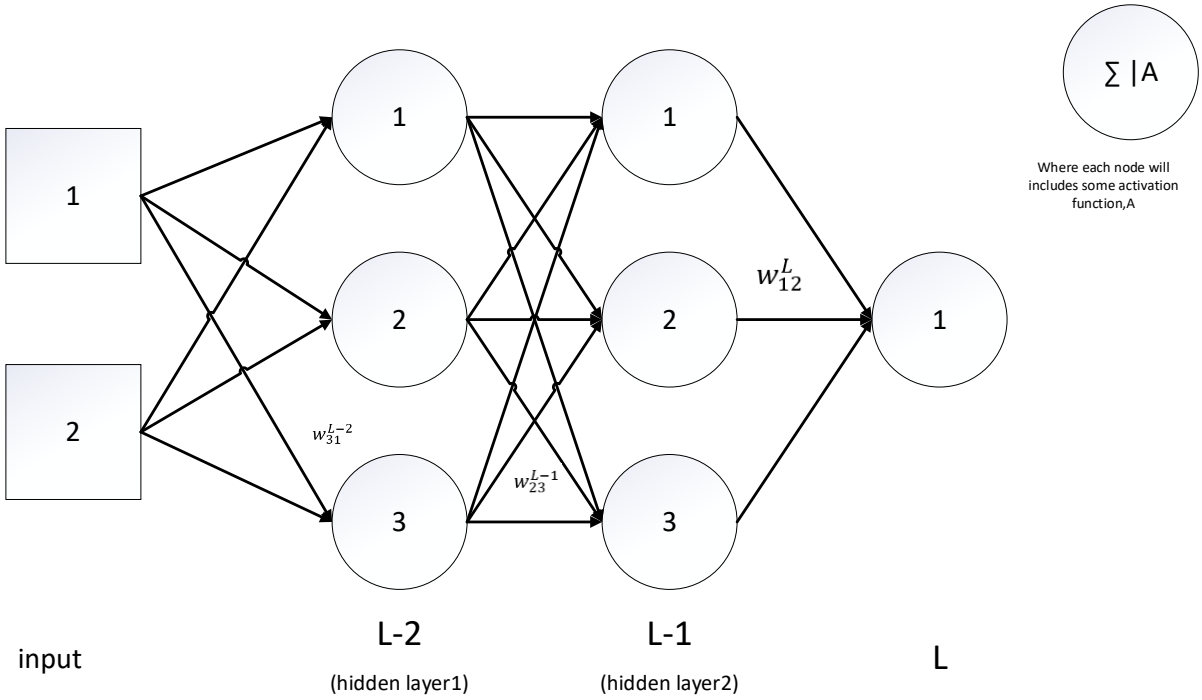
Network

The diagram we have drawn below shows a neural network with one input layer, two hidden layers, and an output layer. Each layer is fully connected to the next by a set of weights. The nodes comprising the hidden and output layers also include an activation function which is applied to the value computed at that node.



Feeding forward

Feeding forward is to use the input to do the calculation at each layer and to get the result at the output layer. Feed-forward should keep directed and no circle structure, which means output is the deterministic function of inputs. In our case, we will use the inputs to calculate the activations of each node at the hidden layer 1, and then use the activations of hidden layer one as input to calculate the activations of hidden layer 2. Finally, we will use activations of hidden layer 2 as input to calculate the output.

In our case, d =2, j =3, k =3 and n =1

for the input $x_1, x_2, x_2, \cdots x_{\mathrm{d}}$

Let a_j be the weighted sum based on input, at jth node of the hidden layer 1

We will have:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(L-2)} x_i + w_{j0}^{(\mathrm{L}-2)}$$

in this formula
j represents the index of current node, and i represents index of previous node, and then D represents the length of the input.

Then we will use an activation function $h(\cdot)$to give:

$$z_j = h(a_j)$$

at the second layer.
Let a_k be the weighted sum based on the activations of layer one, and at k th node of the hidden layer 2 we will have:

$$a_k = \sum_{j} w_{kj}^{(L-1)} z_j + w_{k0}^{(L-1)}$$

Then we will use an activation function $\sigma(\cdot)$ to give:

$$C_k = \sigma(a_k)$$

at the output layer.
Let a_n be the weighted sum based on the activations of layer two, and at n th node of the output layer we will have:

$$a_n = \sum_{k} w_{nk}^{(\mathrm{L})} c_k + w_{n_0}^{(L)}$$

Then we will use an activation function $f(\cdot)$ to give:

$$y_n = f(a_n)$$

To summary the output:

$$y_n = f(\sum_{k} w_{nk}^{(\mathrm{L})} \sigma\left(\sum_{j} w_{kj}^{(L-1)} h(a_j) + w_{ko}^{(L-1)}\right) + w_{n_0}^{(L)})$$

Where:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(L-2)} x_i + w_{j0}^{(\mathrm{L}-2)}$$

Back Propagation

Backpropagation is the opposite of feed-forward; We can do propagation since we can calculate the loss based on the feedforward step;here, we take the loss of the output and move "backwards" through the net to minimize the cost. To figure out what the "effect" of each node on the error is, we use the gradient, or vector of partial derivatives of the function. For the node, we look at the partial derivative and adjust it to reduce the error function. Therefore, backpropagation improved the model by minimizing the error function of it.

