Muhammad Abdullah

SE(7A) | 19F-0916

Applied Machine Learning

TASK 2: Perceptron Learning

CODE COMPLETION

from sklearn import datasets  
import matplotlib.pyplot as plt  
import numpy as np  
  
X, y = datasets.make\_blobs(n\_samples=150, n\_features=2, centers=2, cluster\_std=1.05, random\_state=2)  
  
  
# Plotting dataset  
def visualizeData():  
 fig = plt.figure(figsize=(10, 8))  
 plt.plot(X[:, 0][y == 0], X[:, 1][y == 0], 'r^')  
 plt.plot(X[:, 0][y == 1], X[:, 1][y == 1], 'bs')  
 plt.xlabel("feature 1")  
 plt.ylabel("feature 2")  
 plt.title('Random Classification Data with 2 classes')  
 plt.show()  
  
  
# Threshold Function  
def step\_func(z):  
 return 1.0 if (z > 0) else 0.0  
  
  
def perceptron(X, y, lr, epochs):  
 m, n = X.shape  
  
 # Initializing parapeters(theta) to zeros.  
 # +1 in n+1 for the bias term.  
 theta = np.zeros((n + 1, 1))  
  
 # Empty list to store how many examples were  
 # misclassified at every iteration.  
 n\_miss\_list = []  
  
 # Training.  
 for epoch in range(epochs):  
  
 # variable to store #misclassified.  
 n\_miss = 0  
  
 # looping for every example.  
 for idx, x\_i in enumerate(X):  
 x\_i = np.insert(x\_i, 0, 1).reshape(-1, 1)  
 y\_hat = step\_func(np.dot(x\_i.T, theta))  
 if (np.squeeze(y\_hat) - y[idx]) != 0:  
 theta = theta + lr \* ((y[idx] - y\_hat) \* x\_i)  
 n\_miss += 1  
  
 return theta, n\_miss\_list  
  
  
def plot\_decision\_boundary(X, theta):  
 # X --> Inputs  
 # theta --> parameters  
  
 # The Line is y=mx+c  
 # So, Equate mx+c = theta0.X0 + theta1.X1 + theta2.X2  
 # Solving we find m and c  
 x1 = [min(X[:, 0]), max(X[:, 0])]  
 m = -theta[1] / theta[2]  
 c = -theta[0] / theta[2]  
 x2 = m \* x1 + c  
  
 # Plotting  
 fig = plt.figure(figsize=(10, 8))  
 plt.plot(X[:, 0][y == 0], X[:, 1][y == 0], "r^")  
 plt.plot(X[:, 0][y == 1], X[:, 1][y == 1], "bs")  
 plt.xlabel("feature 1")  
 plt.ylabel("feature 2")  
 plt.title("Perceptron Algorithm")  
 plt.plot(x1, x2, 'y-')  
 plt.show()  
  
  
# # Press the green button in the gutter to run the script.  
if \_\_name\_\_ == '\_\_main\_\_':  
 theta, miss\_l = perceptron(X, y, 0.5, 100)  
 print(theta)  
 visualizeData() # Visualizing raw dataset  
 plot\_decision\_boundary(X, theta) # Visualizing Classification dataset

Running Code Screenshots:

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Manually Debugging of Code Epochs:

A screenshot of a computer

Description automatically generated with medium confidence

For 1st Epoch:

A screenshot of a computer

Description automatically generated

For the 2nd Epoch:

A screenshot of a computer

Description automatically generated with medium confidence