BACK PROPAGATION ALGORITHM ON THE CONNECTION MACHINE

Xiru Zhang

Michael Mckenna
• Back propagation is the most widely used "neural network" learning algorithm.

• The new CM implementation:
  
  ▪ Use NEWS as main communication operations

  ▪ > 40 million weight updates per second on a 64k CM

  ▪ No special requirement (size, topology, etc.) is needed in order for a network to run at the above speed.
A Layered Network

Output Nodes

Hidden Nodes

Input Nodes

\[ O_k = f(\sum_{j} w_{kj} o_j) \]

\[ O_i = f(\sum_{j} w_{ji} o_j) \]
THE ALGORITHM

1. Compute Output (forward pass):

\[ O_j = f(\sum_i W_{ji} \cdot O_i) \]

where:

\[ f(x) = \frac{1}{1 + e^{-x}}; \]

\( O_j \) – the output of node J;
\( W_{ji} \) – the weight from node I to node J.
2. Compute Error (backward pass):

- For output layer:
  \[ \delta_k = O_k \cdot (1 - O_k) \cdot (T_k - O_k) \]

- For hidden layer(s):
  \[ \delta_j = O_j \cdot (1 - O_j) \cdot \sum_k W_{kj} \cdot O_j \]

where:

- \( T_k \) – the ideal output for output node K;
- \( \delta_k \) – the error at output node K;
- \( \delta_j \) – the error at hidden node J;
- \( W_{kj} \) – the weight from node J to node K.
3. Change Weights:

- For a particular training pattern $P$:

$$
\Delta W^p_{ji} = \eta \cdot \delta^p_j \cdot O^p_i
$$

- Change weights after one cycle through all training patterns:

$$
W_{ji} = W_{ji} + \sum_p \Delta W^p_{ji}
$$

where:

$\eta$ – learning rate;

$\Delta W^p_{ji}$ – weight change caused by a training pattern.
Example: Character Mapping

- Output nodes
- Hidden nodes
- Input nodes
The Old Implementations

\{ Spread, Multiply, Scan \}^+
A Simple Network

Output Nodes

Hidden Nodes

Input Nodes
The New Implementation

- The simplest case

{ Multiply, News, Add }+
The New Implementation

- Replicated networks
- Shared weights