A diagram of a function

Description automatically generatedTran Hai Long 5130203/20102

Report about Neural Networks

**Perceptron Algorithm**

1. Model Architecture:

Input layer: A set of features x=[,,,…,].

Weights: Each input has a corresponding weight w=[,,,…,].

Bias: A constant b

Activation Function: A step function that outputs either +1 or −1.

1. Mathematical Formulation

Linear Combination: z=​⋅​+​⋅+...+⋅​+b

Activation Function:

Loss Function:L(y,)= where y is the true label.

1. Prediction Calculation

The prediction ​ is computed as:

1. Gradient and Weights/Bias Updates

Gradient of Loss with respect to weight:

Weight update rule:

5. Gradient Descent Explanation

* The Perceptron uses a gradient descent algorithm to update weights iteratively. The weights are updated based on the error in prediction.
* The learning process involves adjusting the weights to minimize the error in the predicted output.

**Logistic Regression Algorithm**

1. Model Architecture:

Input layer: A set of features x=[,,,…,].

Weights: Each input has a corresponding weight w=[,,,…,].

Bias: A constant b

Activation Function: The sigmoid function, which outputs a probability between 0 and 1.

1. Mathematical Formulation

Linear Combination: z=​⋅​+​⋅+...+⋅​+b

Activation Function (Sigmoid):

Loss Function (Cross-Entropy Loss):L(y,)=-[ylog()+(1-y)log(1-]

where y is the true label (0 or 1).

1. Prediction Calculation

The prediction is the output of the sigmoid function:

1. Gradient and Weights/Bias Updates

Gradient of Loss with respect to weight:

Weight update rule: where is the learning rate.

1. Gradient Descent Explanation

* Gradient Descent is used to minimize the cross-entropy loss by updating the weights.
* It adjusts the weights iteratively to reduce the error in predicting the probability of the class label.

**Multilayer Perceptron (MLP)**

* 1. Model Architecture

Input layer: A set of features x=[,,,…,].

Hidden layers: Multiple layers of neurons where each neuron applies a weight to the input and passes it through an activation function (e.g., ReLU, sigmoid).

Output layer: A final layer that produces the prediction.

The architecture includes connections between the layers, where each layer is fully connected to the next.

* 1. Mathematical Formulation

Linear Combination for each layer l:

where is the activation from the previous layer.

Activation Function:

Loss Function: L(y,) =Cross-Entropy or MSE

* 1. Prediction Calculation

The prediction​ is the output from the final layer:

=f(+)

* 1. Gradient and Weights/Bias Updates

Gradient of Loss with respect to weights and biases:

Weight update rule:

* 1. Gradient Descent Explanation
* In MLP, Gradient Descent is applied to each layer to minimize the loss.
* Backpropagation is used to compute the gradients for each weight and bias in the network.