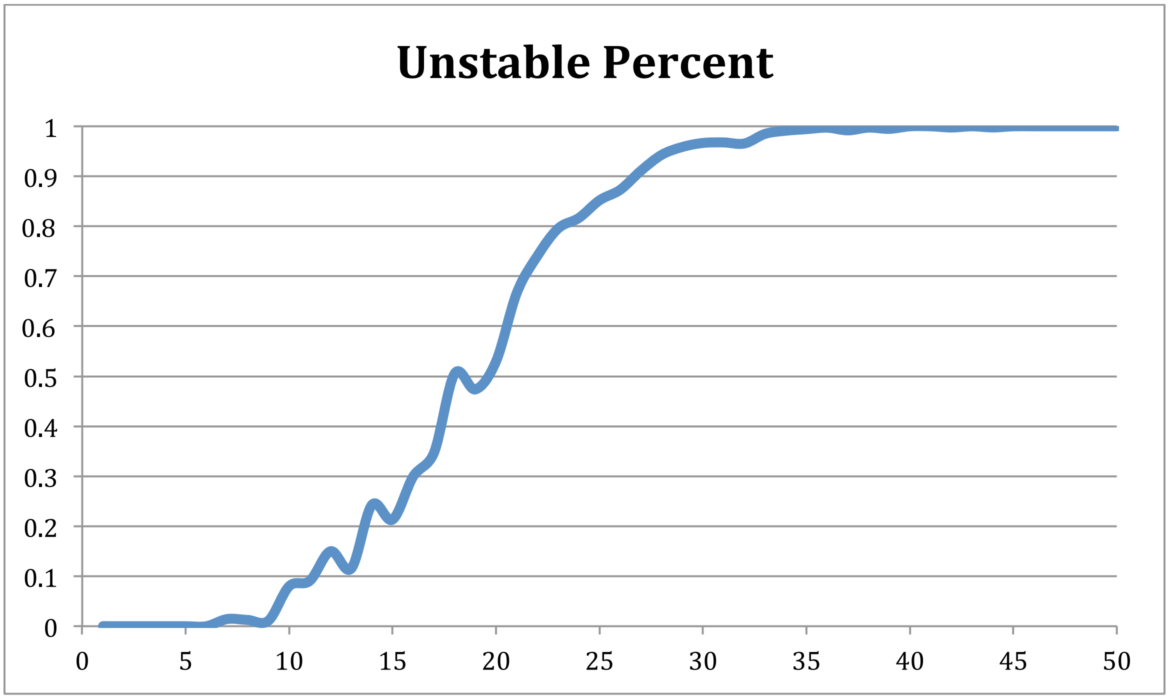
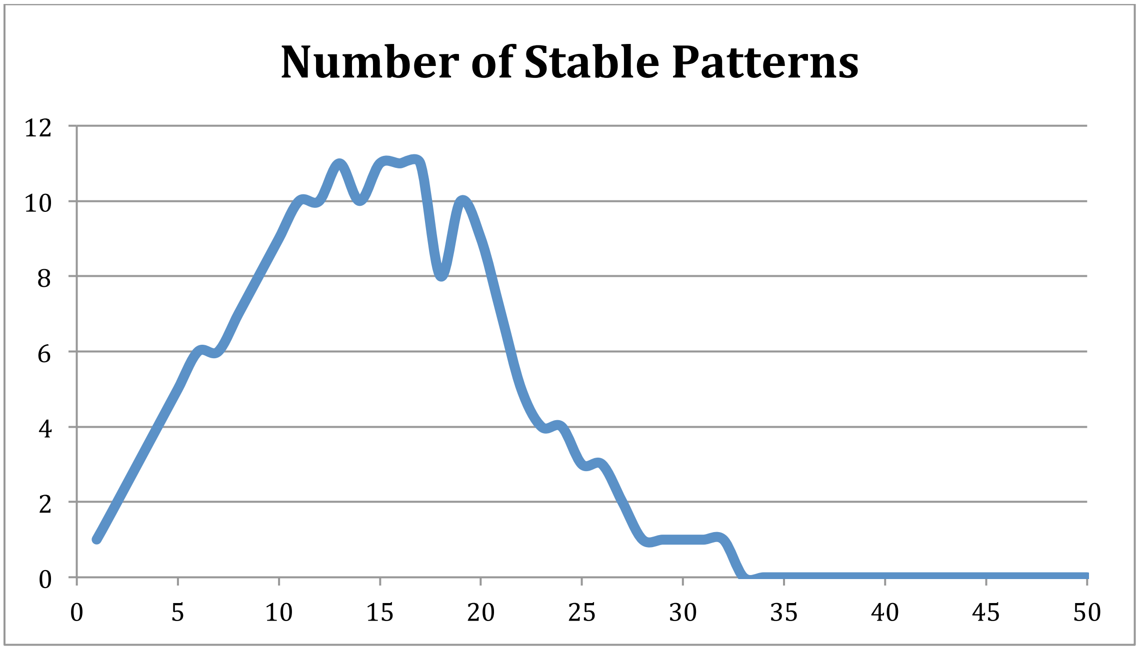
Hopfield Networks

