Deep learning Practical Assignment #1:

Linear Classifier : Perceptron

Lab date : October 5th, 2021
Deadline to submit : October 25th, 2021 - 11:59 PM (23:59)

Instructions

Read all the instructions below carefully before you start working on the assignment.

- You must do this assignment in groups of **at most 3** students.
- You must implement all algorithms **from scratch**.
- Please submit a jupyter notebook that contains:
  - A section (multiple cells) for your answer to the theoretical questions.
  - A section (multiple cells) for the installation of the used packaged. please specify the exact package version to avoid possible conflicts.
  - A section (multiple cells) for you implementation.
- Please submit your notebook to your corresponding git branch in our git work-space.
- Late submissions will be graded as follows:
  - On time submission grade SG.
  - $SG = 0.95 * SG$ for each day. Capped at $\frac{2}{3} * SG$
- Early submissions will be graded as follows:
  - On time submission grade SG.
  - $SG = 1.02 * SG$ for each day. Capped at $\frac{5}{4} * SG$
- Code that does not work will be graded accordingly. Please focus on the algorithms you are implementing.

Practical assignment objective

- To implement the perceptron algorithm from scratch in order to classify then predict data.

1 Working with perceptron algorithm

We can implement the perceptron algorithm in different ways. One of these implementation is the followings:

Implementation of Perceptron algorithm

input: A training set $\mathbb{S} = \{(\mathbf{x_1}, y_1), \dots, (\mathbf{x_m}, y_m) | \mathbf{x_i} \in \mathbb{R}^d\}$
initialize: $\mathbf{w}^{(0)} \leftarrow (0, \dots, 0)$
for t = 1,2,... , some upper bound **n do**
    for $(\mathbf{x_i}, y_i) \in \mathbb{S}$ **do**
        if $y_i \langle \mathbf{w}^{(t)}, \mathbf{x_i} \rangle \leq 0$ **then**
            $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + y_i \mathbf{x_i}$
        end if
    end for
    if $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)}$ **then**
        return $\mathbf{w}^{(t)}$
    end if
end for

In this assignment please implement this version of perceptron algorithm.

Question 1

In this section, we aim to understand the space complexity and time complexity of the simple perceptron algorithm.

Question 1.a

What is the computational/time complexity of the basic perceptron algorithm.
Please define the complexity as function of the variables defined in the provided algorithm

Question 1.b

What is the space complexity of the basic perceptron algorithm.
Please define the complexity as function of the variables defined in the provided algorithm

2 To understand you need to implement !!

In this segment we implement the perceptron algorithm.

Implement from scratch only the perceptron algorithm, for data splitting and suffling and other operations, you can use 3rd party libraries

2.1 Toy data set

Consider a data set $\mathbb{S} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{250}$ consisting of 250 points $\mathbf{x_i} = (x_1, x_2)$ and their labels y_i.

- The first 125 $\mathbf{x_i}$ have label $y_i = -1$ and are generated according to a Gaussian distribution $\mathbf{x_i} \sim \mathcal{N}(\mu_1, \sigma_1^2 \mathbf{I})$, where

$$\mu_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

- The last 125 $\mathbf{x_i}$ have label $y_i = 1$ and are generated according to a Gaussian distribution $\mathbf{x_i} \sim \mathcal{N}(\mu_2, \sigma_2^2 \mathbf{I})$, where

$$\mu_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

After shuffling the data-set, split it into train and test set, containing 80% and 20% of the dataset.

2.2 Experiments

Implement the perceptron algorithm

Experiment 1

Generate one data set for $\sigma_1^2 = \sigma_2^2 = 0.25$.

