Artificial Neural Networks and Deep Architectures DD2437

Lab Assignment 3 – Hopfield networks

Rithika Harish Kumar, Suhas Sheshadri

Aim

To explain the principles underlying the operation and functionality of auto-associative networks. Train the Hopfield network and explain the attractor dynamics in it along with the concept of energy function. We also demonstrate how auto-associative networks can do pattern completion and noise reduction. We study the storage capacity and explain features that help increase it in associative memories.

Tools used

Jupyter Notebook is used to carry out the simulations and learning.

Results

3.1

Can the memory recall the stored patterns from distorted inputs patterns?

Pattern 1 and 3 can converge, but pattern 2 is stuck in local minimum

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

Patterns can converge towards stored patterns.

How many attractors are there in this network?

8 attractors.

What happens when you make the starting pattern even more dissimilar to the stored ones (e.g. more than half is wrong)?

It converges after a few iterations.

Not any input can converge to stored patterns, only that have similarities. If many of the bit are erroneous, no convergence and attractors can't be found.

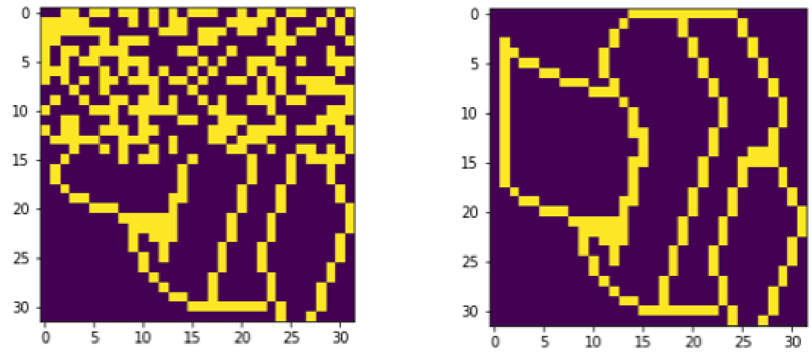
3.2

Check that the three patterns are stable.

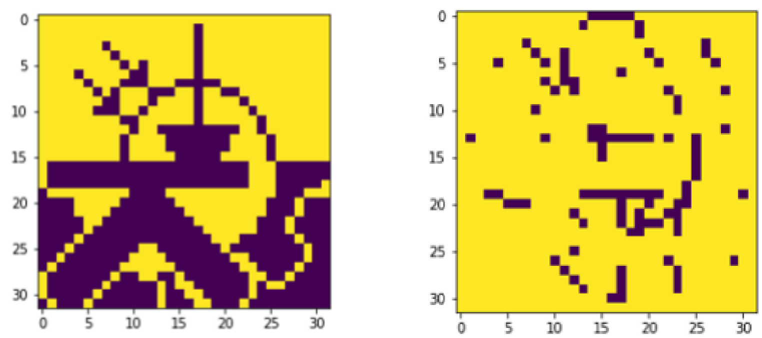
Yes, the patterns are stable.

Can the network complete a degraded pattern? Try the pattern p10, which is a degraded version of p1, or p11 which is a mixture of p2 and p3.

input: p10, output: same as the pattern stored in p1

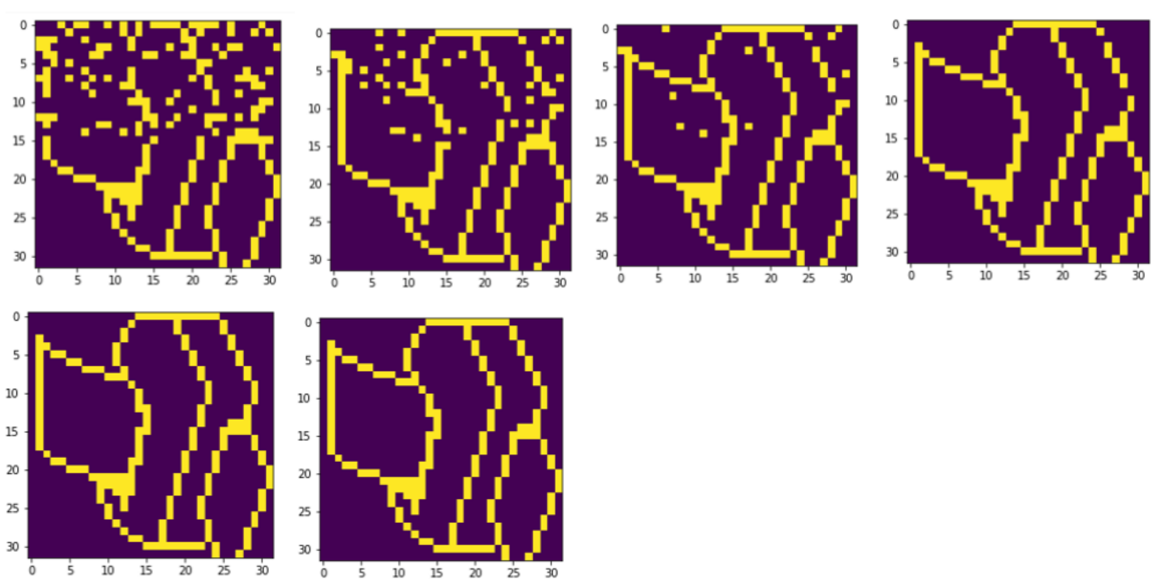


input: p11, output: same as the pattern stored in p3



Clearly convergence is practically instantaneous. What happens if we select units randomly? Please calculate their new state and then repeat the process in the spirit of the original sequential Hopfield dynamics. Please demonstrate the image every hundredth iteration or so.

