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 compare two different learning algorithms L_A, L_B.

Given a dataset D, a learning alg. L, and a natural number $K \in \mathbb{N}$, the algorithm is the following:

let $D_1 \cdots D_K$ to be a partition of D
For $i = 1, 2 \cdots K$ {
  $T_i = D \setminus D_i$
  $h_i = L(T_i)$ // train on T_i
  $\delta_i = \text{accuracy}_{D_i}(h_i)$ // accuracy on D_i
}
return $\frac{1}{K}\sum_{i=1}^{K} \delta_i$

to compare L_A and L_B:

let $D_1 \cdots D_K$ to be a partition of D
$\delta = 0$
For $i = 1, 2 \cdots K$ {
  $T_i = D \setminus D_i$
  $h_A = L_A(T_i)$ // train on T_i
  $h_B = L_B(T_i)$
  $\delta \mathrel{+}= \text{accuracy}_{D_i}(h_A) - \text{accuracy}_{D_i}(h_B)$
}
if $\delta > 0$ { L_A is better }
else { L_B is better }

EXERCISE 2

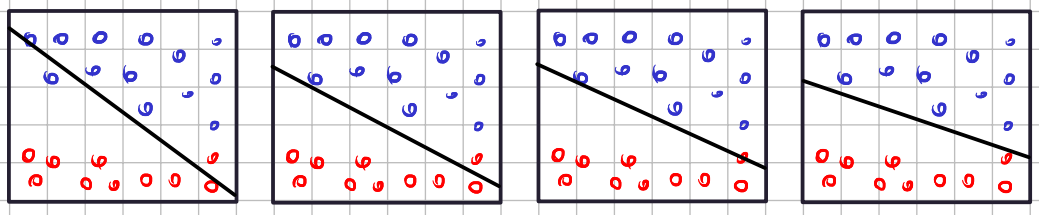
1. Describe the perceptron model for classification and its training rule.
2. Draw a graphical representation of a linearly separable 2D data set for binary classification and provide a qualitative graphical example of a possible evolution of perceptron training with two different values of learning rate: one very small, one large. Draw 4 images for each parameter (very small and large), in each image show the same data set and the current model computed from the algorithm during its steps. Start with the same initial configuration of the model that is not a solution of the classification problem. Show different evolution from this situation according to the different values of the learning rate. The last image of the sequence should show a situation corresponding to a termination condition.

$$w_1 x_1 \cdots + w_j x_j + w_0$$
$$\uparrow$$

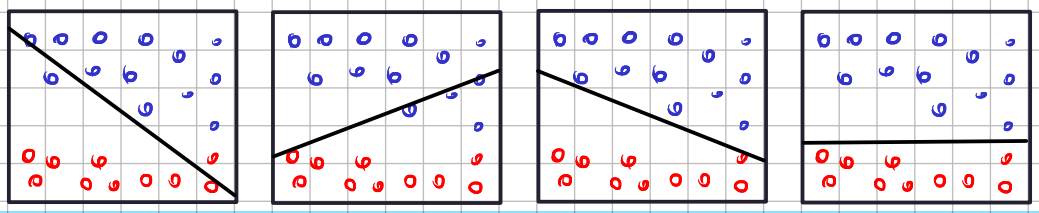
parameters w, model: $\xi(x) = \text{sign}(w^T x)$

Error function: $E(w) = \frac{1}{2}\sum_{i=1}^{N} (t_i - \xi(x_i))^2 \Rightarrow \frac{\partial E}{\partial w_j} = -\sum (t_i - \xi(x_i))\, x_{ij}$

$\Rightarrow$ training rule: $w \leftarrow w - \eta \nabla E$ with $\eta \in \mathbb{R}^+$



η small



η large

EXERCISE 3

Consider a setting where the input space I is the set of finite strings over the characters a, b, c. Notice that input strings can be of different length. Given the dataset on the table on the right:

1. Identify the learning problem at hand, in particular the form of the target function.
2. Define a suitable kernelized linear model for this problem.
3. Define a kernel function suitable to measure the similarity of data sample for this problem.

x	t
b	1
a	2
ab	2
caba	4
abca	4
aabba	8
aaa	8
babaa	8
abcaaca	16
bcaaaca	16
abcbabacca	16

We notice how the label is always 2 to the power of number of a's.

So we want to learn $\delta(s) = 2^{|\#a \text{ in } s|}$. We consider the model

$$\xi(x) = \sum_{i=1}^{N} w_i K(x_i, x) \quad \text{where } \{w_i\} \text{ are the parameters.}$$

The kernel function is:

$$K(s_1, s_2) = \left(1 + |\#\text{of a's in } s_1 - \#\text{ of a's in } s_2|\right)^{-1}$$

where 1 means max similarity.

