Graph Degree Sequence Solely Determines the Expected Hopfield Network Pattern Stability

Daniel Berend
derend@cs.bgu.ac.il
Department of Computer Science and Department of Mathematics, Ben-Gurion University of the Negev, Beer-Sheva, 84105 Israel

Shlomi Dolev
dolev@cs.bgu.ac.il
Ariel Hanemann
hanemann@cs.bgu.ac.il
Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, 84105 Israel

We analyze the effect of network topology on the pattern stability of the Hopfield neural network in the case of general graphs. The patterns are randomly selected from a uniform distribution. We start the Hopfield procedure from some pattern \( v \). An error in an entry \( e \) of \( v \) is the situation where, if the procedure is started at \( e \), the value of \( e \) flips. Such an entry is an instability point. Note that we disregard the value at \( e \) by the end of the procedure, as well as what happens if we start the procedure from another pattern \( v' \) or another entry \( e' \) of \( v \). We measure the instability of the system by the expected total number of instability points of all the patterns.

Our main result is that the instability of the system does not depend on the exact topology of the underlying graph, but rather only on its degree sequence. Moreover, for a large number of nodes, the instability can be approximated by

\[
m \sum_{i=1}^{n} (1 - \Phi(\sqrt{\frac{\delta_i m}{n-1}})),
\]

where \( \Phi \) is the standard normal distribution function and \( \delta_1, \ldots, \delta_n \) are the degrees of the nodes.

1 Introduction

The Hopfield model is one of the classic mathematical models of neuron functionality, playing an important role in computational neuroscience. The capacity of the Hopfield network is measured by the number of bit vectors that can be stored and retrieved in the neural network while preserving a certain level of stability, namely, a certain bound on retrieval errors for the stored vectors. Previous investigations of the capacity of neural networks examined the relation between the (neural) network topology and the...
capacity and stability of the network. Analyses of the capacity of the Hopfield network in a complete graph appeared in the seminal works (Amit, Gutfreund, & Sompolinsky, 1985; McEliece, Posner, Rodemich, & Venkatesh, 1987). In addition, Löwe and Vermet (2011, 2014) and Bovier and Gayrard (1992) found the capacity of different types of random graphs. However, very little theoretical work has been done on other, more realistic, topologies. The only work that has considered other topologies is based on computer simulations rather than theoretical calculations of the capacity and stability. In this work, we calculate exactly the Hopfield network instability for a general topology network and prove that it is solely dependent on the graph degree sequence.

The Hopfield network is an autoassociative neural network, introduced by Hopfield (1982). In this letter, we refer to the original Hopfield model, a graph $G$ with $n$ nodes that saves $m$ vectors (patterns) in $\{-1, 1\}^n$ such that $W_{ii} = 0$ and $W_{ij} = W_{ji} = \sum_u u_i u_j$ for $1 \leq i, j \leq n$, $i \neq j$, where $W_{ij}$ is the weight of the edge between node $i$ and node $j$; $u_i$, $u_j$ are the values of the $i$th and $j$th entries in $u$; and $\sum_u$ indicates summation over all saved vectors $u$.

The recall starts with an input vector and ends after reaching a stable state, where no node $v_i$ changes its value according to $v_i = \text{sgn}(\sum_{j \neq i} W_{ij} v_j)$.

As in other mechanisms of data storage, the Hopfield network has some capacity. In this case, the capacity is defined in terms of the number of vectors that can be saved without having errors when recalling any saved vector. Another way of defining the capacity can be error dependent—how many vectors can be saved so that the number of errors will be less than some predetermined threshold. As can be understood from the definition of capacity, the probability of an error depends on the number of vectors saved in the net.

There are several ways of counting errors in the Hopfield network. One way is to start the Hopfield algorithm from a random vector and measure the Hamming distance (Hamming, 1950) of the result from the closest initially stored vector. Another way is to start from an initially saved vector, run the Hopfield algorithm, and measure the Hamming distance of the result from the initial vector. McEliece et al. (1987) showed that, asymptotically, no more than $\frac{n}{4 \log n}$ vectors can be saved so that all of them may be retrieved perfectly for the complete graph. Amit et al. (1985) showed using the replica trick that for the complete graph, the level of error (as defined there, different from our notion of error) increases very fast when the number of saved vectors is greater than approximately $0.138n$. Following that, other work was done to find lower bounds with rigorous proofs. Newman (1988) first obtained a lower bound of $0.056n$. This was improved by Loukianova (1997) to $0.072n$, Talagrand (1998) to $0.08n$, and finally by Feng, Shcherbina, and Tirozzi (2001) to $0.11n$. One of the interesting aspects of the study of the Hopfield network is the effect of graph topology when dealing with a noncomplete graph. Graphs with the same number of
vertices and edges but different topologies may differ substantially in their capacity of the Hopfield model. Simulations performed by Jianquan, Juan, Jinde, and Zhiqiang (2006) and McGraw and Menzinger (2003) estimated the capacities for random, regular, small-world (Watts & Strogatz, 1998) and scale-free (Barabási & Albert, 1999) graph topologies. Löwe and Vermet have calculated the capacity of the Erdős-Rényi random graphs $G(n,p)$ in ranges of $p$ that yield sparse graphs (Löwe & Vermet, 2011) as well as the case of zero errors in various random graph topologies (Löwe & Vermet, 2014).

