The Stability of Hopfield Networks

Samuel Leonard

COSC 420

March 24, 2019

Introduction:

Hopfield Networks are artificial neural networks that act as associative memory systems. The purpose of this document is to determine the stability of Hopfield Networks. Several patterns will be imprinted in this network to see how many patterns can be recalled before losing reliability. Graphs, calculations and a program will be used to show a representation of the Hopfield Network's stability.

Theory:

On average, how many patterns can be imprinted before one is not accurate? Given that there are 50 patterns, taking the mean at 25 patterns and calculating one standard deviation from there, may give a good indication of how many patterns can be successfully recalled before missing one. In this case that number would be around 8 patterns. What are the most patterns the system can handle given fifty patterns, each with one hundred elements? A good guess would be to take half of the pattern amount, so possibly 25 stable patterns. What will the graphs for number of stable patterns and fraction of unstable imprints look like? The graph for stable patterns will look like a gaussian curve because in the beginning it will be more accurate, then it will peak and then decline. The graph for the fraction of unstable imprints will increase until no more stable patterns are produced. Certain methods and tools will be used to test these theories.

Methods and Calculations:

To demonstrate Hopfield Network stability, some calculations needed to be made. These calculations can be quite cumbersome so, a computer program that could store matrices with the calculated data was developed using Python. The program would begin by generating 50 arrays with 100 elements each, and each element would randomly choose either a 1 or -1 as that elements value. Next, each element of each array was given a weight using the formula in **Figure 1** below. This formula takes each element and compares it with the other elements in the same array. It assigns a weight value that is stored in a weight matrix (**W**). This step is repeated for each array (**p**) and each value is averaged and kept in the weight matrix.

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{k=1}^{p} s_i s_j & i \neq j \\ 0 & i = j \end{cases}$$

Figure 1: Weights Formula

Now, to test the each of the (**p**) patterns for stability, **Figure 2** below was needed. It computes the local field for each neuron in the pattern by multiplying the weight associated with that neuron and its current state. Then it assigns a value of 1 or -1 based on whether the result was negative or positive as shown in the last equation of **Figure 2** below.

$$h_i = \sum_{j=1}^{N} w_{ij} s_j$$

$$s'_i = \sigma(h_i)$$

$$\sigma(h_i) = \begin{cases} -1, & h_i < 0 \\ +1, & h_i \geq 0 \end{cases}$$

Figure 2: Stability Formula

After assigning the values for each neuron based on the weight, each element in the arrays new state was compared with the old state. If any element in the array changed states, then the entire pattern (**p**) was deemed unstable. If the pattern was stable, then the Python program kept a counter for the number of stable imprints. Also, this number was used to determine fraction of unstable imprints by dividing the number of stable imprints by the total number of imprints and then subtracted from 1. The number of stable imprints and the fraction unstable imprints was placed in graphs below by starting with one imprint, using the calculations above, and then iteratively adding another imprint, until reaching 50. The program that was developed allowed for this to happen several times, each time added to a 50-element array for each pattern, and then averaged. The array held the number stable imprints over the total number of imprints and was used to compose the graph in **Image 2** below. Another 50-element array was used to average the fraction of unstable imprints and it composed the graph in **Image 1**.

Graphs:

The graph in **Image 1** below shows the fraction of unstable patterns as a function of the number of imprints. Basically, it shows how unstable the network becomes as more imprints are added.

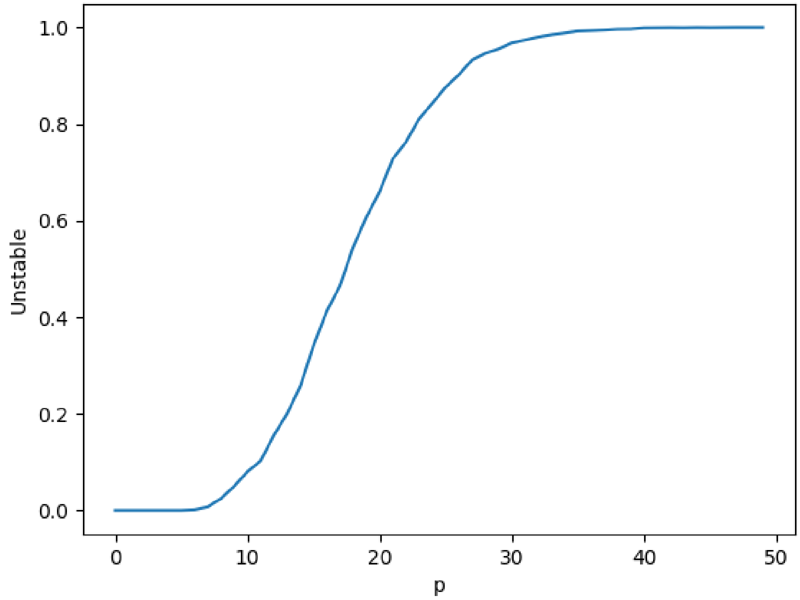

Image 1: Fraction of Unstable Imprints Graph

The graph in **Image 2** below shows the number of stable imprints as a function of the number of imprints added. It shows how many imprints are stable as a new imprint is added consecutively until it reaches 50 patterns.

