CS420 Project 3:
Hopfield Net

Robby Ferguson

March 11, 2016

Introduction/Methodology

For Project 3, I investigated the associative memory capacity of a Hopfield neural network composed of 100 artificial neurons (*N = 100*). In order to gather more reasonable results, I repeated the simulation 15 times with different Hopfield networks and averaged the data. Within each of these simulations, I took an arbitrary neural network and began individually imprinting 50 random patterns onto it (each of the 100 elements in a pattern was either -1 or 1). As each new pattern was imprinted, I recalculated the weight values (using *Equation 1*) between each pair of neurons in the network. Within one of my Hopfield networks, each neuron was connected to every other neuron in the network (but there was no self-coupling; i.e., the weight between a neuron and itself was 0). As such, a weight matrix was constructed to show the numeric correlation of state values between any two neurons *i* and *j* with either a positive or a negative weight in the following way:

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

Equation 1

Next, I checked the stability of each imprinted pattern. For each of these imprinted patterns, I separately reinitialized the network to match the current pattern. Once the Hopfield network was initialized to an imprinted pattern, I used *Equation 2* to calculate the local field (*h*) of each neuron *i* and then used the sigmoid function (*Equation 3*) to determine what the state of the neuron would become at the next time step.

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

Equation 2

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

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

Equation 3

For each neuron i, the calculated next state was compared to the current state value. If the next state and the current state matched, the next neuron in the network was tested in the same way. If all of the neuron states matched their calculated next state, then the imprinted pattern was considered stable for that number of total imprinted patterns thus far. If a single neuron's current state did not match the next state determined by *Equation 2* and *Equation 3*, then that pattern was instantly considered unstable.

In this way, I kept a counter of how many imprinted patterns were stable for each number of total imprinted patterns (1-50). Using those values, I also calculated the probability of an unstable pattern occurring with respect to the number of patterns that had been imprinted.

To build on those results, I went on to investigate the size of the basins of attraction within the neural networks as a function of the number of imprinted patterns. According to the project description, the procedure for estimating basin sizes is defined as follows:

1. "Generate 50 random patterns.
2. For p = 1, …, 50, imprint the first p test patterns, as for your stability tests.
3. For each of the p imprinted patterns, estimate the size of its basin of attraction, as follows.
4. If the pattern is unstable, then the size of its basin is 0.
5. Otherwise, you will estimate how many bits you can change in the imprinted pattern and still have it converge to the correct stable state in a certain fixed number of steps (n = 10).
6. More specifically, to estimate the size of the basin of imprinted pattern k (k = 1, …, p), generate a random permutation of the numbers 1, …, 100; let L_i be the i-th number in this permutation. This list gives you the order in which you will change the bits of pattern k. You will change at most 50 bits, since that is the maximum size of a basin. Therefore, all you need is L_1 to L_{50}.
7. For the i-th iteration, you will flip bits L_1 , L_2, …, L_i of pattern k, initialize the net to the resulting modified pattern, and then see if the net converges to pattern k within 10 cycles (that is, 10 updates of all the cells).
8. Estimate the size of the basin to be the minimum i for which the modified pattern doesn't converge to pattern k.
9. Repeat steps 5–7 for several different random permutations of 1, …, 100, where L_i is the i-th number in this permutation. Average the results to estimate the basin size for pattern k.
10. Repeat steps 3–8 for each imprinted pattern k. Compute a *basin histogram*, which counts the number of imprinted patterns with each different basin size 0, 1, …, 50.
11. Repeat steps 3–9 for p = 1, …, 50. Plot the basin histogram for each value of p."

As described in step 9, I recalculated the basin size for each imprinted pattern *k* 10 different times and recorded the average size. For each of those 10 iterations, I reshuffled a vector containing indices of bits to flip in the neural network, such that each iteration used a new permutation of the indices.

