The Perceptron Algorithm

1. Overview

The Perceptron is a foundational algorithm in machine learning, used primarily for binary linear classification. It attempts to find a separating hyperplane for data labeled in two classes.

2. Mathematical Formulation

Given a dataset $\{(\mathbf{x}_i, y_i)\}$ where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, +1\}$, the Perceptron learns a weight vector $\mathbf{w}$ and bias b.

$$\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b).$$

3. Learning Rule

1. Initialize $\mathbf{w} = \mathbf{0}$, $b = 0$.
2. For each training example $(\mathbf{x}_i, y_i)$:
   (a) Predict $\hat{y}_i = \text{sign}(\mathbf{w} \cdot \mathbf{x}_i + b)$.
   (b) If $y_i \hat{y}_i \leq 0$ (misclassified), update:

$$\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i, \quad b \leftarrow b + y_i.$$

4. Simple Demo Code (Python)

```
import numpy as np

def perceptron_train(X, y, max_iter=100):
    w = np.zeros(X.shape[1])
    b = 0
    for _ in range(max_iter):
        for i in range(len(X)):
            pred = np.sign(np.dot(w, X[i]) + b)
            # handle sign(0) case
            if pred == 0:
                pred = 1
            if y[i] * pred <= 0:
                w += y[i] * X[i]
                b += y[i]
    return w, b
```

```
if __name__ == "__main__":
    X = np.array([[2,1], [1,2], [2,2], [-1,-2], [-2,-1], [-2,-2]])
    y = np.array([1, 1, 1, -1, -1, -1])
    w, b = perceptron_train(X, y)
    print("Weights:", w)
    print("Bias:", b)
```

5. Key Points

- Converges if the data is linearly separable.
- Often considered the simplest neural network model.
- Variations include different learning rates and more advanced convergence criteria.