Ann4

import numpy as np

class ART1:

def \_\_init\_\_(self, input\_size, num\_categories, rho=0.5):

self.input\_size = input\_size

self.num\_categories = num\_categories

self.rho = rho

self.weights = np.ones((num\_categories, input\_size))

self.reset\_categories()

def reset\_categories(self):

self.activated = np.zeros(self.num\_categories)

def match(self, input\_pattern, j):

"""Check if input matches the template within vigilance"""

wj = self.weights[j]

intersection = np.minimum(input\_pattern, wj).sum()

if (intersection / input\_pattern.sum()) >= self.rho:

return True

return False

def train(self, input\_patterns):

for input\_pattern in input\_patterns:

input\_pattern = np.array(input\_pattern)

self.reset\_categories()

for j in range(self.num\_categories):

if self.match(input\_pattern, j):

self.weights[j] = np.minimum(self.weights[j], input\_pattern)

break

else:

print("No suitable category found for pattern:", input\_pattern)

def predict(self, input\_pattern):

input\_pattern = np.array(input\_pattern)

for j in range(self.num\_categories):

if self.match(input\_pattern, j):

return j

return -1

if \_\_name\_\_ == "\_\_main\_\_":

patterns = [

[1, 0, 0, 1, 1],

[1, 0, 0, 1, 0],

[0, 1, 1, 0, 0],

[0, 1, 1, 0, 1]

]

art = ART1(input\_size=5, num\_categories=3, rho=0.6)

art.train(patterns)

test = [1, 0, 0, 1, 1]

result = art.predict(test)

print(f"Pattern {test} matched with category {result}")