BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI – HYDERABAD CAMPUS
BITS F464 : MACHINE LEARNING
FIRST SEMESTER 2023-2024

A worked-out example on Perceptron

Here's a step-by-step solved numerical problem on the perceptron learning algorithm.

Let's take the dataset related to the behaviour of the AND gate, which returns true if and only if both inputs are true:

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

We want to perform the perceptron learning algorithm to learn a linear decision boundary that separates true outputs from false outputs.

Step 1: Initialize the weights and bias:

We start by initializing the weights w_1, w_2 and bias b to small random values.

Let's assume we start with: $w_1=0.4$, $w_2=0.2$, and $b=-0.3$.

Step 2: Calculate the weighted sum and predict the output:

For each data point, calculate the weighted sum z and predict the output $\hat{y}$:

(i) **For Input Pair (0, 0) [$y = 0$]:**

$$z = w_1 \times 0 + w_2 \times 0 + b = 0 \times 0.4 + 0 \times 0.2 - 0.3 = -0.3$$

$$\hat{y} = \begin{cases} 1, & if\ z > 0 \\ 0, & if\ z \leq 0 \end{cases} = 0$$

Correctly classified.

(ii) **For Input Pair (0, 1) [$y = 0$]:**

$$z = w_1 \times 0 + w_2 \times 1 + b = 0 \times 0.4 + 1 \times 0.2 - 0.3 = -0.1$$

$$\hat{y} = 0$$

Correctly classified.

(iii) **For Input Pair (1, 0) [$y = 0$]:**

$z = w_1 \times 1 + w_2 \times 0 + b = 1 \times 0.4 + 0 \times 0.2 - 0.3 = 0.1$

$\hat{y} = 1$

Misclassified.

(iv) For Input Pair (1, 1) [*y = 1*]:

$z = w_1 \times 1 + w_2 \times 1 + b = 1 \times 0.4 + 1 \times 0.2 - 0.3 = 0.3$

$\hat{y} = 1$

Correctly classified.

Step 3: Update the weights and bias:

For misclassified input pairs ($y \neq \hat{y}$), update the weights and bias using the perceptron learning rule:

$$w_j := w_j + \alpha \times (y - \hat{y}) \times x_j$$

$$b := b + \alpha \times (y - \hat{y})$$

where α is the learning rate, j is the index of the weight, y is the actual output, and $\hat{y}$ is the predicted output. Taking the learning rate $\alpha=0.1$, we have:

For Input Pair (1, 0) [$y = 0, \hat{y} = 1$]:

$w_1' := 0.4 + 0.1 \times (0 - 1) \times 1 = 0.3$
$w_2' := 0.2 + 0.1 \times (0 - 1) \times 0 = 0.2$
$b' := -0.3 + 0.1 \times (0 - 1) = -0.4$

$z' = w_1' \times 1 + w_2' \times 0 + b' = 1 \times 0.3 + 0 \times 0.2 - 0.4 = -0.1$

$\hat{y}' = 0 \quad (= y)$

Similarly, we can check that these updated weights correctly classify all the four input pairs.

Step 4: Repeat steps 2 and 3:

Repeat steps 2 and 3 for multiple iterations until all input pairs are either classified correctly or a stopping criterion is met. The weights and bias are updated based on the misclassified points, and the algorithm continues to update the parameters until a decision boundary is learned that can accurately classify the points as false (0) or true (1).

In this example, there was only one misclassified input pair and thus required only one round of updates. However, depending upon the choice of initial weights, there can be multiple misclassified points. The update step has to be repeated for all such points.