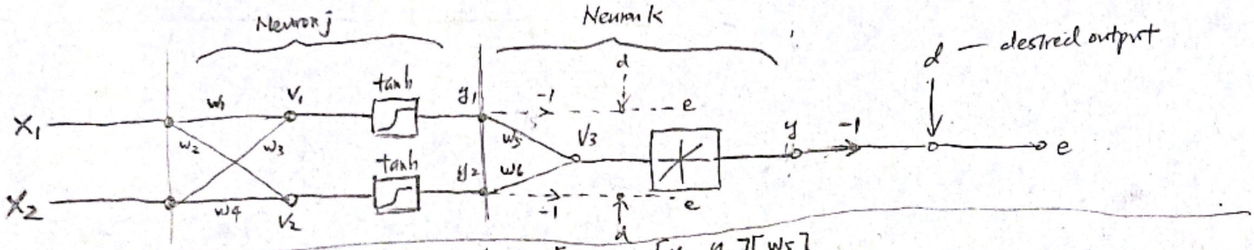
- Let j be an output neuron and let $e_j^{\xi}(n) = d_j^{\xi}(n) - y_j^{\xi}(n)$
- Let $\mathcal{E}_j^{\xi}(n) = \frac{1}{2}(e_j^{\xi}(n))^2$ – an instantaneous error cost
- Let $\mathcal{E}^{\xi}(n) = \sum_j \mathcal{E}_j^{\xi}(n)$ – a total instantaneous error cost
- Let $\mathcal{E}(n) = \frac{1}{N}\sum_{\xi=1}^{N} \mathcal{E}^{\xi}(n) = \sum_{\xi=1}^{1} \mathcal{E}^{\xi}(n) = \underline{\mathcal{E}^{\xi}(n)}$ – an empirical risk / persample error cost

n = iteration
ξ = data sample
N = batch size

✓ $\underline{N = 1}$



$$\begin{cases} [v_1\ v_2] = [x_1\ x_2]\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \\ [y_1\ y_2] = \tanh([v_1\ v_2]) \end{cases}$$

$$\begin{cases} v_3 = [y_1\ y_2]\begin{bmatrix} w_5 \\ w_6 \end{bmatrix} \\ y = v_3 \\ e = d - y \end{cases}$$

estimate value

desire value

i = input neuron, $\underline{j = \text{output layer neuron}}$

→ To update an output neuron weight w_{ji}

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$$

$\underline{\text{chain rule}}$:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$e_j(n)$ -1 $\tanh'(v_j(n))$ $x_i(n)$

$\underline{\text{Let}}$ $\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \boxed{e_j(n)\tanh'(v_j(n))}$, Then $\boxed{\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \eta\,\delta_j(n)\,x_i(n)}$

$\underline{\text{local gradient}}$

(i = "input" neuron)

Now, this case with an output neuron k connected to a $\underline{\text{hidden neuron } j}$:

To update weight $w_{ji}(n)$ of a neuron in a hidden layer

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \longrightarrow \text{chain rule: } \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$= \frac{\partial \mathcal{E}(n)}{\partial y_j(n)}\left(\tanh'(v_j(n))\,x_i(n)\right)$$

$$\text{Let } \delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)}\tanh'(v_j(n))$$

To be computed, K output neurons are installed to generalize

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

$$\text{Then } \Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \eta\,\delta_j(n)\,x_i(n)$$

local gradient

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

since
$e_k(n) = d_k(n) - v_k(n)$
$= d_k(n) - y_k(n)$

$$\frac{\partial e_k(n)}{\partial V_k(n)} = -y_k'(n) \qquad \text{since } V_k(n) = \sum_{j=0}^{m} w_{kj}(n)\, y_j(n)$$

$$= -V_k'(n)$$

$$\text{and} \quad \frac{\partial V_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

$$\Rightarrow \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = -\sum_k \underbrace{e_k(n)\overset{=1,\ \text{since } V_k = y_k \text{ (Linear)}}{\boxed{V_k'(n)}}}_{\delta_k(n)} w_{kj}(n) = -\sum_k \delta_k(n)\, w_{kj}(n)$$

Therefore $\boxed{\delta_j(n) = \tanh'(V_j(n)) \sum_k \delta_k(n)\, w_{kj}(n)}$

$$\therefore \boxed{\begin{cases} \Delta w_{ji}(n) = \eta\, \delta_j(n)\, x_i(n) \\ \Delta w_{kj}(n) = \eta\, \delta_k(n) \end{cases}}$$

