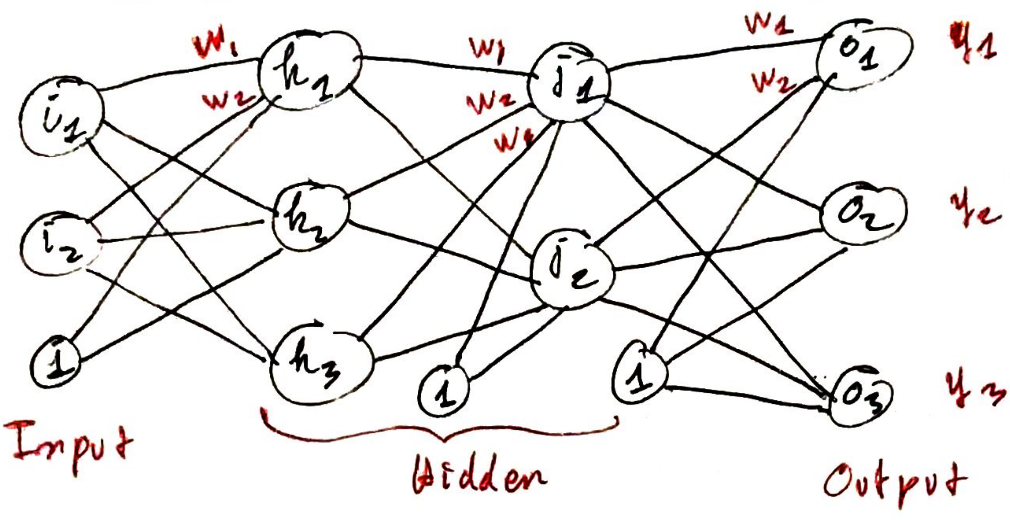
#: Neural Net - Forward/Back Propagation



1 training data point

→ Forward:

* $net_{h_1} = w_1 i_1 + w_2 i_2 + w_{bias}$

activation: $h_1 = \text{sigmoid}(net_{h_1})$

* net_{j_1} · $h_2 + w_3 h_3$
  ↳ 1
* net_{o_1} · j_2
  ↳ 1

$$\boxed{\text{Total Err} = \sum_{i=1}^{3} \frac{1}{2}(y_i - o_i)^2} = E_1 + E_2 + E_3$$

→ Backward Pass:

For w_1 at o_1:

$$\frac{\partial Err}{\partial w_1} = \frac{\partial E_1}{\partial w_1} = \underbrace{\frac{\partial E_1}{\partial o_1} * \frac{\partial o_1}{\partial net_{o_1}}}_{\delta_{o_1}} * \frac{\partial net_{o_1}}{\partial w_1}$$

← just 1 error

$$\frac{\partial E_1}{\partial o_1} = \left[\frac{1}{2}(y_1 - o_1)^2\right]' = -(y_1 - o_1)$$

$$\boxed{\frac{\partial o_1}{\partial net_{o_1}} = \text{sigmoid}' = o_1(1 - o_1)}$$

$$\boxed{\Rightarrow \delta_{o_1} = \frac{\partial Err}{\partial net_{o_1}} = -(y_1 - o_1) * o_1(1 - o_1)}$$

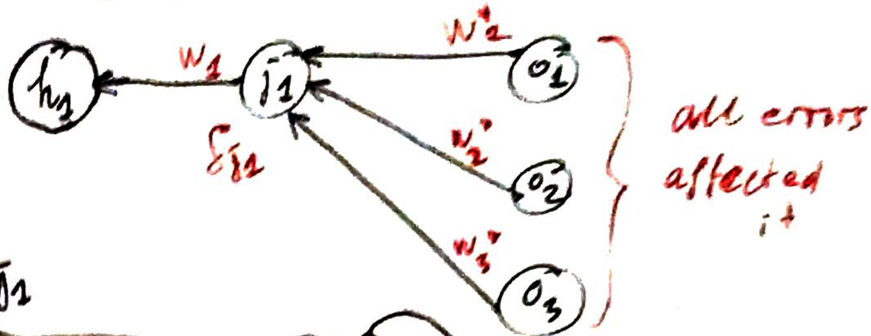
sigmoid'

$$\frac{\partial net_{o_1}}{\partial w_1} = (w_1 j_1 + w_2 j_2 + bias)' = j_1$$

$$\Rightarrow \frac{\partial Err}{\partial w_1} = \delta_{o_1} * j_1 \qquad \boxed{+\Delta w_1 = \mu * \delta_{o_1} * j_1}$$

for w_1 at o_1

- For w_1 @ j_1 (Hidden layer)



$$\frac{\partial Err}{\partial w_1} = \overbrace{\underbrace{\frac{\partial Err}{\partial j_1}}\cdot\frac{\partial j_1}{\partial net_{j_1}}}^{\delta_{j_1}} * \underbrace{\frac{\partial net_{j_1}}{\partial w_1}}_{= h_1}$$

$$\hookrightarrow = \frac{\partial E_1}{\partial j_1} + \frac{\partial E_2}{\partial j_1} + \frac{\partial E_3}{\partial j_1}$$

