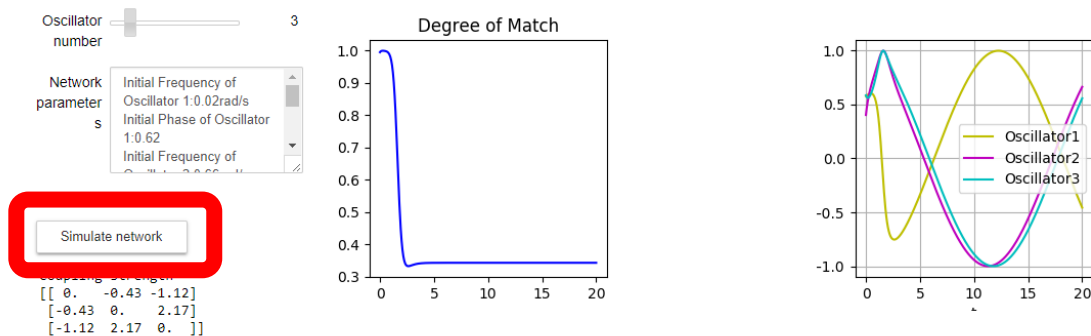


## Section 1: Try oscillator network solver

- Select oscillator and press button in red rectangle.
- The oscillation state and initial random conditions will be shown.

### Simulate network of MEMS oscillator

#### Simulate a oscillator network with random initial phase and random coupling weights

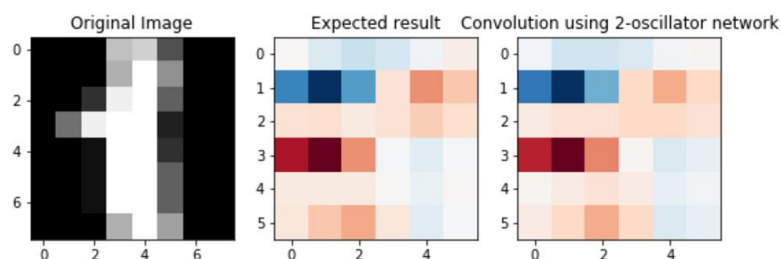
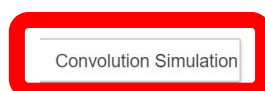


## Section 2: Image convolution

- Press the convolution simulation button to simulate convolution using oscillator network.

### Simulate Convolution

- Notice the regression is performed over the range  $(-0.05, 0.05)$ . We need to preprocess the number before convolution.



## Section 3: Pattern recognition

- Step 1: Select the patterns to be remembered by the network. Interact with the red area

from up to down.

## Pattern recognition

This following is for testing pattern recognition.

• Execute cells one by one!!!

1 Select patterns to remember (Use Ctrl and shift key to select multiple patterns)

Select picture index to remember

1
2
3
4
5
6

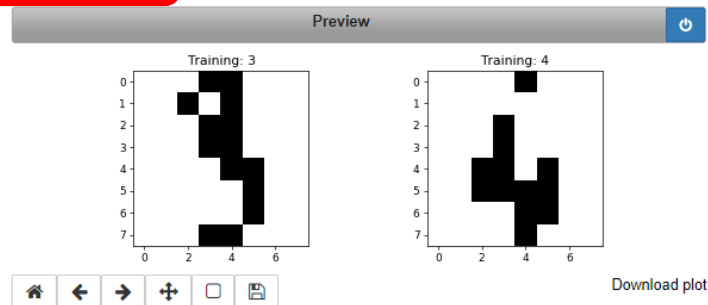
2 Confirm selected pattern indexes (execute the cell and click the button)

Confirm

You have selected index:  
[3, 4]

3 Preview patterns to be remembered

Preview



4 Select the image to be reconstructed

Select image to reconstruct

4
3

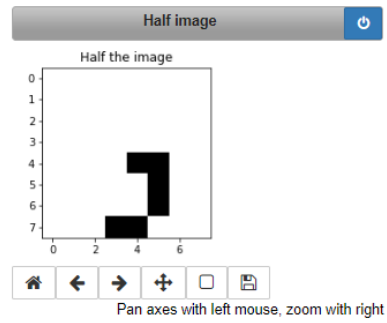
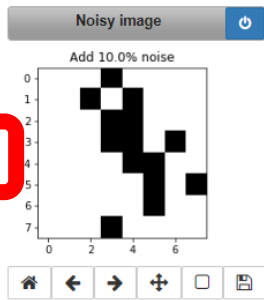
- Step 2:
- Reconstruct the noisy image or half of the image.

## 5 Select image to modify before reconstructing

### 6 Select the noise percent to add

Noise percent  **First**

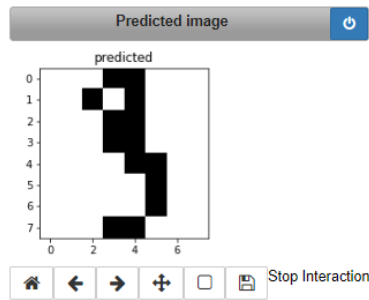
Add noise  **Second**



## 7 Show the reconstructed image

Reconstruct: ☒ Noisy image **Third**  
☐ Half of the image

Predict **Fourth**



- Play with the network energy function.
- The pattern is located at the local minimum of the energy function.

### Hopfield energy function:

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j$$

Let's pretend that vector  $a = (a_1, a_2)$  and  $b = (b_1, b_2)$  is stored in the network.

And as you can see in the figure,  $a$  and  $b$  are located as the local minimum.

a1  -1.00  
a2  -1.00  
b1  1.00  
b2  1.00

Trained energy function

