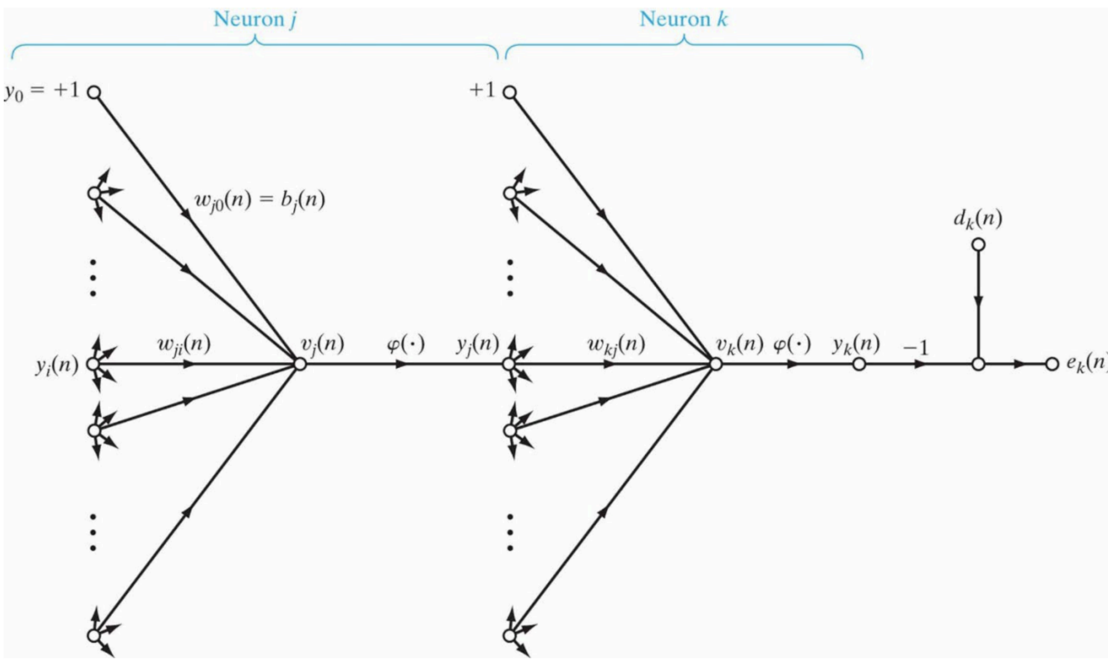
Homework - 11
COEN 240-Machine Learning

Pujitha Kallu
ID : W1653660
pkallu@scu.edu

Consider the following signal-flow graph of a fully-connected neural network that consists of an input layer, one hidden layer and an output layer. y_i is the i^{th} input node in the input layer. Neuron j is the j^{th} neuron in the hidden layer and neuron k is the k^{th} output neuron. Assume the activation function $\varphi(.)$ is sigmoid.



(a). Use back-propagation on the output neuron k to show that the weight correction Δw_{kj} for the n^{th} iteration is given by

$$\Delta w_{kj}(n) = \eta . \delta_k(n) . y_j(n)$$

Where η is the learning rate and the local gradient $\delta_k(n) = [d_k(n) - y_k(n)].\left[y_k(n)\left(1 - y_k(n)\right)\right]$

Notice $y_k(n)$ is obtained from the forward pass:

$$y_k(n) = \varphi\left(v_k(n)\right)$$

$$v_k(n) = \sum_j w_{kj}(n) y_j(n)$$

Where $y_j(n)$ is the output of the hidden neuron j and $w_{kj}(n)$ is the synaptic weight of $y_j(n)$.

Sol:

Sol: For output neuron k.

$$\mathcal{E}(n) = \frac{1}{2} \sum_k e_k^2(n)$$

$$= \frac{1}{2} \sum_k (d_k(n) - y_k(n))^2$$

we need to find:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{kj}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial w_{kj}(n)}$$

$$\frac{\partial \mathcal{E}(n)}{\partial e_k(n)} = e_k(n) \quad , \quad \frac{\partial e_k(n)}{\partial y_k(n)} = -1$$

$$y_k(n) = \phi(v_k(n)) \longrightarrow \frac{\partial y_k(n)}{\partial w_{kj}(n)} = y_j(n)$$

Thus,

$$\frac{\partial \mathcal{E}(n)}{\partial w_{kj}(n)} = -e_k(n)\, \phi'(v_k(n))\, y_j(n)$$

Defining weight connection $\Delta w_{kj}(n)$

$$\Delta w_{kj}(n) = \frac{-\eta\, \partial \mathcal{E}(n)}{\partial w_{kj}(n)} = \eta e_k(n) \phi'(v_k(n)) . y_j$$

defining local gradient $\delta_k(n)$

$$\delta_k(n) = e_k(n) \cdot \phi'(v_k(n))$$

Hence ,

$$\Delta w_{kj}(n) = \eta \cdot \partial_k(n) \cdot y_j(n)$$

if $\phi(v_k(n))$ is a sigmoid function
then

$$\phi'(v_k(n)) = \phi(v_k(n))(1-\phi(v_k(n)))$$

But $\phi(v_k(n))$ is $y_k(n)$

$$\therefore \quad \partial_k(n) = e_k(n)\,\phi'(v_k(n))$$

$$= [d_k(n) - y_k(n)][y_k(n)(1-y_k(n))]$$

(b). Use back-propagation on the hidden neuron j to show that the weight correction Δw_{ji} for the n^{th} iteration is given by