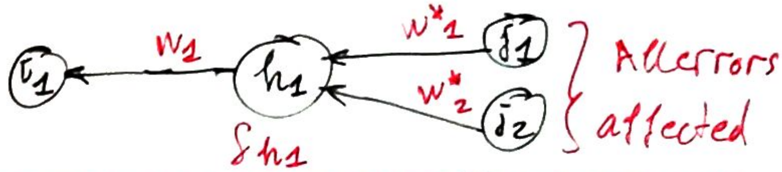
$$\hookrightarrow \frac{\partial E_1}{\partial j_1} = \frac{\partial E_1}{\partial net_{o_1}} * \frac{\partial net_{o_1}}{\partial j_1} = \boxed{\delta_{o_1} * w^*_1}$$

$$\boxed{\Rightarrow \frac{\partial Err}{\partial j_1} = \delta_{o_1} w^*_1 + \delta_{o_2} w^*_2 + \delta_{o_3} w^*_3}$$

$$\boxed{\frac{\partial j_1}{\partial net_{j_1}} = sigmoid' = j_1(1-j_1)}$$

$$\boxed{\begin{aligned} \Rightarrow \delta_{j_1} &= (\delta_{o_1} w^*_1 + \delta_{o_2} w^*_2 + \delta_{o_3} w^*_3) * \underbrace{j_1(1-j_1)}_{sigmoid'} \\ &= \frac{\partial Err}{\partial net_{j_1}} \end{aligned}}$$

$$\Rightarrow \frac{\partial Err}{\partial w_1} = \delta_{j_1} * h_1 \qquad \boxed{+\Delta w_1 = \mu\, \delta_{j_1} * h_1}$$

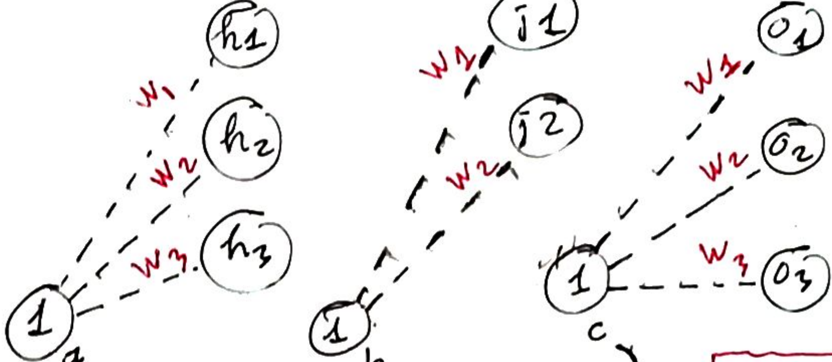


For w_1 @ h_1 :

$$\boxed{\Rightarrow \delta h_1 = (\delta j_1 w^*_1 + \delta j_2 w^*_2) * \underbrace{h_1(1-h_1)}_{\text{sigmoid}'} \qquad = \frac{\partial Err}{\partial net h_1}}$$

$$\Rightarrow \frac{\partial Err}{\partial w_1} = \delta h_1 * i_1 \qquad \boxed{+\Delta w_1 = \mu \, \delta h_1 * i_1}$$

??? Bias weights update at each epoch (training iteration)



$$\Delta w_3 = \mu * \delta o_3$$

$$\boxed{+\Delta w_2 = \mu * \delta o_2 * 1}$$

$$\boxed{+\Delta w_1 = \mu * \delta j_1}$$

$$\boxed{+\Delta w_3 = \mu * \delta h_3}$$

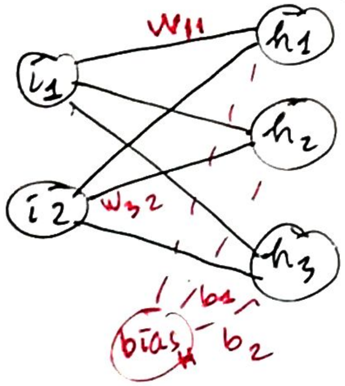
Very easy & similar!

