CS6375
ASSIGNMENT-I

Saketh Annamalla
Revanth Chandra Gampa
Omar Mukhtar Shaik

1.1)

We know that Weight delta,

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$, where E_d = error for example d

$E_d(w) = \frac{1}{2} \sum_{k \in outputs} (t_k - O_k)^2$, i.e sum of squared error for all output units.

Let's compute $\frac{\partial E_d}{\partial w_{ji}}$

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} \quad \text{— (I)}$$

as $net_j = \sum_i w_{ji} x_{ji}$, $\frac{\partial net_j}{\partial w_{ji}} = x_{ji}$ — (II)

$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial O_j} \frac{\partial O_j}{\partial net_j}$ → Case 1, j is an output unit

$\frac{\partial E_d}{\partial net_j} = \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j}$ → Case 2, j is a hidden unit

Case 1,

$$\frac{\partial E_d}{\partial O_j} = \frac{\partial}{\partial O_j}\left[\frac{1}{2} \sum_{k \in outputs} (t_k - O_k)^2\right] = \frac{1}{\cancel{2}} \times (-\cancel{2}(t_j - O_j))$$

$$\frac{\partial E_d}{\partial O_j} = -(t_j - O_j) \quad \text{— (III)}$$

We are yet to find $\frac{\partial O_j}{\partial net_j}$

Let's solve $\frac{\partial O_j}{\partial net_j}$ for our 2 Questions using tanh activation function & ReLu activation function.

a) tanh activation function

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

Let's try computing $\frac{\partial o_j}{\partial net_j}$ in this case

$$\frac{d(\tanh(x))}{dx} = \frac{(e^x+e^{-x})(e^x+e^{-x}) - (e^x-e^{-x})(e^x-e^{-x})}{(e^x+e^{-x})^2}$$

$$= 1 - \left(\frac{(e^x-e^{-x})}{(e^x+e^{-x})}\right)^2 = 1 - \tanh^2(x)$$

Therefore, value of $\frac{\partial o_j}{\partial net_j} = 1 - o_j^2$ — (IV)

From (I)

$$\Rightarrow \frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial net_j}$$

From (III), (IV)

$$\Rightarrow \frac{\partial E_d}{\partial net_j} = (o_j - t_j)(1 - o_j^2) \quad \text{— (V)}$$

Combining (II), (V)

Therefore, In Case 1 $\frac{\partial E_d}{\partial w_{ji}} = (o_j - t_j)(1 - o_j^2) x_{ji}$

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = \eta (t_j + t_j)(1 - o_j^2) x_{ji}$$

Let's assume $(t_j - o_j)(1 - o_j^2) = \delta_j$

Let's try computing for hidden unit

$$\frac{\partial E_d}{\partial net_j} = \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j}$$ / In previous step we assumed $\frac{\partial E_d}{\partial net_k} = -\delta_k$

$$= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial net_j}$$

$$= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial O_j} \frac{\partial O_j}{\partial net_j}$$

In the above $\frac{\partial net_k}{\partial O_j} = \omega_{kj}$

& we know from (IV) $\frac{\partial O_j}{\partial net_j} = 1 - O_j^2$

$$\Rightarrow \frac{\partial E_d}{\partial net_j} = \sum_{k \in Downstream(j)} -\delta_k \omega_{kj} \cdot (1-O_j^2) = \underbrace{-(1-O_j^2) \sum_{k \in Downstream(j)} \delta_k \omega_{kj}}_{\text{lets assume this as } -\delta_j}$$

$$\Rightarrow \Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = -\eta(-\delta_j)(x_{ji})$$

$$\Rightarrow \Delta w_{ji} = \eta \delta_j x_{ji}$$

Therefore,

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

where $\delta_j = \begin{cases} (t_j - O_j)(1-O_j^2), & \text{if } j \text{ is output unit} \\ (1-O_j^2) \sum_{k \in Downstream(j)} \delta_k \omega_{kj}, & \text{if } j \text{ is hidden unit} \end{cases}$

b) Relu activation function

$$ReLu(x) = \max(0, x)$$

$$\frac{d(ReLu(x))}{dx} = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}$$