Data/Graphs

The estimated basin sizes and the associated graphs across all 15 simulations were recorded and kept in the same Microsoft Excel file (*Average_Imprint_Stability.xls*). In that same file, data and graphs can also be found for the average number of stable imprints as a function of the total number of imprints and the fraction of unstable imprints as a function of the total number of imprints. Although larger graphs (and the associated data tables) can be found in *Average_Imprint_Stability.xls*, smaller versions can be seen in *Figure 1*, *Figure 2*, and *Figure 3* below. The graph of basin sizes (*Figure 3*) is only shown for even numbers of imprinted patterns up to 14 imprinted patterns (p = 2, … , p = 14) to avoid cluttering the graph by showing up to p = 50.

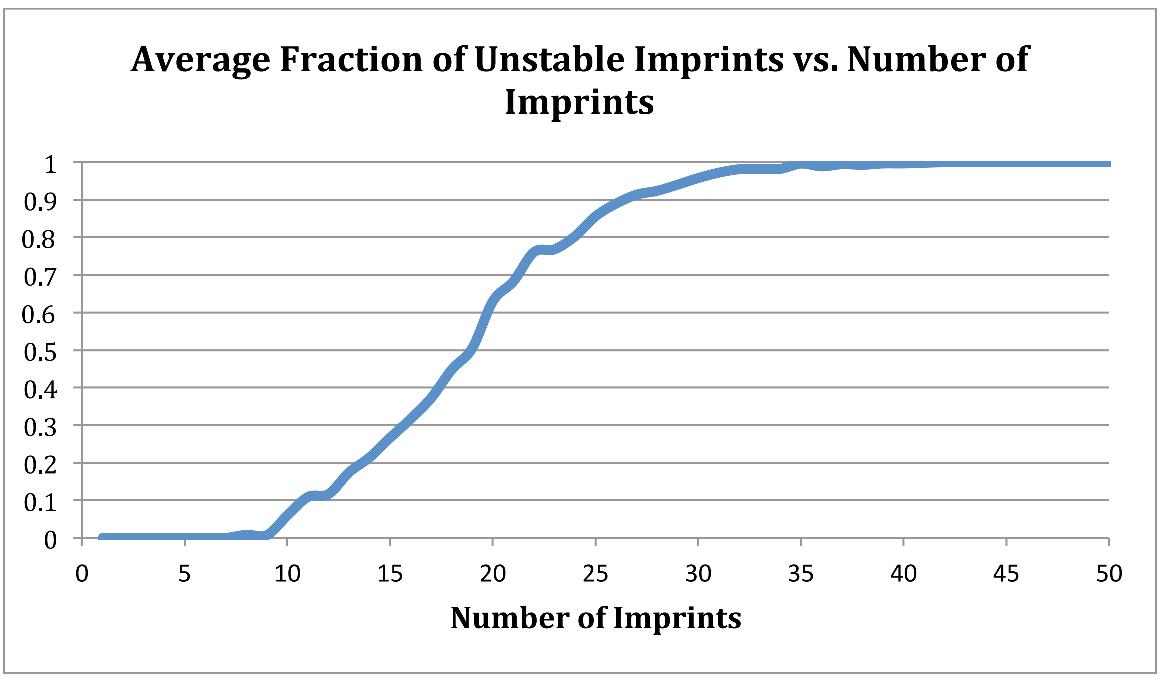

Figure 1

