ML Assignment-6

Date: 13/05/2021

**Implement perceptron for AND, OR gate using perceptron training rule.**

CODE:

# Perceptron using perceptron traing Rule

#https://medium.com/@thomascountz/19-line-line-by-line-python-perceptron-b6f113b161f3

import numpy as np

class Perceptron(object):

    def \_\_init\_\_(self, no\_of\_inputs, threshold=100, learning\_rate=0.01):

        self.threshold = threshold

        self.learning\_rate = learning\_rate

        self.weights = np.zeros(no\_of\_inputs + 1)

    def predict(self, inputs):

       summation = np.dot(inputs, self.weights[1:]) + self.weights[0]

       #summation=0

       #for i in range(0,len(inputs)):

        # summation+=inputs[i]\*self.weights[i+1]+self.weights[0]

       if summation > 0:

          activation = 1

       else:

          activation = 0

       return activation

    def train(self, training\_inputs, labels):

        for \_ in range(self.threshold):

            for inputs, label in zip(training\_inputs, labels):

                prediction = self.predict(inputs)

                self.weights[1:] += self.learning\_rate \* (label - prediction) \* inputs

                self.weights[0] += self.learning\_rate \* (label - prediction)

#AND

training\_inputs = []

training\_inputs.append(np.array([1, 1]))

training\_inputs.append(np.array([1, 0]))

training\_inputs.append(np.array([0, 1]))

#training\_inputs.append(np.array([0, 0,1]))

#training\_inputs.append(np.array([1, 1,0]))

#training\_inputs.append(np.array([1, 0,0]))

#training\_inputs.append(np.array([0, 0,0]))

training\_inputs.append(np.array([0,0]))

labels = np.array([1, 0, 0, 0])

perceptron = Perceptron(2)

perceptron.train(training\_inputs, labels)

inputs = np.array([1, 1])

print(perceptron.predict(inputs))

#=> 1

inputs = np.array([1, 0])

print(perceptron.predict(inputs))

inputs = np.array([0, 1])

print(perceptron.predict(inputs))

inputs = np.array([0, 0])

print(perceptron.predict(inputs))

#OR

training\_inputs = []

training\_inputs.append(np.array([1, 1]))

training\_inputs.append(np.array([1, 0]))

training\_inputs.append(np.array([0, 1]))

#training\_inputs.append(np.array([0, 0,1]))

#training\_inputs.append(np.array([1, 1,0]))

#training\_inputs.append(np.array([1, 0,0]))

#training\_inputs.append(np.array([0, 0,0]))

training\_inputs.append(np.array([0,0]))

labels = np.array([1, 1, 1, 0])

perceptron = Perceptron(2)

perceptron.train(training\_inputs, labels)

inputs = np.array([1, 1])

print(perceptron.predict(inputs))

#=> 1

inputs = np.array([1, 0])

print(perceptron.predict(inputs))

inputs = np.array([0, 1])

print(perceptron.predict(inputs))

inputs = np.array([0, 0])

print(perceptron.predict(inputs))

#XOR

training\_inputs = []

training\_inputs.append(np.array([1, 1]))

training\_inputs.append(np.array([1, 0]))

training\_inputs.append(np.array([0, 1]))

#training\_inputs.append(np.array([0, 0,1]))

#training\_inputs.append(np.array([1, 1,0]))

#training\_inputs.append(np.array([1, 0,0]))

#training\_inputs.append(np.array([0, 0,0]))

training\_inputs.append(np.array([0,0]))

labels = np.array([0, 1, 1, 0])

perceptron = Perceptron(2)

perceptron.train(training\_inputs, labels)

inputs = np.array([1, 1])

print(perceptron.predict(inputs))

#=> 1

inputs = np.array([1, 0])

print(perceptron.predict(inputs))

inputs = np.array([0, 1])

print(perceptron.predict(inputs))

inputs = np.array([0, 0])

print(perceptron.predict(inputs))

**OUTPUT:**

1

0

0

0

1

1

1

0

1

1

0

0

**Implement perceptron on Iris data set using Delta rule.**

CODE:

#Perceptron with delta rule

#https://learnai1.home.blog/2019/11/16/perceptron-delta-rule-python-implementation/

class Perceptron:

    def \_\_init\_\_(self):

        self.weights=[]

    #activation function

    def activation(self,data):

        #initializing with threshold value

         activation\_val=self.weights[0]

         activation\_val+=np.dot(self.weights[1:],data)

         return 1 if activation\_val>=0 else 0

    def fit(self,X,y,lrate,epochs):