Therefore, $$\frac{\partial o_j}{\partial net_j} = \begin{cases} 0, & \text{if } net_j < 0 \\ 1, & \text{if } net_j > 0 \end{cases}$$

$$\frac{\partial E_d}{\partial net_j} = \begin{cases} (o_j - t_j), & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases}$$

In case 1 $$\frac{\partial E_d}{\partial w_{ji}} = \begin{cases} (o_j - t_j)x_{ji}, & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases}$$

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = \begin{cases} \eta(t_j - o_j)x_{ji}, & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases} //$$

Let's assume $$\delta_j = \begin{cases} (t_j - o_j), & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases}$$

Let's try computing for hidden layer

Case 2, $$\frac{\partial E_d}{\partial net_j} = \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j} = \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial net_j}$$

$$= \sum_{k \in Downstream(j)} \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j}$$

In the above $$\frac{\partial net_k}{\partial o_j} = w_{kj}$$

& we know that $\frac{\partial o_j}{\partial net_j} = \begin{cases} 0, & \text{if } net_j < 0 \\ 1, & \text{if } net_j > 0 \end{cases}$

$$\Rightarrow \frac{\partial E_d}{\partial net_j} = \begin{cases} \sum_{k \in Downstream(j)} - \delta_k w_{kj}, & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases}$$

$$\Rightarrow \Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} \cancel{\eta(-\delta_j)(x_{ji})}$$

Therefore,

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

where $\delta_j = \begin{cases} (t_j - o_j), & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases}$, for output j unit

$\delta_j = \begin{cases} \sum_{k \in downstream(j)} \delta_k w_{kj}, & \text{if } net_j > 0 \\ 0, & \text{if } net_j < 0 \end{cases}$, for ~~h~~hidden units

Following is the algorithm for both Tanh & ReLu activation functions

For each training example,

1. Input it to network & compute network outputs } Forward pass

2. For each output unit k

$$\delta_k \leftarrow (t_k - O_k)(1 - O_k^2) \quad , \text{ for Tanh}$$

$$\delta_k \leftarrow \begin{cases} t_k - O_k , & \text{if } net_k > 0 \\ 0 , & \text{if } net_k < 0 \end{cases} \quad , \text{ for ReLu}$$

3. For each hidden unit

$$\delta_h \leftarrow (1 - O_h^2) \sum_{k \in Downstream(h)} \delta_k w_{kh} \quad , \text{ for Tanh}$$

$$\delta_h \leftarrow \begin{cases} \sum_{k \in Downstream(h)} \delta_k w_{kh} , & \text{if } net_k > 0 \\ 0 , & \text{if } net_k < 0 \end{cases} \quad , \text{ for ReLU}$$

4. Update each network weight $w_{i,j}$

$$w_{ij} \leftarrow w_{i,j} + \Delta w_{i,j} ,$$

where $\Delta w_{i,j} = \eta \delta_j x_{ij}$

1.2

$$O = w_0 + w_1(x_1 + x_1^2) + \cdots \quad w_n(x_n + x_n^2)$$

$$E_d = \frac{1}{2}\sum_d (t_d - O_d)^2$$

$$\frac{\partial E_d}{\partial w_i} = \cancel{\frac{2}{2}}\sum_d (t_d - O_d) \cdot \frac{\partial (t_d - O_d)}{\partial w_i}$$

$$= \sum_d (t_d - O_d) \cdot \frac{\partial}{\partial w_i}\left(t_d - (w_0 + w_1(x_1 + x_1^2) + \cdots w_n(x_n^2 + x_n)\right)$$

$$= \sum_d (t_d - O_d) \cdot (-(x_i + x_i^2))$$

$$\Rightarrow \frac{\partial E_d}{\partial w_i} = -\sum_d (t_d - O_d) \cdot (x_i + x_i^2)$$

