Tutorial Sections 11-12

1. Imagine you are developing a simple machine learning model to classify two types of plants in an ecology lab based on their height and leaf size. The plants are either 'Type A' or 'Type B'. The data you have available are as follows:

Sample	Height	Leaf Size	Type
A	1	2	A
B	2	2	B
C	3	1	A
D	3	3	B
E	2	1	A

   (a) Consider a classification model so that $f(w, x) = \text{sign}\left(w^T x\right)$, with $w_0 = [-0.5, 1]^T$, where the first and second elements refer to the weight of the height and leaf size, respectively (i.e., the model has no bias). Assuming that Type A is assigned $y = +1$ and Type B, $y = -1$, what would be the value of the $0 - 1$ loss function for these model parameters?

   (b) Starting with the same model parameters as the previous item, utilise the perceptron algorithm to classify the plants based on the two features above. Assume the maximum number of iterations $R = 4$ and a learning rate $\alpha = 0.1$.

   (c) If a new sample is collected with height of 2.5 and leaf size of 1.6, what would be the prediction of the above algorithm?

   (d) Plot the data points and the decision boundary in a 2D plot, and identify the new sample on the plot. Is the current model a good classifier? What could be changed in the problem to improve the boundary?

Solution: This classification problem covers topics from both L11 and L12. The tutorial is focused on this problem as we can use several iterations to discuss different characteristics of the perceptron algorithm.
Given the number of data points ($N = 5$) and no bias, we can write the features vector as:

$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix},$$

where x^1 represents the plant height and x^2, the leaf size. Using this notation, x_i represents the data for each sample ($i = 1, 2, ..., 5$), therefore, x_i^1 refers to the height of the i^{th} sample, and x_i^2 represents the leaf size for the i^{th} sample.

(a) The 0-1 loss function is defined as

$$F(w) = \sum_{i=1}^{N} \mathbb{I}\left[f(w, x_i) \neq y_i\right],$$

where

$$\mathbb{I}[P] = \begin{cases} 1 & \text{if logical condition } P \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

For the case $w_0^T = [w_0^1, w_0^2] = [-0.5, 1]$, the classification model yields

$$f(w_0, x) = \text{sign}\left(w_0^T x\right) = \text{sign}\left(w_0^1 x^1 + w_0^2 x^2\right)$$

for each data sample. If we consider the whole dataset,

$$f(w_0, X) = \text{sign}\left(\begin{bmatrix} -0.5 \times 1 + 1 \times 2 \\ -0.5 \times 2 + 1 \times 2 \\ -0.5 \times 3 + 1 \times 1 \\ -0.5 \times 3 + 1 \times 3 \\ -0.5 \times 2 + 1 \times 1 \end{bmatrix}\right) = \text{sign}\left(\begin{bmatrix} 3/2 \\ 1 \\ -1/2 \\ 3/2 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} +1 \\ +1 \\ -1 \\ +1 \\ 0 \end{bmatrix}$$

Considering $y = +1$ and $y = -1$ for Types A and B, respectively, the labelled samples can be written as

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \end{bmatrix},$$

and we can compare Y with the predicted values by the classification model, $f(w_0, X)$. Then, the value of the $0 - 1$ loss function is

$$\begin{aligned} F(w) &= \sum_{i=1}^{5} \mathbb{I}\left[f(w_0, x_i) \neq y_i\right] \\ &= \mathbb{I}\left[f(w_0, x_1) \neq y_1\right] + \mathbb{I}\left[f(w_0, x_2) \neq y_2\right] + \mathbb{I}\left[f(w_0, x_3) \neq y_3\right] + \mathbb{I}\left[f(w_0, x_4) \neq y_4\right] + \mathbb{I}\left[f(w_0, x_5) \neq y_5\right] \\ &= 0 + 1 + 1 + 1 + 1 = 4 \end{aligned}$$

