1.1) $\frac{\partial E_d}{\partial net_j} = \underbrace{\frac{\partial E_d}{\partial o_j}} \cdot \frac{\partial o_j}{\partial net_j}$

$$\frac{\partial E_d}{\partial o_j} = \frac{1}{2}\sum_k (t_k - o_k)^2$$

$$= 2\left(\frac{1}{2}\right)\frac{\partial}{\partial o_j}(t_j - o_j)$$

$$= -(t_j - o_j)$$

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

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

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

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

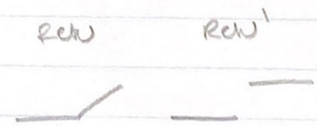
$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j}$$

$$= -(t_j - o_j)(1 - \tanh^2(o_j))$$

Update weight rule

$$\Delta w_{ji} = -\alpha \frac{\partial E_d}{\partial w_{ji}} \rightarrow \alpha(t_j - o_j)(1 - \tanh^2(o_j))\, x_{ji}$$

b) $\text{ReLU}(x) = \max(0, x) \rightarrow \frac{d}{dx}(\max(0, x))$



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

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} \rightarrow -(t_j - o_j)\frac{d}{do_j}(\text{ReLU})(o_j)$$

$$\Delta w_{ji} = -\alpha \frac{\partial E_d}{\partial w_{ji}} \rightarrow \alpha(t_j - o_j)\left(\frac{d}{do_j}(\text{ReLU})(o_j)\right)$$

1.2 Gradient Descent

Derive a gradient descent training rule for a single unit neuron with output o, defined as: $o = w_0 + w_1(x_1 + x_1^2) + \dots + w_n(x_n + x_n^2)$

where $x_1, x_2, \dots, x_n$ are the inputs, $w_1, w_2, \dots w_n$ are the corresponding weights

Error $= \frac{1}{2}\Sigma(t-o)^2$ Gradient descent: $\frac{\partial}{\partial w_i}\left(\frac{1}{2}\Sigma(t-o)^2\right)$

$$\frac{\partial E}{\partial w_i} = \Sigma(t-o)\frac{\partial}{\partial w_i}(t-o)$$

$$= \sum (t-o)\frac{\partial}{\partial w_i}\left(t - w_0 - \sum_{i=1}^{n} w_i(x_i + x_i^2)\right)$$

$$= \sum_{i=1}^{n}(t-o)(-x_i - x_i^2)$$

1.3. Comparing Activation function

(a). Write down the output the neural net y_5 in terms of weights, inputs and a general activation function $h(x)$.

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

(b). Input layer: x

Hidden layer: $h(w^{(1)}(x))$

Output layer: $h(w^{(2)}h(w^{(1)}(x)))$

$y_5 = h(w^{(2)}h(w^{(1)}(x)))$

(c) Sig $S = \frac{1}{1+e^{-x}}$

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

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

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

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

$$= 2S(-2x) - 1$$

We can show that neural nets created using the above two activation functions can generate the same function, with the parameter differing only by linear transformations and constants.