In our case, d =2, j =3, k =3 and n =1

Let the loss function be the sum of square error

$$E_m = \frac{1}{2}\sum_n (y_n - t_n)^2$$

t_n represents corrosponding target for a particular input X_m

Let the activation function h, σ, f be sigmoidal function tanh function

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

The derivative of it is:

$$\begin{aligned}(\tanh(a))' &= \frac{(e^a + e^{-a})(e^a + e^{-a}) - (e^a - e^{-a})(e^a - e^{-a})}{(e^a + e^{-a})^2} \\ &= \frac{(e^a + e^{-a})^2 - (e^a - e^{-a})^2}{(e^a + e^{-a})^2} \\ &= 1 - (\tanh(a))^2\end{aligned}$$

Let

$$\begin{aligned}e_m &= \frac{1}{2}\sum_n (y_n - t_n)^2 \\ y_n &= \tanh(a_n) \\ a_n &= \sum_k \omega_{nk}^{(L)} \mathrm{c_k} + w_{n0}^{(L)}\end{aligned}$$

At output layer

We will update the weight of the output layer according to the loss, we want the correct node's activations to increase, the rest decrease. We will update the weight of output nodes based on the total error. For each weight, the partial derivative of loss with respect to just that weight is used to update that weight using the chain rule to calculate the correspond partial derivative.

For specific direction of $W_{12}^{(L)}$ of the output layer

$$\frac{\partial e_m}{\partial w_{12}^{(L)}} = \left(\frac{\partial e_m}{\partial y_1^{(2)}}\right)\left(\frac{\partial y_1^{(L)}}{\partial a_1^L}\right)\left(\frac{\partial a_1^{(L)}}{\partial u_{12}^{(L)}}\right)$$

For $\left(\frac{\partial e_m}{\partial y_1^{(1)}}\right)$:

$$\left(\frac{\partial e_m}{\partial y_1^{(1)}}\right) = \frac{\partial}{\partial y_1^{(L)}}\frac{1}{2}\sum_n (y_n^L - t_n)^2$$
$$= \sum_n (y_n^L - t_n)\frac{\partial}{\partial y_1^{(2)}}(y_n^L - t_n)$$

Since:

$$\frac{\partial (y_n^{\mathrm{L}} - t_n)}{\partial y_1^2} = \begin{cases} 1 & n = 1 \\ 0 & \text{else} \end{cases}$$

Thus,

$$\left(\frac{\partial e_m}{\partial y_1^{(1)}}\right) = (y_1^L - t_1)$$

For $\left(\frac{\partial y_1^{(\mathrm{L})}}{\partial a_1^{(L)}}\right)$:

$$\left(\frac{\partial y_1^{(\mathrm{L})}}{\partial a_1^{(L)}}\right) = \frac{\partial}{\partial a_1^{(L)}}\tanh\left(a_1^{(\mathrm{L})}\right)$$

Since:

$$\tanh(x)' = 1 - (\tanh(x))^2$$

Thus,

$$\left(\frac{\partial y_1^{(\mathrm{L})}}{\partial a_1^{(L)}}\right) = 1 - \left(\tanh\left(a_1^{(\mathrm{L})}\right)\right)^2$$

For:

$$\left(\frac{\partial a_1^{(\mathrm{L})}}{\partial W_{12}^{(\mathrm{L})}}\right) = \frac{\partial}{\partial W_{12}^{(\mathrm{L})}} \sum_k \omega_{1k}^{(L)} C_k + w_{10}^{(L)} = C_2$$

Thus,

$$\frac{\partial e_m}{\partial w_{12}^{(L)}} = (y_1^L - t_1)(1 - \left(\tanh\left(a_1^{(\mathrm{L})}\right)\right)^2) C_2$$

When we do the update, we will let

$$w_{12}^L = w_{12}^L - \eta \frac{\partial e_m}{\partial w_{12}^L}$$

For more general case:

$$\frac{\partial e_m}{\partial w_{\mathrm{nk}}^{(L)}} = (y_{\mathrm{n}}^L - t_{\mathrm{n}})(1 - \left(\tanh\left(a_{\mathrm{n}}^{(\mathrm{L})}\right)\right)^2) C_{\mathrm{k}}$$

$$w_{\mathrm{nk}}^L = w_{\mathrm{nk}}^L - \eta \frac{\partial e_m}{\partial w_{\mathrm{nk}}^L}$$

From the perspective of gradient, for each output we can calculate it's gradient and update it correspond weights.

