ANN method

Error Backpropagation

E = data [x1,x2….xi] C = Expected Output

Number of hidden nodes: n/2 to 2n (requires experimentation)

Algorithm:

* Set weights and biases to small value (between -2/n and 2/n)
  + Weight for bias starts at 1
* Calculate weighted sum for each node
  + Sum (Input x weight) +(weight x bias)
  + Repeat until output node
  + Calculate activation function
* Work backwards and calculate delta for each node
  + F’(S) = output x (1 – output)
  + output Delta = (C – output) x f’(S)
  + other delta = weight to output x delta of output x f’(S)
* calculate new weight/bias of node i,j
  + weight(I,j) + p x delta j x output j
    - output is always 1 for bias
* Continue to next E
* Find error (difference in expected and actual output)
* Repeat until a condition is met
  + Weight changes become small
  + Changes in error become small
  + X number of epochs (full data passes)
  + Error of an independent validation set increases

Add in improvements to reduce error further

* Momentum
  + New weight = weight + (p x delta x output) +(Value a x change in weight)
* Bold driver
  + If error increases undo change and decrease learning rate (change in weight)
    - Usually decrease by 50%
  + If error decreases undo change and increase learning rate
    - Usually increase by 10%
  + Do this every 1000 epochs or so