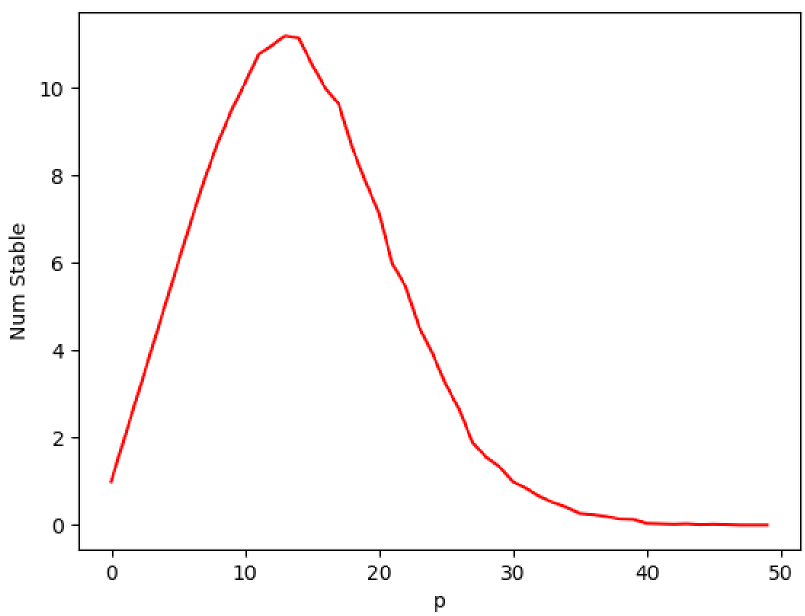

Image 2: Number of Stable Imprints Graph

Results and implications of the graphs and experiment are discussed below.

Discussion:

What do the graphs determine about Hopfield Networks? Both graphs show that there is a finite number of imprints that the network can handle before losing the ability to accurately reproduce the patterns. The graph in **Image 1** demonstrates that as the number of imprints increase, so does the fraction of unstable imprints. The fraction continues to increase until the point of saturation in which 100 percent of the imprints are unstable. This occurs after about 35 to 40 imprints have been registered. The graph in **Image 2** shows similar overall behavior given that as the number of imprints increase, there are less reproducible patterns towards the end. However, **Image 2** does show a maximum number of reproducible patterns for a given number of imprinted patterns. This Hopfield Network can reproduce at most 11 to 12 patterns when imprinted with about 15 total patterns.

So, how many patterns were able to be imprinted before some became unstable. In the theory section, a guess of 8 was made using one standard deviation to the left given a mean of 25, half of the overall patterns. The guess seems to be accurate given the information from both graphs above.

How many patterns can this Hopfield Network reproduce accurately? A guess of 25 was made initially in the theory section of this report because it was half of the overall patterns. This seemed reasonable but was inaccurate. The graph in **Image 2** determined that a maximum of 11 or 12 patterns could be reproduced.

What does this information imply? This implies that Hopfield Networks perform well when the number of patterns stored are fewer. When the number of patterns stored become too many, it becomes difficult for the network to associate with any of the patterns. This occurs because each new pattern's weights are added and averaged. The more this occurs, the more likely one or more neurons in a pattern will experience a change of state, and the entire pattern will be unstable.

Conclusion:

Hopfield Networks provide a way to store and recall patterns. They act as an associative memory system. This experiment used a Python program to simulate a Hopfield Network to determine the network's stability. Patterns were incrementally stored in the system and the calculations in **Figures 1 and 2** were used to determine the weights and stability of each of those patterns. The results of the Python program created graphs that plotted the fraction of unstable imprints and the number of stable imprints as a function of the number of imprints. These graphs showed that the more patterns that were imprinted, the more unstable the network became. Hopfield Networks are a great way to store and recall patterns if the patterns are few. In this experiment the network could only handle about a quarter of the total number of patterns.