In this work we focus on the measure of pattern stability. An instability point is an entry $v_i$ in a stored vector $v$, for which the value of the vertex $v_i$ is different from $\text{sgn}\left(\sum_{j \neq i} W_{ij} v_j\right)$ when starting the system from $v$. Denoting by $I$ the number of instability points, we provide a convenient approximation for its expected value $E(I)$ for large $n$ when the number of saved vectors is $m = \Theta(n)$, which includes the range of $m$ where the model functions as an associative memory with a small number of errors as well as the range where the number of errors increases and the model no longer functions as an associative memory (Hertz, Palmer, & Krogh, 1991). Note that $E(I)$ depends only on the degree sequence of $G$.

2 Instability Solely Depends on the Degree Sequence

We calculate the expected number of instability points first in the complete graph case and then for a general graph. Denote by $R_{v,i}$ the event that the $i$th entry $v_i$ of one of the initial vectors $v$ is an instability point. This event occurs if

$$v_i \neq \text{sgn}\left(\sum_{j \neq i} W_{ij} v_j\right) = \text{sgn}\left(\sum_{j \neq i} u_j v_j\right). \quad (2.1)$$

It will be convenient to separate on the right-hand side of equation 2.1 the contribution of $v$ itself from that of all other patterns:

$$\text{sgn}\left(\sum_{j \neq i} u_j v_j\right) = \text{sgn}\left(\sum_{j \neq i} v_j + \sum_{u \neq v} u_j v_j\right) = v_i \cdot \text{sgn}\left(n - 1 + v_j \sum_{u \neq v} u_j v_j\right). \quad (2.2)$$

Thus, $v_i$ is an instability point if the argument of the $\text{sgn}$ function on the right-hand side of equation 2.2 is negative. Namely, the probability that $v_i$
is an instability point is

\[ P(v_i \neq \text{sgn} \left( \sum_{j \neq i} W_{ij} v_j \right)) = P \left( n - 1 + v_i \sum_{j \neq i} \sum_{u \neq v} u_i u_j v_j < 0 \right). \]

For \( u \neq v \) and \( 1 \leq j \leq n, j \neq i \), define random variables \( T_{uj} \) by \( T_{uj} = v_i v_j u_j \). The four variables in the product are independent, and each assumes the values 1 and \(-1\) with probabilities \( \frac{1}{2} \) each. Hence, so does the variable \( T_{uj} \). Note that while some of the four factors in each \( T_{uj} \) appear in other \( T_{uj'} \)s as well, the factor \( u_j \) appears only in \( T_{uj} \). Thus, the \( T_{uj} \)s are independent. Due to symmetry,

\[ P \left( n - 1 + \sum_{j \neq i} \sum_{u \neq v} T_{uj} < 0 \right) = P \left( \sum_{j \neq i} \sum_{u \neq v} T_{uj} > n - 1 \right). \]

Now the variables \( \frac{(T_{uj} + 1)}{2} \) are \( B(1, \frac{1}{2}) \)-distributed, and it will be more convenient to express the probabilities in terms of these variables:

\[ P(R_{v,i}) = P \left( \sum_{j \neq i} \sum_{u \neq v} \frac{T_{uj} + 1}{2} > \frac{(n - 1)(m - 1)}{2} + \frac{n - 1}{2} \right). \]