        #initializing weight vector

        self.weights=[0.0 for i in range(len(X.columns)+1)]

        #no.of iterations to train the neural network

        for epoch in range(epochs):

            print(str(epoch+1),"epoch has started...")

            for index in range(len(X)):

                x=X.iloc[index]

                predicted=self.activation(x)

                #check for misclassification

                if(y.iloc[index]==predicted):

                    pass

                else:

                    #calculate the error value

                    error=y.iloc[index]-predicted

                    #updation of threshold

                    self.weights[0]=self.weights[0]+lrate\*error

                    #updation of associated self.weights acccording to Delta rule

                    for j in range(len(x)):

                        self.weights[j+1]=self.weights[j+1]+lrate\*error\*x[j]

    def predict(self,x\_test):

        predicted=[]

        for i in range(len(x\_test)):

            #prediction for test set using obtained weights

            predicted.append(self.activation(x\_test.iloc[i]))

        return predicted

    def accuracy(self,predicted,original):

          correct=0

          lent=len(predicted)

          for i in range(lent):

             if(predicted[i]==original.iloc[i]):

                 correct+=1

          return (correct/lent)\*100

    def getweights(self):

        return self.weights

import pandas as pd

from sklearn.model\_selection import train\_test\_split

#read data from .csv file

data=pd.read\_csv("iris.csv")

data.columns=["petal\_length","petal\_width","sepal\_length","sepal\_width","class"]

classes=data["class"]

data=data.drop(columns="class")

#splitting test and train data for iris

x\_train,x\_test,y\_train,y\_test=train\_test\_split(data,classes)

#training the percpetron

model=Perceptron()

model.fit(x\_train,y\_train,0.5,10)

pred=model.predict(x\_test)

print("accuracy: ",model.accuracy(pred,y\_test))

print("weights: ",model.getweights())

**OUTPUT:**

1 epoch has started...

2 epoch has started...

3 epoch has started...

4 epoch has started...

5 epoch has started...

6 epoch has started...

7 epoch has started...

8 epoch has started...

9 epoch has started...

10 epoch has started...

accuracy:  100.0

weights:  [-0.5, -0.9499999999999997, -2.0999999999999996, 3.3999999999999995, 1.7999999999999998]

**Build Neural Network on pima Indian diabetes dataset using tensorflow and keras.**

CODE:

# first neural network with keras tutorial

from numpy import loadtxt

from keras.models import Sequential

from keras.layers import Dense

# load the dataset

dataset = loadtxt('pima-indians-diabetes.csv', delimiter=',')

# split into input (X) and output (y) variables

X = dataset[:,0:8]

y = dataset[:,8]

# define the keras model

model = Sequential()

model.add(Dense(12, input\_dim=8, activation='relu'))

model.add(Dense(8, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

# compile the keras model

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

# fit the keras model on the dataset

model.fit(X, y, epochs=150, batch\_size=10)

# evaluate the keras model

\_, accuracy = model.evaluate(X, y)

print('Accuracy: %.2f' % (accuracy\*100))

**OUTPUT:**

Epoch 1/150

77/77 [==============================] - 1s 1ms/step - loss: 9.5398 - accuracy: 0.5944

Epoch 2/150

77/77 [==============================] - 0s 1ms/step - loss: 2.1210 - accuracy: 0.5173

Epoch 3/150

77/77 [==============================] - 0s 1ms/step - loss: 1.7582 - accuracy: 0.5767

Epoch 4/150

77/77 [==============================] - 0s 1ms/step - loss: 1.3638 - accuracy: 0.5566

Epoch 5/150

77/77 [==============================] - 0s 1ms/step - loss: 1.1762 - accuracy: 0.6067

Epoch 6/150

77/77 [==============================] - 0s 1ms/step - loss: 0.9546 - accuracy: 0.6159

Epoch 7/150

77/77 [==============================] - 0s 2ms/step - loss: 0.8462 - accuracy: 0.6244

Epoch 8/150

77/77 [==============================] - 0s 1ms/step - loss: 0.8050 - accuracy: 0.6380

Epoch 9/150

77/77 [==============================] - 0s 1ms/step - loss: 0.7450 - accuracy: 0.6406

Epoch 10/150

77/77 [==============================] - 0s 1ms/step - loss: 0.7639 - accuracy: 0.6186

Epoch 11/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6425 - accuracy: 0.7151

