Using Hopfield Nets for Basic Handwriting Recognition

Joe Christianson, Neeraj Joshi, Wilbert Lam.

1 What is the problem?

The goal of this project is to create a machine that can learn to translate handwritten letters (minimum of 26 for all upper case letters, with potential for 52 if we include uppercase) despite any additional noise into the equivalent ASCII character.

2 Our solution

We plan to use a Hopfield net to do the task of pattern recognition and noise reduction. A Hopfield net is modeled after the biological method of pattern recognition. A set of nodes that represent neurons are connected by a fully connected graph. Each edge in the graph represents a connection between neurons and has a weight associated with it. In biology stimulated neurons fire, sending a single out towards other neurons. We model this process by choosing one node to be the neuron receiving the signal and the rest to be the sources. Then calculating the signal is the vector product of the weights and the state of the other neurons. Several avenues then present themselves to calculating the next state of the net. A single matrix vector product could be calculated, this is equivalent to all neurons firing at once. It is clean from an engineering perspective, but not accurate of how neurons function in real life. Instead we could update the nodes iteratively. This could be done purely randomly, or a fixed random order. The fixed random order ensures that no individual node gets updated disproportionately, so we will likely go this route.

3 Potential Difficulties or stages of development.

We will need to be able to create a dataset to test and train the Hopfield net. We can provide the handwritten letters as the sample set, as well as provide noise physically (scribbling out parts of the letter, spill coffee, different handwriting styles) or through the use of computer tools. To ensure the clearest readings, we also plan on normalizing all the images of the letters through high contrast, as well as ensuring all images are of the same size. One assumption we will make as well is that all images are intended to contain one and only one letter, with all other aspects of the image as stray noise.

Another difficulty that may arise is when the Hopfield network has to classify very similar characters (i and j, I and T, etc.). These characters may converge to the same local minimum leading the network to believe that multiple characters could fit the same input sequence. A possible solution to this problem could be to randomize the order of updating the nodes and run various sets of these updates to see if one output character fits the model better than another. Another way around this problem is to hard code certain checks for similar characters and provide certain methods to distinguish the differences between the characters in consideration.

4 Future investigations

Many extensions are available on this project, some which we may or may not have time to investigate. A quick addition could be including letters and/or punctuation, to increase the range of symbols we can process with our model. Also, one of our initial assumptions is that all images are intended to contain only one symbol. By adjusting our model, we could potentially look into images with more than one letter, with the potential for words or entire pages. We could also compare the accuracy of this model to our basic single letter model and investigate the advantages and disadvantages of investigating words to letters.

5 Sources

5.1 Hopfield explanations:

http://web.cs.ucla.edu/ rosen/161/notes/hopfield.html

http://www.comp.leeds.ac.uk/ai23/reading/Hopfield.pdf

5.2 Coding Example:

http://www.codeproject.com/Articles/15949/Hopfield-model-of-neural-network-for-pattern-recog