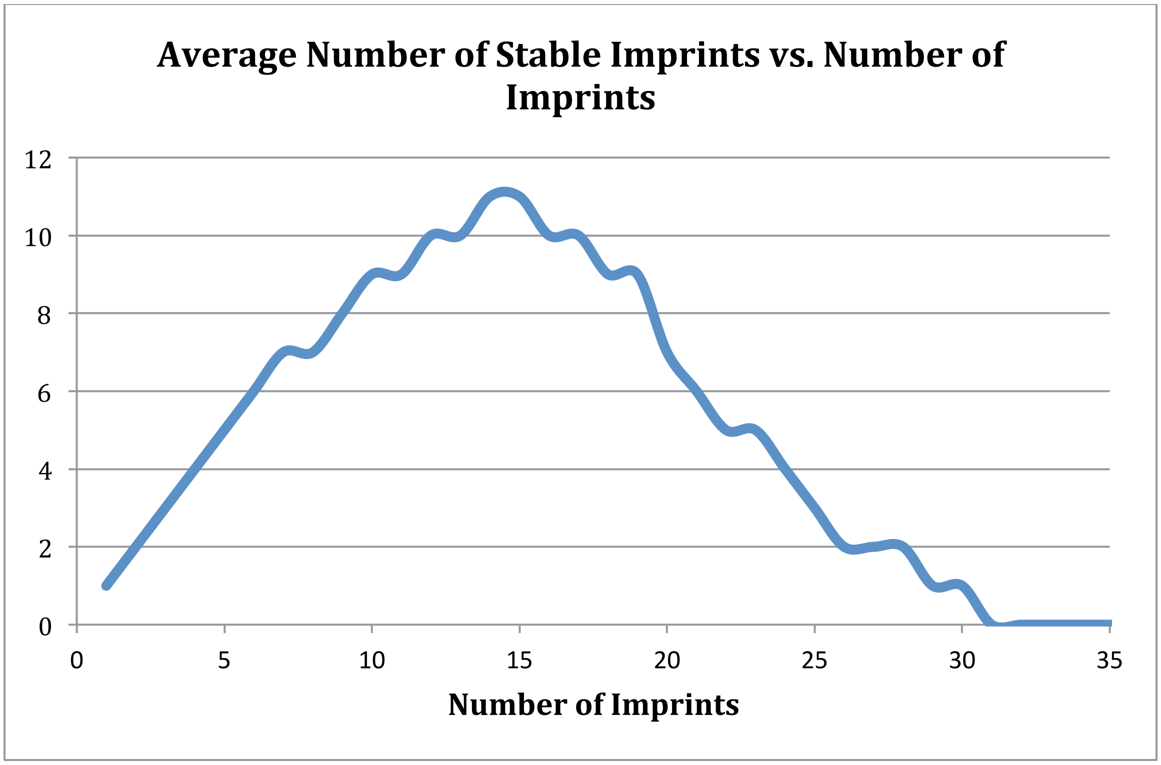

Figure 2

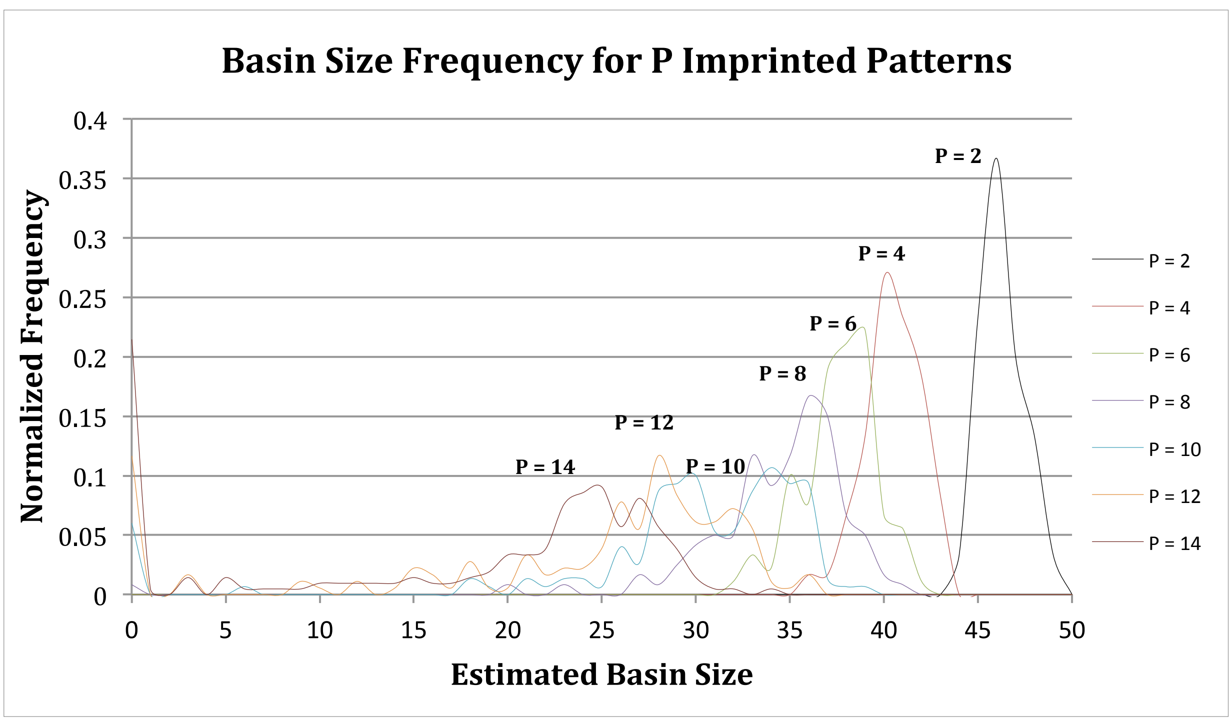

Figure 3

Although the average stability data is included in *Average_Imprint_Stability.xls* and is shown in *Figure 1* and *Figure 2*, data for each of the 15 individual Hopfield simulations (each set of 50 random patterns) can also be found in *Imprint_Stability.xls*, which contains individual data tables (for every simulation) and graphs (for most of the simulations) of the fraction of unstable imprints as a function of the number of imprints and the number of stable imprints as a function of the number of imprints. For example, the graphs for the first and the last simulation (Run 1 and Run 15, respectively) are shown below. Run 1 is associated with *Figure 4* and *Figure 5*, and Run 2 is associated with *Figure 6* and *Figure 7*.

