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]}')
```