- If j is an output layer neuron:

$$\boxed{\delta_j(n) = e_j(n) \tanh'(V_j(n))}$$

- If j is a hidden layer neuron:

$$\boxed{\delta_j(n) = \tanh'(V_j(n)) \sum_k \delta_k(n)\, w_{kj}(n)}$$

※ derivative of $\tanh(x)$
$= 1 - \tanh^2(x) = (1+\tanh(x)) \times (1-\tanh(x))$

The weight update equations

$$\frac{\Delta w_{ji}(n)}{= \eta\, \delta_j(n)\, x_i(n)} \begin{cases} \Delta w_1(n) = \eta(d(n)-y(n))\, w_5(n)[1-y_1(n)][1+y_1(n)]\, x_1 \\ \Delta w_2(n) = \eta(d(n)-y(n))\, w_6(n)[1-y_2(n)][1+y_2(n)]\, x_1 \\ \Delta w_3(n) = \eta(d(n)-y(n))\, w_5(n)[1-y_1(n)][1+y_1(n)]\, x_2 \\ \Delta w_4(n) = \eta\underbrace{(d(n)-y(n))\, w_6(n)[1-y_2(n)][1+y_2(n)]}_{\delta_j(n)}\, x_2 \end{cases}$$

$$\frac{\Delta w_{kj}(n)}{= \eta\, \delta_k(n)} \begin{cases} \Delta w_5(n) = \eta(d(n)-y(n))\, y_1 \\ \Delta w_6(n) = \eta(d(n)-y(n))\, y_2 \end{cases}$$

$\boxed{n=1}$ $x' = (0.5, 0.7)$, $y' = 0.3$ ← desire value

$w_1(0) = -0.02$, $w_2(0) = -0.05$, $w_3(0) = 0.01$, $w_4(0) = 0.1$, $w_5(0) = 0.2$, $w_6(0) = -0.05$

$\eta = 0.1$ (learning rate)

$$[V_1\ V_2] = [0.5\ 0.7]\begin{bmatrix} -0.02 & -0.05 \\ 0.01 & 0.1 \end{bmatrix} = [-0.003\ 0.045]$$

$$[y_1\ y_2] = [\tanh(-0.003)\ \tanh(0.045)] = [-0.003\ 0.045]$$

$$V_3 = y = [-0.003\ 0.045]\begin{bmatrix} 0.2 \\ -0.05 \end{bmatrix} = -0.0029$$ ← estimate value

$$\Rightarrow \begin{cases} \Delta w_1(0) = (0.1)(0.3+0.0029)(0.2)(1-(-0.003))(1+(-0.003))(0.5) = 0.003 \\ \Delta w_2(0) = (0.1)(0.3+0.0029)(-0.05)(1-0.045)(1+0.045)(0.5) = -0.00075572 \\ \Delta w_3(0) = (0.1)(0.3+0.0029)(0.2)(1-(-0.003))(1+(-0.003))(0.7) = 0.0042 \\ \Delta w_4(0) = (0.1)(0.3+0.0029)(-0.05)(1-0.045)(1+0.045)(0.7) = -0.0011 \\ \Delta w_5(0) = (0.1)(0.3+0.0029)(-0.003) = -0.00009087 \\ \Delta w_6(0) = (0.1)(0.3+0.0029)(0.045) = 0.0014 \end{cases}$$

$\boxed{n=1}$

$$\Rightarrow \begin{cases} w_1(1) = w_1(0) + \Delta w_1(0) = -0.017 \\ w_2(1) = w_2(0) + \Delta w_2(0) = -0.05075572 \\ w_3(1) = w_3(0) + \Delta w_3(0) = 0.0142 \\ w_4(1) = w_4(0) + \Delta w_4(0) = 0.0989 \\ w_5(1) = w_5(0) + \Delta w_5(0) = 0.1999 \\ w_6(1) = w_6(0) + \Delta w_6(0) = -0.0486 \end{cases}$$

\#

$\boxed{n=2}$ $x^2 = (-0.2, 0.1)$, $y^2 = 0.1$

$$[v_1\ v_2] = [-0.2\ 0.1]\begin{bmatrix} -0.017 & -0.05075572 \\ 0.0142 & 0.0989 \end{bmatrix} = [0.0048\ 0.02]$$

$$[y_1\ y_2] = [\tanh(0.0048)\ \tanh(0.02)] = [0.0048\ 0.02]$$

$$v_3 = y = [0.0048\ 0.02]\begin{bmatrix} 0.1999 \\ -0.0486 \end{bmatrix} = \underline{-0.00001248}$$

$$\Rightarrow \begin{cases} \Delta w_1(1) = (0.1)(0.1-(-0.00001248))(0.1999)(1-0.0048)(1+0.0048)(-0.2) = -0.00039984 \\ \Delta w_2(1) = (0.1)(0.1-(-0.00001248))(-0.0486)(1-0.02)(1+0.02)(-0.2) = 0.000097173 \\ \Delta w_3(1) = (0.1)(0.1-(-0.00001248))(0.1999)(1-0.0048)(1+0.0048)(0.1) = 0.00019992 \\ \Delta w_4(1) = (0.1)(0.1-(-0.00001248))(-0.0486)(1-0.02)(1+0.02)(0.1) = -0.000048587 \\ \Delta w_5(1) = (0.1)(0.1-(-0.00001248))(0.0048) = 0.000048006 \\ \Delta w_6(1) = (0.1)(0.1-(-0.00001248))(0.02) = 0.00020002 \end{cases}$$

$$\Rightarrow \begin{cases} w_1(2) = w_1(1) + \Delta w_1(1) = -0.01739984 \\ w_2(2) = w_2(1) + \Delta w_2(1) = -0.050658547 \\ w_3(2) = w_3(1) + \Delta w_3(1) = 0.01439992 \\ w_4(2) = w_4(1) + \Delta w_4(1) = 0.098851413 \\ w_5(2) = w_5(1) + \Delta w_5(1) = 0.199957136 \\ w_6(2) = w_6(1) + \Delta w_6(1) = -0.04839998 \end{cases}$$

