Neuroscience

Homework4 Report

Sana Aminnaji 98104722

1. Hopfield network

1.1 Investigation of Hopfield network theory

1.1.1

We know that:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{M} p_i^{\mu} p_j^{\mu}$$

Now we have to define the parameters N, p^{μ}, M; N is the number of the neurons, so $N = 4$. M is the number of the patterns, so $M = 3$. And finally p^{μ} stands for each pattern.

Calculation:

$$w_{11} = w_{22} = w_{33} = w_{44} = \frac{1+1+1}{4} = \frac{3}{4} = 0.75$$

$$w_{12} = w_{21} = \frac{1}{4}\sum_{\mu=1}^{3} p_i^{\mu} p_j^{\mu} = \frac{(1)\times(1)+(1)\times(1)+(1)\times(1)}{4} = \frac{3}{4} = 0.75$$

$$w_{13} = w_{31} = \frac{1}{4}\sum_{\mu=1}^{3} p_i^{\mu} p_j^{\mu} = \frac{(1)\times(1)+(1)\times(1)+(1)\times(-1)}{4} = \frac{1}{4} = 0.25$$

$$w_{14} = w_{41} = \frac{1}{4}\sum_{\mu=1}^{3} p_i^{\mu} p_j^{\mu} = \frac{(1)\times(1)+(1)\times(-1)+(1)\times(-1)}{4} = \frac{-1}{4} = -0.25$$

$$w_{23} = w_{32} = \frac{1}{4}\sum_{\mu=1}^{3} p_i^{\mu} p_j^{\mu} = \frac{(1)\times(1)+(1)\times(1)+(1)\times(-1)}{4} = \frac{1}{4} = 0.25$$

$$w_{24} = w_{42} = \frac{1}{4}\sum_{\mu=1}^{3} p_i^{\mu} p_j^{\mu} = \frac{(1)\times(1)+(1)\times(-1)+(1)\times(-1)}{4} = \frac{-1}{4} = -0.25$$

$$w_{34} = w_{43} = \frac{1}{4}\sum_{\mu=1}^{3} p_i^{\mu} p_j^{\mu} = \frac{(1)\times(1)+(1)\times(-1)+(-1)\times(-1)}{4} = \frac{1}{4} = 0.25$$

$$\rightarrow w = \begin{pmatrix} 0.75 & 0.75 & 0.25 & -0.25 \\ 0.75 & 0.75 & 0.25 & -0.25 \\ 0.25 & 0.25 & 0.75 & 0.25 \\ -0.25 & -0.25 & 0.25 & 0.75 \end{pmatrix}$$

1.1.2

$$s_j^1 = sgn\,[\sum_j w_{ij} s_j^0] \rightarrow$$

$$s_1^1 = sgn[0.75 - 0.75 - 0.25 - 0.25] = sgn[-0.5] = -1$$

$$s_2^1 = sgn[0.75 - 0.75 - 0.25 - 0.25] = sgn[-0.5] = -1$$

$$s_3^1 = sgn[0.25 - 0.25 - 0.75 + 0.25] = sgn[-0.5] = -1 \; s_4^1$$

$$= sgn[-0.25 + 0.25 - 0.25 + 0.75] = sgn[0.5] = +1$$

$$-1 \qquad\qquad 1$$

$$\rightarrow s^1 = (^{-}{-}^{1}1) = -1(\,^{1}1\,) = -x_2$$

$$1 \qquad\qquad -1$$

We know that in this formula if we replace the s^i with one of the initial prototypes it will give the same prototype as result. So the answer will converge to $-x_2$. **1.1.3**

$$s^{t+1} = sgn[Ws^t]$$

$$\rightarrow s^{t+1} = sgn[W's^t] = sgn[(W + N1)s^t] = sgn[Ws^t + N1s^t]$$

Let's find the result for each of the prototypes:

$$x_1 \rightarrow sgn\left[\begin{pmatrix} 1.5 + 4N \\ 1.5 + 4N \\ 1.5 + 4N \\ 0.5 + 4N \end{pmatrix}\right] = sgn\left[\begin{pmatrix} 0.375 + N \\ 0.375 + N \\ 0.375 + N \\ 0.125 + N \end{pmatrix}\right]$$