Let w_n represent the nth output's correpond weights which has k elements correspond to node's number of previous layer, which is a vector.

We will have:

$$\boldsymbol{\nabla w_n^{(L)}} = (\frac{\partial e_m}{\partial w_{\mathrm{n1}}^L}, \frac{\partial e_m}{\partial w_{\mathrm{n2}}^L}, \dots, \frac{\partial e_m}{\partial w_{\mathrm{nk}}^L})$$

The update process can be written as

$$\boldsymbol{w_n^L = w_n^L - \eta \nabla w_n^{(L)}}$$

At the hidden two layer

At this level, the work almost all the same except some term depends on the loss of all output nodes in layer L.

For specific direction of $W_{23}^{(L-1)}$

We will have:

$$\frac{\partial e_m}{\partial w_{23}^{(\mathrm{L}-1)}} = \left(\frac{\partial e_m}{\partial C_2}\right)\left(\frac{\partial C_2}{\partial a_2^{(L-1)}}\right)\left(\frac{\partial a_2^{(L-1)}}{\partial w_{23}^{(L-1)}}\right)$$

For:

$$\frac{\partial e_m}{\partial C_2} = \sum_n \left(\frac{\partial e_m}{\partial y_n^L}\right)\left(\frac{\partial y_n^{(L)}}{\partial a_n^L}\right)\left(\frac{\partial a_n^L}{\partial C_2}\right)$$

$$\frac{\partial e_m}{\partial y_n^L} = (y_n^L - t_n)$$

$$\frac{\partial y_n^L}{\partial a_n^L} = \left(1 - \left(\tanh\left(a_n^L\right)\right)^2\right)$$

$$\frac{\partial a_n^L}{\partial C_2} = \frac{\partial}{\partial C_2}\sum_k W_{nk}^L C_k + W_{n0}^{(L)} = W_{n2}^L$$

$$\frac{\partial C_2}{\partial a_2^{(\mathrm{L}-1)}} = \frac{\partial}{\partial a_2^{(\mathrm{L}-1)}}\tanh\left(a_2^{(L-1)}\right) = \left(1 - \left(\tanh\left(a_2^{L-1}\right)\right)^2\right)$$

$$\frac{\partial a_2^{(L-1)}}{\partial W_{23}^{L-1}} = \frac{\partial}{\partial W_{23}^{L-1}}\sum_j W_{2j}^{(L-1)} z_j + W_{k0}^{(L-1)} = z_3$$

Thus:

$$\frac{\partial e_m}{\partial w_{23}^{(\mathrm{L}-1)}} = \left(\frac{\partial c_2}{\partial a_2^{(\mathrm{L}-1)}}\right)\left(\frac{\partial a_2^{(\mathrm{L}-1)}}{\partial w_{23}^{(\mathrm{L}-1)}}\right)\sum_n \left(\frac{\partial e_m}{\partial y_n^L}\right)\left(\frac{\partial y_n^{(L)}}{\partial a_n^{(L)}}\right)\left(\frac{\partial a_n^L}{\partial C_2}\right)$$

When we do the update, we will let

$$W_{23}^{L-1} = W_{23}^{L-1} - \eta \frac{\partial e_m}{\partial w_{23}^{(\mathrm{L}-1)}}$$

For more general case:

$$\frac{\partial e_m}{\partial w_{\mathrm{kj}}^{(\mathrm{L}-1)}} = \left(\frac{\partial c_{\mathrm{k}}}{\partial a_{\mathrm{k}}^{(\mathrm{L}-1)}}\right)\left(\frac{\partial a_{\mathrm{k}}^{(\mathrm{L}-1)}}{\partial w_{\mathrm{kj}}^{(\mathrm{L}-1)}}\right)\sum_n \left(\frac{\partial e_m}{\partial y_n^L}\right)\left(\frac{\partial y_n^{(L)}}{\partial a_n^{(L)}}\right)\left(\frac{\partial a_n^L}{\partial C_{\mathrm{k}}}\right)$$

$$w_{\mathrm{kj}}^{L-1} = w_{\mathrm{kj}}^{L-1} - \eta \frac{\partial e_m}{\partial w_{\mathrm{kj}}^{L-1}}$$

From the perspective of gradient, let w_k represent the kth output's correpond weights which has j elements correspond to node's number of previous layer, which is a vector.