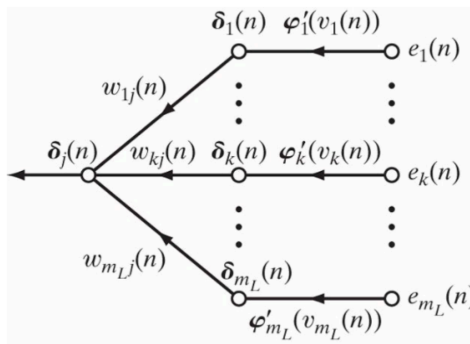
$$\Delta w_{ji}(n) = \eta . \delta_j(n) . y_i(n)$$

Where the $\delta_j(n)$ is the overall backpropagated gradient from the layer to the immediate right (i.e., the output layer) given by

$$\delta_j(n) = \sum_k \delta_k(n) . w_{kj}(n) . y_j(n) . \left(1 - y_j(n)\right)$$

and $y_j(n)$ is the output of the hidden neuron j.

Note that the effect of all e_k's must be included, hence the summation over k.



$y_j(n)$ is obtained from the forward pass:

$$y_j(n) = \varphi\left(v_j(n)\right)$$

$$v_j(n) = \sum_i w_{ji}(n) y_i(n)$$

Where $y_i(n)$ is the i-th input of neuron j (input node i).

Sol:

b) Hidden neuron j.

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial w_{ji}(n)} \quad \text{where,}$$

$$\mathcal{E}(n) = \frac{1}{2} \sum_k e_k^2(n) = \frac{1}{2} \sum_k \left(d_k(n) - y_k(n) \right)^2$$

Hence,

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)}$$

$$\frac{\partial \mathcal{E}(n)}{\partial y_i(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_i(n)}$$

Where $e_k(n) = d_k(n) - y_k(n)$

$$= d_k(n) - \phi(v_k(n))$$

$$\rightarrow \cancel{\frac{\partial e_k(n)}{\partial v_k(n)}}$$

$$= \frac{\partial e_k(n)}{\partial v_k(n)} = -\phi'(v_k(n))$$

$$v_k(n) = \sum_j w_{kj} - (n) y_j(n)$$

$$\rightarrow \frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

Thus,

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}$$

$$= -\sum_k e_k(n) \cdot \phi'(v_k(n)) \cdot w_{kj}(n)$$

Next,

$$\frac{\partial y_j(n)}{\partial w_{ji}(n)} = \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

Where

$$y_j(n) = \phi(v_j(n)) \rightarrow \frac{\partial y_j(n)}{\partial v_j(n)} = \phi'(v_j(n))$$

$$V_j(n) = \sum_i w_{ji}(n) y_i(n)$$

$$\rightarrow \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

So, $$\frac{\partial y_j(n)}{\partial w_{ji}(n)} = \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)} = \phi'(v_j(n)) y_i(n)$$

Hence,
$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial w_{ji}(n)}$$

$$= -\sum_k e_k(n) \cdot \phi'(v_k(n)) w_{kj}(n) \cdot \phi'(v_j(n)) y(n)$$

$$= -\sum_k \delta_k(n) w_{kj}(n) \cdot \phi'(v_j(n)) \cdot y_i(n)$$

Where from part $\delta_k(n) = e_k(n) \phi'(v_k(n))$

Now defining the overall gradient $\delta_j(n)$ which is back propagated gradient from the layer to immediate right (output layer)

$$\delta_j(n) = \sum_k \delta_k(n) \cdot w_{kj}(n) \phi'(v_j(n))$$

Thus,
$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = -\delta_j(n) y_i(n)$$

if $\phi(v_j(n))$ is a signed function

then $\phi(v_j(n)) = \phi(v_j(n))(1-\phi(v_j(n))$

$$= y_j(n)\,(1-y_j(n))$$

so,

$$\partial_j(n) = \sum_k \partial_k(n)\, \omega_{kj}(n)\, y_j(n)\,(1-y_j(n))$$

Now defining weight correction $\Delta w_{ji}(n)$:

$$\Delta w_{ji}(n) = \frac{-n\,\partial \epsilon(n)}{\partial w_{kj}(n)} = n \cdot \partial_j(n)\, y_i(n)$$

Recall $y_j(n)$ is the output of the hidden neuron j $y_i(n)$ is the input node & $\partial_k(n)$ is the local gradient for the output neuron k described in part (a)

So, $\partial_k(n) = [\partial_k(n) - y_k(n)]\,[y_k(n)(1-y_k(n))]$

References:

1. Class notes:Back propagation
2. https://www.geeksforgeeks.org/backpropagation-in-neural-network/
3. https://www.sciencedirect.com/topics/veterinary-science-and-veterinary-medicine/backpropagation