$$2 \qquad\qquad + 2N \qquad 1 + N$$

$$2 + 2N \qquad\qquad 1 + N$$

$$x_2 \rightarrow sgn\,[(\,1 + 2N\,)] = sgn\,[(\,0.5 + N\,)]$$

$$-1 + 2N \qquad\qquad -0.5 + N$$

$$1.5 \qquad\qquad 1$$

$$1.5 \qquad\qquad 1$$

$$x_3 \rightarrow sgn\,[(-0.5)] = (-1) = x_3$$

$$-1.5 \qquad\qquad -1$$

If the answers want to be equal to the initial prototypes the below conditions should be true:

$$1.\ 0.125 + N > 0$$

$$2.\ -0.5 + N < 0$$

$$1\ ,2 \rightarrow\ -0.125 < N < 0.5$$

$$\rightarrow probability = \frac{0.5 - (-0.125)}{2 - (-2)} = \frac{0.625}{4} = 0.15625$$

1.1.4

According to the previous inequality we have reached in part 1.1.3 if $N = 0.125$ then the probability will be 1.

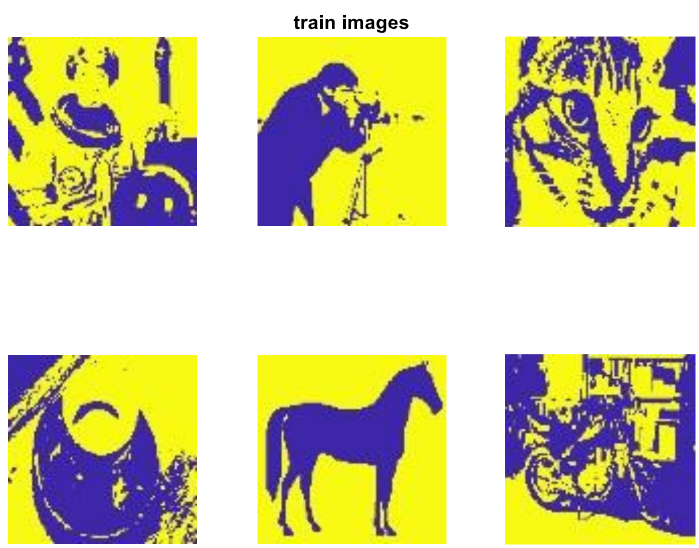
As it is obvious even in this example in the third prototype adding noise would not affect the result so if the initial prototypes will be same as this example we have, the noise won't be effective.

1.1.5

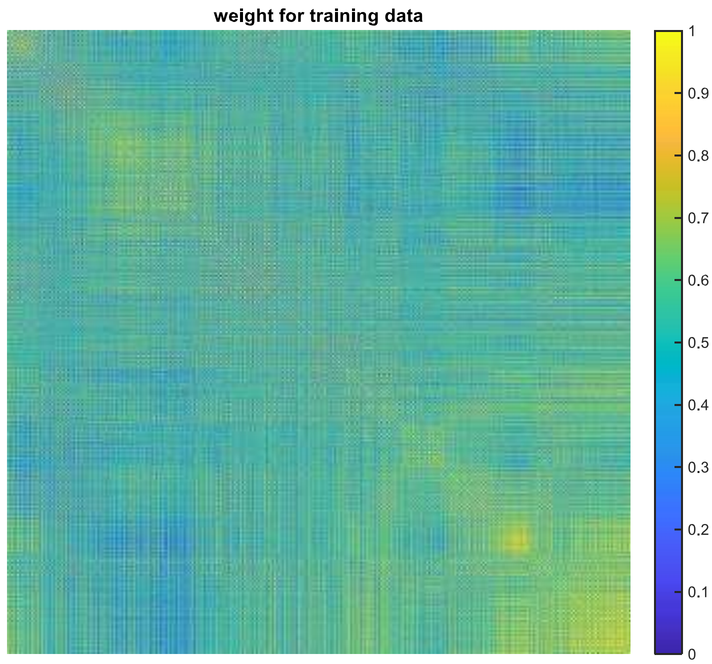
By increasing the number of network's neurons we will mostly face more conditions for noise resistance, but also there is more data about the system in the weight matrix so totally it can helps. if the noises for each weight is independent so we face more conditions that have to be true at the same time and the probability may decrease.

