Neural Networks Report

Gorelkin Alexey 5130203/20102

November 2024

1 Perceptron

1.1 Model Architecture

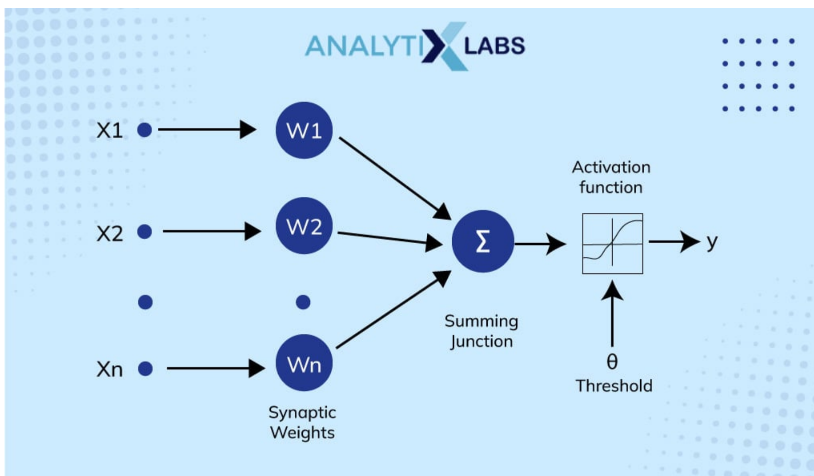

Figure 1: Model Architecture

- **Vector Representation of Data**
  - Inputs: $\mathbf{x} = [x_1, x_2, \ldots, x_n]$
  - Output: $y \in \{0, 1\}$
- **Mathematical Formulation**
  1. Linear Combination:
     $$z = \mathbf{w}^T \mathbf{x} + b$$
     where $\mathbf{w} = [w_1, w_2, \ldots, w_n]$ are the weights and b is the bias.

2. Activation Function (Step Function):

$$\hat{y} = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}$$

3. Loss Function (Binary Cross-Entropy):

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

1.2 Predictions Calculation

$$\hat{y} = \phi(z) = \begin{cases} 1, & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0 \\ 0, & \text{if } \mathbf{w}^T\mathbf{x} + b < 0 \end{cases}$$

1.3 Gradient Descent Algorithm

Gradient descent updates weights and biases to minimize the loss function iteratively. The update rule is:

$$\mathbf{w} = \mathbf{w} - \eta \nabla L, \quad b = b - \eta \frac{\partial L}{\partial b}$$

where η is the learning rate.

1.4 Gradients and Updates

1. Gradient of the Loss:

$$\frac{\partial L}{\partial w_i} = (\hat{y} - y)x_i, \quad \frac{\partial L}{\partial b} = \hat{y} - y$$

2. Weights and Biases Update:

$$w_i \leftarrow w_i - \eta \frac{\partial L}{\partial w_i}, \quad b \leftarrow b - \eta \frac{\partial L}{\partial b}$$

2 Logistic Regression

2.1 Model Architecture

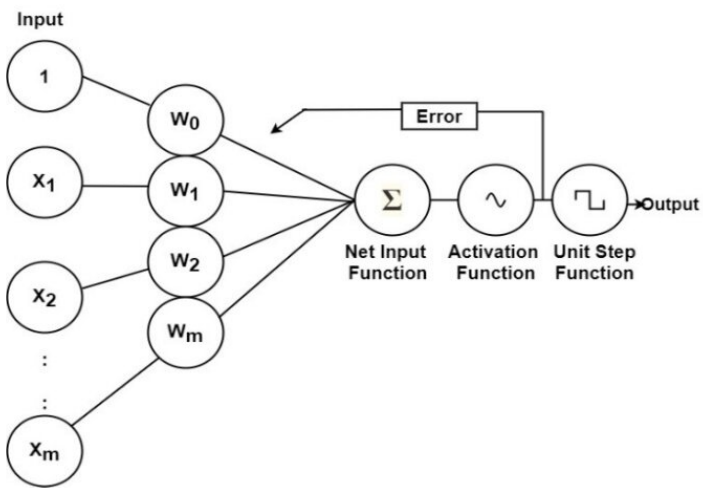

Figure 2: Model Architecture

- **Vector Representation of Data**
  - Inputs: $\mathbf{x} = [x_1, x_2, \ldots, x_n]$
  - Output: Probability $\hat{y} \in [0, 1]$
- **Mathematical Formulation**
  1. Linear Combination:
  $$z = \mathbf{w}^T \mathbf{x} + b$$
  2. Activation Function (Sigmoid):
  $$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$
  3. Loss Function (Binary Cross-Entropy):
  $$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

2.2 Predictions Calculation

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

2.3 Gradient Descent Algorithm

The weights and biases are updated using the loss function's gradient to minimize errors.

