EXERCISE 1

1. Describe with pseudo-code the K-Fold Cross Validation method to estimate the accuracy of a learning algorithm L on a dataset D.
2. Describe how the method can be extended to comparing two different learning algorithms L_A, L_B.

```
K-Fold (L, D={(xi,ti)}, K){
    let D1 ... DK to be a partition of D
    S = 0
    For i = 1 ... K {
        Ti = D \ Di
        h = L(Ti)
        S += error_Di(h)
    }
    return 1 - S/K
}
```

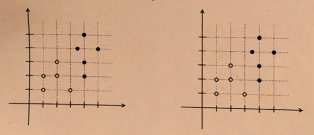
= For comparison ⇒

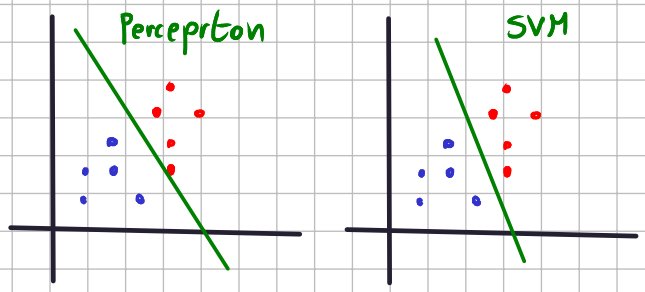
```
K-Fold (LA,LB, D={(xi,ti)}, K){
    let D1 ... DK to be a partition of D
    S = 0
    For i = 1 ... K {
        Ti = D \ Di
        hA = LA(Ti)
        hB = LB(Ti)
        S += error_Di(hA) - error_Di(hB)
    }
    if S ≤ 0 { return "LA is better" }
    return "LB is better"
}
```

EXERCISE 2

Consider the following data set for binary classification, where the two classes are represented with white and black circles.

1. Draw in each of the diagrams below a possible solution for a method based on Perceptron with very small learning rate and a possible solution for a method based on SVM.
2. Describe the difference between the two solutions and briefly explain how these are obtained with the two methods.
3. Discuss which solution would you prefer and why.





The perceptron solution starts with a given random solution and then adjust it to reduce the error through the gradient descent, with a small step size, the convergence will be very low, and is plausible that the solution will be near some sample since, the termination condition may stop the algorithm few steps before the discriminant became a linear separator. The SVM find a discriminant that maximize the margin, i.e. the distance between the separator and the nearest sample to it, this is a better solution, safer due to the margin, and easier to compute, the small step size makes the convergence of the perceptron method too slow.

EXERCISE 3

1. Describe the k-armed bandit problem (also known as One-state MDP).
2. Describe the Reinforcement Learning procedure to compute the optimal policy in the k-armed bandit problem.

There is a MDP with only one state, and K possible actions:

$X = \{x\}$ 1 state. $A = \{a_1 \cdots a_K\}$, $\delta(x, a_i) = x$ $\forall i$ and there is a reward function $r : X \times A \to \mathbb{R}^+$. We directly write $r(a_i)$. We want a policy π that maximize V^π I r is deterministic, we can try the K action and choose $a^* = \max_{a \in A} r(a)$. Let r to be non deterministic, in particulare:

$\mathbb{P}(r(a_i) = \alpha) = D_i(\alpha)$ $\forall i$, where D_i is a probability distribution. We have to choose the action that maximize the expected value: $\mathbb{E}(r_i) = \int_R \alpha \cdot D_i(\alpha) d\alpha$ where R is the set of possible rewards. A possible strategy, is to try a lot of times the action trying to estimate the expected value

let $E_1 \cdots E_K$
For i in $1 \cdots K$ {
  For j in $1 \cdots n$ {
    $E_i += r(a_i)$
  }
  $E_i = \frac{1}{n} E_i$
}
return a_i with max E_i.

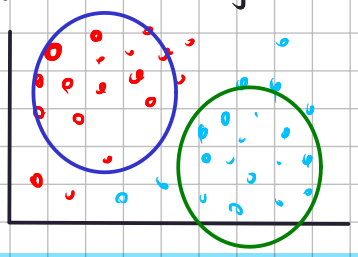
EXERCISE 4

Given a dataset D for a classification problem with classes $\{C_1, \ldots, C_n\}$.

1. Describe the difference between generative and discriminative probabilistic models for classification.
2. Draw a 2D dataset for a binary classification problem and show (also in a graphical form) a possible solution using a probabilistic generative model.

A generative model assumes that there is a normal distribution on the classes: $P(c_i|x) = N(x, \mu_i, \Sigma)$ and use the likelihood of the dataset to estimate the prior probabilities of the classes, the means μ_i and the covariance matrix Σ, after that, uses these values to define w and w_0 and predicts with the softmax function $\text{softmax}(w^T x + w_0)$.

The discriminative model, consider the cross-entropy error function given by the likelihood to find directly w and w_0 by solving an optimization problem (without finding the distributions).



the circles are the classificator, notice how the distributions have the same covariance matrix.

EXERCISE 5

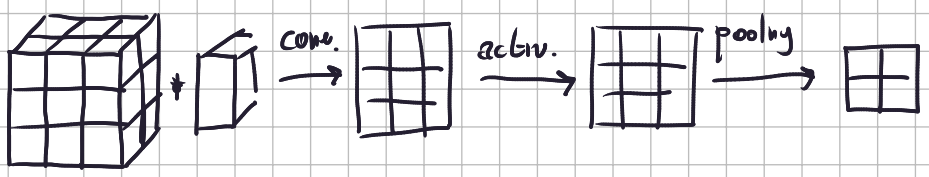
1. Describe the convolution stage of a Convolutional Neural Network (CNN).
2. Discuss the properties of sparse connectivity and parameter sharing for CNN.

The convolution stage have 3 components:

1) application of a convolution with a kernel: $I * M(i,j) = \sum_x \sum_m I(x,m) M(x+i, m+j)$

2) use of an activation function on the result of the convolution

3) pooling: reduction of the size of the result (average or max)



EXERCISE 6

Machine learning problems can be categorized in supervised and unsupervised.

1. Explain the difference between them providing a precise formal definition (not only explanatory text) in terms of input and output of the two categories of problems.
2. Describe an application problem that can be modelled and solved with an unsupervised learning method.

Let $\delta : X \to Y$ to be the target function to learn, if the dataset is $D = \{(x_i, t_i)\}_{i=1}^{N}$ with $x_i \in X$ and $t_i \in Y$, then is supervised learning, since we know the value of δ on the samples. If not : $D = \{x_i\}_{i=1}^{N}$, then is unsupervised learning. In supervised learning we want to define a model $\hat{\delta}$ that approximate δ, in unsupervised learning we want to split the dataset in cathegories. For example, let D to be a set of images representing lakes or forest. These images are fetched from the web and are not labeled by humans since they are too much. We can use unsup. learning methods like K-means (with $K=2$) to classify the images.