The code directory contains 2 MATLAB functions and a Live script implementation

function [W1, W2, W3] = mlp\_train(X, Y, n\_hidden\_nodes, n\_hidden\_layers, eta, i)

% MLP\_TRAIN trains a multi-layer perceptron with an arbitrary number of hidden nodes and up to two hidden layers.

%

% Inputs:

% - X: training data matrix (N x D)

% - Y: training labels vector (N x 1)

% - n\_hidden\_nodes: number of hidden nodes per hidden layer (1 x n\_hidden\_layers)

% - n\_hidden\_layers: number of hidden layers (1 or 2)

% - eta: learning rate for gradient descent

% - i: number of iterations for gradient descent

%

% Outputs:

% - W1, W2, W3: weight matrices for the three layers of the MLP

function Y\_pred = mlp\_test(X, W1, W2, W3)

% MLP\_TEST applies a trained multi-layer perceptron to new data.

%

% Inputs:

% - X: test data matrix (N x D)

% - W1, W2, W3: weight matrices for the three layers of the MLP

%

% Outputs:

% - Y\_pred: predicted labels vector (N x 1)

Then the live script implemented these functions with different inputs.

**RESULTS:**

Attempt 1 (100 epochs):

Using 2 hidden nodes and 1 hidden layer

Y\_pred = 4×1 logical array

0

1

1

1

Expected:

0

1

1

0

Using 5x5 hidden nodes and 2 hidden layer

Y\_pred = 4×1 logical array

0

1

1

1

Expected:

0

1

1

0

Attempt 2 (only changing the epochs to 1000):

Using 2 hidden nodes and 1 hidden layer

Y\_pred = 4×1 logical array

1

0

1

0

Expected:

0

1

1

0

Using 5x5 hidden nodes and 2 hidden layer

Y\_pred = 4×1 logical array

0

1

1

0

Expected:

0

1

1

0