Input: p10 random units and Iterations: 6



3.3

Can we be sure that the network converges, or will it cycle between different states forever?

If the input patterns are slightly similar, they tend to converge. If it is not similar, it will end up in local minima.

What is the energy at the different attractors?

Energy at attractors:

1 = -1439.390625

2 = -1365.640625

3 = -1462.25

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

The energy for distorted pictures:

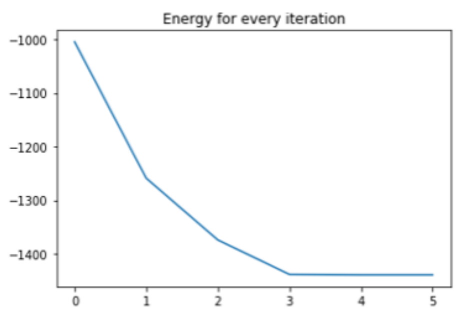
8 = -171.5468

9 = -267.5117

10 = -415.9804

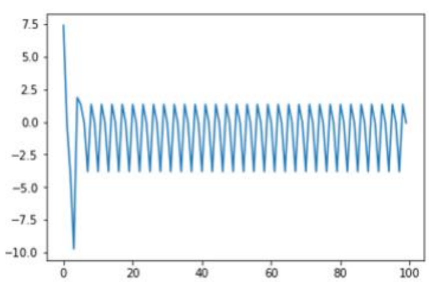
11 = -173.5

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



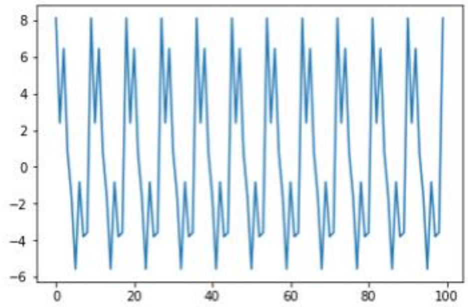
Generate a weight matrix by setting the weights to normally distributed random numbers and try iterating an arbitrary starting state. What happens?

Normally distributed random weight matrix is hard to converge.



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

Normally distributed random but symmetric weight matrix converges but not really only at some points.

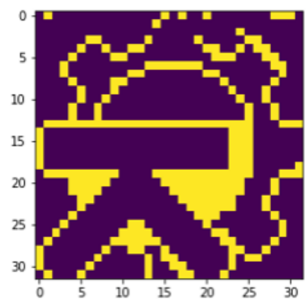


3.4

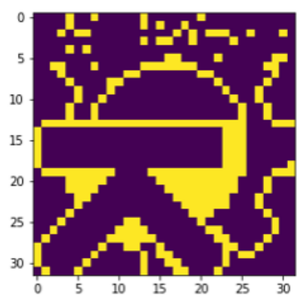
How much noise can be removed? Is there any difference between the three attractors with regard?

For attractor 1

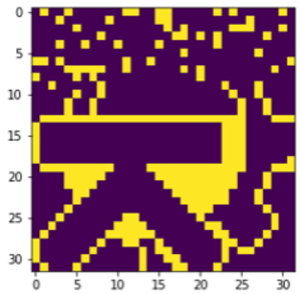
Noise ratio = 0.1(recalled with 1 iteration)



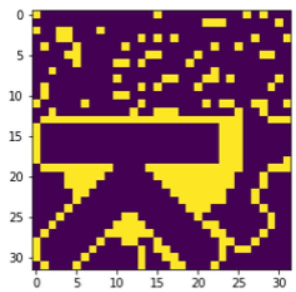
Noise ratio = 0.2(recalled with 1 iteration)



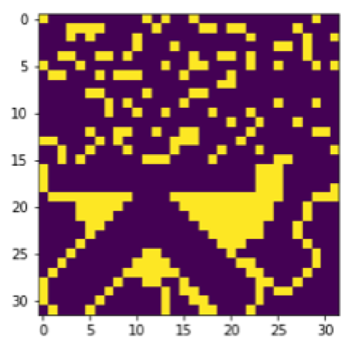
Noise ratio = 0.3(recalled with 1 iteration)



