AD18511- DEEP LEARNING LABORATORY

DATE:

PERCEPTRON MODEL WITHOUT USING PACKAGE.

EX.NO: 3(a)

AIM:

To implement perceptron from scratch and use it to for classification in the iris dataset.

PERCEPTRON DISCRIPTION:

- A Perceptron is an algorithm used for supervised learning of binary classifiers. Binary classifiers decide whether an input, usually represented by a series of vectors, belongs to a specific class.

- In short, a perceptron is a single-layer neural network. They consist of four main parts including input values, weights and bias, net sum, and an activation function.

- Working of Perceptron:

```
Step 1: X1W1 +X2W2  +......... +XnWn
  Σ Wi *Xi

Step 2: Unit step activation function

  If Σ Wi *Xi+ b > 0:
      Output = 1
  else:
      Output =0
```

ALGORITHM:

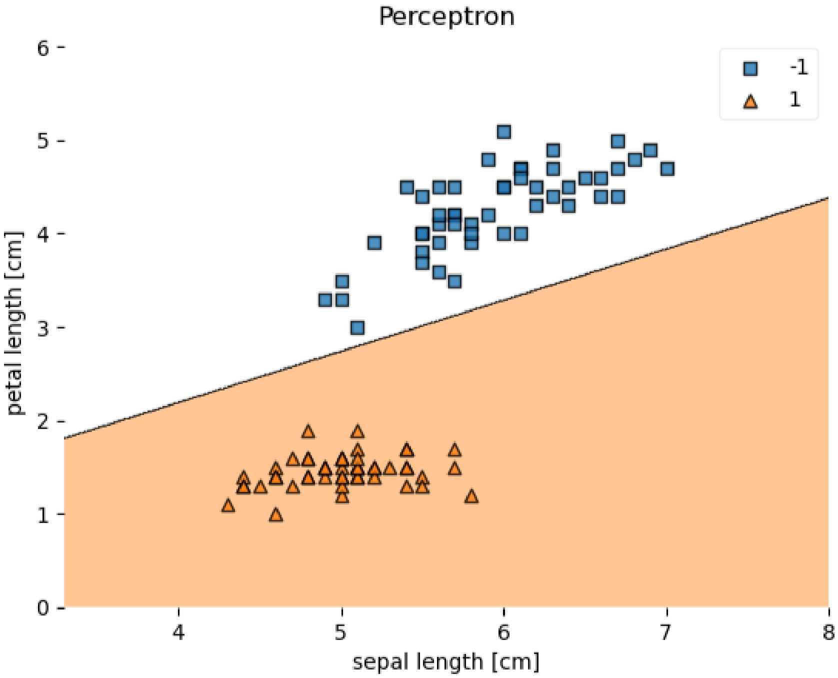
1. Start
2. import numpy
3. declare class perceptron with learning rate = 0.01, epochs = 50.
4. Define functions train, net input and predict.
5. import pandas and read the csv file into a dataframe.
6. plot the graph.
7. Stop.

