Prac8

import numpy as np

class ART1:

def \_\_init\_\_(self,input\_size,rho,alpha):

self.W=np.zeros(input\_size) #weight vector

self.V=np.ones(input\_size) #vigilance vector

self.rho=rho

self.alpha=alpha

self.input\_size=input\_size

def train(self,X):

for x in X:

y=x/(self.rho+np.linalg.norm(self.W))

j=np.argmax(y)

if y[j]>= self.alpha\*np.sum(y) and self.V[j]>0:

self.W+=self.V[j]\*x

self.V[j]\*=0.5

else:

self.V[j]+=0.5

def classify(self,X):

classes=[]

for x in X:

y=x/(self.rho+np.linalg.norm(self.W))

j=np.argmax(y)

classes.append(j)

return classes

X\_train=np.array([

[0,1,1,0],

[1,0,0,1],

[1,0,0,0]

])

X\_test=np.array([

[0,0,1,0],

[1,1,1,0]

])

input\_size=X\_train.shape[1]

rho=0.5

alpha=0.9

art=ART1(input\_size,rho,alpha)

art.train(X\_train)

classes=art.classify(X\_test)

for i, data in enumerate(X\_test):

print(f'Test data: {i+1} Predicted class: {classes[i]}')