Cyber Labs

Report

Dheemant Dixit | WoC 4.0 | ML Bootcamp

Table of Contents

Linear Regression

Colab Notebook: Linear Regression

- Mean-Squared Error (MSE) was used as the Cost Function.
- Gradient Descent was used to minimize the Cost Function.
- Standardization was used to scale the data.

S.No.	Learning Rate	No. of epochs	Accuracy
1.	0.01	2500	99.99998618132689 %
2.	0.03	2500	99.99998618132699 %
3.	0.1	2500	99.99998618132699 %
4.	0.3	2500	99.99998618132699 %

Accuracy on the Testing Dataset using the LinearRegression model from sklearn was 99.99998618132689 %.

Polynomial Regression

Colab Notebook: Polynomial Regression

- Mean-Squared Error (MSE) was used as the Cost Function.
- Gradient Descent was used to minimize the Cost Function.
- Standardization was used to scale the data.
- Regularization with coefficient 0.01 was used when degree 5 polynomial was trained.
- Degree 3 was done manually but sklearn's PolynomialFeature was used for degrees greater than 3.

S.No.	Learning Rate	No. of epochs	Accuracy (Degree 3)	Accuracy (Degree 5)
1.	0.01	1500	99.998713570943 %	99.30686653264766 %
2.	0.03	1500	99.99854583120596 %	99.95962634686077 %
3.	0.1	1500	99.99854091828138 %	99.99017573544458 %
4.	0.3	1500	99.99854091837233 %	Diverges

Accuracy on the Testing Dataset using the LinearRegression model from sklearn was 99.99854091837233 % on Degree 3 and 99.99999537507315 % on Degree 5.

Logistic Regression

Colab Notebook: Logistic Regression

- EMNIST Dataset was given. It contains the pixel brightness value of a 28X28 greyscale image. Each training example corresponds to an English alphabet (26 unique labels).
- Since the number of classes was 26, the one-vs-all strategy was used to compute the predictions. It was vectorized.
- The dataset was scaled by dividing the whole dataset by 255 as the smallest pixel brightness was 0 and the largest was 255.
- Gradient descent was used to minimize the Cost function.

S.No.	Learning Rate	No. of epochs	Accuracy
1.	0.01	3000	60.9527027027027 %
2.	0.03	3000	65.05405405405405 %
3.	0.05	3000	69.75675675675676 %
4.	0.1	3000	68.01351351351352 %

Accuracy on the Testing Dataset using the LogisticRegression model from sklearn was 70.71621621621622 %.

K-Nearest Neighbors

Colab Notebook: K-Nearest Neighbors

- EMNIST Dataset was given. It contains the pixel brightness value of a 28X28 greyscale image. Each training example corresponds to an English alphabet (26 unique labels).
- The dataset was scaled by dividing the whole dataset by 255 as the smallest pixel brightness was 0 and the largest was 255.
- Euclidean distance was used as the distance formula.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

