Random CTLNs

Eric Han, Caitlin Lienkaemper

July-August 2024

1 Project Summary

The goal of this project was to observe the effects of symmetry in random combinatorial linear-threshold networks (CTLNs). The simulation model was built in Python. A major step taken this summer was the completion of a small but complete program to generate averaged heatmaps of how network dynamics behaved across various probability parameters. Additionally, at the tail-end of the project, a more mathematically focused thrust to the problem was conceptualized and a more concrete vision of tackling the symmetry problem was realized.

2 Model

The CTLN model is a simplified mathematical model of neural networks that focuses on the fine connectivity between neurons as the key factor affecting the resulting dynamics of the network. CTLNs exhibit strongly nonlinear dynamics, showcasing behavior such as multistability, chaotic attractors, and limit cycles. Physically, these models are similar to other classical attractor neural networks that model associative memory. For instance, the manifestation of limit cycles can indicate central pattern generators controlling periodic behaviors in animals. The main method of analysis for the CTLN model is through a graph theoretical lens. One can look directly at the structure and specific connectivity of the graph to determine the underlying dynamics of the network.

3 Code

3.1 Algorithm

The goal of this model is to generate random graphs with set parameters for symmetry and edge connection probability. This is done through the generation of a random square adjancency matrix of size n. Only the top diagonal of the matrix is generated with values of $\{-2, -1, 1, 2\}$. These values denote no edge, a unidirectional edge from j to i, a unidirectional edge from i to j, and a bidirectional edge between j and i, respectively. The respective weight table for the values is $\{1-p, p(\frac{1-q}{2}), p(\frac{1-q}{2}), pq\}$, where p denotes edge connection probability and q is the symmetry parameter. As q increases, general symmetry of the graph increases linearly.

Definition 3.1. Let i, j be any two vertices with an edge between them in a directed graph. i and j are considered *symmetric* if there exists a bidirectional edge between them. A graph is considered fully symmetric if every edge in the graph is a bidirectional edge; or if there exist no edges in the graph.

A consequence of this generation method is that the generation of any bidirectional edge is controlled exclusively by the value $\{2\}$, that is, a bidirectional edge cannot be generated by the creation of 2 unidirectional edges in opposing direction between vertices i and j.

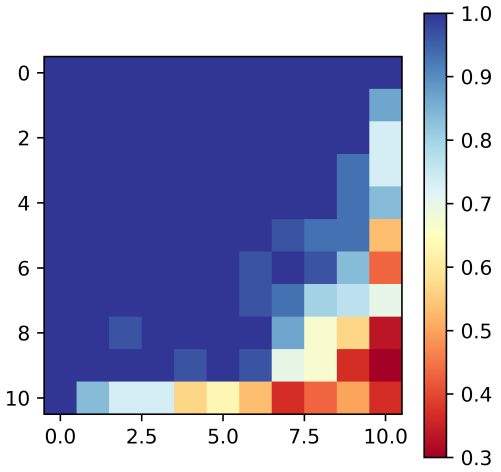

Figure 1: Heatmap for graph $n = 100$

3.2 Application

We use this algorithm to simulate dynamics for the full spectrum of edge connection probabilities and symmetry parametrizations. We generate random weight matrices for every unique combination of p, q. p and q are taken in intervals of 0.1, but we do have the capacity for finer simulation. A stability analysis is included in the code but was ultimately too slow, as it attempted to analyze the stability of every node in the network.

3.3 Heatmaps

Due to the construction of the main algorithm, it is possible to generate heatmaps showcasing the average dynamics of single neurons. We use a threshold-based system to detect fixed points, that is, if the oscillation of the dynamics of a single neuron does not escape the bounds of $[-0.1, 0.1]$ for a long timesecale $t = (0, 600)$, $dt = 0.1$, we consider it a fixed point. More specifically, we observe the last 50 entries in the list of generated values using the RK4 method to see if the neuron's dynamics fall outside the threshold. If the dynamics fall outside the threshold, we consider it a non-fixed point. These non-fixed points may take the form of limit cycles, but from observation through simulation, they are more likely to be chaotic.

We assign a neuron that settles to a fixed point the value $k = 1$. We assign a neuron that is *not* a fixed point the value $k = 0$. We take the average of all $k_{p,q}$ for all combinations of p, q.

We average the "fixed-point-edness" across 30 trials 7 times, for a total of 210 trials for a matrix of size m. A heatmap is generated in this method, using a spectrum for $\bar{k}_{p,q}$ from $[0, 1]$.

The central issue with this approach is the tolerance used to check whether a neuron settles to a fixed point. Because the energy in the system is shared amongst all firing neurons, the more neurons that exist, if the system is predisposed to being chaotic, the smaller the oscillations will be for every neuron. For large networks, these oscillations become small enough that non-fixed point neurons are marked as fixed points.

4 Problem

We are looking at the expected number of stable fixed points in any size n graph by using the expected number of target free cliques as a function of the edge connection probability

p and the symmetry parameter q.

Because the CTLN model can be easily analyzed through a graph-theoretic lens, we aim to analyze stability through clique formation in a random graph.

A recent, interesting result concerns the expected number of maximal cliques in any random n-sized graph. This is important because one can discern the dynamics of the graph through analyzing its graph structure. Because edge generation of our random graph is independent, that is, the generation of an edge does not have any impact on the generation of other edges, we can look for the probability of a single node in the graph destroying maximality.

5 Key Definitions

Definition 5.1. A *clique* is defined as a subset of vertices on a graph such that every two distinct vertices in the clique share a bidirectional edge.

Definition 5.2. A *maximal clique* is a clique that is not contained in a larger clique.

Definition 5.3. The *target of a clique* σ is defined as a clique that receives an edge from every node in σ. A clique with no targets is called *target-free*.

A clique σ is the support of a fixed point if and only if it is target free. That is, if the given clique is maximal, then it is a stable fixed point support.

6 Findings

Proposition 6.1. *Let E be the event that adding a node does not destroy maximality. Given the existence of a k-clique in a size N Erdös-Renyi graph, the probability that adding a node does not destroy maximality is given by*

$$P(E|k\text{-clique exists}) = (1-p^k)^{N-k} \tag{1}$$

where p is the edge connection probability.

Proof. Let N denote the size of a random Erdös-Renyi graph G. Let p denote the edge connection probability of G. Let $\sigma_k \in G$ be any clique of size k in G. Consider the probability of event X, being the event that adding any individual vertex $g_i \in G$ ruins the maximality of σ_k. The vertex would require an edge from every vertex in σ_k. For an edge connection with probability p to be generated between every vertex in σ_k and a single vertex g, we require k edges to be independently drawn, obtaining the following result:

$$P\left(g \text{ ruins maximality}|\sigma_k \text{ exists}\right) = p^k. \tag{2}$$

So, $P(X^c|\sigma_k \text{ exists}) = 1-p^k$, where X^c denotes the event that adding an individual vertex will *not* destroy the maximality of G.

We would like to generalize this result to any possible $g_i \in \sigma$. Note that for any σ_k, there exist $N-k$ unique vertices that are not in σ_k. Additionally, edge generation is independent between vertices contained in σ and g_i. So, the probability that any single $g_i \in G$ does not ruin the maximality of k is $(1-p^k)^{N-k}$. That is,

$$P\left(X^c|\sigma_k \text{ exists}\right) = (1-p^k)^{N-k}. \tag{3}$$

□