Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 3

Hopfield Networks

Jordan Wu, Daniel Janischowsky

Oktober 1, 2019

1 Main objectives and scope of the assignment

- Implement a Hopfield Network and observer thei attractor and energy dynamics
- Investigate how Hopfield Networks react to noise and self connection removal in terms of capacity
- Investigate on different ways to deal with sparse patterns.

2 Methods

The entire lab was done in Python. We used Numpy to implement our own Hopfield class.

3 Tasks

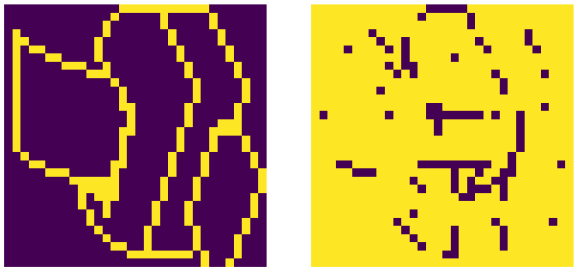
3.1 Convergence and Attractors

All of the distorted patterns converged to a stable pattern, but only $x1d$ and $x2d$ converged to their non-distorted trained pattern.

By converting numbers from 0 to 2^8 to binary (-1 and 1), then iterating to a stable point, we found 15 attractors.

If half of the pattern is wrong, the pattern will not converge to the correct attractor.

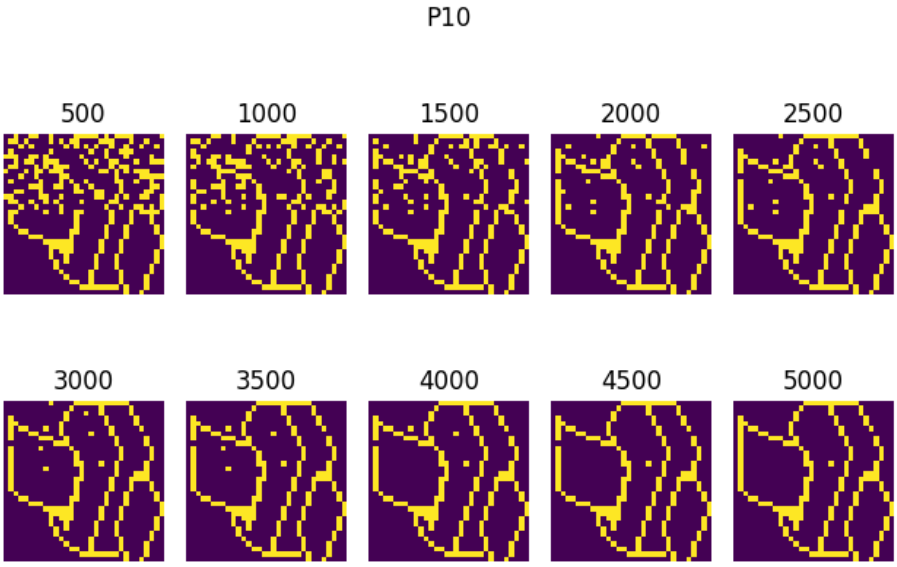
3.2 Sequential Update



P10 P11

Figur 1: "Predicted" Image

Network was able to recover the trained imaged from a noisy pattern for P10, but not for P11

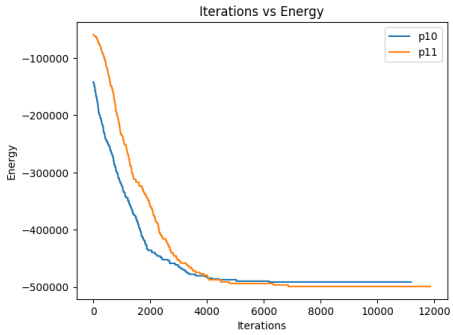


Figur 2: Sequential/Async Update (Every 500 Iteration)

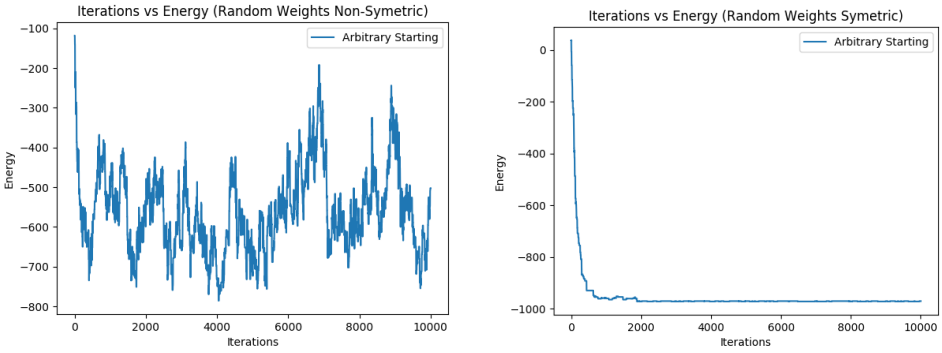
3.3 Energy

Picture	Type	Energy
p1	attractors	-491312.0
p2	attractors	-466138.66667
p3	attractors	-499114.66667
p10	distorted	-141988.0
p11	distorted	-59221.33333