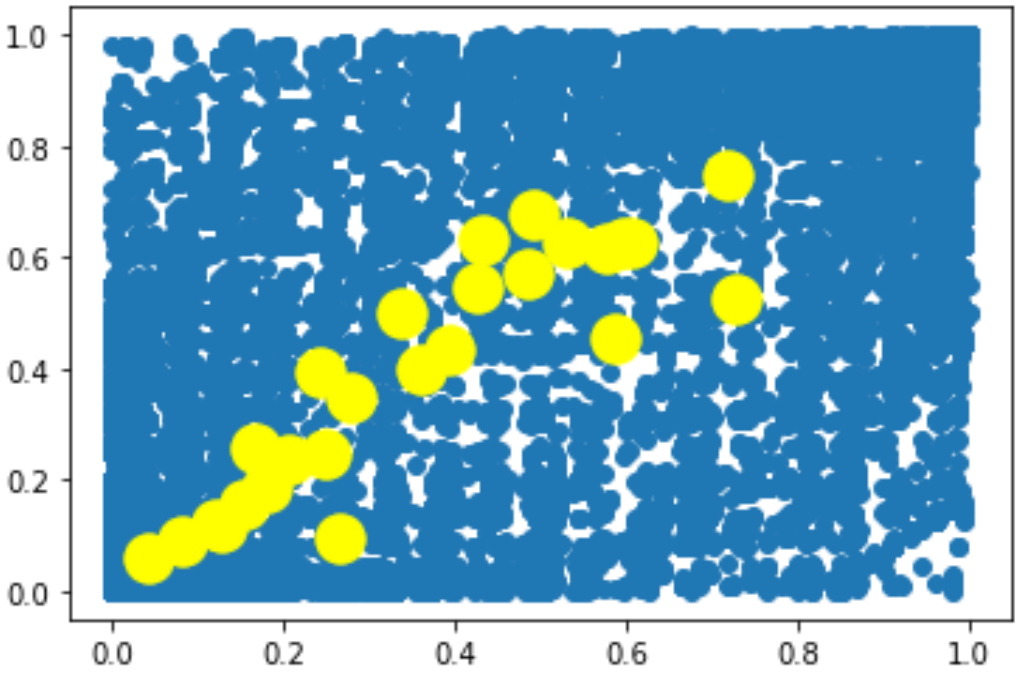
  -

S.No.	Value of K	Accuracy	Accuracy using sklearn's model
1.	1	83.08108108108108 %	83.08108108108108 %
2.	3	84.18918918918918 %	84.18918918918918 %
3.	5	84.28378378378379 %	84.28378378378379 %
4.	7	84.20945945945945 %	84.20945945945945 %

K-Means Clustering

Colab Notebook: K-Means Clustering

- Use the number of unique entries in the labels vector as the number of clusters.
- Calculate the distance of each training example from the centroids.
- Then assign the centroid least farther from the training example to it.
- Then create a vector for each centroid and update the centroid with the mean of all the cluster members.



Neural Network

REGRESSION

Colab Notebooks: Neural Network - Linear Regression

Neural Network - Polynomial Regression

- Abstract Base Class: Layer
    - The abstract class Layer, which all other layers will inherit, handles simple properties which are input, an output, and both a forward and backward methods
- Fully Connected Layer

- FC layers are the most basic layers as all input neurons are connected to every output neuron.
- Forward Propagation:

$$y_j = b_j + \sum_i x_i w_{ij}$$

- Backward Propagation:

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} W^t$$
$$\frac{\partial E}{\partial W} = X^t \frac{\partial E}{\partial Y}$$
$$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$

- Activation Layer
  - Sigmoid is used as the activation function.
- Loss Function
  - Mean-Squared Error is used as the Cost function.

$$E = \frac{1}{n} \sum_i^n (y_i^* - y_i)^2$$

- The Neural Network for Linear Regression contains 20 neurons in the input layer, 10 in the first Hidden layer and 5 in the second hidden layer, and one output neuron in the output layer.
- The Neural Network for Polynomial Regression contains 3 neurons in the input layer, 10 in the first Hidden layer and 5 in the second hidden layer, and one output neuron in the output layer.
- In both Neural Networks, an Activation layer is inserted before each Hidden layer.

S.No.	Learning Rate	No. of epochs	Accuracy (Linear Regression)	Accuracy (Polynomial Regression)
1.	0.01	50	99.99582152469425%	99.98801494807691%
2.	0.03	50	99.9982010925592%	99.9922239143759%
3.	0.1	50	99.9991270973671 1%	99.99442681696597%
4.	0.3	50	99.9981261416254%	99.7534628191681%

CLASSIFICATION

Colab Notebook: Neural Network - Classification

- Abstract Base Class: Layer
    - The abstract class Layer, which all other layers will inherit, handles simple properties which are input, an output, and both a forward and backward methods
- Fully Connected Layer
    - FC layers are the most basic layers as all input neurons are connected to every output neuron.
    - Forward Propagation:
    - $$y_j = b_j + \sum_i x_i w_{ij}$$
    - Backward Propagation:
    - $$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} W^t$$
    - $$\frac{\partial E}{\partial W} = X^t \frac{\partial E}{\partial Y}$$
    - $$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$
- Activation Layer
    - Sigmoid is used as the activation function.
- Loss Function
    - Mean-Squared Error is used as the Cost function.
    - $$E = \frac{1}{n} \sum_i^n (y_i^* - y_i)^2$$

- The Neural Network for Classification contains 784 neurons in the input layer, 100 in the first Hidden layer and 50 in the second hidden layer, and 26 output neurons in the output layer.

- In this Neural Network, an Activation layer is inserted before each Hidden layer and before the Output layer.

S.No.	Learning Rate	No. of epochs	Accuracy
1.	0.03	35	74.97297297297297%
2.	0.1	35	82.46621621621621%
3.	0.3	35	85.35135135135135%
4.	0.5	35	85.89864864864865%