1. Single Layer Perceptron

January 21, 2026

Author : Vaibhav Thakur

Numerical Example on Single Layer Perceptron

Inputs: $f(x) = [x_1, x_2] = [1.0, 2.0]$ **Target:** $y = 1$ **Weights:** $[w_1, w_2] = [0.2, -0.5]$ **Bias:** $b = 0.1$ **Learning rate:** $\eta = 0.1$ **Prediction rule:** If the computed value is ≤ 0, then predict **0**, otherwise predict **1**.

[12]:
```
import numpy as np
```

[19]:
```
# Given data

x = np.array([1.0, 2.0])   # Inputs
y = 1                      # Target
w = np.array([0.2, -0.5])  # Initial weights
b = 0.1                    # Initial bias
eta = 0.1                  # Learning rate
```

[18]:
```
# Run for 3 epochs

for epoch in range(1,4):
    print(f'\nEpoch {epoch}: ')
    net_input = np.dot(w, x) + b
    y_pred = 1 if net_input > 0 else 0
    error = y - y_pred

    print(f' Net input = {net_input:.2f}')
    print(f' Predicted output = {y_pred}')
    print(f' Error = {error}')

    # Update rule

    w = w + eta * error * x
    b = b + eta * error

    print(f' Updated weights = {w}')
    print(f' Updated bias = {b:.2f}')
```

```
Epoch 1:
 Net input = 0.50
 Predicted output = 1
 Error = 0
 Updated weights = [ 0.4 -0.1]
 Updated bias = 0.30

Epoch 2:
 Net input = 0.50
 Predicted output = 1
 Error = 0
 Updated weights = [ 0.4 -0.1]
 Updated bias = 0.30

Epoch 3:
 Net input = 0.50
 Predicted output = 1
 Error = 0
 Updated weights = [ 0.4 -0.1]
 Updated bias = 0.30
```

[16]:
```python
# ---- Testing phase ----

x_test = np.array([2.0, 1.0])
net_test = np.dot(w, x_test) + b
y_test_pred = 1 if net_test > 0 else 0

print('\nTesting phase: ')
print(f'  Test Input = {x_test}')
print(f'   Net input = {net_test: .2f}')
```

```
Testing phase:
  Test Input = [2. 1.]
   Net input =  1.00
```