Forward / Back Propagation - Matrix Algebra

$$\underset{\text{2 neurons}}{I} \longrightarrow \underset{3}{H} \longrightarrow \underset{2}{J} \longrightarrow \underset{3}{O}$$

4 training examples

⇒ Forward $I \rightarrow H$



$$\vec{I} = \begin{bmatrix} i^1_1 & i^1_2 \\ i^2_1 & i^2_2 \\ i^3_1 & i^3_2 \\ i^4_1 & i^4_2 \end{bmatrix}$$

4 × 2

data # neuron #

$$\underset{\downarrow}{W_{HI}} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix}$$

2 by 3

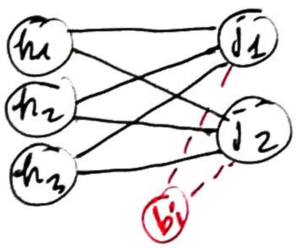
neuron # (prev layer) neuron #

$$\vec{bias}_H = \underbrace{[b_1 \quad b_2 \quad b_3]}_{\text{neuron \#}}$$

$$\underbrace{\underbrace{I \text{ dot } W_{HI}}_{4 \text{ by } 3} + \underbrace{\vec{bias}_H}_{\text{by } 3}}_{4 \text{ by } 3} \xrightarrow[\text{sigmoid}]{\text{activation}} \begin{bmatrix} h^1_1 & h^1_2 & h^1_3 \\ h^2_1 & h^2_2 & h^2_3 \\ h^3_1 & h^3_2 & h^3_3 \\ h^4_1 & h^4_2 & h^4_3 \end{bmatrix}$$

$H = \boxed{\text{4 by 3} \quad \text{data \#} \quad \text{neuron \# } H}$

⇒ Forward $H \rightarrow J$



H = 4 by 3

data # neuron #

W_{JH} = 3 by 2

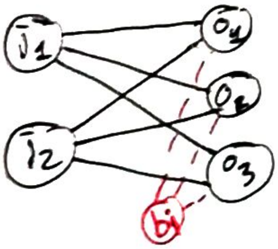
neuron # (prev layer) neuron #

$$\vec{bias}_J = \underbrace{[b_1 \quad b_2]}_{\text{neuron \#}}$$

$$\underbrace{\underbrace{H \cdot W_{JH}}_{4 \times 2} + \underbrace{\vec{bias}_J}_{\text{by } 2}}_{4 \text{ by } 2} \xrightarrow[\text{sigmoid}]{\text{activation}} \boxed{\text{4 by 2} \quad \text{data \#} \quad \text{neuron \# } J}$$

J =

⇒ Forward $J \rightarrow O$



$J = \underset{\text{data \#}}{4}$ by $\underset{\text{neron \#}}{2}$

$W_{OJ} = \underset{\text{neuron \# (prev layer)}}{2}$ by $\underset{\text{neron \#}}{3}$

$\vec{bias}_o = \underbrace{[b_1 \quad b_2 \quad b_3]}_{\text{neuron \#}}$

$\underbrace{\underbrace{J \cdot W_{OJ}}_{4\times 3} + \underbrace{\vec{bias}_o}_{\text{by}3}}_{4\times 3} \xrightarrow[\text{sigmoid}]{\text{activate}}$ $O = \boxed{\underset{\text{data \#}}{4} \text{ by } \underset{\text{neuron \# } O}{3}}$

$I \rightarrow H \rightarrow J \rightarrow O$ -vs- Y

2 neurons, 3, 2, 3

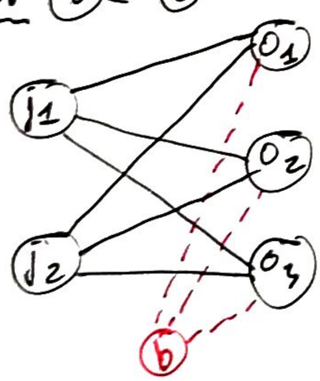
4 training examples

~~Each~~ neuron Err $= \begin{bmatrix} E_1^1 & E_2^1 & E_3^1 \\ E_1^2 & E_2^2 & E_3^2 \\ E_1^3 & E_2^3 & E_3^3 \\ E_1^4 & E_2^4 & E_3^4 \end{bmatrix}$ $\underset{\text{data \#}}{4}$ by $\underset{\text{neuron \# } O}{3}$

~~Sum~~ Err $= \begin{bmatrix} E_1^1 + E_2^1 + E_3^1 \\ E_1^2 + E_2^2 + E_3^2 \\ E_1^3 + E_2^3 + E_3^3 \\ E_1^4 + E_2^4 + E_3^4 \end{bmatrix}$ $\underset{\text{data \#}}{4}$ by 1