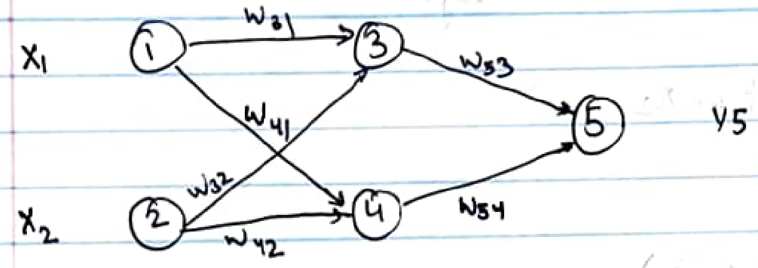
$$\Delta w_i = -\eta \cdot \partial E / \partial w_i = -\eta \cdot \sum_d (t_d - O_d) \cdot (x_i + x_i^2)$$

$$\boxed{w_i^{new} = w_i^{old} + \Delta w_i}$$

where $\Delta w_i = \eta \cdot \sum_d (t_d - O_d) \cdot (x_i + x_i^2)$

$\eta \rightarrow$ learning rate

1.3 Given neural network has 2 input layer neurons, one hidden layer with 2 neurons, and 1 output layer neuron.
Activation function of the input layer is identity function and each neuron of hidden layers and output layer use activation function $h(x)$.



Input layer hidden layer output layer.

a) Output y_5 in terms of weights is

output at 1,2 neurons:

$$net1 = f(x_1) = x_1$$
$$net2 = f(x_2) = x_2$$

output at 3,4 neurons:

$$net3 = h(w_{31}x_1 + w_{32}x_2)$$
$$net4 = h(w_{41}x_1 + w_{42}x_2)$$

Final output is $y_5 = h(w_{53}\,net3 + w_{54}\,net4)$

$$= h(w_{53}(h(w_{31}x_1 + w_{32}x_2)) + (w_{54}(h(w_{41}x_1 + w_{42}x_2))$$

$\therefore$ output $y_5 = h(w_{53} \cdot h(w_{31} \cdot x_1 + w_{32} \cdot x_2) + w_{54} \cdot h(w_{41} \cdot x_1 + w_{42} \cdot x_2))$

b)

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad w^{(1)} = \begin{pmatrix} w_{31} & w_{32} \\ w_{41} & w_{42} \end{pmatrix}$$

Input layer

$$w^{(2)} = \begin{pmatrix} w_{53} & w_{54} \end{pmatrix}$$

hidden layer.

$net1 = x_1 \qquad net2 = x_2$

output layer:

$$H = \begin{pmatrix} net3 \\ net4 \end{pmatrix}$$

$$= h \begin{pmatrix} w_{31} & w_{32} \\ w_{41} & w_{42} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$= h \left(w^{(1)} \cdot X \right)$$

$$y_5 = h \left(\begin{pmatrix} w_{53} & w_{54} \end{pmatrix} \cdot H \right)$$

$$= h \left(w^{(2)} \cdot h \left(w^{(1)} \cdot X \right) \right)$$

c) Sigmoid function

$$h_s(x) = \frac{1}{1+e^{-x}}$$

Tan h

$$h_t(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Need to show that neural nets created using the above two activation functions can generate same function

$$h_s(x) = \frac{1}{1+e^{-x}}$$

$$h_t(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

$$= \frac{e^{x}(1-e^{-2x})}{e^{x}(1+e^{-2x})}$$

$$= \frac{1-e^{-2x}}{1+e^{-2x}}$$

$$= \frac{2}{1+e^{-2x}} - \left(\frac{1+e^{-2x}}{1+e^{-2x}}\right)$$

$$h_t(x) = \frac{2}{1+e^{-2x}} - 1$$

we have $h_s(2x) = \dfrac{1}{1+e^{-2x}}$

$\therefore$ we have $h_t(x) = 2h_s(2x) - 1$

$\therefore$ we have shown that tanh and sigmoid functions, both activation functions generate the same output

one is a rescaled form of other activation function