

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # PATTERNS
5 x1 = np.array([ [ -1, -1, -1, -1, -1, -1, -1, -1, -1
, -1],[ -1, -1, -1, 1, 1, 1, 1, -1, -1],[ -1, -1
, 1, 1, 1, 1, 1, 1, -1, -1],[ -1, 1, 1, 1, -1, -1, 1
, 1, 1, -1],[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1],[ -1
, 1, 1, 1, -1, -1, 1, 1, 1, -1],[ -1, 1, 1, 1, -1, -1
, 1, 1, 1, -1],[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1
],[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1],[ -1, 1, 1, 1
, -1, -1, 1, 1, 1, -1],[ -1, 1, 1, 1, -1, -1, 1, 1, 1
, -1],[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1],[ -1, 1, 1
, 1, -1, -1, 1, 1, 1, -1],[ -1, -1, 1, 1, 1, 1, 1, 1
, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, 1, -1, -1],[ -1
, -1, -1, -1, -1, -1, -1, -1, -1, -1] ])
6 x2 = np.array([ [ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1
],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1],[ -1, -1, -1
, 1, 1, 1, 1, -1, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, -1
, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1],[ -1
, -1, -1, 1, 1, 1, 1, -1, -1, -1],[ -1, -1, -1, 1, 1
, 1, 1, -1, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, -1, -1
, -1],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1],[ -1, -1
, -1, 1, 1, 1, 1, -1, -1, -1],[ -1, -1, -1, 1, 1, 1, 1
, -1, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1
],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1],[ -1, -1, -1
, 1, 1, 1, 1, -1, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, -1
, -1, -1],[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1] ])
7 x3 = np.array([ [ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1],[ 1
, 1, 1, 1, 1, 1, 1, 1, -1, -1],[ -1, -1, -1, -1, -1, -1
, 1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1
, -1],[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1],[ -1, -1
, -1, 1, 1, 1, 1, -1, -1],[ -1, -1, -1, -1, -1, -1, 1
, 1, 1, -1],[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1],[ 1, 1
, 1, 1, 1, 1, 1, 1, -1, -1],[ 1, 1, 1, -1, -1, -1, -1
, -1, -1, -1],[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1
],[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1],[ 1, 1, 1, -1
, -1, -1, -1, -1, -1, -1],[ 1, 1, 1, -1, -1, -1, -1, -1
, -1, -1],[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1],[ 1, 1
, 1, 1, 1, 1, 1, 1, -1, -1] ])
8 x4 = np.array([ [ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1
],[ -1, -1, 1, 1, 1, 1, 1, 1, 1, -1],[ -1, -1, -1, -1
, -1, -1, 1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1, 1, 1
, 1, 1],[ 1, -1],[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1],[ -1

```

```

8  , -1, -1, -1, -1, -1, 1, 1, 1, -1],[ -1, -1, -1, -1
  , -1, -1, 1, 1, 1, -1],[ -1, -1, 1, 1, 1, 1, 1, 1, -1
  , -1],[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1],[ -1, -1, -
  1, -1, -1, -1, 1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1
  , 1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1
  ],[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1],[ -1, -1, -1
  , -1, -1, -1, 1, 1, 1, -1],[ -1, -1, 1, 1, 1, 1, 1, 1
  , 1, -1],[ -1, -1, 1, 1, 1, 1, 1, 1, 1, -1, -1] ]))
9 x5 = np.array([ [ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1
  ],[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1],[ -1, 1, 1, -
  1, -1, -1, -1, 1, 1, -1],[ -1, 1, 1, -1, -1, -1, -1,
  1, 1, -1],[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1],[ -1
  , 1, 1, -1, -1, -1, -1, 1, 1, -1],[ -1, 1, 1, -1, -1
  , -1, -1, 1, 1, -1],[ -1, 1, 1, 1, 1, 1, 1, 1, 1, -1
  ],[ -1, 1, 1, 1, 1, 1, 1, 1, 1, -1],[ -1, -1, -1, -1
  , -1, -1, -1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1, -1
  , 1, 1, -1],[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1
  ],[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1],[ -1, -1, -
  1, -1, -1, -1, -1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1
  , -1, 1, 1, -1],[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -
  1] ]))
10
11 original_shape = x1.shape
12 patterns = [x1, x2, x3, x4, x5]
13 fig = plt.figure()
14 N = original_shape[0]*original_shape[1]
15 rounds_per_iter = 10000
16
17
18 # RESHAPE
19 for i, val in enumerate(patterns):
20     patterns[i] = np.reshape(val, (N))
21
22 # WEIGHT MATRIX
23 pattern_matrix = np.matrix(patterns)
24 W = np.matmul(pattern_matrix.T, pattern_matrix)
25 np.fill_diagonal(W, 0)
26
27 def ms_distance(x,y):
28     return np.sum(np.square((x - y)))
29
30 def draw_digit(x):
31     ax1 = fig.add_subplot(121)
32     ax1.imshow(x, interpolation='nearest')

```

