An Investigation into the Associative Memory Capacity of a Hopfield Network

Ronald Randolph
CS420: Biologically-Inspired Computation
March 25, 2019

Introduction

The Hopfield network – popularized by John Hopfield in 1982 - is a form of a recurrent artificial neural network. These types of neural networks use their internal state similarly to human memory to process multiple inputs. These networks are extremely affective at handwriting and speech recognition. The Hopfield network provides a form of an associative memory system which can be extremely useful for examining and understanding human memory.

Hopfield networks are commonly used for auto-association and optimization tasks. It was originally created to mimic or simulate the synaptic connection between neurons of the human brain. Due to its ability to retain information, it is also useful for image processing.

Theory and Methodology

A Hopfield network is composed of only one layer of nodes which are connected to every node excluding itself. The connections are represented by different "weights". Furthermore, the state of each node eventually converges – or becomes fixed – after a certain number of updates. This experiment will be exploring the general capacity of a Hopfield network.

To explore this, a total of 50 different randomly-initialized bipolar patterns will be created and used to examine the ability of the associative memory created by the network. To examine the networks memory capacity, a varying fraction of the 50 random patterns will be imprinted and checked for their overall stability. A pattern is defined as stable if none of the bits are changed while updating. If any one bit is unstable, then the entire pattern is unstable.

To imprint the designated number of patterns, a NxN matrix of weights is created using the formula shown in **figure 1** below. Each entry in this weight matrix represents a pair of neurons. In other words, each value is the weight of the connection between two neurons, i and j.

To prevent self-coupling every value where i = j is 0. This designates that no neuron has a weight or connection to itself.

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

Figure 1: Rule for imprinting *p* patterns

After imprinting the designated number of patterns, *p*, the next step is to observe the stability of the *p* imprinted patterns. As mentioned earlier, a pattern is stable if there are no bit changes after the pattern has been updated. If a single neuron state is changed during the update process, then the imprinted pattern is declared unstable. A neuron of a pattern is updated by changing it to the calculated local field for that neuron. The rules for calculating local field is shown below.

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

Figure 2: Rule for calculating local field, h_i

Using the calculated local field acquired from the rule above, the corresponding neuron can now be updated with the designated rules below. If the overall local field is negative, then the neuron's new state is -1. Otherwise, if the local field is nonnegative, then the new state will be updated to +1.

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

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

Figure 3: Rules for updating current neuron, *i*

If any of the N elements of the neural net differ from its corresponding new state value, then that imprinted pattern is not stable. Using these rules and definitions for stable and unstable patterns, the probability of stable imprints for each value, *p,* can be calculated and used for examination of the algorithm's associative memory capacity.

The included C++ program, *p3.cpp*, executes each of the steps above. This process is executed for each value of *p* from 1 to 50. For each value of *p*, the number of stable imprints and the fraction of unstable imprints is calculated and recorded. The main functions of the program operate as follows.

- **Initialize** – randomly initializes 50 bipolar valued vectors. This function also initializes an 100x100 weight matrix to 0.
- **Imprint** – imprints *p* of the 50 random patterns using rule in **figure 1.**
- **Stability** – determines stability of *p* imprinted patterns using the local field calculation rule (**figure 2**) and the state update rules (**figure 3**).
- **PrintData** – outputs calculated data (number of stable patterns and fraction of unstable patterns) to a .cvs document, *data.cvs*.

The program first initializes 50 vectors (patterns) of N=100 randomly initialized neurons. Additionally, a 100x100 weight matrix is initialized to 0. Next, the program imprints *p* patterns and checks each one for its stability. Once *p* has iterated from 1 to 50, the program then prints out the data for each iteration and executes this process with a new set of randomly initialized patterns several (20) times.

Results and Findings

The data produced by the program, p3.cpp, can be found in the included file data.cvs. After a quick observation of the output data, it is clear that imprints only result in an increase of stable patterns for a small range. The program ran a total of 20 separate iterations and averaged the stability values for p = 1 to 50. The max average amount of stable imprints occurs when p = 5. At p = 27, the average number of stable imprints converges to 0 and never rises again.