\#

$\boxed{n=3}$ $x' = (0.5, 0.7)$, $y' = 0.3$

$$[v_1\ v_2] = [0.5\ 0.7]\begin{bmatrix} -0.01739984 & -0.050658547 \\ 0.01439992 & 0.098851413 \end{bmatrix} = [0.0014\ 0.0439]$$

$$[y_1\ y_2] = [\tanh(0.0014)\ \tanh(0.0439)] = [0.0014\ 0.0439]$$

$$v_3 = y = [0.0014\ 0.0439]\begin{bmatrix} 0.199957136 \\ -0.04839998 \end{bmatrix} = \underline{-0.0018}$$

$$\Rightarrow \begin{cases} \Delta w_1(2) = (0.1)(0.3-(-0.0018))(0.199957136)(1-0.0014)(1+0.0014)(0.5) = 0.0030173 \\ \Delta w_2(2) = (0.1)(0.3-(-0.0018))(-0.04839998)(1-0.0439)(1+0.0439)(0.5) = -0.0007289 \\ \Delta w_3(2) = (0.1)(0.3-(-0.0018))(0.199957136)(1-0.0014)(1+0.0014)(0.7) = 0.0042243 \\ \Delta w_4(2) = (0.1)(0.3-(-0.0018))(-0.04839998)(1-0.0439)(1+0.0439)(0.7) = -0.0010205 \\ \Delta w_5(2) = (0.1)(0.3-(-0.0018))(0.0014) = 0.000042252 \\ \Delta w_6(2) = (0.1)(0.3-(-0.0018))(0.0439) = 0.0013249 \end{cases}$$

$\boxed{n=3}$

$$\Rightarrow \boxed{\begin{cases} w_1(3) = w_1(2) + \Delta w_1(2) = -0.01438254 \\ w_2(3) = w_2(2) + \Delta w_2(2) = -0.051387447 \\ w_3(3) = w_3(2) + \Delta w_3(2) = 0.01862422 \\ w_4(3) = w_4(2) + \Delta w_4(2) = 0.097830913 \\ w_5(3) = w_5(2) + \Delta w_5(2) = 0.199999388 \\ w_6(3) = w_6(2) + \Delta w_6(2) = -0.04707508 \end{cases}} \#$$

$\boxed{n=4}$ $\quad x^2 = (-0.2, 0.1)$, $y^2 = 0.1$

$$[v_1 \ v_2] = [-0.2 \ \ 0.1]\begin{bmatrix} -0.01438254 & -0.051387447 \\ 0.01862422 & 0.097830913 \end{bmatrix} = [0.0047 \ \ 0.0201]$$

$$[y_1 \ y_2] = [\tanh(0.0047) \ \ \tanh(0.0201)] = [0.0047 \ \ 0.0201]$$

$$V_3 = y = [0.0047 \ \ 0.0201]\begin{bmatrix} 0.199999388 \\ -0.04707508 \end{bmatrix} = \underline{-0.000006212}$$

$$\Rightarrow \begin{cases} \Delta w_1(3) = (0.1)(0.1-(-0.000006212))(0.199999388)(1-0.0047)(1+0.0047)(-0.2) = -0.0004 \\ \Delta w_2(3) = (0.1)(0.1-(-0.000006212))(-0.04707508)(1-0.0201)(1+0.0201)(-0.2) = 0.000094118 \\ \Delta w_3(3) = (0.1)(0.1-(-0.000006212))(0.199999388)(1-0.0047)(1+0.0047)(0.1) = 0.0002 \\ \Delta w_4(3) = (0.1)(0.1-(-0.000006212))(-0.04707508)(1-0.0201)(1+0.0201)(0.1) = -0.000047059 \\ \Delta w_5(3) = (0.1)(0.1-(-0.000006212))(0.0047) = 0.0000470029 \\ \Delta w_6(3) = (0.1)(0.1-(-0.000006212))(0.0201) = 0.00020101 \end{cases}$$

$$\Rightarrow \boxed{\begin{cases} w_1(4) = w_1(3) + \Delta w_1(3) = -0.01478254 \\ w_2(4) = w_2(3) + \Delta w_2(3) = -0.051293329 \\ w_3(4) = w_3(3) + \Delta w_3(3) = 0.01882422 \\ w_4(4) = w_4(3) + \Delta w_4(3) = 0.097783854 \\ w_5(4) = w_5(3) + \Delta w_5(3) = 0.200046391 \\ w_6(4) = w_6(3) + \Delta w_6(3) = -0.04687407 \end{cases}} \#$$