Hopfield networks are a form of artificial neural network where a number of initialized patterns may be recognized and recreated when the system is given a similar pattern. One Hopfield network can hold several patterns, but its associative memory decreases as the number of imprinted patterns arises; to quantify this hypothesis I created a program implementing a Hopfield network with one hundred neurons and tested its stability for any number of patterns ranging from one to fifty. Furthermore, I calculated the basin of attraction for each set of imprinted patterns to further understand how the number of imprinted patterns affects this basin of attraction, thus giving further insight on why a Hopfield network may or may not converge to the correct pattern, given an input pattern with enough noise.

Implementing the Hopfield network began by imprinting patterns, then calculating a weight vector to store how each neuron affected each other; this implementation was created to have no self coupling, so a neuron's weight to itself was zero. As the number of patterns increased, the value of each weight decreased. Next, each of the patterns was tested for stability to ensure the weights could properly create the pattern. To do this, the local field was calculated on each neuron by finding the summation of all the other neurons' states multiplied by their corresponding weights; this value was inserted into a function to determine the final value of that neuron. If the local field was above zero, the new value was one. If the local field was below zero, the new value was negative one. To test the stability of these patterns, the new value was compared against the old value, if they differed even once across all the patterns, then the Hopfield network's weights could not create a stable system. This process was run ten times, and the average values for the unstable percent versus the number of patterns was calculated.



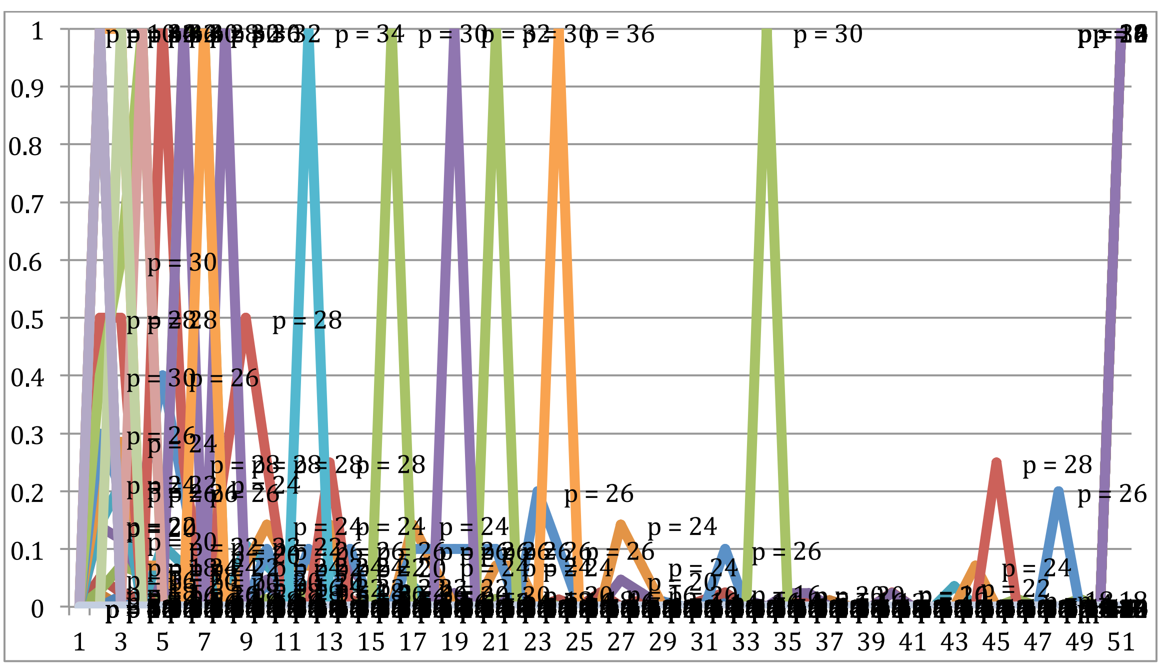
As this chart clearly points out, the percent of unstable patterns increases as the number of patterns increases; this clearly makes sense within the idea of neural networks, as if there are too many patterns the weights will not be able to determine which specific pattern is being analyzed as easily. For fewer than nine imprinted patterns, the Hopfield network appears to do a very good job, the unstable percent maxes out at 1.42% unstable. After this point, the unstable percent increases quickly. At ten imprinted patterns, 8.00% are unstable, by twenty imprinted patterns, 53.00% of the patterns are unstable, and by thirty imprinted patterns nearly all the patterns are unstable with an unstable percentage of 96.67%. In this case, the percent of unstable states increases rapidly, as there are only one hundred total neurons in the system, although if there were more, the system should maintain stability longer. With an understanding of how the number of imprinted patterns affects the percent of unstable patterns, the number of stable patterns per imprinted pattern was tested.



