EXERCISE 1

1. Describe the principle of maximal margin used by SVM classifiers. Illustrate the concept with a geometric example.
2. Draw a linearly separable dataset for binary classification of 2D samples. Draw two solutions (i.e., two separation lines): one corresponding to the maximum margin, the other one can be any other solution.
3. Discuss why the maximum margin solution is preferred for the classification problem.

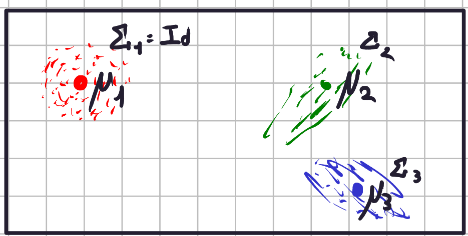
EXERCISE 2

Given an unsupervised dataset $D = \{\mathbf{x}_n\}$

1. Define the Gassian Mixture Model (GMM) and describe the parameters of the model.
2. Draw an example of a 2D data set (i.e., $D \subset \Re^2$) generated by a GMM with $K = 3$, qualitatively showing in the picture also the parameters of the model.
3. Determine the size of the model (i.e., number of independent parameters) for the dataset illustrated above.

In the GMM we assume that the samples are generated by K Normal distributions.

$P(C_i | x) = \mathcal{N}(x ; \mu_i, \Sigma_i)$, with p_i = at priori probability of having a model of class C_i.



For each class we have

p_i, one real number.

μ_i, two real numbers.

$\Sigma_i \in Mat(2 \times 2)$, Four real numbers.

So the total size is $3 \cdot (1 + 2 + 4) = 21$.

EXERCISE 3

1. Discuss the following statement: *"Accuracy is not always a good performance metric for classification"*.
2. Provide a numerical example to motivate your answer.

Accuracy is not a good performance metric if the dataset is unbalanced.

Consider a binary classification problem with $D = \{(x_i, t_i)\}_{i=1}^N$ and $t_i = 1 \ \forall i$

The model $y(x) = 1$ will have 100% accuracy on D, but is not necessarily a good model.

EXERCISE 5

1. Describe the difference between exploitation and exploration in reinforcement learning and discuss why it is important to properly balance between the two strategies.

2. Describe how exploitation and exploration are generally implemented in RL algorithms. Illustrate this step with an abstract pseudo-code.

Exploitazion: We prioritize the actions for which we know they gives higher rewards.

Exploration: We choose random actions to look for high rewarding but still unknown actions. We might start by choosing random actions with probability ε and best action with prob. $1-\varepsilon$, with ε decreasing over time.

```
ε-greedy_strat(
    Q(0) := 0
    let x a state
    For i := 1 ..... T {
        ε := 1/T
        with prob. ε {
            a = random action
            Q(i) = r̄ + Q(i-1)(x', a)
        }                                      // where x' = δ(x, a)
        else {                                 // r̄ = r(x, a)
            Q(i) = r̄ + max_{a'} (x', a')
        }
        x = x'
    }
    return  π*(x) ∈ argmax_{a∈A} Q(T)(x, a)
}
```