**5.3 Energy**

**• How do you express this calculation in Matlab? (Note: you do not need to use any loops!)**

e = - diag(x \* w \* x')

**• What is the energy at the different attractors?**

|  |  |
| --- | --- |
| **Test on T5.1:**  e =  -68  -68  -72 | **Test on picture:**  e =  -1473936 (p1)  -1398416 (p2)  -1497344 (p3) |

**• What is the energy at the points of the distorted patterns?**

|  |  |
| --- | --- |
| **Test on T5.1:**  ed\_ini =  -40  -36  -24 | **Test on picture:**  ed\_ini =  -425964 (p11)  -177664 (p12) |

**• Follow how the energy changes from iteration to iteration when you use the sequential update rule to approach an attractor.**

**Test on T5.1:**

|  |  |
| --- | --- |
| e\_out =  -40 -68 -68  -36 -56 -68  -24 -72 -72 | When stuck in local minima:  e\_out =  -40 -68 -68 -68  -36 -56 -68 -68  -24 -56 -68 -68 …… |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**• Generate a weight matrix by setting the weights to normally distributed random numbers, and try iterating an arbitrary starting state. What happens?**

**• Make the weight matrix symmetric (e.g. by setting w=0.5\*(w+w')). What happens now? Why?**

Tset on T5.1

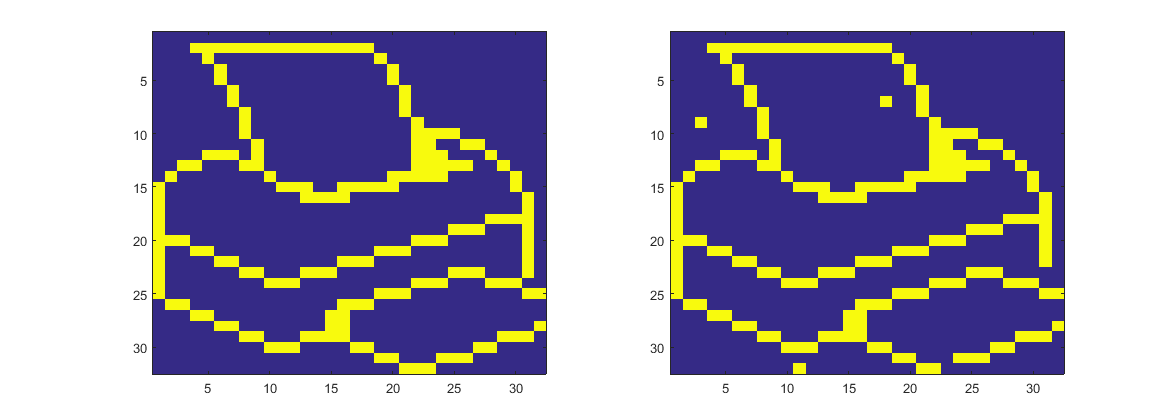
|  |  |
| --- | --- |
| asymmetric | symmetric |

Test on picture:

|  |  |
| --- | --- |
|  |  |

**5.4 Distortion Resistance**

function of flip: flip(p1,5): randomly choose 5 pixels in the picture and turn it to the opposite value, i.e. turn 1 to 0 and 0 to 1.



Noise test:

noise = [0, 0.2, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.8, 1];

Trained with p1, p2, p3, and also noise is added to p1, p2 and p3

|  |  |
| --- | --- |
| picture 1 | picture 2 |
| picture 3 |  |

Conclusion: a good restoration when noise < 0.4, and when noise > 0.5, the picture tends to restore to an inverse of colour.