Epoch 12/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6915 - accuracy: 0.6687

Epoch 13/150

77/77 [==============================] - 0s 1ms/step - loss: 0.7165 - accuracy: 0.6417

Epoch 14/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6597 - accuracy: 0.6849

Epoch 15/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6189 - accuracy: 0.6987

Epoch 16/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6422 - accuracy: 0.6849

Epoch 17/150

77/77 [==============================] - 0s 2ms/step - loss: 0.6197 - accuracy: 0.6909

Epoch 18/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5669 - accuracy: 0.7199

Epoch 19/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6196 - accuracy: 0.7019

Epoch 20/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5626 - accuracy: 0.7323

Epoch 21/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5920 - accuracy: 0.7023

Epoch 22/150

77/77 [==============================] - 0s 2ms/step - loss: 0.5945 - accuracy: 0.7084

Epoch 23/150

77/77 [==============================] - 0s 2ms/step - loss: 0.6046 - accuracy: 0.6814

Epoch 24/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6057 - accuracy: 0.7015

Epoch 25/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6067 - accuracy: 0.7116

Epoch 26/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5599 - accuracy: 0.7245

Epoch 27/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6481 - accuracy: 0.7108

Epoch 28/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6013 - accuracy: 0.7049

Epoch 29/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5538 - accuracy: 0.7224

Epoch 30/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5427 - accuracy: 0.7334

Epoch 31/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5421 - accuracy: 0.7199

Epoch 32/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5698 - accuracy: 0.7173

Epoch 33/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5543 - accuracy: 0.7129

Epoch 34/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5569 - accuracy: 0.7300

Epoch 35/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5450 - accuracy: 0.7468

Epoch 36/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5773 - accuracy: 0.7158

Epoch 37/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5769 - accuracy: 0.7088

Epoch 38/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5294 - accuracy: 0.7293

Epoch 39/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5866 - accuracy: 0.7023

Epoch 40/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5507 - accuracy: 0.7479

Epoch 41/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5551 - accuracy: 0.7205

Epoch 42/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6532 - accuracy: 0.6933

Epoch 43/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5168 - accuracy: 0.7643

Epoch 44/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5752 - accuracy: 0.6998

Epoch 45/150

77/77 [==============================] - 0s 1ms/step - loss: 0.6013 - accuracy: 0.7065

Epoch 46/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5563 - accuracy: 0.7248

Epoch 47/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5343 - accuracy: 0.7578

Epoch 48/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5094 - accuracy: 0.7420

Epoch 49/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4888 - accuracy: 0.7719

Epoch 50/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5578 - accuracy: 0.7085

Epoch 51/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5902 - accuracy: 0.7171

Epoch 52/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5164 - accuracy: 0.7439

Epoch 53/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5306 - accuracy: 0.7562

Epoch 54/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5139 - accuracy: 0.7550

Epoch 55/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5479 - accuracy: 0.7273

Epoch 56/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5470 - accuracy: 0.7317

Epoch 57/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5015 - accuracy: 0.7616

Epoch 58/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5264 - accuracy: 0.7373

Epoch 59/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5289 - accuracy: 0.7589

Epoch 60/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5689 - accuracy: 0.7110

Epoch 61/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5742 - accuracy: 0.6886

Epoch 62/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5276 - accuracy: 0.7696

Epoch 63/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5607 - accuracy: 0.7068

Epoch 64/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5274 - accuracy: 0.7547

Epoch 65/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5286 - accuracy: 0.7334

Epoch 66/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5390 - accuracy: 0.7260

Epoch 67/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5771 - accuracy: 0.7292

Epoch 68/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5221 - accuracy: 0.7506

Epoch 69/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5479 - accuracy: 0.7330

Epoch 70/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5322 - accuracy: 0.7389

Epoch 71/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4947 - accuracy: 0.7552

Epoch 72/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5278 - accuracy: 0.7420

Epoch 73/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4866 - accuracy: 0.7623

Epoch 74/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5261 - accuracy: 0.7481

Epoch 75/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5227 - accuracy: 0.7433

Epoch 76/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4702 - accuracy: 0.7864

Epoch 77/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5300 - accuracy: 0.7475

Epoch 78/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5066 - accuracy: 0.7798

Epoch 79/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5728 - accuracy: 0.7115

Epoch 80/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5225 - accuracy: 0.7701

Epoch 81/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5330 - accuracy: 0.7300

