5.0 :

Check if the network was able to store all three patterns.

- Yes all 3 patterns are stored.

5.1

Apply the update rule repeatedly until you reach a stable fixpoint. Did all the patterns converge towards stored patterns?

* Yes.

How many attractors are there in this network?

* 14
* What happens when you make the starting pattern even more dissimilar to the stored ones (e.g. more than half is wrong)?
* Did not converge

5.2

* Can the network complete a degraded pattern? Try the pattern p11, which is a degraded version of p1, or p22 which is a mixture of p2 and p3.
* With synch update procedure only p11 converged. Img:5\_2\_a

What happens if we select units randomly, calculate their new state and then repeat the process (the original sequential Hopfield dynamics)?

* With sequential update, both of them converged.

5.3

* How do you express this calculation in Matlab?
* -(x \* w \* x')
* What is the energy at the different attractors?
* Energy at p1 = -1473936.000000
* Energy at p2 = -1398416.000000
* Energy at p3 = -1497344.000000
* What is the energy at the points of the distorted patterns?
* Energy at p11 = -425964.000000
* Energy at p22 = -177664.000000
* Follow how the energy changes from iteration to iteration when you use the sequential update rule to approach an attractor.
* Img:5\_3\_a\_1 & 2

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

* Img: 5\_3\_b\_1 & 2

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

* Img: 5\_3\_c\_1 & 2
* Energy monotonically goes down.

5.4

How much noise can be removed?

* Img: 5\_4

5.5

How many patterns could safely be stored? Was the drop in performance gradual or abrupt?

* Img: 5\_5\_a\_1 & 2

Try to repeat this with learning a few random patterns instead of the pictures and see if you can store more. You can use sgn(randn(1,1024)) to easily generate the patterns.

* Img:5\_5\_b
* It has been shown that the capacity of a Hopfield network is around 0.138N. How do you explain the difference between random patterns and the pictures?
* When number of -1 and +1 are almost equally probable in patterns then we go towards limit of capacity.

What happens with the number of stable patterns as more are learned?

* 5\_5\_c
* What happens if convergence to the pattern from a noisy version (a few flipped units) is used? What does the different behavior for large number of patterns mean?
* 5\_5\_c

What is the maximum number of retrievable patterns for this network?

5\_5\_d : without bias

What happens if you bias the patterns, e.g. use sign(0.5+randn(300,100)) or something similar to make them contain more +1? How does this relate to the capacity results of the picture patterns?

5\_5\_e : with bias and tends toward more +1

5.6

Try generating sparse patterns with just 10% activity and see how many can be stored for different values of θ (use a script to check different values of the bias).

5\_6\_rho\_0\_1

What about even sparser patterns (ρ = 0.05 or 0.01)?

5\_6\_rho\_0\_01

5\_6\_rho\_0\_05