The number of stable patterns sees a peak of eleven, given the total number of imprinted patterns at thirteen, fifteen, sixteen, and seventeen. On the climb before this peak is reached, nearly every imprinted pattern is stable; the slope declining after this peak displays the loss in reliability of the Hopfield network. Despite having three values share the highest number of stable patterns for the Hopfield network, it is clear that the most effective number of imprinted patterns for this max is thirteen, as after the percent of unstable patterns continues to rise.

From this data, it is clear that after roughly ten imprinted patterns, the stability of the Hopfield network begins to decline, given one hundred neurons in the network. Furthermore, in these experiments, after thirteen imprinted patterns there is no addition benefit to imprinting more patterns, as the number of stable patterns no longer increases, while the percent of unstable patterns does increase.

Next, the implementation tested the basin of attraction size for all the stable patterns. It does this by calling a function whenever a stable pattern is found to increment a two dimensional vector storing how many times a certain basin size has been found in a particularly sized pattern. This function created a random array of size one hundred containing integers values ranging from zero to one hundred. It randomly shuffled this array, and used it as an index to swap values of an original pattern. The Hopfield network was then handed a pattern that had been changed one through fifty times and given ten iterations of updating to try to correct the network. The first iteration of the network that was unable to reach the stable pattern indicated the basin size. This process was taken over ten runs of the program, and the collected output was normalized between the values of zero and one.



Even with the chaotic appearance of the histogram above, lots of revealing evidence is apparent. The fact that each row was normalized creates the interesting behavior of showing several peaks reaching one where individual patterns had equal likelihood of having the same basin of attraction size. Imprinting less than nine patterns often lead to the maximum basin size, showing the efficiency of those systems. Basin size begins to start appearing at lower values by the time fifteen patterns are imprinted, showing several different basin of attraction sizes with highest probability happening around a size of two. By twenty-five, the basin of attraction size has reduced to be most likely around one, and by thirty-three it is unlikely for there to be many stable patterns at all. The varying basin of attraction sizes help show how likely a system is to be stable by showing how far a system can reach to obtain stability, and it is clear that more patterns imprinted leads to a smaller basin of attraction.

The number of imprinted patterns affects the efficiency of a Hopfield network dramatically. When increasing past around thirteen imprinted patterns, the number of stable patterns never increases much more and the percent of unstable patterns just continues to rise, given one hundred neurons in the network. Furthermore, the size of the basin of attraction begins to greatly lower around this point, leading to an understanding of this decreased efficiency.