2.4 Gradients and Updates

1. Gradient of the Loss:

$$\frac{\partial L}{\partial w_i} = (\hat{y} - y)x_i, \quad \frac{\partial L}{\partial b} = \hat{y} - y$$

2. Weights and Biases Update:

$$w_i \leftarrow w_i - \eta \frac{\partial L}{\partial w_i}, \quad b \leftarrow b - \eta \frac{\partial L}{\partial b}$$

3 Multilayer Perceptron (MLP)

3.1 Model Architecture

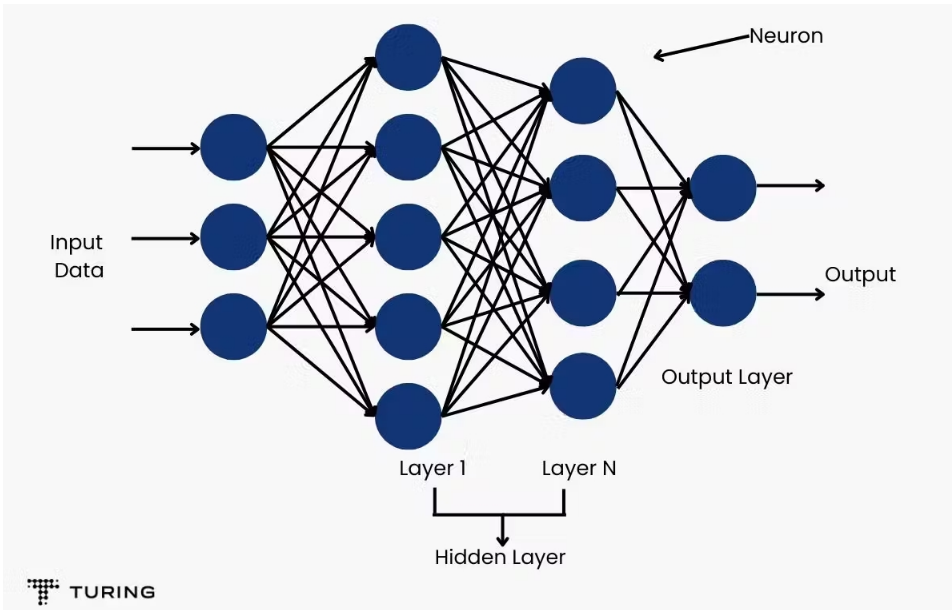

Figure 3: Model Architecture

- **Vector Representation of Data**
  - Inputs: $\mathbf{x}^0 = [x_1, x_2, \ldots, x_n]$ (input layer)
  - Outputs: $\hat{y}$ from the output layer (single value or vector for multiple outputs)
- **Mathematical Formulation**

  Trap Teacher, [01.12.2024 0:30]

  1. Linear Combination in Layer l:
  $$z^l = \mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l$$
  2. Activation Function (ReLU/Sigmoid):
  $$\hat{y}^l = \phi(z^l)$$
  3. Loss Function (Mean Squared Error or Cross-Entropy):
  $$L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \quad \text{(for MSE)}$$

3.2 Predictions Calculation

For a general MLP with L layers:

$$\hat{y} = \mathbf{W}^L \cdot \cdots \cdot \phi(\mathbf{W}^2 \phi(\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^L$$

3.3 Gradient Descent Algorithm

Backpropagation is employed for updating weights and biases across the network to minimize the loss function.

3.4 Gradients and Updates

1. Gradient for each layer:
$$\frac{\partial L}{\partial w_{ij}^l} = \delta_j^l x_i^{l-1}$$
   where δ_j^l is the error term for neuron j in layer l.

2. Weights and Biases Update:
$$w_{ij}^l \leftarrow w_{ij}^l - \eta \frac{\partial L}{\partial w_{ij}^l}, \quad b_j^l \leftarrow b_j^l - \eta \frac{\partial L}{\partial b_j^l}$$