Figur 3: Energy At Various Points



Figur 4: Energy vs Iteration



Non-symmetric Symmetric

Figur 5: Energy Curve for Random Weight Initialization

When weight matrix is not symmetric, energy does not decrease generally w.r.t. iterations. Results in random ups and downs. When it is symmetric, energy is decreasing generally, but not monotonically w.r.t iterations. If we set the diagonal of the weight matrix to 0, then it becomes monotonically decreasing. The equation for energy change when the weight matrix is symmetric and the

diagonal is 0 is

$$\Delta E = -\frac{1}{2}(x_j^{new} - x_j^{old}) \sum_i^n w_{ij} x_i \leq 0$$

Our version, where the diagonal is not 0, will result in x_j^{new} being slightly different since the diagonal will contribute to the affine transformation, but just slightly.

3.4 Distortion Resistance

Figur 6: Distortion Resistance for P1-P3

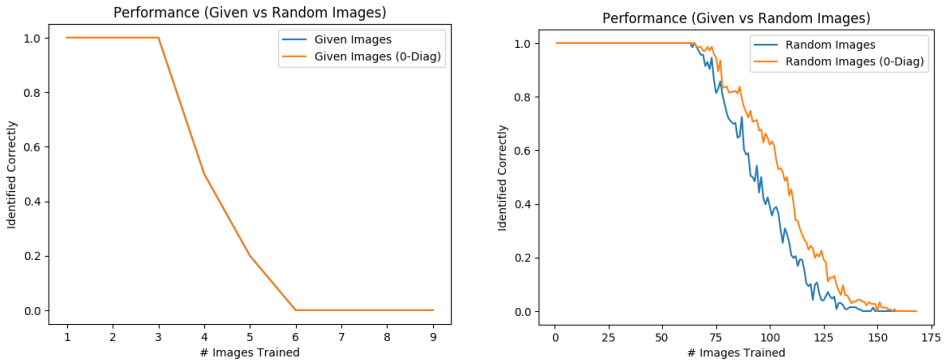
Image	Max Noise Mean	Max Noise Std
P1	0.426999	0.0363455
P2	0.429000	0.0301496
P3	0.414000	0.0436348

Figur 7: Noise Before Network Fails (10 runs each)

Generally, 40% noise is acceptable. We didn't notice a difference in noise tolerance between the images.

The network will not converge to the correct attractors if noise level is too high. Any additional steps after convergence of the little model will not change the result (Weights don't change). We noticed the inverse of the trained images also become attractors. When noise is $> 60\%$, the network converges to these inverted images.

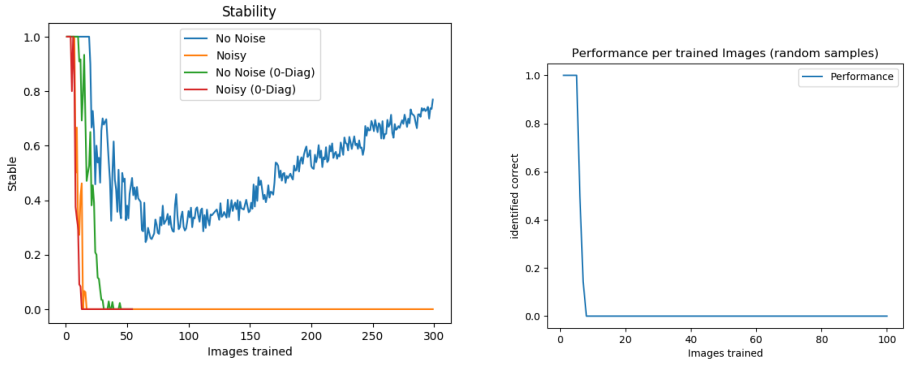
3.5 Capacity



(a) Given Image (b) Random Image

Figur 8: Capacity of Given vs Random Image

The network has a maximum capacity of storing 3 images. If more are stored the performance falls to zero correctly recovered image. With random images, the network can store up to 50 images. More then that results in the same abrupt performance decrease. The main difference lies in the orthogonality of the random samples, which will conclude to more diversity in the weight matrix.



(a) Stability (b) Biased Patterns Capacity

Figur 9: Stability and Bias

When comparing the stable patterns from already trained samples, we found that more patterns get recognised correctly with increasing amount of samples trained. The self connecting diagonal within the weight matrix increases to an enormous amount, creating a "identity matrix". This result in the network more often returning the same input pattern as output. The amount of recognised patterns drops quickly to zero when noise is added.

When removing the self connection in the weight matrix, the amount of stored random increases for this sample to about 65, but stays the same for given images. With a bias of 0.5 added to the randomly generated images, the amount of stored images drops to just about 5 because images are much less orthogonal.

3.6 Sparse Patterns

Figur 10: Capacity vs Activity and Bias

The "peak" performance is occurs where $\theta = \sqrt{p}$ for all 3 activities. The bias allows the $(Wx - \theta)$ term the to hover around 0, where the sign function works. If θ is too high or too low, the activation function will push all nodes to either 0 or 1.