Hopfield Network for TSP

This describes a simple version of the Hopfield network for optimizing the traveling salesman problem.

Hopfield Network Basics

1. The Hopfield network was originally designed for pattern storage and recognition (stores patterns, then recognizes noisy or incomplete versions of the pattern). This is called an auto-associative network.
2. It was originally a binary network (binary inputs and outputs).
3. The Hopfield network is fully interconnected, that is, all neurons are connected to all other neurons (there are no “layers).
4. The weights are symmetric, so that the weight from neuron 1 to neuron 2 is identical to the weight from neuron 2 to neuron 1.
5. The Hopfield networks works on an energy idea, where the network changes its outputs to find the low energy state. The energy equation is of the form of a Lyapunov equation, which only accepts “moves” which lower the energy and theoretically converges to a low energy state.
6. The network calculates the energy through the product of the outputs (0 or 1) and the interconnecting weights.
7. Hopfield networks for pattern storage find their weights through calculation using the patterns to be stored. There is no iterative training.
8. The Hopfield’s capacity for accurate pattern storage is poor and is usually about 0.15 * number of neurons.

Hopfield Version for TSP

This version is based on using a consensus function, which is to be maximized. This is the same as minimizing the energy function. Each neuron, *x*, is either on (output = 1) or off (output = 0). The consensus, C, is calculated by:

$$C = \sum_i \left(\sum_{j \le i} w_{ij} x_i x_j \right)$$

1. The number of neurons in the network is N^2, where N is the number of cities. For a TSP with N cities, there are N! tours where N!/2 are unique.
2. These are arranged in an N x N matrix where the row is the city and the column is the position in the tour.
3. A valid tour will have one and exactly one neuron "on" in each row and each column.
4. Connecting the "on" cities in order constructs the tour.
5. Weights are set according to the problem. There are three main types of weights:
    a. Each neuron has its own positive bias weight.
    b. Each neuron has a negative (inhibitatory) weight to each of the other neurons in its row and its column.
    c. Each neuron has a negative weight to other possible cities just before it and just after it on the tour. This weight is equal to (or correlated with) Euclidean distance. Since the tour makes a loop, the final column is connected to the first column by distance weights.
6. Outputs of nodes change one by one and only changes that increase consensus are accepted. (A probabilistic version, the "Boltzmann Machine", accepts changes that lower consensus probabilistically using a simulated annealing function).