EXERCISE 4

Consider the k-armed bandit problem (also known as One-state MDP) with stochastic Gaussian behavior and unknown reward functions.

1. **Formally** describe the *model*, provide mathematical definitions and explanations of all the elements of the model.
2. **Formally** describe the *problem* and the *solution concept* (i.e., what is the problem to be solved with the model described in the previous point), provide mathematical definitions and explanations of all the elements of the problem.
3. **Formally** describe the Reinforcement Learning algorithm to compute the optimal policy for this problem, provide explanations of all the terms of the algorithm.

We have an MDP $:= (\{x_0\}, A = \{a_1 \cdots a_K\}, \delta, r)$ where x_0 is the state and A the actions so that $\delta(x_0, a_i) = x_0 \ \forall i$ and $r(x_0, a_i, x_0)$ is the reward function. There is a distribution:

For each i : $\mathbb{P}(r(a_i) = \rho) \overset{\text{normal}}{=} \mathcal{N}(\rho; \mu_i, \sigma_i)$ with μ_i and σ_i unknown.

The goal is to find the a_i that maximizes $\mathbb{E}(r(a_i))$ that is μ_i, so we need to estimate μ_i. We do this by sampling:

$\hat{\mu}_1 \cdots \hat{\mu}_K = 0$
For $i = 1, 2 \cdots K$
  For $t = 1, 2 \cdots T$ {
    $\bar{r} = r(a_i)$
    $\hat{\mu}_i \mathrel{+}= \bar{r}$
  }
  $\hat{\mu}_i = \frac{1}{T} \hat{\mu}_i$
}

$\Rightarrow$ The optimal policy is $\pi^*(x_0) = a_j$ where

$j = \arg\max_i \hat{\mu}_i$

EXERCISE 5

Consider a layer of a CNN with input feature maps of size $96 \times 96 \times 10$.

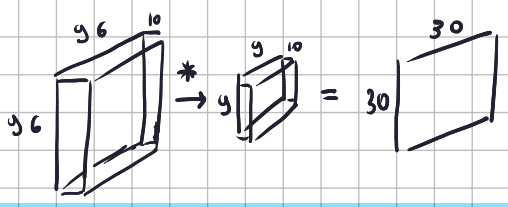
1. Provide the design of the layer (including all details about kernel, pooling and activation functions) to obtain output feature maps of size $30 \times 30 \times 25$. Note: kernel and pooling size should be small (less than 10×10).
2. How many trainable parameters are produced by this layer?

The kernel should have a depth of size 10 and there will be 25 kernels to produce a feature map of depth 25. We know that

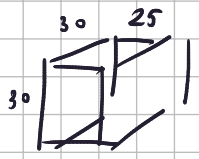
$$\frac{W_{in} - W_K + 2p}{s} + 1 = 30 \Rightarrow \frac{96 - W_K}{3} = 29 \Rightarrow 96 - W_K = 87 \Rightarrow W_K = h_K = 9$$

(padding p; stride s; $s = 3$, $p = 0$)

So we have 25 $(9 \times 9 \times 10)$ kernels. We apply an average pooling without stride or padding. So the kernel total size is 20250.



with 25 kernels $\Rightarrow$



For the gating we choose the relu.

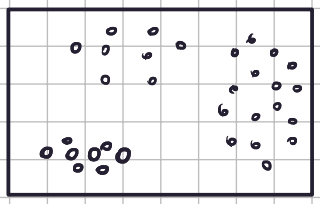
We have 25 more parameters For the biases.

EXERCISE 6

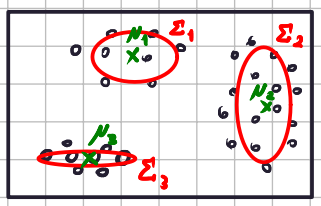
Given an unsupervised dataset $D = \{\mathbf{x}_n\}$

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 $K = 3$ with significant difference of all the parameters of the model.
3. Draw a possible solution obtained by the Expectation-Maximization algorithm algorithm on the data set shown in point 2.

In the GMM we assume that there is a normal distr.: $\mathbb{P}(x \mid c_i) = \mathcal{N}(x; \mu_i, \Sigma_i)$ For each class c_i and $\mathbb{P}(c_i) = p_i$. In this model the parameters are $\{\mu_i, \Sigma_i, p_i\}$ For each i. And $\mathbb{P}(x) = \sum_{i=1}^{K} p_i \mathcal{N}(x; \mu_i, \Sigma_i)$ A possible data set is:



a possible solution might be:



with
$p_1 = 1/4$
$p_2 = 1/2$
$p_3 = 1/4$