draw a dataset for binary classification of 2D samples and show two solutions based on SVM with and without soft margin constraints. Choose a proper example that illustrates well the concept, i.e., in which the two solutions are significantly different.

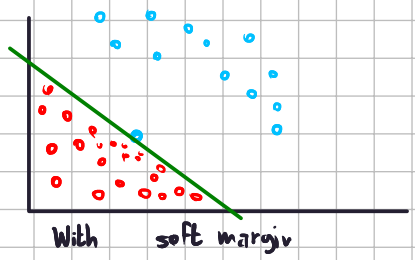
We consider as margin, the distance between the linear discriminant and the closest sample to it: $\min_{x_i \in D} \frac{|\xi(x_i)|}{\|w\|}$. The SVM aims to find the w that maximize such margin, while classificating correctly all the samples:

$D = \{(x_i, t_i)\}_{i=1}^{N}$, $t_i \in \{+1, -1\} \Rightarrow \xi(x_i) t_i \geq 0 \ \forall i$. We can add a relaxation in the optimization problem:

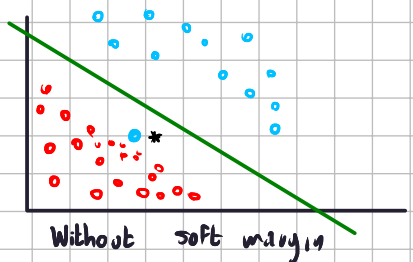
$$\begin{cases} w^* = \arg\max_w \left(\min_{x_i \in D} \frac{|\xi(x_i)|}{\|w\|} \right) + C \sum_{i=1}^{N} \xi_i^+ \\ \text{s.t.} \quad t_i \xi(x_i) \geq 0 \quad \forall i \\ \qquad \xi_i \geq 0 \quad \forall i \end{cases}$$

$\Rightarrow$ we allow some samples to overcome the margin

$$\begin{cases} \xi_i = 0 & \text{in his side} \\ \xi_i \in [0,1] & \text{violates margin} \\ \xi_i > 1 & \text{misclassified} \end{cases}$$



With soft margin



Without soft margin

* may be a very noisy sample.

EXERCISE 5

Let $\mathcal{D}$ be a dataset containing the following input values $X = \{(3.3, 1.6), (7.5, 48.2), \ldots, (98.3, 43.5), (87.2, 92.4)\}$ and target values $T = \{0, 2, \ldots, 4, 3\}$.
Consider designing a Feedforward Neural Network for learning the function $t = f(x)$.

1. Explain what is a valid choice for the activation function of the output layer and for the loss function.
2. Provide some valid options for the activation functions of the hidden units.
3. Formally describe the Stochastic Gradient Descent (SGD) algorithm and illustrate its hyper-parameters.

Since the target values are not limited in $[0, 1]$, a linear activation for the output layer is mandatory. For the hidden units, a good choiche might be the sigmoid, we observe how the samples have different scales for the attributes, so a normalization procedure of X would benefit the training process. Given the parametric net $\xi(w, x)$ with w parameters, the SGD follows:

```
SGD {
    initialize w with small rando values
    while (!termination condition) {
        X_k = sample k points from X (randomly)
        g = 0
        For each w_i ∈ w {
            g = g + (∂E/∂w_i) e_i   // e_i = [0 0 ... 1 ... 0]^T ← i-th pos.
        }                                    std basis vector
        w = w - ηg
    }
}
```

the hyper parameters are: the choice of the loss function $E(w)$, the choice of the termination condition, the choice of the batch size $k \in \mathbb{N}$, the choice of the step size $\eta \in \mathbb{R}^+$.

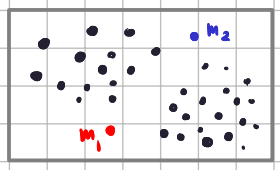
EXERCISE 6

1. Describe the K-means algorithm in a **formal** way (i.e., with precise mathematical formulas and equations), including: input and output of the algorithm, its main steps, and the termination condition.
2. Draw a suitable 2-D data set for K-means.
3. **Qualitatively** simulate the execution of K-means in such 2-D data, showing at least three steps of the algorithm and the final output.

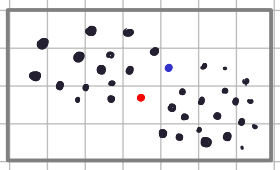
The K-means is an algorithm that assumes that the samples belongs to K classes and have a gaussian distribution

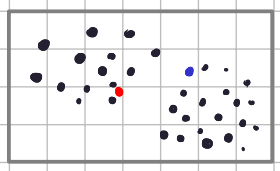
$P(x \mid c_i) = N(x, \mu_i, I)$ with unitary variance matrix (a circle). The algorithm aims (given K) to estimates the K means μ_i.

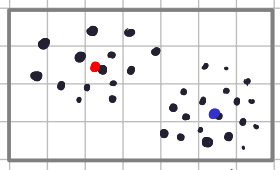
```
K-means {
    let m_1 ··· m_K, K random points called MEANS
    while (termin. condition){
        define X_1 ···· X_K as K empty sets
        For i=1···K {
            D_i = {point whose closest MEAN is m_i}
            m_i = 1/|D_i| Σ_{x∈D_i} x
        }
    }
    return m_1 ··· m_K
}
```



step 1


step 2


step 3


Final result