NNTI Assignment 1

Released: 23.10.2024

Deadline: 30.10.2024 23:59

Guidelines: You are expected to work in a group of upto 3 students. Only one member of the group should submit the assignments. While submitting the assignments, please make sure to include the following information for all our teammates in your PDF/python script:
Name:
Student ID (matriculation number):
Email:

The first assignment can be submitted alone if you have not had time to find a group. After this, please only submit in a group.

Your submissions should be zipped as **Name1_id1_Name2_id2_Name3_id3.zip** when you have multiple files, as the case might be for the assignments. For assignments where you are submitting a single file, use the same **naming convention** without creating a zip. For any clarification, please reach out to us on the **CMS forum**.

Exercise 1

We will use the NumPy python library to generate some data, and in the process get used to the library functions. Do the following exercise using vectorised numpy operations instead of python loops.

1. Write a function to generate two sets of points that can be linearly separated in 2D space:
   - Use np.random.randn() to create two clusters of 100 points each, each representing a different class. (This function can only generate the standard normal distribution).
   - Move the mean of each of these clusters to [-2, -2] and [2, 2] respectively.
   - Assign classes to these clusters as -1 and 1.
   - Combine and shuffle the points from different clusters. Make sure to combine and shuffle the assigned classes accordingly.
   - Return the points and their classes as separate numpy arrays of shape (N, 2) and (N, 1) respectively.

   (2 points)
2. Write a function to similarly generate the XOR dataset:

- It consists of 4 points as follows: [[0, 0], [0, 1], [1, 0], [1, 1]].
- The class for each point is decided as the XOR of its x and y coordinate.
- Instead of [0, 1], change the class labels to [-1, 1].
- Return the points and their classes as separate numpy arrays of shape (N, 2) and (N, 1) respectively.

(1 point)

3. Write a function to visualise these clusters and their labels as different colors using matplotlib.pyplot.plot **(0.5 points)**

Exercise 2

This exercise will use the scikit-learn library to find a decision boundary to separate your clusters.

1. Use sklearn.svm.LinearSVC to find a linear decision boundary for your classification. Initialise it with the default parameters and use the .fit() method to optimise the linear model. **(2 points)**
2. Write a function to visualise the decision boundary predicted by your model. **(1 point)**
3. Optimise the model and visualise the decision boundary for both your datasets.

Exercise 3

Analyse what you have done so far.

1. Explain your observations from Exercise 2. Explain why the model worked in one case and not the other. **(1.5 points)**
2. Is this decision boundary unique ? **(1 point)**
3. Create some outliers by flipping the class of randomly selected 8 points in the first dataset. What was the change in the predicted decision boundary ? **(1 point)**