pratical-9-hopfield-4-vector

May 9, 2024

[1]:
```python
import numpy as np

# Define the 4 vectors to be stored
vectors = np.array([[1, 1, -1, -1],
                    [-1, -1, 1, 1],
                    [1, -1, 1, -1],
                    [-1, 1, -1, 1]])


# Calculate the weight matrix
weights = np.zeros((4, 4))
for i in range(4):
    for j in range(4):
        if i == j:
            weights[i, j] = 0
        else:
            weights[i, j] = np.sum(vectors[i] * vectors[j])


# Define the activation function (in this case, a sign function)
def activation(x):
    return np.where(x >= 0, 1, -1)


# Define the Hopfield network function
def hopfield(input_vector, weights):
    output_vector = activation(np.dot(weights, input_vector))
    return output_vector


# Test the Hopfield network with one of the stored vectors as input
input_vector = vectors[3]
output_vector = hopfield(input_vector, weights)
print("Input vector:")
print(input_vector)
print("Output vector:")
print(output_vector)
```

```
Input vector:
[-1  1 -1  1]
Output vector:
[-1  1 -1  1]
```

[ ]: