THE EFFECT OF WATTS-STROGATZ AND BARABÁSI-ALBERT GRAPHS
ON MEMORY FORMATION

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Ethan Wolfe

June 2024
COMMITTEE MEMBERSHIP

TITLE: The Effect of Watts-Strogatz and Barabási-Albert graphs on Memory Formation

AUTHOR: Ethan Wolfe

DATE SUBMITTED: June 2024

COMMITTEE CHAIR: Mugizi Robert Rwebangira, Ph.D.
Assistant Professor of Computer Science

COMMITTEE MEMBER: Jonathan Ventura, Ph.D.
Associate Professor of Computer Science

COMMITTEE MEMBER: Daniel P. S. Frishberg, Ph.D.
Assistant Professor of Computer Science
Understanding higher level cognitive processes is a central problem in neuroscience. The Neuroidal model provides a useful framework for posing these problems in a computer science context. There has been significant recent work trying to understand memory capacity in the Neuroidal model but this work was done assuming that the network of neurons was an Erdős–Rényi random graph. However, the network of neurons in the brain has been shown to exhibit small-world properties, which are not present in Erdős–Rényi graphs. In this research we explore replacing Erdős–Rényi graphs with Watts–Strogatz and Barabási–Albert graphs in order to more accurately model the biological reality. We aim to investigate the implications for memory capacity and interference within the Neuroidal model.
ACKNOWLEDGMENTS

Thanks to:

- My advisor, Dr. Mugizi Rwebangira, for his guidance and support
- My fellow researchers, for their collaboration and feedback
- My family, for their encouragement and understanding
- My friends, for their companionship and humor
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>5.2</td>
</tr>
<tr>
<td>Results of the Neuroidal model with parameters ( n = 10k, d = 128, k = 32 )</td>
<td>33</td>
</tr>
<tr>
<td>Results of the Neuroidal model with parameters ( n = 10k, d = 2048, k = 64 )</td>
<td>33</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>5.1</td>
<td>Edge Probability ($p$) vs. Mean Path Length ($L$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
<tr>
<td>5.2</td>
<td>Edge Probability ($p$) vs. Global Clustering Coefficient ($C$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
<tr>
<td>5.3</td>
<td>Edge Probability ($p$) vs. Mean Local Clustering Coefficient ($C_i$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
<tr>
<td>5.4</td>
<td>Edge Probability ($p$) vs. Mean Degree Distribution ($k$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
<tr>
<td>5.5</td>
<td>Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Erdős–Rényi graphs of size 1k for varying edge probabilities $p$</td>
</tr>
<tr>
<td>5.6</td>
<td>Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Barabási–Albert graphs of size 1k for varying edge probabilities $p$</td>
</tr>
<tr>
<td>5.7</td>
<td>Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Barabási–Albert graphs of size 1k for varying edge probabilities $p$</td>
</tr>
<tr>
<td>5.8</td>
<td>Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Barabási–Albert graphs of size 1k for varying nearest neighbors $k$</td>
</tr>
<tr>
<td>5.9</td>
<td>Edge Probability ($p$) vs. Edge Count ($e$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
<tr>
<td>5.10</td>
<td>Edge Probability ($p$) vs. Mean Local Efficiency ($E_{loc}$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
<tr>
<td>5.11</td>
<td>Edge Probability ($p$) vs. Mean Local Efficiency ($E_{glob}$) averaged over 100 simulations on graphs of size 10000</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

An increasingly discussed problem in neuroscience is the modeling of the human brain. More specifically, research has been focused on answering the question of “How many memories can the human brain hold?”. Recent attempts to answer this question have turned to computer science to simulate the problem.

A graph theory model attempting to solve this problem was first proposed in 2005, using Erdős–Rényi graphs to model the neurons and synapses in the brain, and subgraphs within to model memories [47]. This model proposed two main algorithms for the formation of memories, JOIN for creating new memories from existing neurons, and LINK to connect two pre-existing memories. Within this model, Valiant explored both a disjoint representation, where memories are not allowed to share neurons, and a shared version where they can form intersecting connections. While the capacity of the disjoint model is easily calculated, as within the model, a primary constraint is the uniform size of memories within the graph. The shared memory version, however, requires insight into memory interference. This is when one memory being fired, or activated, causes the unintended firing of another memory that has become too interconnected. Valiant left the characterization of this interference within the Neuroidal Model to future work.

Valiant revisited the problem in 2017, looking at interference within the model concerning the LINK algorithm [49]. Perrine later expanded this work to include the JOIN algorithm along with the creation of a generalized framework for simulating the Neuroidal model [36]. Chowdhury also investigated interference by utilizing set
theory to derive a set of equations for calculating capacity, validated by simulations within Perrine’s framework [9]. There have also been studies to quantify the memory capacity of the Assembly Calculus, a derivative of Valiant’s Neuroidal model [52]. These studies have focused almost exclusively on the creation of memories and formal definitions for the capacity of the models.

The term “small-world” was first coined by Milgram in 1967, when he conducted an experiment to determine the average path length between two people in the United States [29]. Since then, many neuroscience studies have investigated the structure of different brain networks and found that they exhibit small-world properties [43, 41, 4, 56, 19]. There have also been studies that have utilized small-world graphs to model brain networks themselves [30].

Small-world properties have not yet been applied to the Neuroidal model. Many graph models have been proposed that exhibit small-world properties, most notably the Watts–Strogatz and Barabási–Albert models. Watts and Strogatz proposed the Watts–Strogatz model in 1998, to attempt to bridge the gap between regular and random graphs [51]. Albert and Barabási proposed the Barabási–Albert model in 2002, to model the natural growth of chemical reactions and internet topology [1].

Small-world graphs have yet to be applied to the Neuroidal model, and this work aims to bridge the gap, connecting findings of real world networks to the Neuroidal model. To reach this goal, we first examine previously defined constraints and assumptions and prove that they will be unchanged within the frame of Watts–Strogatz and Barabási–Albert graphs. We then utilize the framework put forth by Perrine to empirically evaluate the capacity of memory within these models.
Findings indicate that both Watts–Strogatz perform similar to Erdős–Rényi graphs and Barabási–Albert graphs due to a higher degree distribution, cause a higher memory size with a higher interference rate.

This work builds upon the Neuoridal model by introducing two distinct graph back-ends that can be used to simulate the creation of memories. These advancements move the simulation closer toward biological accuracy and open up possibilities for further research to better understand memory capacity.
Chapter 2

BACKGROUND

In this chapter, we first discuss the characteristics of graphs that we use to measure small-world properties. Then, we discuss Erdős–Rényi, Watts–Strogatz, and Barabási–Albert graphs, and how they differ from each other. Finally, we discuss the Neuroidal model and its constraints upon the underlying graph.

2.1 Graph Measurements

When discussing graphs in the context of small-world properties, there are four main measurements that can be used to describe the graph: average path length, clustering coefficient, degree distribution, and efficiency. All of these measurements were collected using the adjacency matrix given to the Neuroidal model within the framework, as a directed graph measurement as well as an undirected measurement by transposing the adjacency matrix along the diagonal.

For all discussion of graphs and their properties, a node in the graph will also be commonly referred to as a neuron, or a neuroid, which is the term used within neural network models. Edges in the graph will also be commonly referred to as synapses.

2.1.1 Average path length

The average path length is the mean of all shortest path lengths, and is a measure of how well-connected the graph is. It was first defined by Milgram in 1967 as the number of intermediaries between two people in a social network [29].
It is commonly defined as:

$$L = \frac{1}{n \cdot (n - 1)} \sum_{i \neq j} L_{ij} = d(v_i, v_j)$$  \hspace{1cm} (2.1)$$

Where $d(v_i, v_j)$ is the shortest path between nodes $v_i$ and $v_j$ within the graph $G$.

To utilize this equation however, it was first needed to adapt it to work with adjacency matrices, to fit the data structures used in the framework developed for this work. Doing so, the equation was adapted to:

$$L = \frac{1}{n \cdot (n - 1)} \sum_{i \neq j} \min \{A_{ij}^1, \ldots, A_{ij}^k\}$$  \hspace{1cm} (2.2)$$

Where $A$ is the adjacency matrix of the graph $G$ with all non-existent edges replaced with infinity. This is derived from the fact that the $ij$’th entry of the matrix $A^k$ is the number of paths of $k$ length paths between nodes $i$ and $j$.

2.1.2 Clustering coefficient

The clustering coefficient of a graph measures the cliquishness, or the connectedness of neighborhood subgraphs. This is the second defining characteristic of small-world networks as introduced by Watts and Strogatz in 1998 [51]. There are two variations of the clustering coefficient, local and global. The local clustering coefficient is the ratio of the existing edges between the neighbors of a node to the total possible edges between them. The global clustering coefficient is the ratio of all triangles to all triples in the graph. A triangle is a set of three nodes where each node is connected to the other two nodes and a triple is a set of three nodes where each node is connected to at least one other node.

The local clustering coefficient was first defined by Watts and Strogatz as:
\[ C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i \cdot (k_i - 1)} \quad (2.3) \]

Where the edge between \(v_j\) and \(v_k\) is denoted as \(e_{jk}\), \(N_i\) is the set of neighbors of \(v_i\), and \(k_i\) is the degree of \(v_i\). To compute the local clustering coefficient upon the adjacency matrix, the equation was adapted to:

\[ C_i = \frac{\sum_{j,h} A_{ij}A_{ih}A_{jh}}{k_i \cdot (k_i - 1)} \quad (2.4) \]

Which uses the property of the adjacency matrix that \(A_{ij}A_{ih}A_{jh}\) is 1 if there is a triangle between nodes \(i, j,\) and \(h\), and 0 otherwise [55].

The global clustering coefficient was also defined by Watts as:

\[ C = \frac{3 \cdot \text{number of triangles}}{\text{number of connected triples}} \quad (2.5) \]

Using the same properties of the adjacency matrix, the equation was adapted to:

\[ C = \frac{\sum_{i,j,h} A_{ij}A_{jk}A_{ki}}{\sum_i k_i \cdot (k_i - 1)} \quad (2.6) \]

Although nearly equivalent, \(C \neq \sum_i C_i\) as the number of triangles and triples would be counted multiple times depending on the number of connections between nodes. Although faster, the adjacency matrix method does not correctly count directed edges, and for the clustering coefficient measurements shown in this paper, they were calculated by iterating over every edge and counting the triangles and triples.
Unlike the average path length, Erdős–Rényi graphs do not always have a high clustering coefficient, because it measures local connectivity, not global connectivity. Watts–Strogatz graphs were developed to have a higher clustering coefficient, while still maintaining a low average path length.

2.1.3 Small-world Properties

A graph is said to be a small-world graph, or to exhibit small-world properties when it has both a low average path length and a high clustering coefficient. More specifically, the average path length should be shown to be $L \propto \log N$. When considering the clustering coefficient, it should be shown that $1 > \frac{C}{C_r}$ where $L$ and $C$ are the average path length and clustering coefficient of the graph, and $L_r$ and $C_r$ are the average path length and clustering coefficient of a random graph with the same average degree as the graph being measured.

It is also common to study the scale of the degree distribution of a graph, such that $P(k) \sim k^{-\gamma}$ where $\gamma$ is a parameter within the range $2 < \gamma < 3$, or that the distribution decays to an asymptotic constant.

2.1.4 Degree distribution

The degree distribution of a graph is the occurrence of nodes with a given degree. It can be simply formulated as:

$$P(k) = \frac{n_k}{n} = \frac{\sum_j A_j}{n}$$  \hspace{1cm} (2.7)
Where $P(d)$ is the probability that a randomly selected node has degree $d$, $n_d$ is the number of nodes with degree $d$, and $n$ is the total number of nodes in the graph.

Although Erdős–Rényi, Watts–Strogatz, and Barabási–Albert graphs were not developed with a specific degree distribution in mind, Erdős–Rényi graphs generally have a binomial distribution but can also be shown to have a poisson distribution[11], Watts–Strogatz graphs have a dirac delta degree distribution [3], and Barabási–Albert graphs have a power-law degree distribution [1].

### 2.1.5 Efficiency

The efficiency of a graph is a measure of how well information can be transferred between nodes. The global efficiency $E_{glob}$ measures how quickly information can be transferred within the graph. It was first defined by Latora and Marchiori in 2001 as:

$$E_{glob} = \frac{1}{n \cdot (n-1)} \sum_{i \neq j} \frac{1}{L_{ij}}$$

(2.8)

They also defined a local efficiency $E_{loc}$, which measures how well information can be transferred within the neighbors of a node without passing through the node itself.

$$E_{loc} = \frac{1}{n} \sum_i E_{glob}(G_i)$$

(2.9)

For an unweighted graph, this measure will be well estimated by $\frac{1}{L}$. For the neuroidal model, this is a key factor, and the local efficient $E_{loc}$ will be more enlightening, as it measures the fault tolerance of the graph.
2.2 Erdős–Rényi graphs

There are two different variations of Erdős–Rényi graphs; $G(n, p)$, and $G(n, M)$. These follow the two different approaches to forming a random graph of $n$ nodes, and $M$ edges. This paper is primarily interested in the $G(n, p)$ variation, as it is the one utilized within the Neuroidal Model.

To create a $G(n, p)$ graph, first $n$ nodes are created, and then each node has a $p$ percentage to form a connection to every other node. In practice, almost all $G(n, p)$ graphs are equivalent to the $G(n, M)$ graph where $M = \binom{n}{2}p$. This is the method that Perrine utilized when simulating the Neuroidal model, though this work used the $G(n, p)$ variant, as it was faster to generate when using adjacency matrices.

Erdős–Rényi graphs were selected by Valiant for the Neuroidal model due to the low average path length, and the fact that neural networks are believed to be richly interconnected [47]. Studies since have shown that real-world brain networks exhibit both power-law degree distributions and small-world properties, neither of which are present in Erdős–Rényi graphs [38, 44, 34, 37].

2.3 Watts–Strogatz graphs

Watts–Strogatz graphs are expansions of Erdős–Rényi graphs where instead of forming every edge with probability $p$, each node is first connected to $k$ nearest neighbors before every edge is considered with the probability $p$ to be rewired, or replaced with an edge to a randomly selected edge. For values where $p \to 0$, nodes remain connected to their nearest neighbors, and as $p \to 1$, the graph becomes increasingly similar to an Erdős–Rényi graph.
This process results in Watts–Strogatz graphs with a $p \to 0$ having a higher average clustering coefficient, and graphs with a $p \to 1$ having a lower average path length.

Converting the Watts–Strogatz graph to a directed graph is similar in concept to the conversion of the Erdős–Rényi graph, where the nodes iterated over are the source, and the neighbors are the target. Although this changed the degree distribution of the graph, as can be seen later, it does not remove the small-world properties of the graph.

### 2.4 Barabási–Albert graphs

Barabási–Albert graphs are initialized with only one parameter $m$, the attachment parameter. In practice, a $n$ parameter is also used, to specify the finished size of the network. The graph is first initialized with $m$ nodes, each with a connection to every other node. At each step, a node is added to the graph, which then samples $m$ nodes from the existing network and considers a connection to each with a probability $p_i$. For each node, this probability is calculated as

$$p_i = \frac{k_i}{\sum_j k_j}$$

(2.10)

where $k_i$ is the degree of node $i$ and the sum is over all existing nodes $j$. Nodes are added until the network reaches the required size. This results in nodes with a larger number of edges quickly gaining more, while nodes with fewer edges maintain a lower degree.

For converting both Watts–Strogatz graphs and Barabási–Albert graphs from their default undirected form to a directed form, we took the most direct form of this
conversion, changing what would normally be an undirected edge to a directed edge. This was done to lessen confusion when working with existing graph libraries, and any efforts to better retain the small-world properties of the graph would fall between the current version and the undirected version for each graph characteristic that was measured and shown in the results.
In this chapter, we go over related work that has investigated the neuroidal model and the assembly calculus, as well as studies that have explored the real-world structure of neural models.

3.1 Neuroidal model

Valiant proposed the Neuroidal model in 1994, to study memory formation in neural systems in a biologically accurate manner [45]. The model consists of a collection of neuroids, which are simplified representations of neurons. Each neuroid has a threshold unit for firing, and is connected to other neuroids via directed, weighted synapses. The construction of this network used four parameters observed in the cortex: neuron count, synapses per neuron, relative synaptic strength, and switching time.

Valiant later proposed two algorithms in 2005, JOIN and LINK, that form memories within the Neuroidal model while respecting biological constraints [47]. JOIN implements memory formation, taking existing elements A and B within the network, and modifying it to add a memory C that fires if and only if A and B fire. If during the creating of memory C, another memory is activated, the model is considered to have reached capacity. A memory is considered to be firing if more than half of its associated neurons are firing. The resulting size of the memory C is considered the memory size, and the parameters of the model are tuned to make these sizes the same, and was the primary focus of Valiant’s work [47]. LINK implements memory
association, taking existing elements A and B, and modifying the network so that firing A, causes the firing of B.

To reach the desired measurement of memory capacity, we calculate the number of memories that can be joined from the collection of starting memories. If all possible combinations of starting memories are joined, the model is considered to have memorized all possible information, and if the interference threshold is passed when joining memories, the model is considered to have reached capacity for the given parameters.

### 3.2 Assembly Calculus

Expanding upon the Neuroidal model, Papadimitriou et al., 2020 propose the Assembly Calculus, adapting Valiant’s Neuroidal model to use Assemblies [33]. Assemblies were first introduced by Hebb in 1949 as a way to define large sets of neurons that are believed to imprint memories. Assemblies are better connected than arbitrary sets, which Valiant names the sub-graphs within the Neuroidal model [17].

The authors also propose two new memory formation algorithms, Project, which creates a “copy” of an existing assembly $x$ in a new brain area $B$, and Merge, which combines two existing assemblies $x$ and $y$ to a new assembly $z$.

### 3.3 Connection to this work

Though the Neuroidal Model and Assembly Calculus both currently use Erdős–Rényi graphs, it is believed that they can be modified to utilize the Watts–Strogatz and Barabási–Albert structures without modifying the resultant information or algorithms. Although we don’t simulate the Assembly Calculus in this work, the results
of this study could be applied to the Assembly Calculus, as it is a derivative of the Neuroidal model.

### 3.4 Real-World Neural Networks

There is a growing body of research that has investigated the structure of real-world neural networks. In 2000, Young et al., investigated the connectivity of the macaque and cat cortex, finding them to be highly clustered [54]. At the same time, Sporns et al., investigated dynamical systems in the brain and their relation to complex graph systems [42]. Later, in 2004, Sporns extended his network looking at neural connectivity through the filter of small-world networks [43]. Later that year, R. Chialvo investigated the criticality of the brain, and how it stays at a regime governed by a power-law [38]. In 2006, Bassett et al., investigated the small-world properties of the human brain, finding that the brain behaves in a fractal representation of small-world graphs [4, 5]. At the same time, Sporns and Honey also investigated the small-world properties of the brain [41]. In 2009, Portillo et al., investigated the adaptive properties of the brain, and how it can rewire itself to maintain small-world properties and a hierarchical power-law structure [37]. In 2011, Varshney et al., investigated the structure of the C. Elegans neural network, to help further neural research [50]. In 2017, Tomasi et al., investigated the brain’s network topology, finding that it exhibits small-world properties [44]. There are doubtless other studies that have investigated the structures of the brain, but these are the most relevant to this work, and show the abundance of research in this area in comparison to the Neuroidal model and furthering neural simulation ability.
This chapter will discuss the methods that were used to assess the relevant properties of Watts–Strogatz and Barabási–Albert graphs and convert the Neuroidal model to function upon them.

To convert the neuroidal model to function upon a Watts–Strogatz or Barabási–Albert graph, we must first understand the constraints that the Neuroidal model has upon the underlying graph.

In his landmark paper, Valiant puts forth the assertion, “It is convenient for the sake of analysis to view the edges as being generated in that manner afresh, rather than as fixed at some previous time.” [47]. This was convenient for Valiant, as it allowed him to consider the theoretical, utilizing the binomial probability of an edge existing when needed, instead of undertaking the memory intensive task of creating a graph at large values of $n$. This means however, that after creation, we can assume that the graph $G$ is not being modified, as there was never a graph to begin with, instead modifying a collection of separate parameters for each neuron and synapse. This is further solidified by the fact that the Neuroidal model’s algorithms, JOIN and LINK do not consider the graph itself other than to use it as a basis for the structure of a neuron’s connections [47]. Recently, in his 2023 paper, in which he implemented a working simulation of the Neuroidal model, Perrine puts forth a formation algorithm that creates an Erdős–Rényi graph, then assigned vertex and edge properties to the graph, but at no point modifies the structure of the graph after its initial creation [36].
Now that we have established that the graph is not being modified, we can consider
the parameters of the Neuroidal model and how they are used to create the graph.
The Neuroidal model, as defined by Perrine, has nine parameters:

- $n$ - The number of neurons in the model
- $p$ - The probability of a neuron being connected to another neuron
- $d$ - The expected degree of a neuron
- $k$ - The inverse synaptic strength
- $k_{adj}$ - The synaptic strength modifier
- $r$ - The replication factor
- $T$ - The firing threshold
- $L$ - The number of starting memories
- $F$ - The interference factor
- $H$ - The interference threshold

The Neuroidal model will maintain a list of memories, creating $L$ new memories at the
start of the simulation, each with a random set of $r$ neurons selected from the graph.
The model will also maintain a list of firing states, both for each neuron or node, and
also for each synapse or edge. During simulation, if we set a memory to be firing,
the idea of recalling a memory, the model will set all neurons within that memory to
be firing. Each outgoing synapse from those neurons will then be set to firing, and if
the calculated total synapse weight is over the firing threshold, $T/(k \times k_{adj})$ then the
neuron will be set to firing, with firing states cascading through the graph until no
more neurons are firing. It is this collection of edge weights that are being modified by the JOIN and LINK algorithms to reach the desired memory states.

In his original paper, Valiant puts forth the constraints \( d = p(n - 1) \) for small values of \( n \), and \( d = pn \) for large \( n \). In his recent work, Perrine has shown this cutoff for large \( n \) to be around \( n = 10^5 \) [36]. Perrine chooses to swap these constraints to isolate \( p \), and chose a fixed value of \( d \) to align with the replication factor tables calculated by Valiant in his 2005 paper [47]. In this study we will do the same, as the computational complexity of the Neuroidal model that Perrine faced is also present within our study.

The expected degree of a neuron in the Neuroidal model lines up well with the parameters of the Watts–Strogatz and Barabási–Albert graphs, \( k \) and \( m \), which directly influence the average degree of the graph. We will use the same value for \( k \) and \( m \) as we do for \( d \) in the Neuroidal model.

The remaining parameters of the Neuroidal model define the behavior of the model itself. To ensure that the model is utilizing the variations of Watts–Strogatz and Barabási–Albert graphs that best highlight the effects of small-world properties, we will first investigate the characteristics of the graphs themselves, the then tune to model to perform best upon them.

The Watts–Strogatz graph itself is defined by three parameters, \( n \), \( k \), and \( p \), and the algorithm to create the directed version of the graph is detailed in 1. The Barabási–Albert graph only takes two parameters \( n \), and \( m \), and the algorithm to create the directed version of the graph is detailed in 2. The Barabási–Albert graph starts as an undirected star graph of size \( m \) so that the edge attachment applies evenly at the start of the loop.
In both algorithms, $A$ is the adjacency matrix of the graph, and $n$ is the number of nodes in the graph. A 1 in the adjacency matrix represents an edge between two nodes, with $A_{ij}$ representing the edge between nodes $i$ and $j$, and the reflection across the diagonal, $A_{ji}$, representing the parallel edge. To create the undirected version of the graphs, we can either take the resultant adjacency matrix, and combine it with the transpose of itself, or we can modify the algorithms to assign $A_{ji}$ at the same time as $A_{ij}$.

Within the algorithms, $random$ is a function that returns a random float between 0 and 1 with a uniform distribution, and $choice$ is a function that returns a random integer between 0 and $n$ with a distribution defined by the weights $p$.

**Algorithm 1: Watts-Strogatz Graph Creation**

<table>
<thead>
<tr>
<th>Input</th>
<th>: $n, k, p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>$A$</td>
</tr>
</tbody>
</table>

$A \leftarrow 0_{M_{n,n}}$

for $i \leftarrow 0$ to $n$ do

  for $j \leftarrow 1$ to $k/2$ do

    $j \leftarrow (i + j) \mod n$

    if $random(0, 1) < p$ then

      $j \leftarrow random(0, n)$

      $A_{ij} \leftarrow 1$
Algorithm 2: Barabási-Albert Graph Creation

**Input**: $n, m$

**Output**: $A$

$A \leftarrow 0_{M_{n,n}}$

for $i \leftarrow 0$ to $m$ do

$A_{i+1,0} \leftarrow 1$

$A_{0,i+1} \leftarrow 1$

for $i \leftarrow m + 1$ to $n$ do

$p \leftarrow \frac{\sum A_{0,i+1-n}}{\sum A}$

for $j \leftarrow 0$ to $m$ do

$j \leftarrow \text{choice}(n, p) \quad A_{ij} \leftarrow 1$

While it would be best to showcase how the model performs over a variety of parameters, as Perrine found in his work, the model is susceptible to seemingly innocuous changes in the parameters, the resultant values either falling outside the constraints defined by Valiant or returning inconclusive results [36]. Valiant also found that the JOIN algorithm, which this study will use to reach capacity, doesn’t perform well when the size of a neighborhood is too large, and non-sparse variations of the graphs would likely fail [47].
Chapter 5

RESULTS

In this chapter, we investigate the characteristics of Erdős–Rényi, Watts–Strogatz, and Barabási–Albert graphs as implemented within this work to simulate the Neuroidal model.

First, we will go over the characteristics of the graphs themselves. Of particular interest are the average path length and clustering coefficient, the two properties that define small-world networks.

Figure 5 shows the relationship between edge probability and mean path length for Erdős–Rényi, Watts–Strogatz, and Barabási–Albert graphs on graphs with a node count 1000. For Watts–Strogatz graphs, $p$ was fixed at 0.8, and $k$ was calculated as $k = p(n - 1)$. This was done so that the graph obtains a low average path length from the amount of rewiring while also retaining enough of the ring lattice to also have a high clustering coefficient. For Barabási–Albert graphs, $m$ was calculated as $m = p(n - 2 \ln(n)) + \ln(n)$. It was done this way to adhere to the constraint of $\ln(n) \ll m \ll n$. Interestingly, the average path length is the same for Barabási–Albert graphs both undirected and directed, while Watts–Strogatz graphs have a slightly higher average path length when directed as well as failing to connect when directed. This behaviour could likely be fixed by changing the directed graph creation to connect half the nearest neighbors in each direction, rather than all in one direction. Of particular interest is the behaviour of the graphs where the edge probability $0.01 < p < 0.3$, as this is where most of Valiant’s calculated $r$ values fall. Erdős–Rényi and Watts–Strogatz graphs at this point had difficulty connecting to every node, causing
Figure 5.1: Edge Probability ($p$) vs. Mean Path Length ($L$) averaged over 100 simulations on graphs of size 10000
Figure 5.2: Edge Probability ($p$) vs. Global Clustering Coefficient ($C$) averaged over 100 simulations on graphs of size 10000

an infinite average path length, and the average path length of Watts–Strogatz graphs grows exponentially as it approaches 0, as the lattice is less destroyed. The path length of Barabási–Albert graphs remained consistent, making them seem the most desirable due to the stability they could provide. Watts–Strogatz graphs also seem to have a lot of variation between simulations, while Barabási–Albert and Erdős–Rényi graphs are more consistent.

The other property of interest is the global and local clustering coefficients, as shown in Figure 5 and Figure 5. While the results for the clustering coefficient are not as clear as the path length, Barabási–Albert graphs still do the best at maintaining a higher clustering coefficient within sparse graphs. Interestingly, the inflection point
Figure 5.3: Edge Probability ($p$) vs. Mean Local Clustering Coefficient ($C_i$) averaged over 100 simulations on graphs of size 10000
Figure 5.4: Edge Probability ($p$) vs. Mean Degree Distribution ($k$) averaged over 100 simulations on graphs of size 10000

where Erdős–Rényi and Watts–Strogatz graphs take over is around $p = 0.3$. Also of note is the poor clustering coefficient of directed Watts–Strogatz graphs as compared to Erdős–Rényi graphs at the same edge probability. These results point towards Barabási–Albert graphs being the most desirable for testing the effects of small-world properties on the Neuroidal model, as they maintain a consistent average path length and clustering coefficient across edge probabilities. Their ability to maintain these also speaks to a stability that could be beneficial to the neuroidal model in further research.

The next important property to investigate is the degree distribution of the graphs as shown in Figure 5. The degree distribution itself isn’t nearly as compared to the
degree distribution across edge probabilities, as it illustrates the power law distribution that makes Barabási–Albert graphs scale-free. Nevertheless, the degree distribution across edge probabilities closely resembles the global clustering coefficient, with Barabási–Albert graphs maintaining a higher average degree distribution than Erdős–Rényi and Watts–Strogatz graphs. While this would likely change if you were to plot the graph in terms of percentiles, the average degree distribution is still a good indicator of the overall connectivity of the graph. As can be seen in the overall degree distribution for Erdős–Rényi graphs, the average distribution is a poisson distribution at each edge probability with a quadratic slope. Watts–Strogatz graphs are similar, but in both cases of modifying the $p$ parameter and the $k$ parameter, the average degree distribution is a poisson distribution that follows a power law. Barabási–Albert graphs are somewhat different with two peaks in the distribution, one at the average connected hub degree, and a large spike at the minimum. This speaks to Barabási–Albert graphs tendency to have a few highly connected nodes, and many nodes with only a few connections.

Another interesting correlation to point out is the similarity between the results of the mean degree distribution, clustering coefficient and the edge count of the graphs. Barabási–Albert graphs perform better in sparse graphs, but have a higher edge count on average than Erdős–Rényi and Watts–Strogatz graphs.

The last graph characteristic to touch upon, is the local and global efficiency of the graphs. As shown in Figure 5 and Figure 5, Barabási–Albert graphs have a higher local efficiency than Erdős–Rényi and Watts–Strogatz graphs across sparse probabilities, and have a more stable local and global efficiency than Watts–Strogatz or Erdős–Rényi graphs, which struggle to maintain a high efficiency when sparsely connected.
Figure 5.5: Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Erdős–Rényi graphs of size $1k$ for varying edge probabilities $p$. 
Figure 5.6: Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Barabási–Albert graphs of size 1k for varying edge probabilities $p$
Figure 5.7: Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Barabási–Albert graphs of size 1k for varying edge probabilities $p$
Figure 5.8: Degree Probability ($P(k)$) vs. Degree Distribution ($k$) averaged over 100 simulations on Barabási–Albert graphs of size 1k for varying nearest neighbors $k$
Figure 5.9: Edge Probability ($p$) vs. Edge Count ($e$) averaged over 100 simulations on graphs of size 10000
Figure 5.10: Edge Probability ($p$) vs. Mean Local Efficiency ($E_{loc}$) averaged over 100 simulations on graphs of size 10000
Figure 5.11: Edge Probability ($p$) vs. Mean Local Efficiency ($E_{glob}$) averaged over 100 simulations on graphs of size 10000
Table 5.1: Results of the Neuroidal model with parameters \( n = 10k, d = 128, k = 32 \)

<table>
<thead>
<tr>
<th></th>
<th>Erdős–Rényi</th>
<th>Watts–Strogatz</th>
<th>Barabási–Albert</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ComputationTime(s)</strong></td>
<td>296.1660 ± 26.8</td>
<td>52.2535 ± 16.0</td>
<td>560.6400 ± 9.9</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>973.5000 ± 100.7</td>
<td>255.5000 ± 90.5</td>
<td>4.0000 ± 0.0</td>
</tr>
<tr>
<td><strong>EmpiricalMemorySize</strong></td>
<td>25.6023 ± 1.4</td>
<td>0.1382 ± 0.0</td>
<td>2745.8000 ± 15.7</td>
</tr>
<tr>
<td><strong>Interference</strong></td>
<td>0.0104 ± 0.0</td>
<td>0.0125 ± 0.0</td>
<td>0.0120 ± 0.0</td>
</tr>
</tbody>
</table>

Table 5.2: Results of the Neuroidal model with parameters \( n = 10k, d = 2048, k = 64 \)

<table>
<thead>
<tr>
<th></th>
<th>Erdős–Rényi</th>
<th>Watts–Strogatz</th>
<th>Barabási–Albert</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ComputationTime(s)</strong></td>
<td>41.4118 ± 20.0</td>
<td>63.2755 ± 14.3</td>
<td>578.8455 ± 1.3</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>226.3000 ± 109.2</td>
<td>238.8000 ± 62.5</td>
<td>4.0000 ± 0.0</td>
</tr>
<tr>
<td><strong>EmpiricalMemorySize</strong></td>
<td>7.3873 ± 0.5</td>
<td>0.4177 ± 0.0</td>
<td>2357.1500 ± 52.1</td>
</tr>
<tr>
<td><strong>Interference</strong></td>
<td>0.0108 ± 0.0</td>
<td>0.0121 ± 0.0</td>
<td>0.0120 ± 0.0</td>
</tr>
</tbody>
</table>

