BACK-PROP-LEARNING

AIMA3e

function BACK-PROP-LEARNING(*examples*, *network*) **returns** a neural network
  inputs *examples*, a set of examples, each with input vector **x** and output vector **y**
    network, a multilayer network with L layers, weights $w_{i,j}$, activation function g
  local variables: Δ, a vector of errors, indexed by network node

  repeat
    for each weight $w_{i,j}$ in *network* **do**
      $w_{i,j} \leftarrow$ a small random number
    for each example (**x**, **y**) **in** *examples* **do**
      /* *Propagate the inputs forward to compute the outputs* */
      for each node i in the input layer **do**
        $a_i \leftarrow x_i$
      for $l = 2$ **to** L **do**
        for each node j in layer l **do**
          $in_j \leftarrow \Sigma_i w_{i,j} a_i$
          $a_j \leftarrow g(in_j)$
      /* *Propagate deltas backward from output layer to input layer* */
      for each node j in the output layer **do**
        $\Delta[j] \leftarrow g'(in_j) \times (y_i - a_j)$
      for $l = L - 1$ **to** 1 **do**
        for each node i in layer l **do**
          $\Delta[i] \leftarrow g'(in_i) \Sigma_j w_{i,j} \Delta[j]$
      /* *Update every weight in network using deltas* */
      for each weight $w_{i,j}$ in *network* **do**
        $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
  until some stopping criterion is satisfied
  return *network*

The back-propagation algorithm for learning in multilayer networks.