Report on Neural Networks

Contents

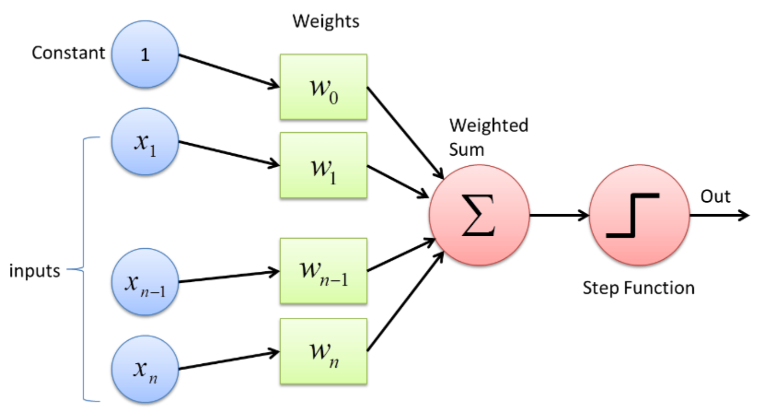
1 Introduction

This report provides a mathematical formulation of three algorithms: Perceptron, Logistic Regression, and Multilayer Perceptron (MLP). Each algorithm is explained in terms of its architecture, data representation, mathematical formulation, and gradient descent optimization.

2 Perceptron

2.1 Model Architecture

The Perceptron is a single-layer neural network model. Its architecture is depicted in Figure 1.



Source

Figure 1: Perceptron Architecture. The model takes inputs, applies weights, computes the weighted sum, and passes it through a step activation function to produce the output.

2.2 Vector Representation of Data

- Input vector: $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$
- Weight vector: $\mathbf{w} = [w_1, w_2, \ldots, w_n]$
- Bias: b
- Output: $y \in \{0, 1\}$

2.3 Mathematical Formulation

2.3.1 Linear Combination

$$z = \mathbf{w}^\top \mathbf{x} + b$$

2.3.2 Activation Function

$$\hat{y} = \text{step}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

2.3.3 Loss Function

No explicit loss function is used; weights are updated based on misclassified examples.

2.4 Gradient Descent and Weight Updates

The weights are updated using the following rule:

$$w_j := w_j + \eta(y - \hat{y})x_j, \quad b := b + \eta(y - \hat{y})$$

3 Logistic Regression

3.1 Model Architecture

The Logistic Regression model is depicted in Figure 2. It computes a weighted sum of inputs, applies the sigmoid activation function to output a probability, and optionally applies a threshold function for classification.

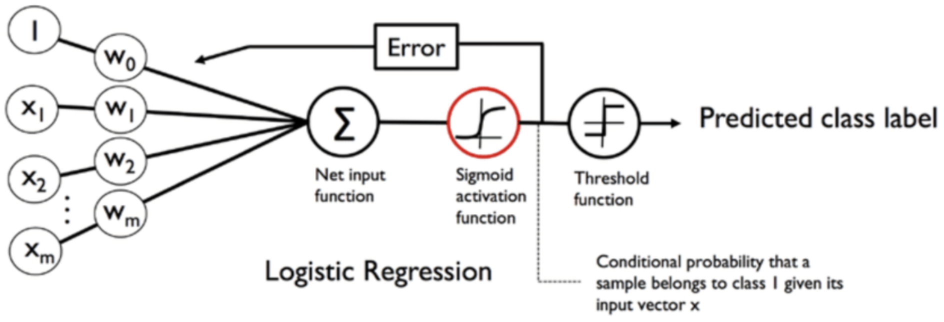


Figure 2: Logistic Regression Architecture. The model predicts the conditional probability of a sample belonging to class 1 given its input vector $\mathbf{x}$.

