Project Ideas for Hopfield Neural Networks

A. New Technique for Using HNNs

1. Use labeled feature vectors (x(1),...,x(N), l) where l is the label for the vector. Use these vectors to set up the weights.

2. For any unknown feature vector, use the vector as (x(1),...,x(N), 0), i.e., we don't know the label so we just put zero in (zero is not a label for any of the classes). The input should feed back until it stabilizes to the feature vector with the correct label.

B. Steepest Descent for Training HNNs

1. Write a program for a Hopfield NN that accepts as inputs

N, Q, and a set of Q class patterns $\{\mathbf{y}^{(q)}\}$

Compute the weights W from the class patterns

2. Express an energy function in terms of the inputs

For a set of distorted input patterns $\{\mathbf{x}^{(p)}: p = 1,...,P\}$ with P > Q, attempt to adjust the weights in the weight matrix W by using steepest descent to minimize the basic energy function

$$E = -(1/2)\sum_{(i<j)} w_{ij}\, s_i\, s_i$$

The problem is that the thresholds are not differentiable, so you will need to replace them by logistic functions (like in BPNNs) that are continuously differentiable.

3. This may lead to something more useful, or maybe not. If not, then try other modifications to make the HNN more useful. Perhaps using real values instead of binary patterns will help. Here we can also try to adjust one weight at a time by adding or subtracting a small value $\in > 0$ and running it until it either converges to a class pattern or not.

4. Experimentation and recording the results often leads to useful results.