Epoch 82/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5852 - accuracy: 0.6941

Epoch 83/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5286 - accuracy: 0.7152

Epoch 84/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5023 - accuracy: 0.7618

Epoch 85/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5360 - accuracy: 0.7234

Epoch 86/150

77/77 [==============================] - 0s 2ms/step - loss: 0.5148 - accuracy: 0.7577

Epoch 87/150

77/77 [==============================] - 0s 2ms/step - loss: 0.5009 - accuracy: 0.7512

Epoch 88/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4816 - accuracy: 0.7503

Epoch 89/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5137 - accuracy: 0.7126

Epoch 90/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4761 - accuracy: 0.7704

Epoch 91/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5487 - accuracy: 0.7093

Epoch 92/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4838 - accuracy: 0.7627

Epoch 93/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5218 - accuracy: 0.7352

Epoch 94/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4784 - accuracy: 0.7786

Epoch 95/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5279 - accuracy: 0.7475

Epoch 96/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4531 - accuracy: 0.7986

Epoch 97/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5029 - accuracy: 0.7656

Epoch 98/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5078 - accuracy: 0.7535

Epoch 99/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4956 - accuracy: 0.7725

Epoch 100/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4904 - accuracy: 0.7584

Epoch 101/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4934 - accuracy: 0.7662

Epoch 102/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4886 - accuracy: 0.7663

Epoch 103/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4889 - accuracy: 0.7707

Epoch 104/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4921 - accuracy: 0.7399

Epoch 105/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4981 - accuracy: 0.7539

Epoch 106/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4752 - accuracy: 0.7797

Epoch 107/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5202 - accuracy: 0.7301

Epoch 108/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4971 - accuracy: 0.7414

Epoch 109/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5108 - accuracy: 0.7576

Epoch 110/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5107 - accuracy: 0.7461

Epoch 111/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4879 - accuracy: 0.7475

Epoch 112/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4788 - accuracy: 0.7599

Epoch 113/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4752 - accuracy: 0.7782

Epoch 114/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5056 - accuracy: 0.7423

Epoch 115/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4970 - accuracy: 0.7699

Epoch 116/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4998 - accuracy: 0.7812

Epoch 117/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4958 - accuracy: 0.7505

Epoch 118/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5064 - accuracy: 0.7840

Epoch 119/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4914 - accuracy: 0.7747

Epoch 120/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4823 - accuracy: 0.7826

Epoch 121/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4782 - accuracy: 0.7789

Epoch 122/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5093 - accuracy: 0.7682

Epoch 123/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4726 - accuracy: 0.7759

Epoch 124/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4704 - accuracy: 0.7682

Epoch 125/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4828 - accuracy: 0.7720

Epoch 126/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4627 - accuracy: 0.7892

Epoch 127/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4909 - accuracy: 0.7448

Epoch 128/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5119 - accuracy: 0.7408

Epoch 129/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4982 - accuracy: 0.7561

Epoch 130/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4802 - accuracy: 0.7605

Epoch 131/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4839 - accuracy: 0.7786

Epoch 132/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4801 - accuracy: 0.7693

Epoch 133/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4762 - accuracy: 0.7727

Epoch 134/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4805 - accuracy: 0.7641

Epoch 135/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5147 - accuracy: 0.7446

Epoch 136/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4735 - accuracy: 0.7573

Epoch 137/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4872 - accuracy: 0.7594

Epoch 138/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4817 - accuracy: 0.7666

Epoch 139/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4952 - accuracy: 0.7583

Epoch 140/150

77/77 [==============================] - 0s 1ms/step - loss: 0.5008 - accuracy: 0.7600

Epoch 141/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4775 - accuracy: 0.7761

Epoch 142/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4950 - accuracy: 0.7615

Epoch 143/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4988 - accuracy: 0.7581

Epoch 144/150

77/77 [==============================] - 0s 2ms/step - loss: 0.4759 - accuracy: 0.7750

Epoch 145/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4590 - accuracy: 0.7894

Epoch 146/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4773 - accuracy: 0.7562

Epoch 147/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4621 - accuracy: 0.7715

Epoch 148/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4628 - accuracy: 0.7694

Epoch 149/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4668 - accuracy: 0.7704

Epoch 150/150

77/77 [==============================] - 0s 1ms/step - loss: 0.4797 - accuracy: 0.7829

24/24 [==============================] - 0s 1ms/step - loss: 0.4562 - accuracy: 0.7786

Accuracy: 77.86