Assignment-2

Yashwanth K Suruneni

Explanation of the Code Snippet of

A module to implement the stochastic gradient descent learning

algorithm for a feedforward neural network

Class: Network

The Class network is initialized with an array `sizes` whose length is the number of layers in the neural network and each element in the array representing the number of neurons in the layer. It has about 4 variables in the constructor function

num\_layers , which calculates the length of the array (number of layers )

sizes , which is the array which has the number of neurons in each layer

biases, An array of matrices with biases (3x1 , 1x1)

weights, An array of matrices with weights (1x3, 3x1)

There about six other methods in the class Network which are

feedforward : feedforward is a method that returns the output of an activation function (sigmoid function) , An activation function is what it makes a neural network capable of learning complex non-linear function. Activations are the nonlinear computations done in each node of a Neural Network.

SGD: This method accepts the training data along with the number of Epoch’s , mini\_batch\_size and the train\_data , The default parameter for test\_data is None , But in case the test data is provided we validate against the test\_data after each iteration i.e, after updating the weights and biases using `update\_mini\_batch` .

update\_mini\_batch: This method is called from SGD , This uses `backprop` to update the random weights and biases generated in the \_\_init\_\_ function.

backprop: The backprop method uses sigmoid function which is declared outside the Network class as the activation function, It also uses cost\_derivative method in the network class the derivative of the output activation and sigmoid\_prime which is the derivative of the sigmoid function to calculate the derivative of loss with respect to the output.

evaluate: This evaluates the test data with respect to train data , it returns the number of test inputs for which the network with particular bias and weight outputs the correct result.

cost\_derivative: This method returns the partial derivative of the output activation.

sigmoid: This function is outside the network class, The sigmoid function or activation function which each layer in the neural network goes through.

sigmoid\_prime: The derivative of the sigmoid function which the derivative of sigmoid function and is used along with cost\_derivative function in Network class to calculate the derivative of loss with respect to output.