**CS331 - Lab Assignment 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Group ID = AyKaGaRo**

**Deep Learning**

1. Implement the following operations (forward and backward pass)

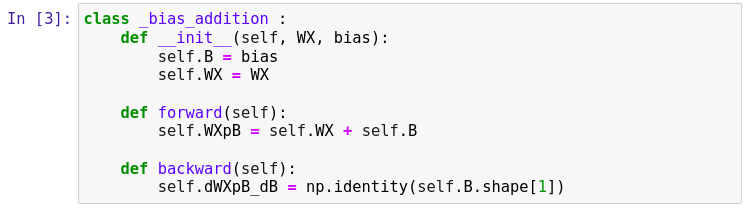
**(a) Matrix multiplication layer W X**



**forward:** Performs matrix multiplication of input X and weight matrix W.

**backward:** Computes the gradient of the loss with respect to the weights W.

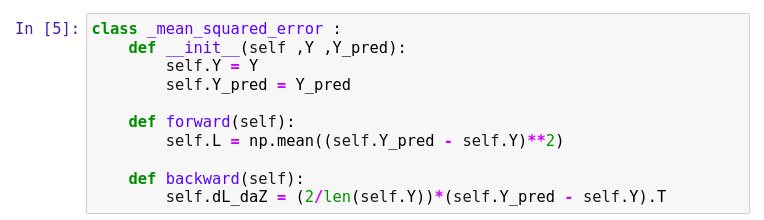
**(b) Bias addition layer**



**forward:** Adds bias to the input from the previous layer.

**backward:** Computes the gradient of the loss with respect to the bias.

**(c) Mean squared loss layer**



**forward:** Computes the mean squared error loss between predicted and actual values.

**backward:** Computes the gradient of the loss with respect to the predicted values.

**(d) Soft max layer**



**forward:** Applies the softmax function to the input.

**backward:** Computes the gradient of the loss with respect to the input.

**(e) Sigmoid layer**



**forward:** Applies the sigmoid activation function to the input.

**backward:** Computes the gradient of the loss with respect to the input.

(f) Cross entropy loss layer

****

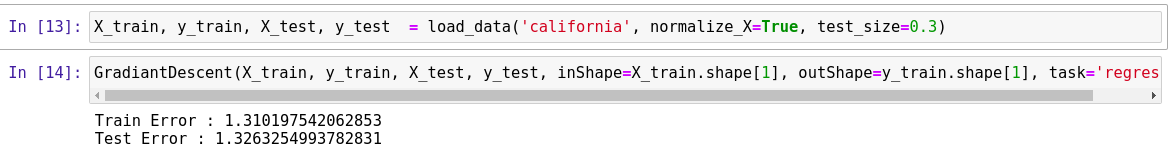
**forward:** Computes the cross-entropy loss between predicted and actual values.

**backward:** Computes the gradient of the loss with respect to the predicted values.

**Question 2. Using the sklearn.load boston() function, obtain boston house pricing dataset.**

**Train a regression model using the operations implemented above. You need to**

**write a stochastic gradient descent function to train.**

****

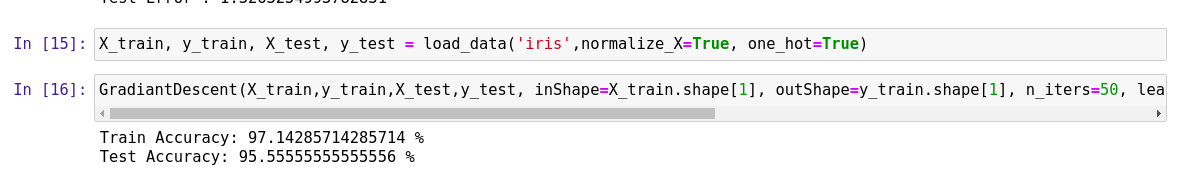
**Explanation:**

Loads the California housing dataset, normalizes the features, and splits it into training and testing sets.

Trains a regression model using stochastic gradient descent with the implemented operations.

**Question 3. Load the iris dataset in sklearn. This data sets consists of 3 different types of**

**irises’ (Setosa, Versicolour, and Virginica) petal and sepal length, stored in a 150x4 numpy.ndarray. Using the operations implemented above create a multi- class classifier (Cross entropy loss + soft max)**

****

**Explanation:**

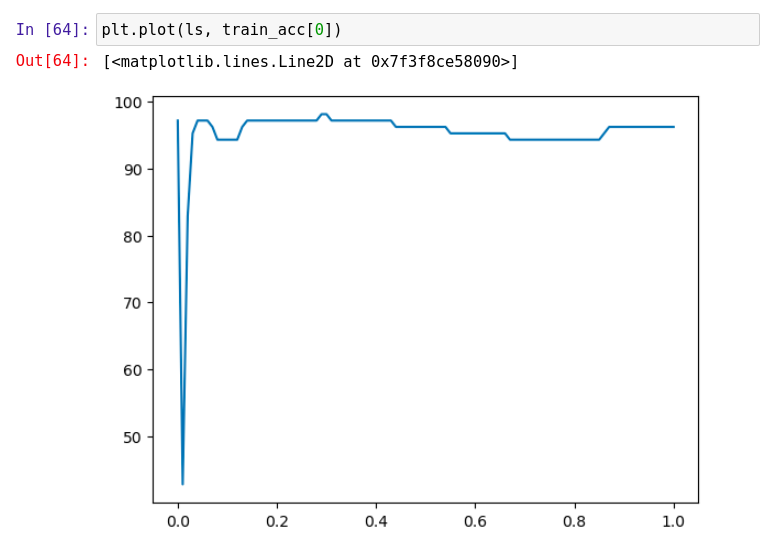
Loads the Iris dataset, normalizes the features, and one-hot encodes the target variable.

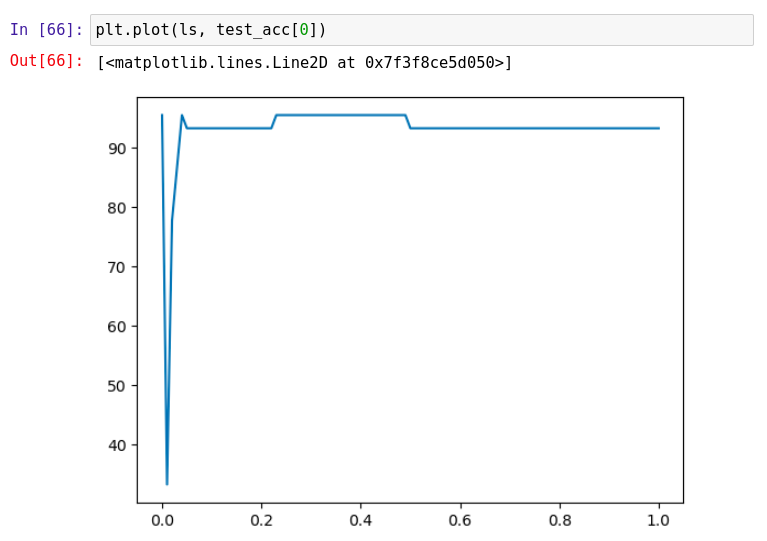
Trains a multi-class classification model using stochastic gradient descent with the implemented operations.

**LEARNING RATE (X AXIS)**

**VS**

**ACCURACY (Y AXIS)**

****

****