A4M33BIA
MLP Implementation

Jiří Kubalík
kubalik@ciirc.cvut.cz

Jan Drchal
drchajan@fel.cvut.cz

Back-Propagation Algorithm

random weight initialization
repeat
  repeat *// epoch*
    choose an instance from the training set
    apply it to the network
    evaluate network outputs
    compare outputs to desired values
    modify the weights
  until all instances selected from the training set
until global error < criterion

FeedforwardANN.java methods

sigmoid(*p*) // calculates logistic sigmoid of the potential *p*

setActivity(*l*, *n*, *v*) // sets input of neuron *n* in layer *l* to value *v*

getActivity(*l* - 1, *j* + 1)

getNumLayers() // the number of layers

getNumNeurons(*l*) // the number of neurons in given layer (including bias neuron)

getNumOutputs() // the number of output neurons (there is no bias neuron in the output layer)

getNumIncomingConnections(*l*, *n*) // the number of connections incoming to neuron *n* in layer *l* (which is same as the number of neurons in layer *l*-1).

getWeight(*l*, *toNeuron*, *fromNeuron*) // returns a weight of connection between *fromNeuron* in layer *l*-1 and *toNeuron* in layer *l*

FeedforwardANN.java methods

propagate(pattern]) // Loads input pattern and propagates it through the whole
// network. All neurons recalculate their activities.

applyDeltaWeights(double [][][] *dw*) // The elements of *dw* array are added to the
// corresponding elements of the weights array.
// Both arrays must have the same shape.

sigmoidDerivative(*y*) // calculates sigmoid derivative of neurons activity *y*

random weight initialization
repeat
  repeat *// epoch*
    choose an instance from the training set
    apply it to the network
    evaluate network outputs
    compare outputs to desired values
    modify the weights
  until all instances selected from the training set
until global error < criterion

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}$$

$$\Delta w_{jk} = \eta \delta_k y_j$$

$$\delta_k = \gamma y_k (1 - y_k)(d_k - y_k)$$

$$\delta_k = \gamma y_k (1 - y_k) \left(\sum_l \delta_l w_{kl} \right)$$