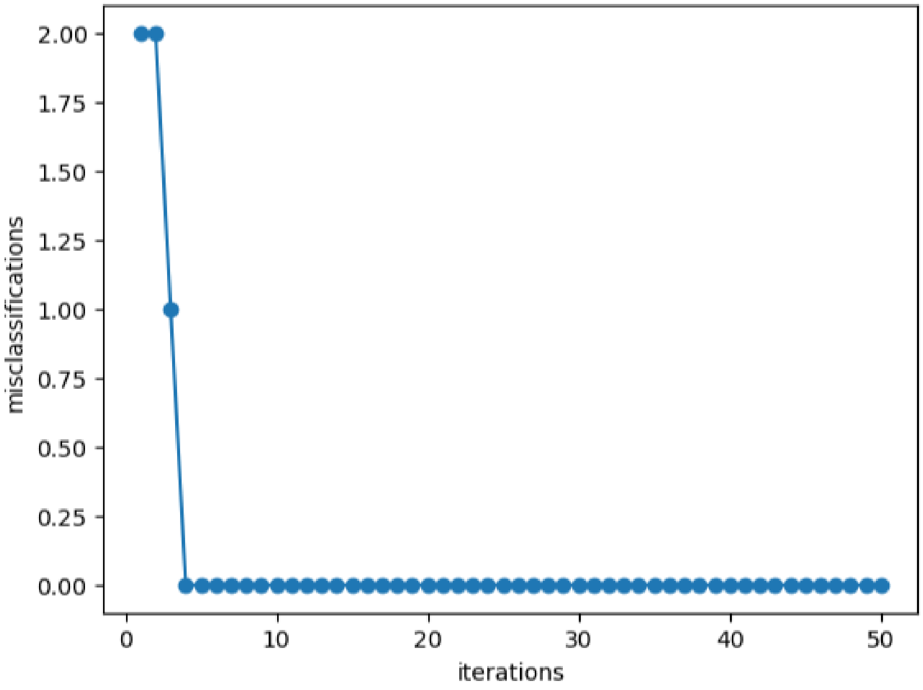
PROGRAM:

```
import numpy as np

class Perceptron(object):

   def_init_(self,eta=0.01,epochs=50):
      self.eta=eta
      self.epochs=epochs

   def train(self,X,y):
      self.w_=np.zeros(1+X.shape[1])
      self.errors_=[]

      for _ in range(self.epochs):
          errors=0
          for xi,target in zip(X,y):
              update=self.eta*(target - self.predict(xi))
              self.w_[1:] += update* xi
              self.w_[0] += update
              errors += int(update !=0.0)
          self.errors_.append(errors)
      return self

   def net_input(self,X):
      return np.dot(X,self.w_[1:] + self.w_[0])

   def predict(self,X):
      return np.where(self.net_input(X) > 0.0,1,-1)

import pandas as pd
df=pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',header=None)

y=df.iloc[0:100,4].values
y=np.where(y=='Iris-setosa',1,-1)

X=df.iloc[0:100,[0,2]].values


import matplotlib.pyplot as plt
from mlxtend.plotting import plot_decision_regions

ppn=Perceptron()

ppn.train(X,y)
print('weights:%s'%ppn.w_)
plot_decision_regions(X,y,clf=ppn)
plt.title('Perceptron')
plt.xlabel('sepal length [cm]')
plt.ylabel('petal length [cm]')
plt.show()


plt.plot(range(1,len(ppn.errors_)+1),ppn.errors_,marker='o')
plt.xlabel('iterations')
plt.ylabel('misclassifications')
plt.show()
```

OUTPUT:

```
weights:[ 0.02 0.026 -0.104]
```





RESULT:

The Perceptron model is implemented using Tensorflow. The model is trained and tested and then the loss and accuracy are displayed.

AD18511- DEEP LEARNING LABORATORY

DATE: **PERCEPTRON MODEL USING PACKAGES.** **EX.NO: 3(b)**

AIM:

To write a program that builds a perceptron model with the use of packages.

DESCRIPTION:

- A Perceptron is an algorithm used for supervised learning of binary classifiers. Binary classifiers decide whether an input, usually represented by a series of vectors, belongs to a specific class.
- In short, a perceptron is a single-layer neural network. They consist of four main parts including input values, weights and bias, net sum, and an activation function.
- Working of Perceptron:

```
Step 1: X1W1 +X2W2  +......... +XnWn
          Σ Wi *Xi
Step 2: Unit step activation function

   else:
         Output =0
```

PROGRAM:

```
import pandas as pd
df = pd.read_csv("/home/user/anaconda3/lib/python3.10/site-
packages/bokeh/sampledata/_data/iris.csv") y = df.iloc[0:100, 4].values
y = np.where(y == "setosa", 1, -1)
X = df.iloc[0:100, [0,2]].values
from sklearn.linear_model import Perceptron as per
from sklearn.model_selection import train_test_split as tts from sklearn.metrics import
accuracy_score as ac x1,x2,y1,y2 = tts(X,y,test_size=0.2, random_state=42) model =
per(alpha=0.001, max_iter=100) model.fit(x1,y1)
pred = model.predict(x2) acc = ac(y2, pred) print("acc=",acc)
```

OUTPUT:

```
acc= 1.0
```

RESULT:

Hence, the program to build a perceptron model using packages is complete and the output is verified.