The preferred model was a single-layer feedforward neural network, also known as a single-layer perceptron. The model consists of an input layer, a hidden layer, and an output layer. The hidden layer neurons feed information to the output layer neurons, and every feature in the input layer is connected to every hidden layer neuron. The sigmoid activation function is used to introduce non-linearity and perform binary classification in the model.

The Learning Process and Equations:

To make predictions and learn from the data, the selected model employs the following equations and a learning process:

Sigmoid activation function:

The model is given non-linearity by using the sigmoid activation function. It converts any real number to the [0, 1] range. It is said that the sigmoid function is

sigmoid(x) is equal to 1 / (1 + exp(-x)).

The loss of binary cross-entropy:

The cost function used to calculate the difference between the true binary labels' (y true) and predicted probabilities' (y pred) similarity is called the binary cross-entropy loss. The binary cross-entropy loss is calculated as follows:

- (y true * log (y pred) + (1 - y true) * log (1- y pred)).

Forward Propagation:

The model uses forward propagation during each training epoch to make predictions based on the training set of data. The weighted sum of the hidden layer's inputs is first determined, and then the sigmoid activation function is used to determine the hidden layer's output. To obtain the final output, it then computes the weighted sum of the inputs to the output layer and applies the sigmoid activation function.

Backward propagation:

The model calculates the binary cross-entropy loss between the true labels and the predicted outputs after forward propagation. After that, backpropagation is used to determine the weights' gradients with respect to the loss. We can learn from the gradients how much the loss varies with each weight. The model's weights are updated using these gradients. in the direction where the loss is the least.

Descent in Gradients:

The weights in the model are updated using gradient descent. An optimization algorithm known as gradient descent modifies weights in accordance with computed gradients. The step size of each weight update is determined by the learning rate (lr).