Report:
A study on node removal and replacement in Hopfield Associative Memory Networks

Kruttika Bhat
Under the guidance of Dr. Sitabhra Sinha, IMSc Chennai

26th December, 2018

Background

Until recently, it was believed that we were born with all the neurons that we will ever have and further neurons are not generated. In 1962, Joseph Altman challenged this belief when he saw evidence of neurogenesis in the hippocampus of an adult rat brain. He also reported that newborn neurons migrated from their birthplace to other parts of the brain. In 1979, Michael Kaplan confirmed Altman's findings and later in 1983, he found neural precursor cells in the forebrain of an adult monkey. In 1998, Fred Gage and Peter Eriksson demonstrated that new neurons are generated in the dentate gyrus of adult humans. Their results indicated that the human hippocampus retains its ability to generate neurons throughout life. This brings in the question of neuron death. During development of the nervous system many more neurons are born than what will be needed. These extra neurons begin to die before we are even born and continue to die every day. This is a programmed death and is termed as apoptosis, which is essential for brain development. These appropriate neuronal deaths occur through the processes of neuron migration and differentiation. An unnatural death may occur due to diseases such as Parkinson's disease, Huntington's disease, Alzheimer's disease, a blow to the brain or a spinal cord injury. When a neuron dies, the existing connections are discarded and new connections are established. If the level of neurodegeneration is very high then it can result in loss of information.

Purpose

We will abstract this process of neurodegeneration and neurogenesis by modelling it in a Hopfield associative memory network which will be having dynamic connection weights i.e. the network will be evolving over time. We will then look at the following questions:

1. If nodes are removed and replaced at some rate and the network is evolving, then will the network degrade or stay robust?
2. What happens when the network is sparse? What would be the optimal sparsity that would give the best network performance?

Procedure

Hopfield associative memory network is considered to be one of the more straight forward neural network models in terms of implementation. A hopfield network is a fully connected auto-associative network with 2 state neurons (either binary or $\pm$ 1) and asynchronous updating. It is a recurrent network since it contains connections allowing output signals to enter as input signals. Associative memory is an application of recurrent networks. Associative memory refers to storing information as dynamically stable configurations. Given a noisy pattern, the network can recall the version it has been trained on. This is called auto-associative memory.
We can represent the network by a symmetric weight matrix. Note that there does not exist an edge from a node to itself.
So, $w_{ij} = w_{ji}$ *and* $w_{ii} = 0$, $\forall i, j \in n$, where n is the number of nodes (neurons) in the graph.

The weights are calculated using Hebbian learning rule. When the neurons are stimulated with some pattern, the synapses between them grow, strengthening the connections. If 2 neurons are anti-correlated, the synapses between them weaken. This is based on the theory that an increase in synaptic strength comes from a presynaptic cell's repeated stimulation of a postsynaptic cell. So, the weight between 2 nodes increases if both are either excitatory(+1) or inhibitory(-1), while it decreases if the 2 nodes are of different types.
The formula for calculating the weight matrix is as follows:

$$w_{ij} = \sum_{k=1}^{m} x_k^i x_k^j \tag{1}$$

where x is a pattern consisting of n values of ± 1 that is being stored in the network and there are m memories.

The network is able to recall stored patterns from distorted input patterns. If a random node i is chosen from the set of nodes then we use the formula

$$y_i = sign(\sum_{j=1}^{n} w_{ij} y_j) \tag{2}$$

This process of randomly choosing a node in y and reassigning its value is performed till y reaches an attractor i.e it converges to a pattern stored in the memory.
This way of learning is called one shot learning. It is not adaptable when new memories need to be stored later on. There also exists a memory limit on the hopfield network using Hebbian learning rule. The network is able to learn and exactly correct up to $\frac{n}{2logn}$ random inputs of length n each.

Two methods were used for ensuring that the network evolves over time.

1. Storkey learning algorithm
2. Iterative learning algorithm

Storkey learning algorithm

The learning rule is as follows:

$$w_{ij}^0 = 0, \forall i,j \text{ and } w_{ij}^v = w_{ij}^{v-1} + \frac{1}{n}x_i^v x_j^v - \frac{1}{n}x_i^v h_{ji}^v - \frac{1}{n}h_{ij}^v x_j^v$$

where $h_{ij}^l = \sum_{k=1,k\neq i,j}^{n} w_{ik}^{l-1} x_k^l$ is a form of local field at neuron i.

