1.1 Revision Backpropagation Algorithm

a) tanh activation function

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

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

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

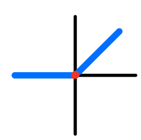
Case I: j is output layer node:

$$\boxed{\delta_j = (t_j - o_j) \cdot \frac{2e^{2x}}{e^{2x}+1}(1 - o_j)}$$

Case II: j is hidden layer node:

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

b) ReLu activation function



$$\mathrm{ReLu}(x) = \max(0, x)$$

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

Case I: j is output layer node:

$$\boxed{\delta_j = (t_j - o_j) \cdot \mathrm{ReLu}'(x)}$$

Case II: j is hidden layer node:

$$\boxed{\delta_j = \mathrm{ReLu}'(x)}$$

1.2 Gradient Descent

Act. Function:
$f(x)=x$

$net_j = \sum_{i=1}^{n} w_{ji} \cdot (x_{ji} + x_{ji}^2)$

$o_j = f(x)\ (net_j)$

Case I: j is output layer node:

$\Delta W_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$

$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ji}}$

$\frac{\partial net_j}{\partial w_{ji}} = x_{ji} + x_{ji}^2$

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

$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} [\frac{1}{2}(t_j - o_j)^2]$
$= -(t_j - o_j)$

$\frac{\partial o_j}{\partial net_j} = f'(x)$

$\frac{\partial E_d}{\partial net_j} = \boxed{-(t_j - o_j) \cdot f'(x) = -\delta_j}$

$\Delta W_{ji} = \eta\ \delta_j\ x_{ji} + x_{ji}^2$

$\boxed{W_{ji}^{new} = w_{ji}^{old} + \Delta W_{ji}}$

Case II: j is hidden layer node:

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

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

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

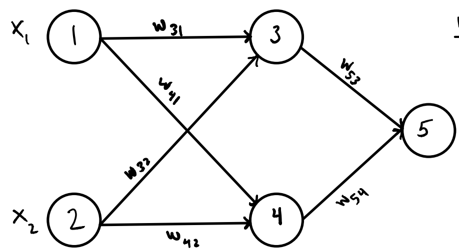
$= \sum_{k \in Downstream(j)}$

$\boxed{\delta_j = -\delta_k\ w_{kj} \cdot f'(x)}$

$\Delta W_{ji} = \eta\ \delta_j\ x_{ji} + x_{ji}^2$

$\boxed{W_{ji}^{new} = w_{ji}^{old} + \Delta W_{ji}}$

1.3 Comparing Activation Function



input act funct. : $f(x)$

hidden/output act funct: $h(x)$

a).

Node	Net	Output
1	$f(x_1) = x_1$	$x_1 = f(x_1)$
2	$f(x_2) = x_2$	$x_2 = f(x_2)$
3	$net_3 = w_{31} x_1 + w_{32} x_2$	$x_3 = h(net_3)(net_3)$
4	$net_4 = w_{41} x_1 + w_{42} x_2$	$x_4 = h(net_4)(net_4)$
5	$net_5 = w_{53} x_3 + w_{54} x_4$	$\boxed{y_5 = h(net_5)(net_5)}$

b.) given :

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

$$W^{(2)} = (w_{53} \;\; w_{54})$$

derived:

$$X^{(2)} = \begin{pmatrix} x_3 = h(x_3)(net_3) \\ x_4 = h(x_4)(net_4) \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}$$

$$\boxed{Y = h(x_5)\begin{pmatrix} x_3 \\ x_4 \end{pmatrix} \cdot (w_{53} \;\; w_{54})}$$

c.) sigmoid :

$$x_1 = x_1 \,, \; x_2 = x_2$$

$$x_3 = [w_{31} \; x_1 + w_{32} \; x_2] \cdot \frac{1}{1+e^{-net_3}}$$

$$x_4 = [w_{41} \; x_1 + w_{42} \; x_2] \cdot \frac{1}{1+e^{-net_4}}$$

$$y_5 = [w_{53} x_3 + w_{54} x_4] \cdot \frac{1}{1+e^{-net_5}}$$

Tanh:

$$x_1 = x_1 \,, \; x_2 = x_2$$

$$x_3 = [w_{31} \; x_1 + w_{32} \; x_2] \cdot \frac{e^{net_3} - e^{-net_3}}{e^{net_3} + e^{-net_3}}$$

$$x_4 = [w_{41} (i_1^2) + w_{42} (i_2^2)] \cdot \frac{e^{net_4} - e^{-net_4}}{}$$

$$y_5 = [w_{53} x_3 + w_{54} x_4] \cdot \frac{e^{net_5} - e^{-net_5}}{e^{net_5} + e^{-net_5}}$$

Sigmoid vs. Tanh

$$y_5 = \frac{[w_{53} x_3 + w_{54} x_4] \cdot \frac{1}{1+e^{-net_5}}}{[w_{53} x_3 + w_{54} x_4]} = \frac{[w_{53} x_3 + w_{54} x_4] \cdot \frac{e^{net_5} - e^{-net_5}}{e^{net_5} + e^{-net_5}}}{[w_{53} x_3 + w_{54} x_4]}$$

$$\boxed{\frac{1}{1+e^{-net_5}} = \frac{e^{net_5} - e^{-net_5}}{e^{net_5} + e^{-net_5}} \qquad h_s(x) = h_t(x)}$$

1.4 Gradient Descent with a Weight Penalty

given:

$$E(w) = \frac{1}{2} \sum_{d \in D} \sum_{k \in outputs} (t_{kd} - o_{kd})^2 + \gamma \sum_{i,j} w_{ji}^2 = \frac{1}{2} \Sigma (t_{kd} - o_{kd})^2 + \gamma \Sigma w_{ji}^2$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ji}}$$

$$= -f(x)(t_{kd} - o_{kd}) = -\delta_j$$

$$\boxed{\Delta w_{ji} = \eta \delta_j \gamma \cdot \partial w_{ji}}$$

$$\frac{\partial E}{\partial net} = \frac{\partial E}{\partial o_{kd}} \frac{\partial o_{kd}}{\partial net_j}$$

$$\frac{\partial net_j}{\partial w_{ji}} = \gamma \cdot \partial w_{ji}$$

$$\frac{\partial E}{\partial o_{kd}} = \frac{\partial}{\partial o_{kd}} \left[\frac{1}{2} \Sigma (t_{kd} - o_{kd})^2 + \gamma \Sigma w_{ji}^2\right]$$

$$= \frac{\partial}{\partial o_{kd}} \left[\frac{1}{2} (t_{kd} - o_{kd})^2 + \gamma w_{ji}^2\right]$$

$$= -(t_{kd} - o_{kd})$$

$\frac{\partial o_{kd}}{\partial net_j} = f(x)$, where $f'(x)$ is the activation function

Summarized Results:

Sigmoid and Tanh activation functions performed the best in my test. I expected ReLu to be the best performer because the positive data, however, it seemed as if many of the weights began to be negated over more iterations. Sigmoid and Tanh performed similarly. Sigmoid seemed to become obsolete sooner. For that reason, Tanh was the best performer. I expect this to dramatically change in the favor of ReLu with an increase of hidden layers. The test I performed included 4-8 hidden nodes (1 layer).