1.2 Practical investigation of Hopfield network

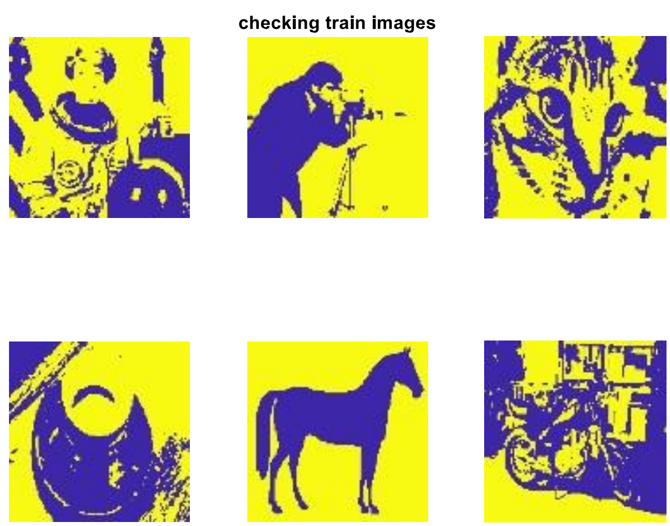
1.2.1



1.2.2



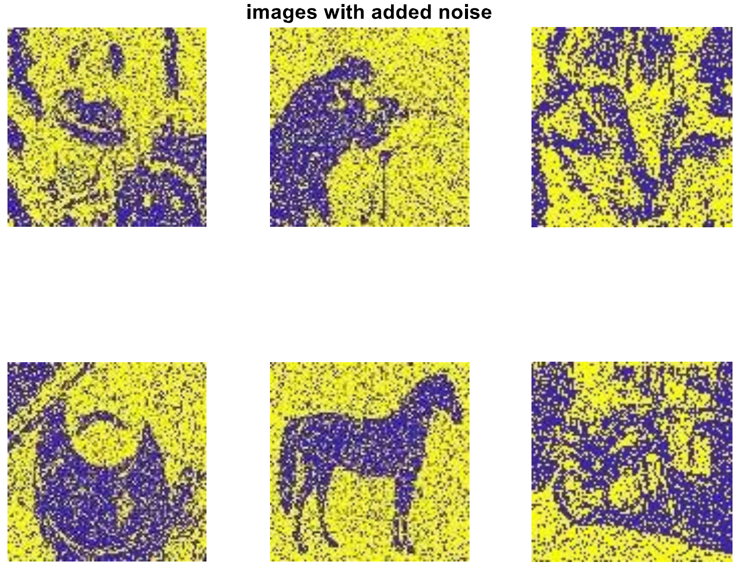
1.2.3



In this section we have defined a function which take the input and weight matrix and give us a matrix that is ready to be shown as an image and a variable called error which specifies the MSE of the function, as we have predicted the error for all six images is equal to zero.

1.2.4

images with added noise



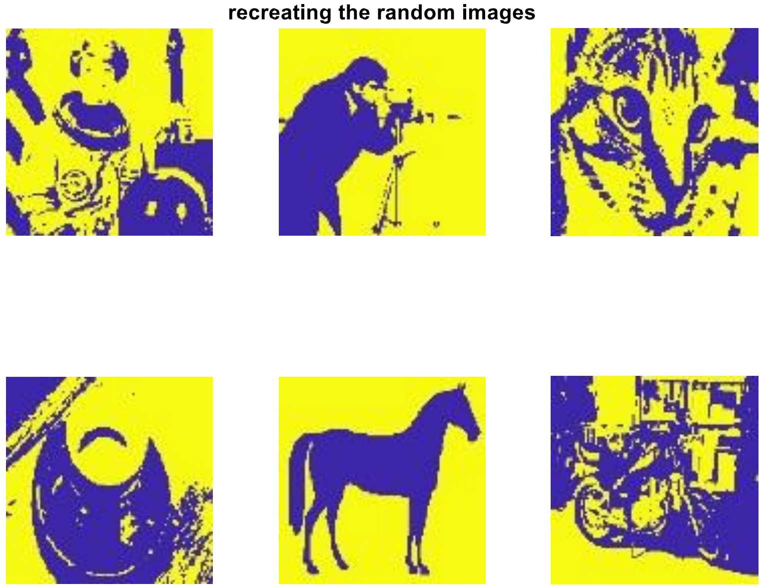
1.2.5

```
corr =

    0.6245    0.6044    0.6299    0.6335    0.6115    0.6320
```