Put \( T = \sum_{j \neq i} \sum_{u \neq v} \frac{T_{uj} + 1}{2} \). Since the variables \( T_{uj} \) are independent, \( T \) is \( B(\frac{(n - 1)(m - 1)}{2}, \frac{1}{2}) \) - distributed. Since \( n \) is large, we may use the central limit theorem to obtain

\[ P \left( T > \frac{(n - 1)(m - 1)}{2} + \frac{n - 1}{2} \right) = P \left( T - E(T) > \frac{n - 1}{2} \right) \]

\[ = P \left( T' > \frac{n - 1}{2} \sqrt{\frac{(n - 1)(m - 1)}{4}} \right) \]

\[ \approx P \left( Z > \frac{n - 1}{2} \sqrt{\frac{(n - 1)(m - 1)}{4}} \right), \]

where \( T' = \frac{T - E(T)}{\sqrt{V(T)}} \) and \( Z \) is a standard normal. Hence:

\[ P(R_{v,i}) \approx P \left( Z > \sqrt{\frac{n - 1}{m - 1}} \right) = 1 - \Phi \left( \sqrt{\frac{n - 1}{m - 1}} \right), \]
which is equal to the probability of an error in Hertz et al. (1991). Due to the fact that all the entries of all the vectors behave in the same way, the expected number of instability points is

\[ E(I) = \sum_u \sum_{i=1}^n P(R_{u,i}) = mnP(R_{u(1),1}), \]

where \( u^{(1)} \) is (say) the first random vector. For large \( mn \) where \( m = \Theta(n) \),

\[ E(I) \approx mn \left( 1 - \Phi \left( \sqrt{\frac{n - 1}{m - 1}} \right) \right). \]

For a general graph \( G \), the calculations repeat verbatim, except that the relevant \( j \)'s for node \( i \) are only its neighbors in \( G \). That is, denoting by \( N(i) \) the set of neighbors of \( i \) and by \( \delta_i = |N(i)| \) the degree of \( i \), the probability of error when starting from \( v_i \) is

\[ P \left( \delta_i + \sum_{j \in N(i)} \sum_{u \neq v} T_{uj} < 0 \right). \]

As before,

\[ P(R_{v,i}) = P \left( \sum_{j \in N(i)} \sum_{u \neq v} T_{uj} > \delta_i \right) = P \left( T > \frac{(m-1)\delta_i}{2} + \frac{\delta_i}{2} \right), \]

where \( T = \sum_{j \in N(i)} \sum_{u \neq v} \frac{T_{uj} + 1}{2} \sim B((m-1)\delta_i, \frac{1}{2}) \). Thus,

\[ P(R_{v,i}) = \sum_{k=\left\lceil \frac{(m-1)\delta_i}{2} \right\rceil}^{(m-1)\delta_i} P(T = k) = \frac{1}{2^{(m-1)\delta_i}} \sum_{k=\left\lceil \frac{(m-1)\delta_i}{2} \right\rceil}^{(m-1)\delta_i} \binom{(m-1)\delta_i}{k}. \tag{2.3} \]

Assuming that \((m - 1)\delta_i\) is large, we have

\[ P(R_{v,i}) \approx 1 - \Phi \left( \sqrt{\frac{\delta_i}{m - 1}} \right). \]

Thus, the expected number of instability points is

\[ E(I) = \sum_u \sum_{i=1}^n P(R_{u,i}) = m \sum_{i=1}^n \frac{1}{2^{(m-1)\delta_i}} \sum_{k=\left\lceil \frac{(m-1)\delta_i}{2} \right\rceil}^{(m-1)\delta_i} \binom{(m-1)\delta_i}{k}. \]
Therefore, 

\[ E(I) \approx m \sum_{i=1}^{n} \left( 1 - \Phi \left( \sqrt{\frac{\delta_i}{m-1}} \right) \right). \]  

We can conclude that, surprisingly, the expected number of instability points depends solely on the set of graph node degrees and is not a function of other parameters of the graph.

Note that in the case of a scale-free graph topology, the expected number \( E(N_k) \) of nodes with degree \( k \) is determined only by three parameters, and for large \( n \), it is approximately \( n \cdot c \cdot k^{-\gamma} \), where \( n \) is the total number of nodes and \( c \) and \( \gamma \) are two constants determined by the method of constructing the scale-free network (Barabási & Albert, 1999). Thus, in this case we can approximate the expected number of instability points by 

\[ E(I) \approx m \sum_{k=k_0}^{n-1} nck^{-\gamma} \left( 1 - \Phi \left( \sqrt{\frac{k}{m-1}} \right) \right), \]  

where \( k_0 \) is a small constant, which is the minimum degree in the graph, and is also determined by the construction method.

One may ask whether, in fact, the degree sequence determines not only \( E(I) \) but the exact distribution of \( I \). The following example shows that this is not the case. Consider saving three vectors on the 2-regular graphs \( G_1 \) and \( G_2 \), depicted in Figure 1. A simple calculation of the probability function of the random variable \( I \) shows that the two distributions are quite different even though both graphs are 2-regular and, in particular, have the same degree sequence. The expected values of both distributions are easily calculated to be 1.125.