(b) Using the perception algorithm, at $n = 0$ we have $w_0^T = \begin{bmatrix} -0.5 & 1 \end{bmatrix}$, with $\alpha = 0.01$ and $R = 4$. The values of $f(w_0, X) = w_0^T X$ were already calculated in part (a). So, we can use them to update the weights at $n = 1$. Using the gradient descent step of the perceptron algorithm, we initially set $w_1 = w_0$ and update the weights at each data point i:

$$i = 1 : \text{sign}\left(w_1^T x_1\right) = +1 = y_1 \rightarrow w_1 = \begin{bmatrix} -0.5 & 1 \end{bmatrix}^T \text{ (no change)}$$

$$i = 2 : \text{sign}\left(w_1^T x_2\right) = +1 \neq y_2 \rightarrow w_1 = w_1 + \alpha y_2 x_2 = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.7 \\ 0.8 \end{bmatrix}$$

$$i = 3 : \text{sign}\left(w_1^T x_3\right) = -1 \neq y_3 \rightarrow w_1 = w_1 + \alpha y_3 x_3 = \begin{bmatrix} -0.7 \\ 0.8 \end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.4 \\ 0.9 \end{bmatrix}$$

$$i = 4 : \text{sign}\left(w_1^T x_4\right) = +1 \neq y_4 \rightarrow w_1 = w_1 + \alpha y_4 x_4 = \begin{bmatrix} -0.4 \\ 0.9 \end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} -0.7 \\ 0.6 \end{bmatrix}$$

$$i = 5 : \text{sign}\left(w_1^T x_5\right) = -1 \neq y_5 \rightarrow w_1 = w_1 + \alpha y_5 x_5 = \begin{bmatrix} -0.7 \\ 0.6 \end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.7 \end{bmatrix}$$

Therefore, after $n = 1$ we have $w_1^T = \begin{bmatrix} -0.5 & 0.7 \end{bmatrix}^T$. For $n = 2$, we initially set $w_2 = w_1$ and reevaluate all data points:

$$i = 1 : \text{sign}\left(w_2^T x_1\right) = +1 = y_1 \rightarrow w_2 = \begin{bmatrix} -0.5 & 0.7 \end{bmatrix}^T \text{ (no change)}$$

$$i = 2 : \text{sign}\left(w_2^T x_2\right) = +1 \neq y_2 \rightarrow w_2 = w_2 + \alpha y_2 x_2 = \begin{bmatrix} -0.5 \\ 0.7 \end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.7 \\ 0.5 \end{bmatrix}$$

$$i = 3 : \text{sign}\left(w_2^T x_3\right) = -1 \neq y_3 \rightarrow w_2 = w_2 + \alpha y_3 x_3 = \begin{bmatrix} -0.7 \\ 0.5 \end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.4 \\ 0.6 \end{bmatrix}$$

$$i = 4 : \text{sign}\left(w_2^T x_4\right) = +1 \neq y_4 \rightarrow w_2 = w_2 + \alpha y_4 x_4 = \begin{bmatrix} -0.4 \\ 0.6 \end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} -0.7 \\ 0.3 \end{bmatrix}$$

$$i = 5 : \text{sign}\left(w_2^T x_5\right) = 0 \neq y_5 \rightarrow w_2 = w_2 + \alpha y_5 x_5 = \begin{bmatrix} -0.7 \\ 0.3 \end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.4 \end{bmatrix}$$

Similarly, for $n = 3$, we have $w_3 = w_2$ and:

$$i = 1 : \text{sign}\left(w_3^T x_1\right) = +1 = y_1 \rightarrow w_3 = \begin{bmatrix}-0.5 & 0.4\end{bmatrix}^T \text{ (no change)}$$
$$i = 2 : \text{sign}\left(w_3^T x_2\right) = -1 = y_2 \rightarrow w_3 = \begin{bmatrix}-0.5 & 0.4\end{bmatrix}^T \text{ (no change)}$$
$$i = 3 : \text{sign}\left(w_3^T x_3\right) = -1 \neq y_3 \rightarrow w_3 = w_3 + \alpha y_3 x_3 = \begin{bmatrix}-0.5\\0.4\end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix}3\\1\end{bmatrix} = \begin{bmatrix}-0.2\\0.5\end{bmatrix}$$
$$i = 4 : \text{sign}\left(w_3^T x_4\right) = +1 \neq y_4 \rightarrow w_3 = w_3 + \alpha y_4 x_4 = \begin{bmatrix}-0.2\\0.5\end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix}3\\3\end{bmatrix} = \begin{bmatrix}-0.5\\0.2\end{bmatrix}$$
$$i = 5 : \text{sign}\left(w_3^T x_5\right) = 0 \neq y_5 \rightarrow w_3 = w_3 + \alpha y_5 x_5 = \begin{bmatrix}-0.5\\0.2\end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}-0.3\\0.3\end{bmatrix}$$