1.2.6

recreating the random images

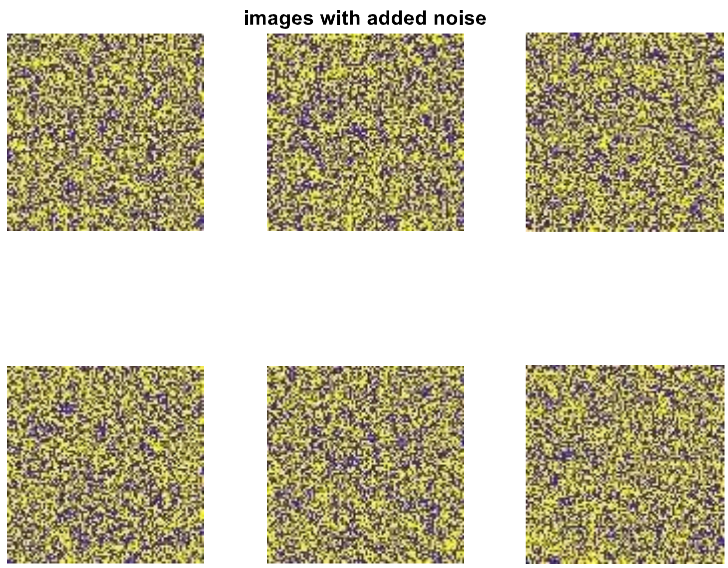


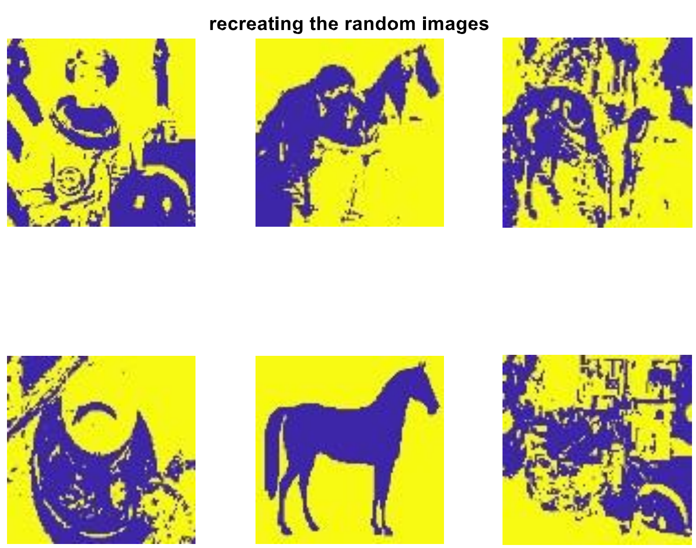
```
e1 =

    0.7324
    0.7324
    0.7324
    0.7324
    0.7324
    0.7324
```

As you can see the error (computed with MSE formula between the input and output) is large because of the random noise but the resolution of results is perfect and that is because of the networks accuracy.

If we calculate the recreation error: It is equal to zero. **1.2.7**





The error in this section:

```
error2 =

    1.8779
    2.2710
    1.9243
    1.9219
    1.9910
         0
```

By continuing the cycle and giving the new output as input to the network we will converge to the initial photos.

Second part:

$$(i):\ W_{ij} = \sum_{s=1}^{n} (2V_i^s - 1) \times (2V_j^s - 1) \quad i \neq j$$

$$\rightarrow for\ n = 1 : W_{ij} = \left(2V_i^1 - 1\right) \times \left(2V_j^1 - 1\right)$$

$$there\ are\ no\ self - connections\ in\ this\ network \rightarrow$$

$$if\ i = j\ \rightarrow W_{ii} = 0$$

$$W_{ij} = \sum_{s=1}^{n} (2V_i^s - 1) \times (2V_j^s - 1) = \sum_{s=1}^{n} (2V_j^s - 1) \times (2V_i^s - 1) = W_{ji}$$

