Machine Learning: Back Propagation

Codie Roelf (RLFCOD001), 07/08/2014

***Introduction***

The backpropagation algorithm is a classic supervised learning algorithm, based on gradient descent. To implement this I have chosen to use a one dimensional pointer to represent the input layer, another to represent the hidden layer, and another to represent the output layer. For the input->hidden layer I have used a 2 dimensional pointer, and another for the hidden->output layer. I used pointers because they generally work faster than vectors do.

I used a mseCutOff of 0.01 with a learning rate of 0.5. The nodes seemed to learn enough in that time to get good results.

***Feed Forward***

**The feed forward method receives the input and saves the memory for it. It then computes the output at hidden layer nodes by making each hidden layer node equal to sigmoid(sum(weight at inputHiddenLayer \* input value)). The sigmoid function I used is 1/(1+ (e^sum).**

**I used the same method to compute the output at the output layer.**

***Propagate Error Backward***

**I first computed the error at the output layer by taking the sigmoid value and multiplying it by the difference between the expected and computed outputs.**

**I then computed the error at the hidden layer by taking the sigmoid value and multiplying it by the sum of the hidden layer node value multiplied by the difference between outputs at the output layer.**

**I adjusted the weights in the hiddenOutputLayer to the learning rate multiplied by the output layer error and the hidden layer value. I adjusted the weights from the making it equal the learning rate multiplied by the error at the hidden layer and the output at the input layer.**

***Classify***

**For each training set size, this method feeds forward, propagates the error backwards. It then adds the mean squared error to the total mean squared errors.**

***Results and Discussion***

**Below is a graph of the average mines collected with series (environment) one, two and three. Environment one with 30 minesweepers. 40 mines and 10 supermines collects the most mines, not surprisingly. The average mines collected over these 21 iterations was 13.095.**

**Environment two with 25 mines and 25 supermines had an average of 8.961. Environment 3 with 10 mines and 40 supermines had the least amount of mines collected with an average of 2.77**

**Below is a graph of the amount of agents destroyed on each iteration. Environment one had an average of 2. Environment two had an average of 9.71 deaths, and environment three had an average of 10 deaths.**