感知机实验报告

姓名：李欣芮 学号：201822090530

# 实验内容

实现感知机算法解决二分类问题，采用训练数据集如下（监督学习方法）:



这组样本是可以进行线性划分的，算法可以收敛，最终训练所得到的模型可以正确分类。

# 实现原理

目标函数及决策线为:w1X+w2Y=0，该决策线可以正确分类这些点。

前向学习过程: 训练集上的样本点(x,y)带入计算w1x+w2y的值，使用自定义激活函数激活得到训练结果。

更新参数过程：当目标值为1，训练结果不正确则使用W=W(old)+(x,y)进行更新W值，当目标值为0，训练结果不正确则使用W=W(old)-(x,y)进行更新参数。

目标矩阵: 

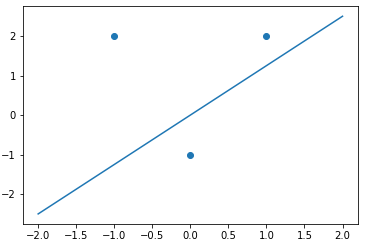
# 实验步骤

1. 给定训练集样本如上。
2. 给定初始参数W=[1，-0.8]
3. 进行前向计算得到训练样本的结果值
4. 反向计算比较更新参数W
5. 生成最终决策线

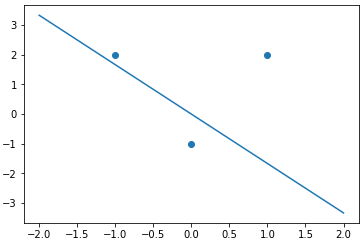
# 实验代码与结果

|  |
| --- |
| #导入模块  import numpy as np  import matplotlib.pyplot as plt  #产生数据集  X=np.array([[1,2],[-1,2],[0,-1]])  #产生目标数据集  Y=np.array([1,0,0])  #产生参数集  #w=np.random.RandomState(seed).rand(1,2)  w=np.array([[1,-0.8]])  print(w)  x=np.linspace(-2,2)  y=-(w[0,0]/w[0,1]\*x)  plt.scatter(X[:,0],X[:,1])  plt.plot(x,y)  plt.show()  #前向传播计算函数  def cacul(x,w):  y=np.dot(x.T,w)  if y>0:  res=1  else:  res=0  return res  #训练反向传播  STEPS=10  for i in range(STEPS):  print('训练了',i,'次')  if cacul(X[0],w[0])!=1:  print('加变换')  w[0]=w[0]+X[0]  print(w[0])  x=np.linspace(-2,2)  y=-(w[0,0]/w[0,1]\*x)  plt.scatter(X[:,0],X[:,1])  plt.plot(x,y)  plt.show()    if cacul(X[1],w[0])!=0:  print('减变换1')  w[0]=w[0]-X[1]  print(w[0])  x=np.linspace(-2,2)  y=-(w[0,0]/w[0,1]\*x)  plt.scatter(X[:,0],X[:,1])  plt.plot(x,y)  plt.show()    if cacul(X[2],w[0])!=0:  print('减变换2')  w[0]=w[0]-X[2]  print(w[0])  x=np.linspace(-2,2)  y=-(w[0,0]/w[0,1]\*x)  plt.scatter(X[:,0],X[:,1])  plt.plot(x,y)  plt.show()    print('结果为：',w[0])  x=np.linspace(-2,2)  y=-(w[0,0]/w[0,1]\*x)  plt.scatter(X[:,0],X[:,1])  plt.plot(x,y)  plt.show() |

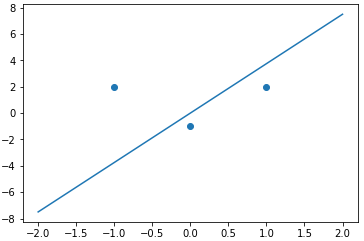
初始决策线,W=[0.8,-1]：



训练1次W=W+X[0]更新为[2：1.2]:



训练2次W=W-X[1]更新为[3,-0.8]:



最终训练结果，W=[3,0.2]，决策线为3x+0.2y=0能准确对训练集进行分类：