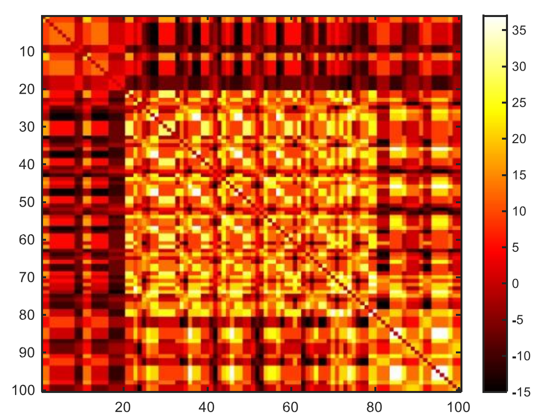
1.4 Forth Question

encoding function → input: $a \times b \times n$ char , output: $(ab) \times n$ double for this example: input: $10 \times 10 \times 5$ char , output: 100×5 double decoding function → input: 100×5 double, 1×2 double (this is the size of patterns) , output: $10 \times 10 \times 5$ char

For this example: inputs: 100×5 double, $[10 , 10]$, output: $10 \times 10 \times 5$ char

1.5 Fifth Question

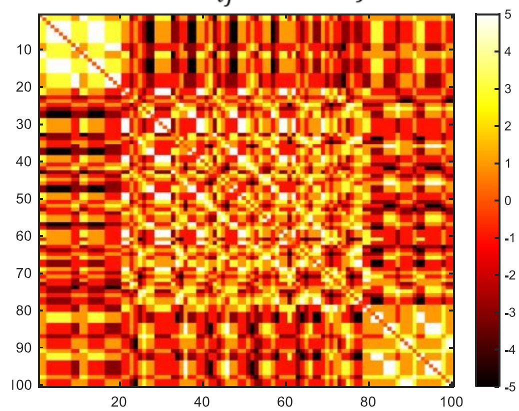
Weights function → input: $t \times n$ double, output: $t \times t$ double for this example: input: 100×5 double, output: 100×100 double if the weights are calculated from equation (i): (w1)



If the weights are calculated from equation below (for the bipolar coding): (w2)

$$(ii):\ W_{ij} = \sum_{s=1}^{n} (V_i^s) \times (V_j^s) \quad i \neq j$$

$$W_{ij} = 0 \quad i = j$$



1.2 1.6 Sixth Question

Reconstruction function → inputs: $(ab) \times (ab)$ double, :$a \times b \times n$ char , output: $a \times b \times n$ char

for this example: inputs: 100×100 double , $10 \times 10 \times 5$ char output: $10 \times 10 \times 5$ char output

for w1:

```
'    OO    '
'    OO    '
'   OOOO   '
'   O  O   '
'  OO  OO  '
'  O    O  '
' OOOOOOOO '
' OOOOOOOO '
'OO      OO'
'OO      OO'

'OOOOOOOOOO'
'OOOO  OOOO'
'  OOOOOOOO'
'  OOOOOOOO'
'  OOOOOOOO'
'  OOOOOOOO'
'  OOOOOOOO'
'  OOOOOOOO'
'  OOOOOOOO'
'  OOOOOOOO'

'        O '
'    OO  OO'
'OO      OO'
'OO        '
'OO        '
'OO        '
'OO        '
'OO        '
'OO        '
'OO        '
```

```
'        o '
'    oo  oo'
'oo      oo'
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '

'        o '
'    oo  oo'
'oo      oo'
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
```

output for w2:

```
'    oo    '
'    oo    '
'   oooo   '
'   o  o   '
'  oo  oo  '
'  o    o  '
' oooooooo '
' oooooooo '
'oo      oo'
'oo      oo'

'oooooo    '
'ooooooo   '
'oo   oo   '
'ooooooo   '
'ooooooo   '
'oo   ooo  '
'oo    oo  '
'oo   ooo  '
'ooooooo   '
'oooooo    '

'oooooooooo'
'oooooooooo'
'oo      oo'
'oo        '
'oo        '
'oo        '
'oo        '
'oo      oo'
'oooooooooo'
'oooooooooo'

'oo      oo'
'oo      oo'
'oo      oo'
'oo      oo'
'oooooooooo'
'oooooooooo'
'oo      oo'
'oo      oo'
'oo      oo'
'oo      oo'

'      ooo '
'    oo  oo'
'oooooooooo'
'ooo       '
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
'oo        '
```