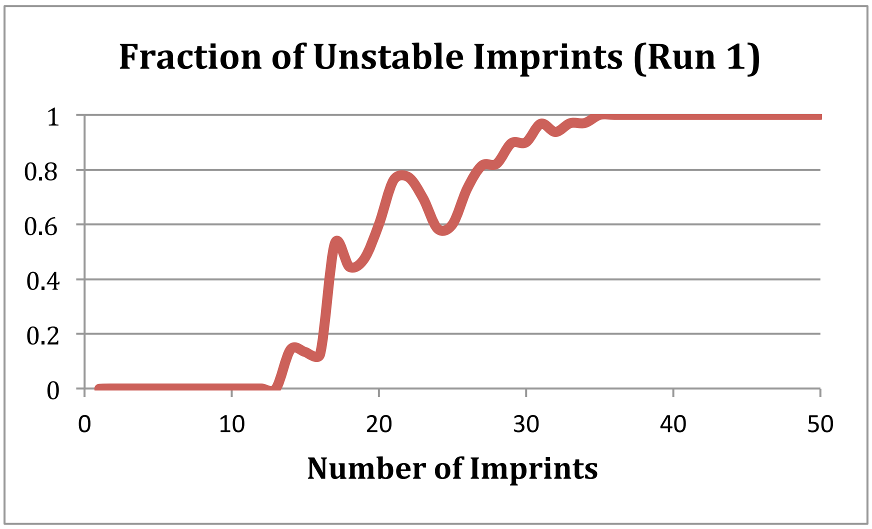

Figure 4

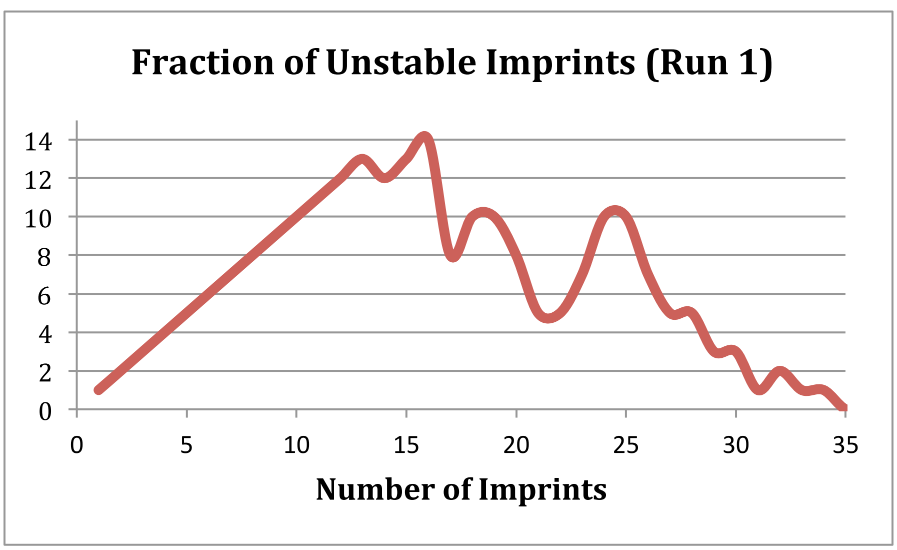

Figure 5

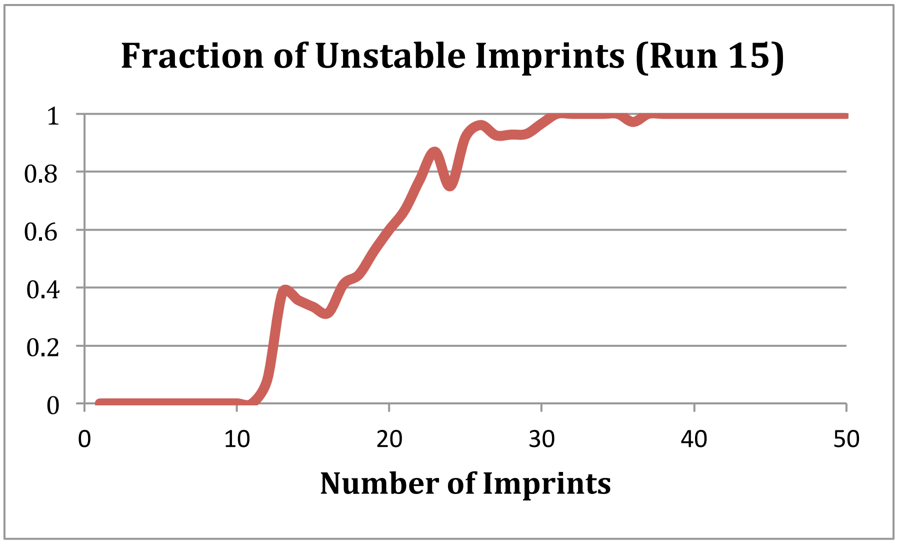

Figure 6

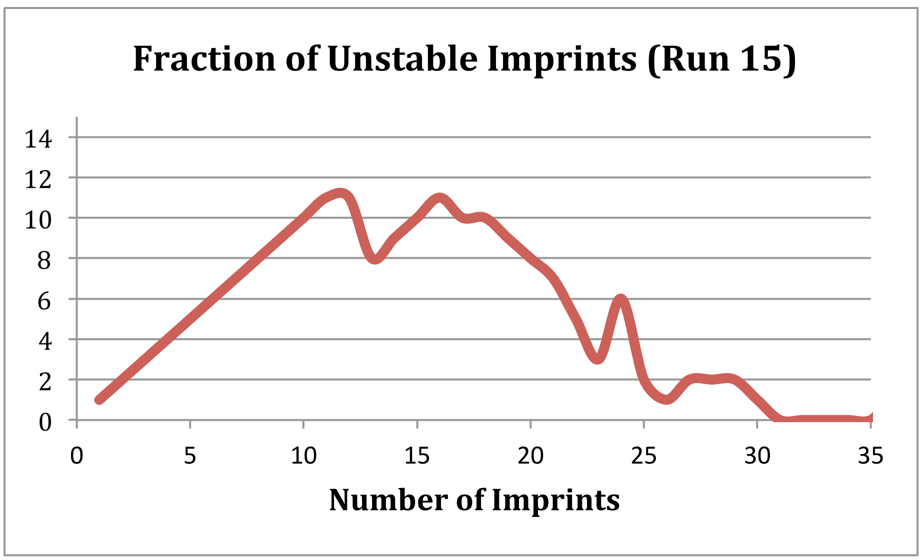

Figure 7

Results/Conclusions

The primary goal of this project was to investigate the associative memory capacity of a Hopfield neural network composed of 100 artificial neurons. To gather a more reasonable estimate of the general behavior of such a network, I ran 15 separate simulations/runs with new sets of 50 randomized pattern vectors (of 100 bits), where each set of patterns was imprinted into a Hopfield network one pattern at a time.

As new patterns were imprinted in each simulation, I individually tested the stability of all the imprinted patterns up to that point and kept a count of how many were stable out of the p imprinted patterns (p = 1, … , p = 50 for each simulation). Those values were then used to also calculate the fraction of unstable patterns at each p value. For most of the individual simulations, graphs were created to show the relationship between the number of imprinted patterns and the stability of those patterns that had been imprinted. For example, the individual results of Run 1 (*Figure 4* and *Figure 5*) and Run 15 (*Figure 6* and *Figure 7*) are shown above. The similarity of the graphs between the two simulations is quite apparent, which reflects the fact that the associative memory capacity of this particular Hopfield setup is relatively consistent for all sets of randomized patterns.

