Back-propagation

Example

We consider the below given network with initial weights assigned. We will perform a forward pass, perform a backward pass and then perform a forward pass again to see how the error reduces by using back propagation.

\[ \text{Input} \]

\[ \text{Output} \]

---

Assumptions: Learning rate, \( \eta = 1.0 \); target output = 0.5

- \( W_{ij} \) = weight of the link from node \( i \) to node \( j \)
- \( x_{ji} \) = input of node \( i \) to node \( j \)

We consider output layer as layer \( k \), hidden as layer \( h \) and input as layer \( i \).

1. **Forward propagation**
2. For the output,
   \[
   o(x_1, x_2, \ldots, x_n) = \sigma(Wx) \\
   = 1 / (1 + e^{-Wx})
   \]

- Input to neuron \( C \) = \((0.35 \times 0.1) + (0.9 \times 0.8) = 0.755\)
  Output from neuron \( C \) = \(1 / 1 + e^{-(0.755)} = 0.68\)

- Input to neuron \( D \) = \((0.9 \times 0.6) + (0.35 \times 0.4) = 0.68\)
  Output from neuron \( D \) = \(1 / 1 + e^{-(0.68)} = 0.6837\)
→ Input to neuron E = (0.3 x 0.68) + (0.9 x 0.6637) = 0.80133
Output from neuron E = \frac{1}{1 + e^{-0.80133}} = 0.69.

Error \( \epsilon = \) target - output from neuron E
= 0.5 - 0.69
\( \text{Error} = -0.19 \) \( \quad \) \( \text{1} \)

2. Backward propagation of errors

→ For each output node \( k \), compute the errors:
\( \delta_k = \sigma_k (1 - \sigma_k) \sum_k W_{kh} \delta_h \) \( \quad \) \( \text{2} \)
where \( \sigma_k \) = target output
\( \delta_k \) = final output from neural network

→ For each hidden node \( h \), calculate the errors:
\( \delta_h = \sigma_h (1 - \sigma_h) \sum_k W_{kh} \delta_k \) \( \quad \) \( \text{3} \)
where, \( \sigma_h \) = output at hidden node \( h \)
\( W_{kh} \) = weight of from node \( h \) to \( k \)
\( \delta_k \) = error on node \( k \)

→ For each network weight update,
\( W_{ji} = W_{ji} + \Delta W_{ji} \) \( \quad \) \( \text{4} \)
where \( \Delta W_{ji} = \eta \delta_j X_{ji} \)
\( \eta \) = learning rate,
\( \delta_j \) = error at node \( j \),
\( X_{ji} \) = input from node \( i \) to \( j \)
According to eq. (2),
\[ \Delta e = O_e (1 - O_e) (t_e - O_e) \]
\[ = 0.69 (1 - 0.69) (0.5 - 0.69) \]
\[ = -0.0406 \]

→ Weight updates for output layer,
\[ W_e^+ = W_e + (\eta \times \Delta e \times X_e) \]
\[ = 0.3 + (1 \times (-0.0406) \times 0.68) \]
\[ = 0.272392 \]

\[ W_{de}^+ = W_{de} + (\eta \times \Delta e \times X_{de}) \]
\[ = 0.91 + (1 \times (-0.0406) \times 0.6637) \]
\[ = 0.87305 \]

→ Errors for hidden layers, (According to eq. (3))
\[ \Delta c = O_c (1 - O_c) (W_{ce} \times \Delta e) \]
\[ = 0.68 (1 - 0.68) (0.272392 \times (-0.0406)) \]
\[ = -2.406 \times 10^{-3} \]

\[ \Delta d = O_d (1 - O_d) (W_{de} \times \Delta e) \]
\[ = 0.6637 (1 - 0.6637) (0.87305 \times (-0.0406)) \]
\[ = -7.916 \times 10^{-3} \]

→ Weight updates for hidden layers,
\[ W_{ac}^+ = W_{ac} + (\eta \times \Delta c \times X_a) \]
\[ = 0.1 + (1 \times (-2.406 \times 10^{-3}) \times 0.35) \]
\[ = 0.09916 \]

\[ W_{ad}^+ = W_{ad} + (\eta \times \Delta d \times X_a) \]
\[ = 0.4 + (1 \times (-7.916 \times 10^{-3}) \times 0.35) \]
\[ = 0.3972 \]
\[ W_{bc}^+ = W_{bc} + (\eta * \delta_c * X_{cb}) = 0.8 + (1 * (-2.406 \times 10^{-3}) * 0.9) = 0.7978 \]

\[ W_{bd}^+ = W_{bd} + (\eta * \delta_d * X_{db}) = 0.6 + (\eta * (-7.916 \times 10^{-3}) * 0.9) = 0.5928. \]

③ Forward pass

- D. Input to neuron C = (0.35 \times 0.9916) + (0.9 \times 0.7978) = 0.752726
  Output from neuron C = \frac{1}{1 + e^{-(0.752726)}} = 0.679477

- D. Input to neuron D = (0.35 \times 0.3972) + (0.9 \times 0.5928) = 0.67254
  Output from neuron D = \frac{1}{1 + e^{-(0.67254)}} = 0.662071

- D. Input to neuron E = (0.67977 \times 0.232392) + (0.662071 \times 0.87305)
  = 0.763184
  Output from neuron E = \frac{1}{1 + e^{-(0.763184)}} = 0.682

Error = target - output from neuron E = 0.5 - 0.682
\[ \text{Error} = -0.182 \] ⑤

From ① & ⑤, we can say that error has reduced by using the backpropagation algorithm.