Deep Learning (IST, 2022-23)

Practical 2: Perceptron

André Martins, Andreas Wichert, Luis Sá-Couto

Question 1

Consider the following linearly separable training set:

$$\boldsymbol{x}^{(1)} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{x}^{(2)} = \begin{bmatrix} 0 \\ 0.25 \end{bmatrix}, \boldsymbol{x}^{(3)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}^{(4)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$y^{(1)} = -1, y^{(2)} = +1, y^{(3)} = +1, y^{(4)} = -1.$$

1. Initialize all weights to zero (including the bias). Assume $\text{sign}(z) = +1$ iff $z \geq 0$, and -1 if $z < 0$. Use a learning rate of one. Apply the perceptron learning algorithm until convergence. How many epochs does it take to converge?
2. Draw the separation hyperplane.
3. What is the perceptron output for the query point $\begin{bmatrix} 0 & 1 \end{bmatrix}^\top$?
4. Change the initialization of weights and biases to be random with a standard normal distribution $\mathcal{N}(0, 1)$. Try multiple times. Does it always converge?

Question 2

1. Generate a balanced dataset with 30 examples in $\mathbb{R}^2$ and 3 classes. Assume each of the 10 inputs associated to class $k \in \{0, 1, 2\}$ is generated as $x \sim \mathcal{N}(\mu_k, \sigma_k^2 I)$, with $\sigma_0 = \sigma_1 = \sigma_2 = 1$, $\mu_0 = [0, 0]^\top$, $\mu_1 = [0, 3]^\top$, and $\mu_2 = [2, 2]^\top$. Plot the data.
2. Implement the multi-class perceptron algorithm and run 100 iterations. Initialize all the weights to zero and use a learning rate of one. What is the training accuracy (fraction of points that are correctly classified)?

Question 3

The perceptron can learn a relatively large number of functions. In this exercise, we focus on simple logical functions.

1. Show graphically that a perceptron can learn the logical NOT function. Give an example with specific weights.
2. Show graphically that a perceptron can learn the logical AND function for two inputs. Give an example with specific weights.

3. Show graphically that a perceptron can learn the logical OR function for two inputs. Give an example with specific weights.

4. Show graphically that a perceptron can not learn the logical XOR function for two inputs.

Question 4

Now it's time to try the perceptron on real data and see what happens.

1. Load the UCI handwritten digits dataset using `scikit-learn`:

   ```
   from sklearn.datasets import load_digits
   data = load_digits()
   ```

   This is a dataset containing 1797 8x8 input images of digits, each corresponding to one out of 10 output classes. You can print the dataset description and visualize some input examples with:

   ```
   print(data.DESCR)

   import matplotlib.pyplot as plt
   plt.gray()
   for i in range(10):
       plt.matshow(data.images[i])
   plt.show()
   ```

   Randomly split this data into training (80%) and test (20%) partitions. This can be done with:

   ```
   from sklearn.model_selection import train_test_split
   X_train, X_test, y_train, y_test = train_test_split(
       X, y, test_size=0.2, random_state=42)
   ```

2. Run your implementation of the multi-class perceptron algorithm on this dataset. Measure the training and test accuracy.

3. Use `scikit-learn`'s implementation of the perceptron algorithm. This can be done with

   ```
   from sklearn.linear_model import Perceptron
   clf = Perceptron(fit_intercept=False, shuffle=False)
   clf.fit(X_train, y_train)
   print(clf.score(X_train, y_train))
   print(clf.score(X_test, y_test))
   ```

   Compare the resulting accuracies.