3 Experimental Results

To test the accuracy of equation 2.4, we conducted a few computer simulations and compared them with our theoretical results. Table 1 compares the
Table 1: Average Number of Instability Points: Experimental versus Theoretical Results.

<table>
<thead>
<tr>
<th>Node Degrees</th>
<th>Simulation</th>
<th>Equation 2.4</th>
<th>Simulation</th>
<th>Equation 2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_1 = \delta_2 = \cdots = \delta_{100} = 10$</td>
<td>29.73</td>
<td>28.46</td>
<td>652.41</td>
<td>648.25</td>
</tr>
<tr>
<td>$\delta_1 = \cdots = \delta_{50} = 16, \delta_{51} = \cdots = \delta_{100} = 4$</td>
<td>47.30</td>
<td>45.35</td>
<td>689.19</td>
<td>685.81</td>
</tr>
<tr>
<td>$\delta_1 = \cdots = \delta_{10} = 37, \delta_{11} = \cdots = \delta_{100} = 7$</td>
<td>42.31</td>
<td>41.88</td>
<td>691.70</td>
<td>689.59</td>
</tr>
<tr>
<td>Barabási-Albert model: $m = m_0 = 30, n = 5000$</td>
<td>23.35</td>
<td>24.09</td>
<td>10,414.6</td>
<td>10,412.81</td>
</tr>
</tbody>
</table>

experimental and theoretical results. We used three groups of 5 graphs, all with the same total number of nodes and edges. The groups have distinct degree sequences. Each group contained five randomly selected graphs with 100 nodes. The average degree in each case was 10. For each graph, 100 experiments were made with 5 stored vectors ($m = 5$) and 100 more experiments with 25 stored vectors ($m = 25$). We also compared simulations on scale-free graphs to the result of equation 2.4. We have tested five scale-free graphs with 5 stored vectors ($m = 5$) and 5 scale-free graphs with 25 stored vectors ($m = 25$). For each of the 10 graphs we made 100 experiments. All the graphs were built according to the Barabási-Albert model (1999) with the same Barabási-Albert model’s parameters: $m = m_0 = 30, n = 5000$. Note that we have not compared the experimental results to equation 2.5 since $E(\mathcal{N}_k)$ becomes close to $n \cdot c \cdot k^{-\gamma}$ only for $n$ much larger than 5000.

4 On the Importance of Instability Points

This work deals with initial instability points, that is, instability points in the original saved vectors. One can consider the capacity of the Hopfield model from the point of view of general instability points—points that will change their value if selected by the algorithm in the nonoriginal vectors. Moreover, by considering the route an original vector with initial instability points does until it reaches a stable state, one can learn about the connection between instability points and the final Hamming distance that the originally saved vector reaches. For example, if $m = 0.138n$, there will be, on average, approximately 0.36% instability points per saved vector. However, the average Hamming distance between the saved and final vector is 1.6% (Hertz et al., 1991). This is due to an avalanche effect that causes new instability points to emerge while flipping the value of the current instability points. Without this avalanche, the final Hamming distance between an originally saved vector and a final stable vector would have been equal to the number of initial instability points. Understanding the dynamics of
this avalanche may help to offer more understanding of the capacity of the Hopfield model. As an example to the dynamics of instability points, our simulations show a strong linear correlation between the sum of inputs of a point (in a complete graph, if this sum is less than zero, the point is an instability point) and the change in this sum after flipping the point with the lowest input in the entire vector. If we sort all the points in a saved vector according to their input and flip the one with the lowest input, \( p_0 \), then the input to all other points changes because it is affected by the new value of the point we flipped. This change in each point \( p_i \) input is linearly correlated with the original input of each \( p_i \). This means that if a point is close to being an instability point, it has a higher probability of actually becoming a new instability point after flipping an original instability point on the way to the final stable vector, which fact can cause the avalanche. It is known that topology influences the final Hopfield capacity (Jianquan et al., 2006; McGraw & Menzinger, 2003). However, given the importance of instability points in the process, the fact that their initial number is identical in graphs with the same degree sequences regardless of topology may provide a better understanding of the process.

Acknowledgments

This work was partially supported by the Rita Altura Trust Chair in Computer Sciences, Lynne and William Frankel Center for Computer Sciences, and Israel Science Foundation (grant number 428/11).

References


Received May 7, 2014; accepted July 22, 2014.