For $n = 4$,

$$i = 1 : \text{sign}\left(w_4^T x_1\right) = +1 = y_1 \rightarrow w_4 = \begin{bmatrix}-0.3 & 0.3\end{bmatrix}^T \text{ (no change)}$$
$$i = 2 : \text{sign}\left(w_4^T x_2\right) = 0 \neq y_2 \rightarrow w_4 = w_4 + \alpha y_2 x_2 = \begin{bmatrix}-0.3\\0.3\end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}-0.5\\0.1\end{bmatrix}$$
$$i = 3 : \text{sign}\left(w_4^T x_3\right) = -1 \neq y_3 \rightarrow w_4 = w_4 + \alpha y_3 x_3 = \begin{bmatrix}-0.5\\0.1\end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix}3\\1\end{bmatrix} = \begin{bmatrix}-0.2\\0.2\end{bmatrix}$$
$$i = 4 : \text{sign}\left(w_4^T x_4\right) = 0 \neq y_4 \rightarrow w_4 = w_4 + \alpha y_4 x_4 = \begin{bmatrix}-0.2\\0.2\end{bmatrix} + 0.1 \times (-1) \times \begin{bmatrix}3\\3\end{bmatrix} = \begin{bmatrix}-0.5\\-0.1\end{bmatrix}$$
$$i = 5 : \text{sign}\left(w_4^T x_5\right) = -1 \neq y_5 \rightarrow w_4 = w_4 + \alpha y_5 x_5 = \begin{bmatrix}-0.5\\-0.1\end{bmatrix} + 0.1 \times (+1) \times \begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}-0.3\\0\end{bmatrix},$$

and since $n = R$, we exit with solution $w^\star = w_4 = \begin{bmatrix}-0.3\\0\end{bmatrix}$.

Note: to those who attended the Thursday tutorial in Edgbaston and attempted to use the gradient descent algorithm considering the perceptron classifier as the objective function (since we hadn't seen the perceptron algorithm at that time), the solution to this problem is given below. However, keep in mind that the correct solution to the problem of the tutorial is given above, using the perceptron algorithm. The value of the parameter updates will change as the gradient will be updated at every iteration as the sum across all data points. Here's the complete solution using SGD with the perceptron classifier:

$$F(w) = \sum_{i=1}^{5} \max\left(0, -y_i w^T x_i\right),$$

whose gradient is given by

$$F_w(w) = -\sum_{i=1}^{5} y_i x_i \mathbb{I}\left[-y_i w^T x_i \geq 0\right].$$

Considering $w_0^T = [-0.5, 1]$, we would have to compute the value of $y_i x_i$ for all the data points whose model doesn't match the label (i.e., points in which $-y_i w^T x_i \geq 0$). For w_0, these are data points $i = 2, 3, 4, 5$. Therefore, we have

$$\begin{aligned} F(w_0) &= \max(0, -1.5) + \max(0, 1) + \max(0, 0.5), \max(0, 1.5) + \max(0, 0) \\ &= 0 + 1 + 0.5 + 1.5 + 0 = 3 \end{aligned}$$

The error function gradient is given by

$$\begin{aligned} F_w(w_0) &= -\left((-1)\begin{bmatrix}2\\2\end{bmatrix} + (+1)\begin{bmatrix}3\\1\end{bmatrix} + (-1)\begin{bmatrix}3\\3\end{bmatrix} + (+1)\begin{bmatrix}2\\1\end{bmatrix}\right) \\ &= -\left(\begin{bmatrix}-2\\-2\end{bmatrix} + \begin{bmatrix}3\\1\end{bmatrix} - \begin{bmatrix}3\\3\end{bmatrix} + \begin{bmatrix}2\\1\end{bmatrix}\right) = -\left(\begin{bmatrix}0\\-3\end{bmatrix}\right) = \begin{bmatrix}0\\3\end{bmatrix} \end{aligned}$$

so that the new model parameter is given by

$$w_1 = w_0 - \alpha F_w(w_0) = \begin{bmatrix}-0.5\\1\end{bmatrix} - 0.1\begin{bmatrix}0\\3\end{bmatrix} = \begin{bmatrix}-0.5\\0.7\end{bmatrix}.$$