3.2 Vector Representation of Data

- Input vector: $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$
- Weight vector: $\mathbf{w} = [w_1, w_2, \ldots, w_n]$
- Bias: b
- Output: $y \in [0, 1]$

3.3 Mathematical Formulation

3.3.1 Linear Combination

$$z = \mathbf{w}^\top \mathbf{x} + b$$

3.3.2 Activation Function

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

3.3.3 Loss Function

The loss function for Logistic Regression is the binary cross-entropy:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{m} \sum_{i=1}^{m} \left(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right)$$

3.4 Gradient Descent and Weight Updates

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) x_j^{(i)}, \quad \frac{\partial \mathcal{L}}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)$$

$$w_j := w_j - \eta \frac{\partial \mathcal{L}}{\partial w_j}, \quad b := b - \eta \frac{\partial \mathcal{L}}{\partial b}$$

4 Multilayer Perceptron (MLP)

4.1 Model Architecture

The architecture of a Multilayer Perceptron (MLP) is depicted in Figure 3. It consists of an input layer, one or more hidden layers, and an output layer. Each layer is fully connected, and biases are added at each layer.

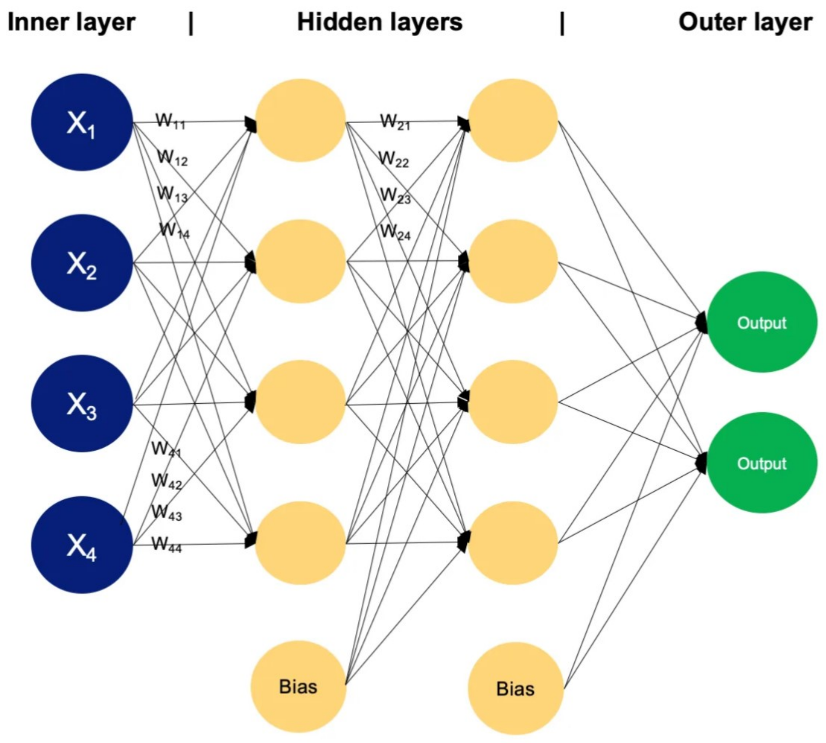


Figure 3: Multilayer Perceptron (MLP) Architecture. The network consists of an input layer, hidden layers, and an output layer with weighted connections and bias terms.

4.2 Vector Representation of Data

- Input vector: $\mathbf{x} \in \mathbb{R}^n$
- Weight matrices: $\mathbf{W}^{[l]}$ for layer l
- Bias vectors: $\mathbf{b}^{[l]}$
- Activations: $\mathbf{a}^{[l]}$

4.3 Mathematical Formulation

4.3.1 Linear Combination

For layer l:

$$z^{[l]} = \mathbf{W}^{[l]}\mathbf{a}^{[l-1]} + \mathbf{b}^{[l]}$$

4.3.2 Activation Function

$$\mathbf{a}^{[l]} = \phi(z^{[l]})$$

Common choices for ϕ: ReLU, sigmoid, or tanh.

4.3.3 Loss Function

For classification: cross-entropy loss. For regression: mean squared error (MSE).

4.4 Gradient Descent and Backpropagation

4.4.1 Backpropagation Rule

$$\delta^{[l]} = (\mathbf{a}^{[l]} - \mathbf{y}) \cdot \phi'(z^{[l]})$$

4.4.2 Weight and Bias Updates

$$\mathbf{W}^{[l]} := \mathbf{W}^{[l]} - \eta\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{[l]}}, \quad \mathbf{b}^{[l]} := \mathbf{b}^{[l]} - \eta\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{[l]}}$$

5 Conclusion

This report provides a detailed mathematical formulation and explanation of the Perceptron, Logistic Regression, and MLP algorithms. These models form the foundation of neural networks and are optimized using gradient descent.