Perceptron

David Robinson

A **perceptron** classifies data into one of two categories by finding a decision boundary that separates the two classes.

Training Process

1. Sees a data point and makes a prediction
2. Checks if the prediction was correct
3. Adjusts its weights to be more accurate next time if the prediction was wrong

Applications

- Spam Detection
- Recommendation Systems
- Predicting User Interest

A **linear decision surface** is a line (or hyperplane for higher dimensions) that separates data into two categories.

$$h_1(x) = w_1 \cdot x + w_0$$

Algorithm

- **Initial conditions**: Initialize the w_1 weight vector to be a vector of all zeros.
- **Mistake on a positive example**: If the algorithm predicts that the example is negative but it's actually positive, the weight vector is increased.

$$w_{t+1} = w_t + x$$

- **Mistake on a negative example**: If the algorithm predicts that the example is positive but it's actually negative, the weight vector is decreased.

$$w_{t+1} = w_t - x$$

Geometric Margin

The **geometric margin** is how far the closest point is to the decision boundary. A larger margin usually shows a more confident and accurate classification, while a smaller margin makes the classification less confident.

Mistake Bound

If the data has margin γ and all points are inside a ball of radius R, then the perceptron makes less than $(R/\gamma)^2 + 1$ mistakes.

Strengths of the Perceptron

- **Simplicity**: The perceptron is easy to understand and implement.
- **Guaranteed Convergence**: If the data is linearly separable, the perceptron is guaranteed to find a correct decision boundary.
- **Online Learning**: The perceptron can be used for online learning, where data points are passed in one by one and the weights are continuously learned.

Limitations of the Perceptron

- **Linearly Separable Data**: The perceptron works only when the data is linearly separable.
- **No Confidence Scores**: Unlike more advanced algorithms, the perceptron doesn't provide confidence or probability estimates for its predictions.
- **Sensitive to Noisy Data**: If the data contains outliers or mislabeled points, the perceptron may struggle to find a good boundary.