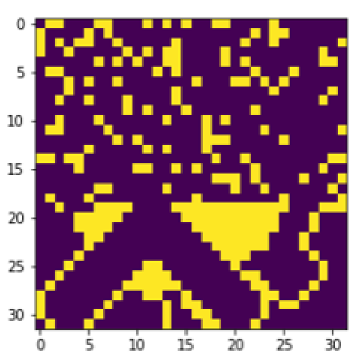
Noise ratio = 0.4



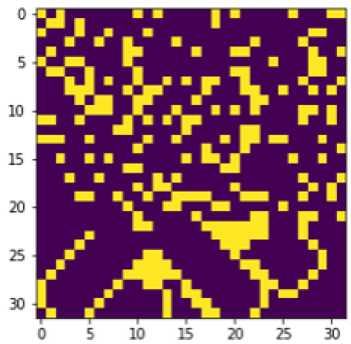
Noise ratio = 0.5



Noise ratio = 0.6



Noise ratio = 0.7



For attractor 2 it does not converge above the noise ratio 0.3

For attractor 3 it does not converge above the noise ratio 0.2

Does the network always converge to the right attractor? Do the extra iterations (beyond a single-step recall) help?

Good restoration when noise ratio is less than 0.4.

3.5

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

Three patterns could be safely stored. Error increased abruptly.

```
1 number of patterns
1 patterns remained stable

2 number of patterns
2 patterns remained stable

3 number of patterns
3 patterns remained stable

4 number of patterns
Not stable

5 number of patterns
Not stable

6 number of patterns
Not stable

7 number of patterns
Not stable

8 number of patterns
Not stable

9 number of patterns
Not stable
```

Try to repeat this with learning a few random patterns instead of the pictures and see if you can store more.

0.138N= 0.138*1024 = 141.312

Around 150 patterns could be safely stored. Much more uncorrelated patterns, increased capacity.

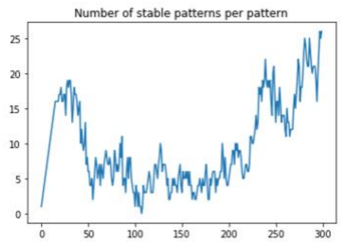
It has been shown that the capacity of a Hopfield network is around 0:138N. How do you explain the difference between random patterns and the pictures?

The random patterns are more dissimilar to each other, which makes the attractors far away from each other. Then, the probability of falling into spurious states is lowered. So, capacity increases.

300 random patterns and 100-unit network:

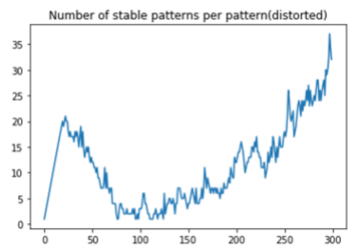
What happens with the number of stable patterns as more are learned?

Decayed stability if patterns increase. If no self-connections, hard to remain at current state and bad stability.



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?

For noise ratio=0.5



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

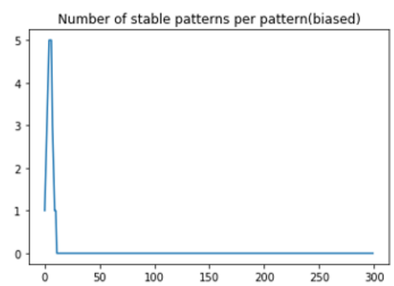
In some cases, it is 40 patterns to be stable.

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?

When bias is introduced into the patterns, there will be more +1 and the similarity between different patterns will be increased, so closer attractors and easier to fall into spurious states. Thus, the capacity reduces.

```
9 number of patterns
2 patterns remained stable

10 number of patterns
1 patterns remained stable

11 number of patterns
1 patterns remained stable

12 number of patterns
Not stable

13 number of patterns
Not stable

14 number of patterns
Not stable

15 number of patterns
Not stable

16 number of patterns
Not stable
```

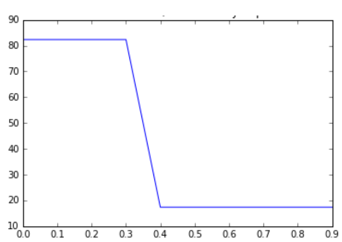
Seen that from 12 number of patterns it does not retrieve anything.



3.6

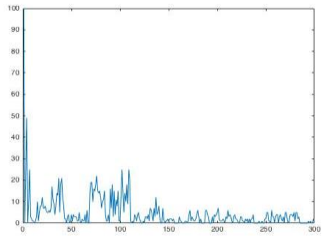
Try generating sparse patterns with just 10% activity and see how many can be stored for different values of theta.

Approx. three patterns can be stored.

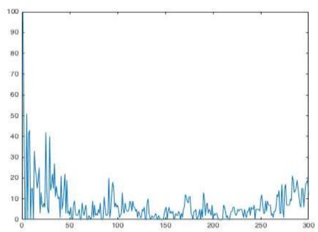


What about even sparser patterns (rho = 0:05 or 0.01)?

0.01



0.05



Capacity first increases and then drops to zero as theta increases.