```

33
34 def feed(x):
35     iteration = 0
36     running = True
37
38     while running:
39         old_x = x.copy()
40         for i in range(N):
41             # Update all neurons
42             neuron_index = i
43             selected_weights = np.matrix(W[
neuron_index]).T
44             new_neuron_value = np.sign(np.dot(x,
selected_weights))
45             new_neuron_value = new_neuron_value.item
()
46             if new_neuron_value == 0:
47                 new_neuron_value = 1
48                 x[0][neuron_index] = new_neuron_value
49                 curr_dist = ms_distance(old_x, x)
50                 print("it. finished : " + str(curr_dist))
51                 running = (curr_dist != 0)
52         x = x.reshape(original_shape)
53         draw_digit(x)
54         print(x.tolist())
55
56
57 # MAIN
58 x = np.array([[1, -1, -1, 1, 1, 1, 1, -1, -1, 1], [1
, -1, -1, 1, 1, 1, 1, -1, -1, 1], [1, -1, -1, 1, 1, 1
, 1, -1, -1, 1], [1, -1, -1, 1, 1, 1, 1, -1, -1, 1
], [1, -1, -1, 1, 1, 1, 1, -1, -1, 1], [1, -1, -1, 1
, 1, 1, 1, -1, -1, 1], [1, -1, -1, 1, 1, 1, 1, -1, -1
, 1], [1, -1, -1, -1, -1, -1, -1, -1, -1, 1], [1, -1
, -1, -1, 1, 1, 1, 1, -1], [-1, -1, -1, -1, -1, -1, -
1, -1, 1, 1, -1], [-1, -1, -1, -1, -1, -1, -1, -1, 1, 1
, -1], [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1], [-1, -
1, -1, -1, -1, -1, 1, 1, -1], [-1, -1, -1, -1, -1
, -1, -1, 1, 1, -1], [-1, -1, -1, -1, -1, -1, -1, 1,
1, -1], [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1]])
59 # x = np.array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1
, 1, -1, -1, -1, -1, 1, 1, 1], [-1, -1, 1, 1, 1, 1, 1,
1, -1, -1], [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1], [-1
, 1, 1, 1, -1, -1, 1, 1, 1, -1], [-1, 1, 1, 1, -1, -1

```

```

59 , 1, 1, 1, -1], [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1
    ], [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1], [-1, 1, 1, 1
    , -1, -1, 1, 1, 1, -1], [-1, 1, 1, 1, -1, -1, 1, 1,
    1, -1], [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1], [-1, 1,
    1, 1, -1, -1, 1, 1, 1, -1], [-1, 1, 1, 1, -1, -1, 1
    , 1, 1, -1], [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1], [-1
    , -1, -1, 1, 1, 1, 1, -1, -1, -1], [-1, -1, -1, -1
    , -1, -1, -1, -1, -1, -1]])
60 # x = np.array([[1, 1, 1, -1, -1, 1, -1, 1, 1, -1
    ], [1, 1, 1, -1, -1, 1, -1, 1, 1, -1], [1, 1, 1, -1
    , -1, 1, -1, 1, 1, -1], [1, 1, 1, -1, -1, 1, -1, 1, 1,
    1, -1], [1, 1, 1, -1, -1, 1, -1, 1, 1, -1], [1, 1, 1
    , -1, -1, 1, -1, 1, 1, -1], [1, 1, 1, -1, -1, 1, -1
    , 1, 1, -1], [1, 1, 1, 1, 1, -1, 1, 1, 1, -1], [1, 1
    , 1, 1, 1, -1, 1, 1, 1, -1], [1, -1, -1, -1, -1, 1
    , -1, 1, 1, -1], [1, -1, -1, -1, -1, 1, -1, 1, 1, -1
    ], [1, -1, -1, -1, -1, 1, -1, 1, 1, -1], [1, -1, -1
    , -1, -1, 1, -1, 1, 1, -1], [1, -1, -1, -1, -1, 1, -
    1, 1, 1, -1], [1, -1, -1, -1, -1, 1, -1, 1, 1, -1
    ], [1, -1, -1, -1, -1, 1, -1, 1, 1, -1]])
61
62 x = np.reshape(x, (1,N))
63 feed(x)
64 plt.show()

```