While the Hebbian learning rule is local and incremental it has a low absolute capacity. This capacity decreases further if patterns are correlated. While, the Storkey rule has a higher capacity than Hebb rule. This capacity is maintained even for correlated patterns.

Iterative learning algorithm

The algorithm uses a learning rate ϵ. The algorithm is given by the equation:

$$w_{ij}(t+1) = (1-\epsilon)w_{ij}(t) + \epsilon x_i(t)x_j(t) \tag{3}$$

w_{ij} depends only on information available at the two adjacent neurons, i and j.

Observations

1. When a node is removed and re-added, the weight matrix will be updated. After which, we check a distorted pattern with the network. This process is performed in a loop. In order to ensure that the patterns are not lost i.e the network doesn't break down entirely, we need to expose the network to the patterns regularly.

   - Removing a node changes the values of the respective row and column of the weight matrix. The new weight may be less than or greater than the previous weight. This in turn affects the sum depending on which node was removed. Based on this, the memory may be given correctly at the output or it may not.
   - The new weights may be assigned as 0 or a random value picked from normal distribution with mean 0 and standard deviation 1. Zero would imply that the connections don't actually exist, so it would be more accurate to initialise with a random value.
   - On another iteration of node removal. The fault caused by the previous node removal may be fixed. This may take multiple iterations.
   - If no nodes are removed and we keep exposing the network to the same set of memories, then the weight matrix will tend towards plus or minus infinity.
   - If the weights are reassigned with random values picked from a normal distribution and $w_{ii} \neq 0$, then after some number of iterations of node removal we find that the network becomes unstable and is not able to reach an attractor.

2. If we keep removing and replacing nodes without exposing the network to the set of patterns:

   - If we remove a node and reassign the respective row and column in the weight matrix with zeros, then the weight matrix eventually converges to a zero matrix and the network reaches an attractor where all the neurons are in an excited state (+1), irrespective of the distorted input pattern.
   - If we remove a node and reassign the weights with random values picked from a normal distribution then the network converges to a random attractor in every iteration. The patterns stored initially are entirely lost.

- If we remove a node and reassign with sign(random values picked from a normal distribution) then the network remains robust for a few iterations but thereafter it degrades.

3. If the network is exposed to the patterns at every ith iteration then the network gives the correct attractor at every (i+1)th iteration. For all the other iterations a different attractor is reached.

4. Iterative algorithm and Storkey algorithm show better performance as compared to Hebb's rule and one shot learning.

Conclusion

- The dynamics and working of Hopfield network have been understood using different learning algorithms.
- As the network evolves, the weight matrix does not converge nor does it regain it's initial value.
- The network remains robust so long as it is repeatedly exposed to the learning patterns. This is not affected by the number of nodes removed.
- The network performs better if the input patterns are sparse.

Further Work

1. The network needs to be made sparse and optimal sparsity needs to be found, if any.
2. The weight matrix can be assigned with random values initially and attractors can be found. Thereafter, on each node removal, the attractors can be compared with the inital attractors.
3. We can check the difference between final attractors and original attractors as a function of sparsity.

References

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities" (1982)

[2] R. Singh, S. Dasgupta and S. Sinha, "Extreme variability in convergence to structural balance in frustrated dynamical systems", *Europhys. Lett.* 105 (2014) 10003.

[3] Amos Storkey and Romain Valabregue, "A Hopfield learning rule with high capacity storage of time-correlated patterns"

[4] Rishidev Chaudhuri and Ila Fiete, "Associative content-addressable networks with exponentially many robust stable states" (2017)

[5] Hee-Kuck Oh "The relaxation method for learning in artificial neural networks" (1992)

[6] Kevin Gurney: Notes on Neural Networks
`http://web.cs.ucla.edu/ rosen/161/NeuralNotes/NeuralNotes.html`

[7] James B Hutchins and Steven W. Barger, "Why Neurons Die: Cell Death in the Nervous System" *Wiley-Liss, Inc.*(1998)