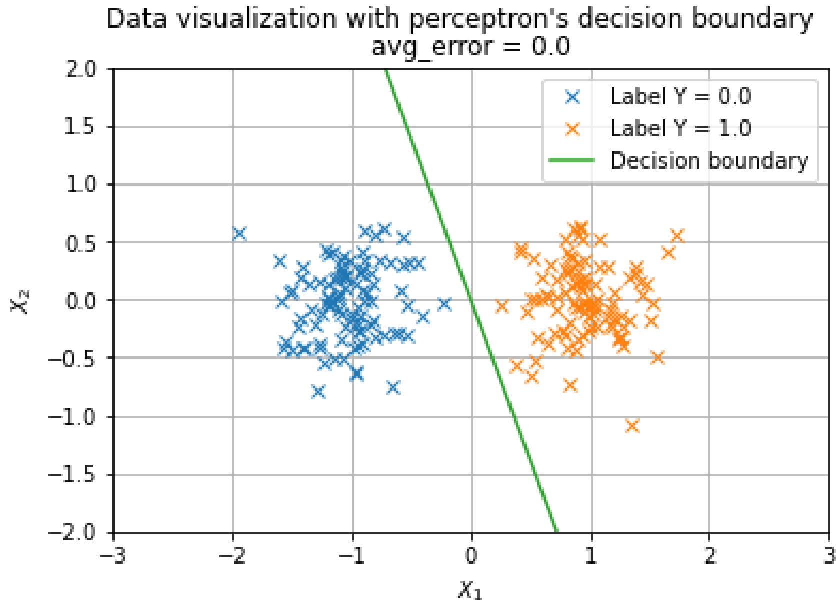
- **Question 1:** Does the algorithm converges? Why?
- **Question 2:** Plot the decision boundary found by your algorithm. Is this decision boundary unique? Does changing the initialisation changes the result of the algorithm?
- **Question 3:** Compute the accuracy of the classification on the test set. Plot the decision boundary on the test set.

Experiment 2

Generate one data set for $\sigma_1^2 = \sigma_2^2 = 0.75$.

- **Question 1:** Does the algorithm converges? Why?
- **Question 2:** Plot the decision boundary found by your algorithm. Is this decision boundary unique? Does changing the initialisation changes the result of the algorithm?
- **Question 3:** Compute the accuracy of the classification on the test set. Plot the decision boundary on the test set.

Comment the results found in the previous two experiments.
The result should be similar to the figure below, varying the variance will impact the result



Experiment 3

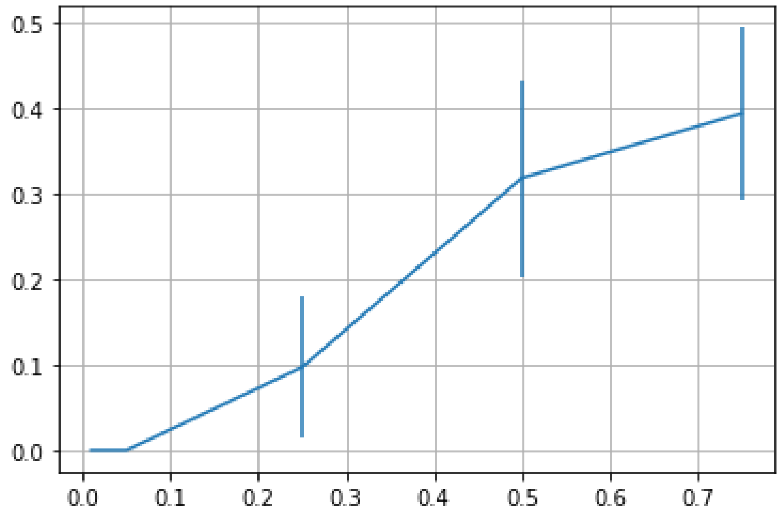
We define one experiment as following

- Generate the data and train your model .
- Compute the error on the test set.

In order to study the impact of varying σ_1^2 and σ_2^2 on the performance of the system, we store the error over multiple experiments (take $nb\_experiment = 30$). Then we compute the mean and the variance of the stored errors.

for each σ_1^2 and $\sigma_2^2 \in [0.01, 0.1, 0.5, 0.7]$ compute the mean and variance then plot the results using $matplotlib.pyplot.errorbar$.

The following figure should be similar to the expected result



Comment the result.