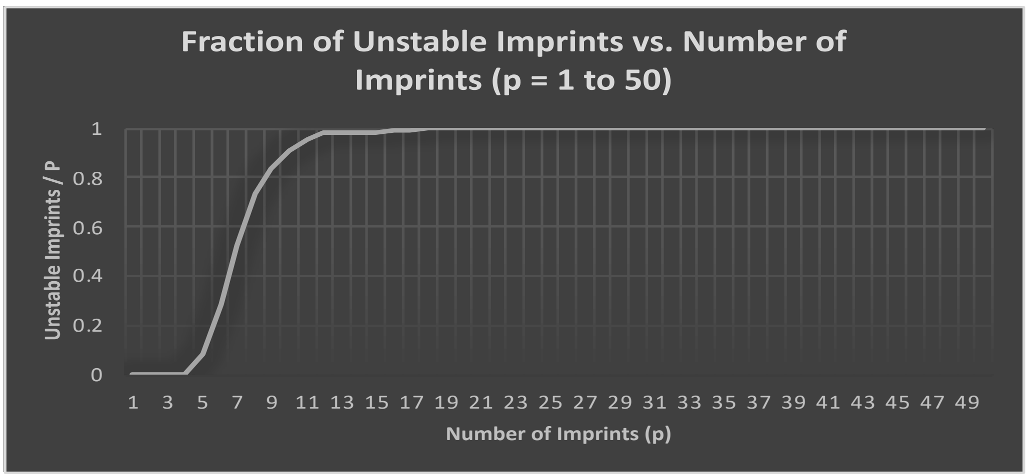

Figure 4: Graph of the Fraction of Unstable Imprints vs. P

The graph above (**figure 4)** illustrates the trend created by the fraction of unstable imprints as a function of the number of imprints (p = 1 to 50). The fraction of unstable imprints begins exponentially increases after p = 5 imprints. After p = 11 imprints, almost all of the imprinted patterns are unstable. This data illustrates the idea that after initially imprinting a few of the patterns, the overall networks begin to become completely unstable.

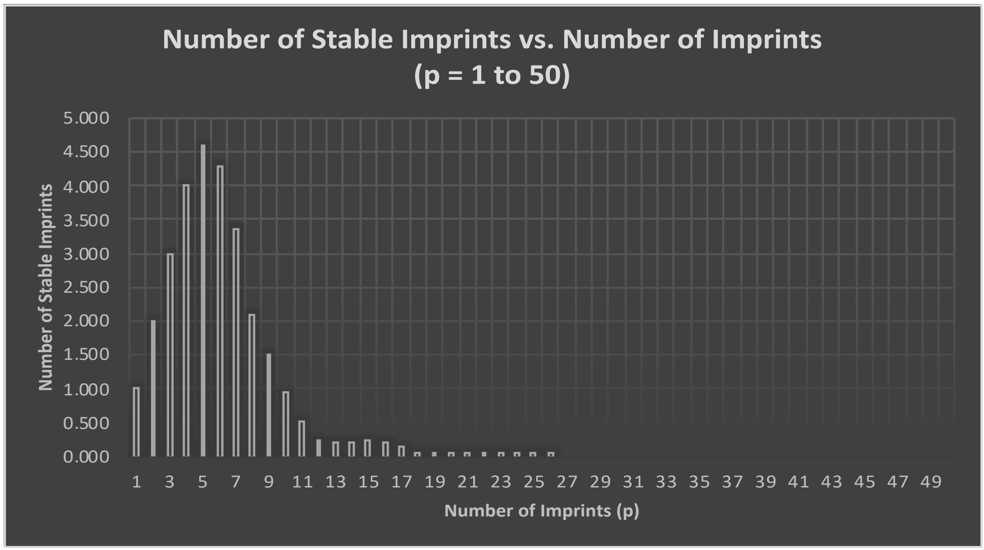

Figure 5: Graph of Number of Stable imprints vs. P

The second graph (**figure 5**) shows the number of stable imprints as a function of the number of overall imprints (p = 1 to 50). As the graph suggests, the highest frequency of stable imprints is found when p ranges from 4 to 6. This aligns with the observations of the previous graph depicting that the number of unstable imprints quickly rises after p = 5.

Conclusion

In conclusion, Hopfield networks clearly have some operating level of associative memory. However, as shown by the produced graphs, this section of memory is only effective in a very small range of imprinted patterns. Hopfield neural networks provide a very surface-level simulation of the actual human brain and memory. While these networks are effective for use in observation and study, there are clearly other neural network algorithms that would be much more efficient for modern day usage with speech recognition, handwriting recognition, image processing, and optimization.