Assignment B-2

CODE :

```
import numpy as np


class ART1:
    def __init__(self, input_size, vigilance=0.8):
        self.input_size = input_size
        self.vigilance = vigilance
        self.weights = []

    def _match(self, input_pattern, category_weights):
        """ Check the similarity between the input and category weights. """
        intersection = np.minimum(input_pattern, category_weights).sum()
        return intersection / input_pattern.sum()

    def _update_weights(self, category_idx, input_pattern):
        """ Update weights for the winning category. """
        self.weights[category_idx] = np.minimum(input_pattern, self.weights[category_idx])

    def train(self, patterns):
        for input_pattern in patterns:
            input_pattern = np.array(input_pattern)

            for category_idx in range(len(self.weights)):
                if self._match(input_pattern, self.weights[category_idx]) >= self.vigilance:
                    self._update_weights(category_idx, input_pattern)
                    print(f"PATTERN {input_pattern} - CATEGORY {category_idx}")
                    break
            else:
                self.weights.append(input_pattern)
```

```
            print(f"PATTERN {input_pattern} - CATEGORY {len(self.weights) - 1}")

    def predict(self, input_pattern):
        input_pattern = np.array(input_pattern)
        for category_idx in range(len(self.weights)):
            if self._match(input_pattern, self.weights[category_idx]) >= self.vigilance:
                return category_idx
        return -1

if __name__ == "__main__":
    patterns = [
        [1, 1, 0, 0, 1, 0],
        [1, 1, 1, 0, 1, 0],
        [0, 0, 1, 1, 0, 0],
        [1, 0, 1, 0, 1, 1],
    ]
    art_net = ART1(input_size=6, vigilance=0.8)
    art_net.train(patterns)

    while True:
        new_pattern = list(map(int, input("ENTER A BINARY PATTERN (6 DIGIT) :
").split()))
        if new_pattern == ['exit']:
            break
        result = art_net.predict(new_pattern)
        if result == -1:
            art_net.weights.append(new_pattern)
            print(f"NEW PATTERN {new_pattern} ASSIGNED to NEW CATEGORY
{len(art_net.weights) - 1}")
        else:
            print(f"NEW PATTERN {new_pattern} BELONGS TO {result}")
```

OUTPUT :

```
PS C:\Users\Admin> & C:/Users/Admin/AppData/Local/Microsoft/Win
PATTERN [1 1 0 0 1 0] - CATEGORY 0
PATTERN [1 1 1 0 1 0] - CATEGORY 1
PATTERN [0 0 1 1 0 0] - CATEGORY 2
PATTERN [1 0 1 0 1 1] - CATEGORY 3
ENTER A BINARY PATTERN (6 DIGIT) : 1 1 1 1 1 1
NEW PATTERN [1, 1, 1, 1, 1, 1] ASSIGNED to NEW CATEGORY 4
ENTER A BINARY PATTERN (6 DIGIT) : 1 0 1 0 1 1
NEW PATTERN [1, 0, 1, 0, 1, 1] BELONGS TO 3
ENTER A BINARY PATTERN (6 DIGIT) :
```