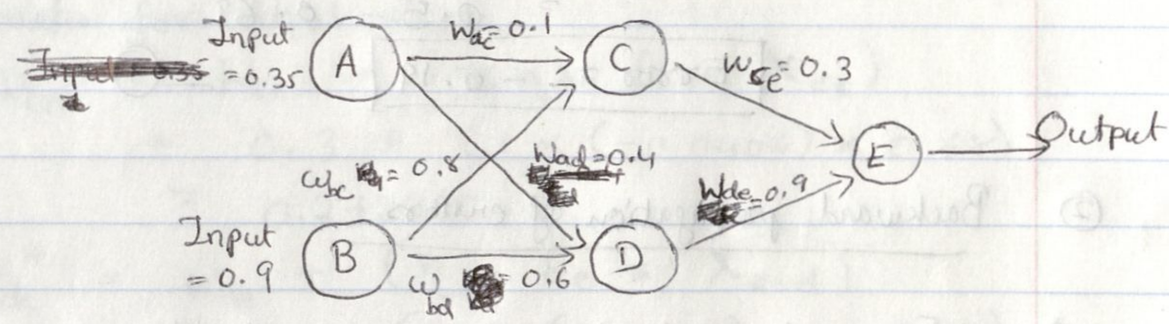
Back-propagation

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.



→ Assumptions : Learning rate, $\eta = 1.0$; target output $= 0.5$

→ W_{ji} = weight of 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 & input & layer i

(1). Forward propagation.

→ 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 = $1/1+e^{-(0.68)} = 0.6637$

→ Input to neuron E $= (0.3 \times 0.68) + (0.9 \times 0.6637) = 0.80133$

Output from neuron E $= 1/1+e^{-(0.80133)} = \underline{\underline{0.69}}$.

→ Error $=$ target $-$ output from neuron E

$= 0.5 - 0.69$

$$\boxed{\text{Error} = -0.19} \quad \text{— ①}$$

② Backward propagation of errors

→ For each output node k, compute the error:

$$\delta_k = O_k (1 - O_k)(t_k - O_k) \quad \text{— ②}.$$

where t_k = target output

O_k = final output from neural network

→ For each hidden node h, calculate the error:

$$\delta_h = O_h (1 - O_h) \Sigma_k W_{kh} \delta_k \quad \text{— ③}.$$

where, O_h = output at hidden node h

W_{kh} = weight from node h to k

δ_k = error on output node k.

→ For each network weight update,

$$W_{ji} = W_{ji} + \Delta W_{ji} \quad \text{— ④}$$

where $\Delta W_{ji} = \eta \delta_j X_{ji}$

η = learning rate, δ_j = error at node j,

X_{ji} = input from node i to j

According to eq. ②,

→ Output error, $\delta_e = O_e (1 - O_e)(t_e - O_e)$

$$= 0.69 (1 - 0.69)(0.5 - 0.69)$$

$$= -0.0406$$

→ Weight updates for output layers,

$$W_{ce}^{+} = W_{ce} + (\eta * \delta_e * X_{ec})$$

$$= 0.3 + (1 \times (-0.0406) \times 0.68)$$

$$= 0.272392$$

$$W_{de}^{+} = W_{de} + (\eta * \delta_e * X_{ed})$$

$$= 0.9 + (1 * (-0.0406) \times 0.6637)$$

$$= 0.87305$$

→ Error 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,

$$W_{ac}^{+} = W_{ac} + (\eta * \delta_c * X_{ca})$$

$$= 0.1 + (1 * (-2.406 \times 10^{-3}) * 0.35)$$

$$= 0.09916$$

$$W_{ad}^{+} = W_{ad} + (\eta * \delta_d * X_{da})$$

$$= 0.4 + (1 * (-7.916 \times 10^{-3}) * 0.35) = 0.3972$$

$$W_{bc}^{+} = W_{bc} + (\eta * \delta_c * X_{cb})$$
$$= 0.8 + (1 * (-2.406 \times 10^{-3}) * 0.9)$$
$$= 0.7978$$
$$W_{bd}^{+} = W_{bd} + (\eta * \delta_d * X_{db})$$
$$= 0.6 + (\eta * (-7.916 \times 10^{-3}) * 0.9)$$
$$= 0.5928.$$

③. <u>Forward pass</u>.

→ Input to neuron C = $(0.35 \times 0.09916) + (0.9 \times 0.7978) = 0.752726$
Output from neuron C = $1/1+e^{-(0.752726)} = 0.67977$

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

→. Input to neuron E = $(0.67977 \times 0.272392) + (0.662071 \times 0.87305)$
$= 0.763184$
Output from neuron E = $1/1+e^{-(0.763184)}$
$= \underline{0.682}$

Error = target − output from neuron E = 0.5 − 0.682
$\boxed{\text{Error} = -0.182}$ — ⑤

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