We will have:

$$\boldsymbol{\nabla w_k^{(L-1)}} = \left(\frac{\partial e_m}{\partial w_{\mathrm{k1}}^{L-1}}, \frac{\partial e_m}{\partial w_{\mathrm{k2}}^{L-1}}, \ldots, \frac{\partial e_m}{\partial w_{\mathrm{kj}}^{L-1}}\right)$$

The update process can be written as

$$\boldsymbol{w_{\mathrm{k}}^{L-1} = w_{\mathrm{k}}^{L-1} - \eta \nabla w_k^{(L-1)}}$$

At the hidden one layer

At this level, the work almost all the same except some term depends on the loss of all output nodes in layer L and intermediary nodes in layer L-1.

For specific direction of $W_{31}^{(L-2)}$

We will have:

$$\frac{\partial e_m}{\partial w_{31}^{(\mathrm{L}-2)}} = \left(\frac{\partial e_m}{\partial z_3}\right)\left(\frac{\partial z_3}{\partial a_3^{(\mathrm{L}-2)}}\right)\left(\frac{\partial a_3^{(L-2)}}{\partial W_{31}^{(1-2)}}\right)$$

For:

$$\left(\frac{\partial e_m}{\partial z_3}\right) = \sum_k \left(\sum_n \left(\frac{\partial e_m}{\partial y_n^{\mathrm{L}}}\right)\left(\frac{\partial y_n^{(L)}}{\partial a_n^{(\mathrm{L})}}\right)\left(\frac{\partial a_n^L}{\partial C_k}\right)\right)\left(\frac{\partial C_k}{\partial a_k^{(\mathrm{L}-1)}}\right)\left(\frac{\partial a_k^{(\mathrm{L}-1)}}{\partial z_3}\right)$$

$$\left(\frac{\partial z_3}{\partial a_3^{(\mathrm{L}-2)}}\right) = 1 - \left(\tanh\left(a_3^{(\mathrm{L}-2)}\right)\right)^2$$

$$\left(\frac{\partial a_3^{(L-2)}}{\partial W_{31}^{(1-2)}}\right) = x_1$$

When we do the update, we will let

$$W_{31}^{L-2} = W_{31}^{L-2} - \eta \frac{\partial e_m}{\partial w_{31}^{(\mathrm{L}-2)}}$$

For general case:

$$W_{\mathrm{jd}}^{(L-2)}$$

$$\frac{\partial e_m}{\partial w_{\mathrm{jd}}^{(\mathrm{L}-2)}} = \left(\frac{\partial e_m}{\partial z_{\mathrm{j}}}\right)\left(\frac{\partial z_{\mathrm{j}}}{\partial a_{\mathrm{j}}^{(\mathrm{L}-2)}}\right)\left(\frac{\partial a_{\mathrm{j}}^{(L-2)}}{\partial W_{\mathrm{jd}}^{(1-2)}}\right)$$

For:

$$\left(\frac{\partial e_m}{\partial z_{\mathrm{j}}}\right) = \sum_k \left(\sum_n \left(\frac{\partial e_m}{\partial y_n^{\mathrm{L}}}\right)\left(\frac{\partial y_n^{(L)}}{\partial a_n^{(\mathrm{L})}}\right)\left(\frac{\partial a_n^L}{\partial C_k}\right)\right)\left(\frac{\partial C_k}{\partial a_k^{(\mathrm{L}-1)}}\right)\left(\frac{\partial a_k^{(\mathrm{L}-1)}}{\partial z_{\mathrm{j}}}\right)$$

$$\left(\frac{\partial z_{\mathrm{j}}}{\partial a_{\mathrm{j}}^{(\mathrm{L}-2)}}\right) = 1 - \left(\tanh\left(a_{\mathrm{j}}^{(\mathrm{L}-2)}\right)\right)^2$$

$$\left(\frac{\partial a_{\mathrm{j}}^{(L-2)}}{\partial W_{\mathrm{jd}}^{(\mathrm{L}-2)}}\right) = x_d$$

$$w_{\mathrm{jd}}^{L-2} = w_{\mathrm{jd}}^{L-2} - \eta \frac{\partial e_m}{\partial w_{\mathrm{jd}}^{L-2}}$$

From the perspective of gradient, let w_j represent the jth output's correpond weights which has d elements correspond to node's number of inputs, which is a vector.

We will have:

$$\boldsymbol{\nabla w_j^{(L-2)}} = \left(\frac{\partial e_m}{\partial w_{\mathrm{j1}}^{L-2}}, \frac{\partial e_m}{\partial w_{\mathrm{j2}}^{L-2}}, \ldots, \frac{\partial e_m}{\partial w_{\mathrm{jd}}^{L-2}}\right)$$

The update process can be written as

$$\boldsymbol{w_{\mathrm{j}}^{L-2} = w_{\mathrm{j}}^{L-2} - \eta \nabla w_j^{(L-2)}}$$