From w_1, we can compute prediction for each data point,

$$f(w_1, X) = \text{sign}\left(w_1^T X\right) = \text{sign}\left(\begin{bmatrix}0.9\\0.4\\-0.8\\0.6\\-0.3\end{bmatrix}\right) = \begin{bmatrix}+1\\+1\\-1\\+1\\-1\end{bmatrix},$$

which yields a perceptron error of $F(w_1) = 2.1$. Repeating the process, we will find that:

$$F_w(w_1) = \begin{bmatrix}0\\3\end{bmatrix}, w_2 = \begin{bmatrix}-0.5\\0.4\end{bmatrix}, f(w_2, X) = \text{sign}\left(\begin{bmatrix}0.3\\-0.2\\-1.1\\-0.3\\-0.6\end{bmatrix}\right) = \begin{bmatrix}+1\\-1\\-1\\-1\\-1\end{bmatrix}, F(w_2) = 1.7$$

$$F_w(w_2) = \begin{bmatrix}-5\\-2\end{bmatrix}, w_3 = \begin{bmatrix}0\\0.6\end{bmatrix}, f(w_3, X) = \text{sign}\left(\begin{bmatrix}1.2\\1.2\\0.6\\1.8\\0.6\end{bmatrix}\right) = \begin{bmatrix}+1\\+1\\+1\\+1\\+1\end{bmatrix}, F(w_3) = 3$$

$$F_w(w_3) = \begin{bmatrix}5\\5\end{bmatrix}, w_4 = \begin{bmatrix}-0.5\\0.1\end{bmatrix}, f(w_4, X) = \text{sign}\left(\begin{bmatrix}-0.3\\-0.8\\-1.4\\-1.2\\-0.9\end{bmatrix}\right) = \begin{bmatrix}-1\\-1\\-1\\-1\\-1\end{bmatrix}$$

(c) For a new sample, $x_6 = [2.5, 1.6]^T$, and considering the optimal solution obtained in part (b) with the perceptron algorithm, $w^\star = \begin{bmatrix}-0.3 & 0\end{bmatrix}^T$, the function would be

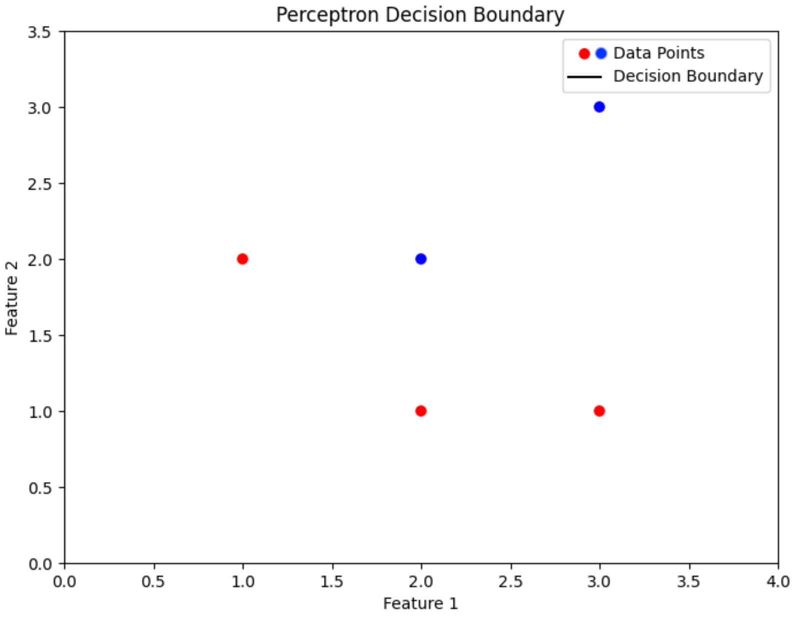
$$\begin{aligned} f(w^\star, x_6) = \text{sign}\left(w_4^T x_6\right) &= \text{sign}\left(w_4^1 x_6^1 + w_4^2 x_6^2\right) \\ &= \text{sign}\left(-0.3 \times 2.5 + 0 \times 1.6\right) = \text{sign}(-0.75 + 0) = -1, \end{aligned}$$

which implies that the new sample would be assigned Type B.

(d) Considering the current optimal solution, the decision boundary would be a vertical line at $x^1 = 0$, independent of x^2. This is not a good decision boundary. We could attempt to run the algorithm with a higher R, start with a different initial guess for the parameters, or change the learning rate.

However, one of the main reasons for not having a good decision boundary in this problem is how we defined the set of weights; without the constant term (bias), we can only find decision boundaries that must pass the $(0, 0)$ point, which severely restricts our possibilities.

This example illustrates (1) the reason for a bias to exist, and (2) the importance of a good model!


Perceptron Decision Boundary
Data Points
Decision Boundary
Feature 2
Feature 1