To see a more general representation of the behavior of this type of system (with randomized imprinted patterns) the average results across all of the simulations can be seen in *Figure 1* and *Figure 2* above. *Figure 2* shows that, on average, the associative memory of this 100-neuron Hopfield network setup is relatively reliable up until a maximum of about 11 random patterns have been imprinted onto it. In individual runs, the maximum number of stable imprints tended to be about 15, but the majority of the simulations stayed within the range of 10 to 15 stable imprints before the reliability of the associative memory always started to decline.

Ultimately, after this maximum point of imprinted patterns was reached in each simulation, the reliability of the system's memory became even more sporadic, and it would always decrease overall until none of the imprinted patterns were stable. As can be seen in *Figure 1*, the average fraction of unstable imprints (relative to the total number of imprinted patterns at each point) across all the simulations would remain constant at zero (no unstable patterns) until a critical point of about 9 imprints where the reliability would become increasingly unreliable (but could still generally hold a few more patterns) up to a maximum point of around 11 imprints. These conclusions are generally consistent with the results shown the graphs for Run 1 and Run 15 shown above.

After determining these results for the reliability/capacity of the associative memory of a Hopfield neural network, I also examined the size of the basins of attraction that occurred within each network relative to the number of imprinted patterns at a given point. A basin of attraction is a section of the phase space in which a particular point within the basin will eventually evolve to a specific attractor. In the case of this project, the phase space is the neural network, the points are the neurons, and the attractors are the imprinted patterns (energy minima). As new patterns are imprinted into the network, these basin sizes shift to reflect the fluctuating tendency of a given bit to converge to a particular state.

To analyze the basin sizes, I created a histogram of basin size counts (from a minimum of 0 to a maximum of 50) for each p value (number of imprinted patterns). As each new pattern was imprinted, I reevaluated the basin size of each previously imprinted pattern up to that point and kept a count of the number of occurrences of each possible basin size, which was used to create the histogram (*Figure 3*). This data was collected over all simulations, not just a single run. The data was then normalized with respect to the number of imprinted patterns p and the number of simulations run. In this way, the overall probability of each basin size occurring for each number of imprinted patterns can be seen. For example, the most probable basin size for p = 2 (two imprinted patterns) is about 46. In general, the probability of a particular basin size occurring seems to decrease as the total number of imprinted patterns increases. Intuitively, this seems to make sense because as more patterns are imprinted, the tendency of an individual bit to go toward one state or another is influenced by an increasing number of other attractors.