perceptron

July 2, 2024

1 Perceptron Implementation

1.0.1 Imports

[30]:
```
import numpy as np
import matplotlib.pyplot as plt
```

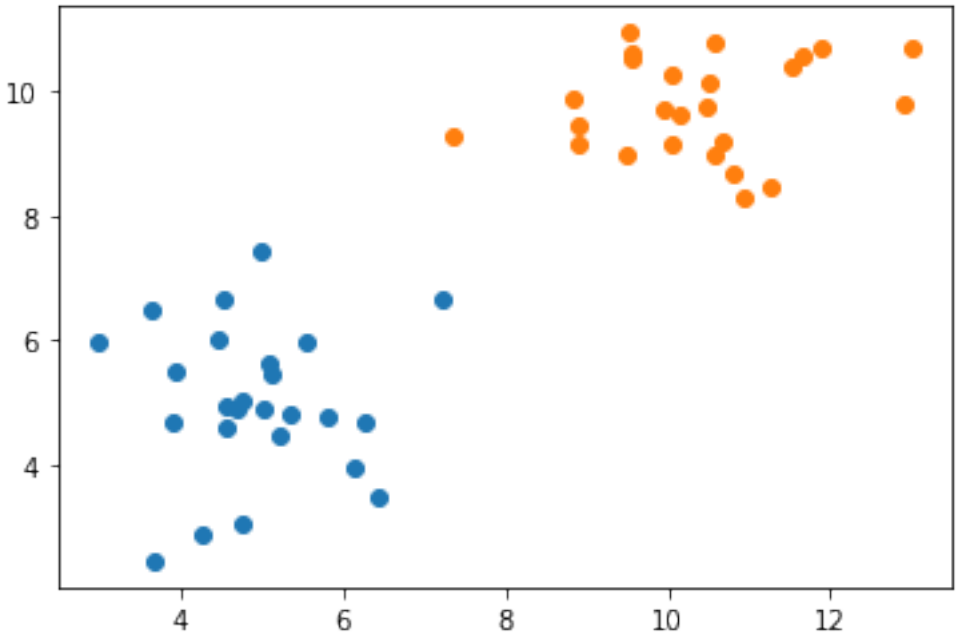
1.0.2 Generating Test Dataset

[31]:
```
class1 = [np.random.normal(5, 1, 2) for x in range(25)]
class2 = [np.random.normal(10, 1, 2) for x in range(25)]
```

[32]:
```
x_data1 = [x[0] for x in class1]
y_data1 = [x[1] for x in class1]
x_data2 = [x[0] for x in class2]
y_data2 = [x[1] for x in class2]
```

[33]:
```
plt.scatter(x_data1, y_data1)
plt.scatter(x_data2, y_data2)
```

[33]:
```
<matplotlib.collections.PathCollection at 0x15daa9c2490>
```



[34]:
```
class1 = list(zip(class1, np.zeros(25)))
class2 = list(zip(class2, np.ones(25)))
data = class1 + class2
```

1.0.3 Perceptron Algorithm

[42]:
```
def predict(point, weights):
    activation = weights[0]
    for i in range(len(point)):
        activation += weights[i + 1] * point[i]
    return 1 if activation >= 0 else 0
```

[47]:
```
def train_perceptron(data, learning_rate, epochs):
    # Step 1: Initialize weights
    weights = np.zeros(len(data[0][0]) + 1)
    for epoch in range(epochs):
        for point in data:
            # Step 2: For each point, make a prediction based on the current␣
 ↪weights and compute error
            prediction = predict(point[0], weights)
            error = point[1] - prediction
            # Step 3: For each point's error, update weights
            weights[0] += learning_rate * error
            for i in range(len(weights) - 1):
                weights[i + 1] += learning_rate * error * point[0][i]
```

```
    return weights
```

1.0.4 Test the Algorithm

[138]:
```
learning_rate = 0.05
epochs = 1000

weights = train_perceptron(data, learning_rate, epochs)
```

[148]:
```
slope = -(weights[0] / weights[2]) / (weights[0] / weights[1])
intercept = -weights[0] / weights[2]

x = np.arange(2, 14, 1)
y = slope * x + intercept
```

[149]:
```
plt.plot(x, y)
plt.scatter(x_data1, y_data1)
plt.scatter(x_data2, y_data2)
```

[149]:
```
<matplotlib.collections.PathCollection at 0x15daabe6f70>
```