Now that we have established the characteristics of the graphs, we can move on to the results of the Neuroidal model on these graphs. Due to the model’s tendency to fail when the average degree of the graph is too high as well as the small frame of allowable parameters, we have decided to test the model under two conditions. The first, a sparse graph with parameters \( n = 10k, d = 128, k = 32 \), resulting in a \( p \) value of 0.0128. The second, approaching what we believe to be the upper limit of the model, with parameters \( n = 10k, d = 2048, k = 64 \), resulting in a \( p \) value of 0.2048. A low value of \( n \) was chosen so that the model would stop due to interference, rather than due to joining all possible combinations of memories, giving us insight to how the different graphs handle interference and capacity. For both simulations, the other parameters were \( F = 0.001, H = 100, t = 1, k_{adj} = 1.55, L = 1000 \). Each simulation was run 10 times on a Ryzen 9–7900X, and the results were averaged. Computational times should be taken with discretion, and only compared between the same simulation, not extrapolated as a general rule. The results of these simulations can be seen in Table 5.1 and Table 5.2
Somewhat surprisingly, in both simulations, Barabási–Albert graphs performed the worst, with the higher average degree of the graph causing the model to both work more for each join sequence, but also with a much higher empirical memory size, likely a direct result of the large hubs, causing the interference to be much higher. Watts–Strogatz graphs actually performed interestingly in 5.2, reaching a similar capacity as Erdős–Rényi graphs but with half the deviation but with an empirical memory size of less than one. This means that the framework created many empty memories, or memories where the size of memory $C$ was zero, as no neurons were connected by $A$ and $B$. This speaks to a sparse local neighborhood, and I would expect to see this result within Barabási–Albert graph results as well, but the memory size of hubs is likely overshadowing this effect. Whether empty graphs should be allowed is a question for future research. In 5.1, they also had a fraction of the empirical memory size, but the interference was much higher leading to less capacity, likely a direct result of the higher clustering. In conclusion, Valiant’s conclusions about the JOIN algorithm only working well with small neighborhoods are clearly shown in our results, and Barabási–Albert graphs showed great promise for stability and clustering, that was actually detrimental to the Neuroidal model. More research should be done to determine the effects of the empirical memory size, as if a lower or higher empirical memory size is more desired, the Watts–Strogatz and Barabási–Albert graphs could help in that regard. It may also be desirable to have less variance between the simulations, allowing for smaller test sets to reach a conclusion, in which case Watts–Strogatz might be the best choice, the additional computational time actually being less on a smaller sample size. Otherwise, it Erdős–Rényi graphs are still the fastest to compute, an important consideration when trying to scale the Neuroidal model to real-world sizes.
By proving that Watts–Strogatz and Barabási–Albert graphs can be substituted for Erdős–Rényi graphs in the Neuroidal model, we have shown that the Neuroidal model can be simulated on a variety of graph structures. We expected that Watts–Strogatz graphs would perform similar to Erdős–Rényi graphs and Barabási–Albert graphs would perform better, yet our results showed that Watts–Strogatz graphs resulted in a much lower memory size with a higher interference rate, while Barabási–Albert graphs resulted in a higher memory size with a higher interference rate. This might actually be desirable on very large graphs, where the sparsity of the graph makes it difficult to cause interference.

We originally expected that model interference would be lower in Watts–Strogatz and Barabási–Albert graphs, but the results showed the opposite. We also found that Barabási–Albert graphs performed well on extremely sparse graphs, when we expected all graphs to perform similarly.

As mentioned previously, the Watts–Strogatz and Barabási–Albert graphs had to be modified to be directed, which may limit its applicability in common frameworks. Our research also only examined capacity within two specific parameters sets for the Neuroidal model, and further research could expand the scope of the study to include more parameters and graph structures.

Specific future work could include continuing to improve the performance of the Neuroidal model to make large graph simulations more feasible, as that would lower the interference rate and may offer more insight into the capacity of the model. Fur-
ther research could also help to better understand real-world neural networks and their structure to better determine if higher degree graphs would be more biologically accurate.

The most impactful future research however would probably be to obtain combinations of $k, d$, and $r$ that are within the constraint space defined by Valiant, beyond an approximation, as obtaining parameters that successfully reached capacity was difficult on small values of $n$ with changing graph parameters.

In conclusion, this work shows the ability to apply small-world properties to the Neuroidal model, and that the Neuroidal model can be generalized to other graph structures. Other neural models that similarly rely on graph structures could also benefit from this research, such as the assembly calculus. We also contributed these improvements to the simulation framework created by Perrine, allowing for further research to utilize our findings and expand upon them.


