Deep Learning Assignment 1 in FS 2022

Perceptron Learning

Microsoft Forms Document:
https://forms.office.com/r/6pxGSqCxzM

Manuel Günther

Distributed: Friday, February 25, 2022
Discussed: Friday, March 4, 2022

The goal of this exercise is to apply the perceptron learning to a total of $N = 100$ automatically generated, separable random data $X = \left\{\vec{x}^{[1]}, \vec{x}^{[2]}, \dots, \vec{x}^{[N]}\right\}$ with each $\vec{x}^{[n]} = \left(x_1^{[n]}, x_2^{[n]}\right)^{\mathrm{T}}$. Each data point $\vec{x}^{[n]}$ is accompanied by an according target value $X = \left\{t^{[1]}, t^{[2]}, \dots, t^{[N]}\right\}$ with $t^{[n]} \in \{-1, +1\}$.

1 Data Generation

The data should be generated such that $\forall n \leq \frac{N}{2} \colon \vec{x}^{[n]} \sim \mathcal{N}_{\vec{\mu}_+, \sigma_+}$. These samples will be our positive data labeled with $t^{[n]} = 1$. Similarly, we generate our negative data with $\forall n > \frac{N}{2} \colon \vec{x}^{[n]} \sim \mathcal{N}_{\vec{\mu}_-, \sigma_-}$ and label them as $t^{[n]} = -1$.

Task 1 Implement a function that generates and returns the data such that the two classes are linearly separable, i.e., that it is possible to define a line:

$$a = f_{\vec{w}}(\vec{x}) = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 \,, \tag{1}$$

with the property $\forall n \colon a^{[n]} \cdot t^{[n]} > 0$ using $a^{[n]} = f_{\vec{w}}(x^{[n]})$.

Task 2 Select proper values for $\vec{\mu}_+, \vec{\mu}_-, \sigma_+, \sigma_-$ that allow you to generate a data set (X, T) for which you can manually define the weights $\vec{w}$ so that the line $f_{\vec{w}}(\vec{x})$ separates the positive and negative data.

Test Case 1 Write a test case that checks if a given set of data (X, T) is linearly separable with a given line parameters $\vec{w}$. Apply this function to the data from Task 1 and the weight vectors from Task 2.

2 Perceptron Learning

The perceptron is defined as the Adaline $a = f_{\vec{w}}(\vec{x})$ in (1), which is then thresholded using the sign function:

$$\mathrm{sign}(a) = \begin{cases} +1 & \text{if } a \geq 0 \\ -1 & \text{otherwise.} \end{cases} \tag{2}$$

Task 3 Implement a function that returns the output of the Perceptron for given $\vec{w}$ and $\vec{x}$.

As provided in the lecture, the perceptron learning rule is defined as follows. First, the weights $\vec{w} = (w_0, w_1, w_2)^{\mathrm{T}}$ are initialized randomly. Then, for each sample it is decided if the sample is correctly classified by checking if $\mathrm{sign}(a^{[n]}) \cdot t^{[n]} > 0$. For incorrectly classified samples, the weights are adapted as:

$$w_0 = w_0 + t^{[n]} \qquad w_1 = w_1 + t^{[n]} \cdot x_1^{[n]} \qquad w_2 = w_2 + t^{[n]} \cdot x_2^{[n]} \tag{3}$$

This is repeated until all samples are correctly classified.

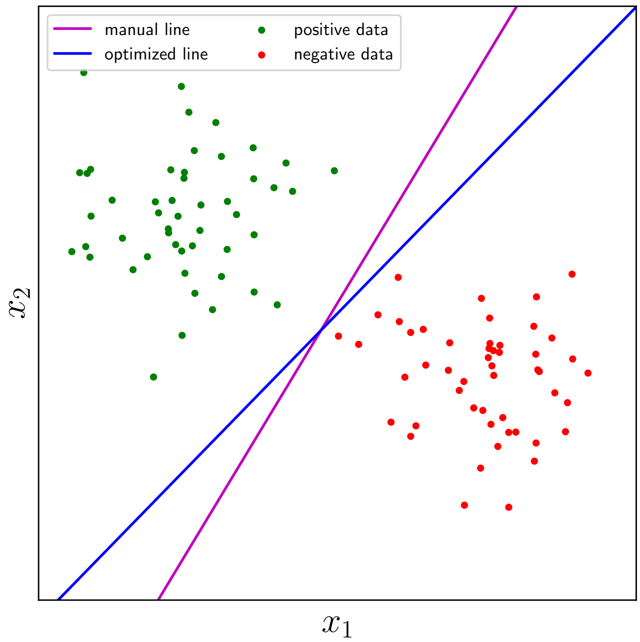

Figure 1: Exemplary plot resulting from Task 6.

Task 4 Implement a function that performs the perceptron learning for a given dataset (X, T) and a given weight vector $\vec{w}$. The final weight vector $\vec{w}^*$ shall be returned from that function. Define a proper stopping criterion for the iteration. Consider in your implementation error cases that could arise.

Test Case 2 Call this function with the data from task 1 and the manually-defined weights from Task 2. What is the expected outcome?

Task 5 Implement a function that generates and returns a randomly initialized weight vector $\vec{w} \in [-1, 1]^3$.

Task 6 Call the perceptron learning function with the data from Task 1 and a random weight vector from Task 4. Store the resulting weight vector $\vec{w}^*$.

Test Case 3 Implement a test case that verifies that the resulting weight vector $\vec{w}^*$ results in a line that separates the positive and negative data.

3 Visualization

We have selected our data to be 2-dimensional to be able to visualize the results. For this purpose, we would like to jointly plot the positive and the negative data from Task 1 together with the decision boundaries of the weight vectors obtained in Tasks 2 and 5. Finally, we want to export our plot into a file so that we can share the results.

Task 7 Use a scatter plot to visualize the positive data as green dots, and the negative data as red dots. Into the same figure, plot the manually designed decision boundary from task 1 as a magenta line and the automatically computed one from task 4 as a blue line. Define proper start and end points of your lines. Optionally add a legend, axis labels and a title to the figure. A sample solution can be found in Figure 1.