Avg Sum Err $= \frac{1}{4}(E^1 + E^2 + E^3 + E^4)$

⇒ δ for $J \leftarrow O$



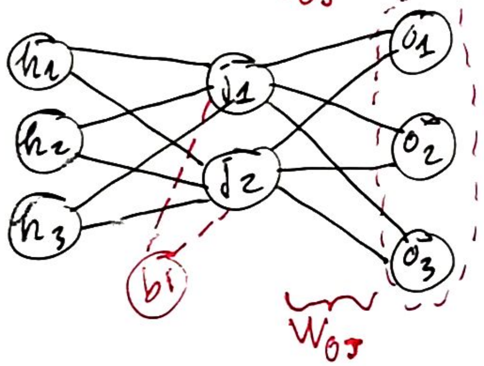
$\delta o = \begin{bmatrix} \delta o_1^1 & \delta o_2^1 & \delta o_3^1 \\ \delta o_1^2 & \delta o_2^2 & \delta o_3^2 \\ \delta o_1^3 & \delta o_2^3 & \delta o_3^3 \\ \delta o_1^4 & \delta o_2^4 & \delta o_3^4 \end{bmatrix}$ $\underset{\text{data \#}}{4}$ by $\underset{\text{neuron \# } O}{3}$

$Y = \underset{\text{data \#}}{4}$ by $\underset{\text{neuron \#}}{3}$

$O = \underset{\text{data \#}}{4}$ by $\underset{\text{neuron \#}}{3}$

$$\boxed{\delta_O = \underbrace{(Y - O)}_{4 \times 3} * \underbrace{\{O * (1 - O)\}}_{4 \times 3}} = \underset{\text{data \#}}{4} \text{ by } \underset{\text{neuron \# } O}{3}$$

$\Rightarrow$ δ for $H \leftarrow J \xleftarrow{W_{OJ}} O$ δ_O



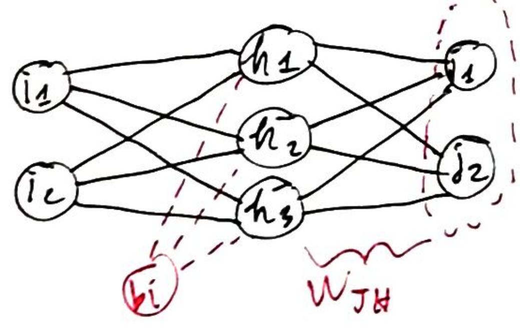
$$W_{OJ} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix}$$

$\underset{\text{neuron \# } J}{2} \times \underset{\text{neuron \# } O}{3}$

$\delta_O = \underset{\text{data \#}}{4} \times \underset{\text{neuron \# } O}{3}$

$$\boxed{\delta_J = \underbrace{\delta_O \cdot W_{OJ}^{+}}_{4 \times 2} * \underbrace{\{J * (1 - J)\}}_{4 \times 2}} = \underset{\text{data \#}}{4} \text{ by } \underset{\text{neuron \# } J}{2}$$

$\Rightarrow$ δ for $I \leftarrow H \xleftarrow{W_{JH}} J$ δ_J



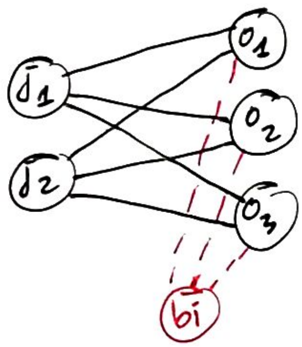
$$W_{JH} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix}$$

$\underset{\text{neuron \# } H}{3} \times \underset{\text{neuron \# } J}{2}$

$\delta_J = \underset{\text{data \#}}{4} \text{ by } \underset{\text{neuron \# } J}{2}$

$$\boxed{\delta_H = \underbrace{\delta_J \cdot W_{JH}^{+}}_{4 \times 3} * \underbrace{\{H * (1 - H)\}}_{4 \times 3}} = \underset{\text{data \#}}{4} \text{ by } \underset{\text{neuron \# } H}{3}$$

⇒ Update weights for O



$$W_{OJ} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix} \qquad \vec{bias}_o = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

2×3 — 2: J#, 3: O# (neuron #)

$\delta o = 4 \times 3$ — 4: data#, 3: O#

$J = 4 \times 2$ — 4: data#, 2: J#

$$\boxed{J^T \cdot \delta_o} = (2 \text{ by } 4) \cdot (4 \text{ by } 3) = \boxed{2 \text{ by } 3}$$

$$\begin{bmatrix} j_1^1 & j_1^2 & j_1^3 & j_1^4 \\ j_2^1 & j_2^2 & j_2^3 & j_2^4 \end{bmatrix} \qquad \begin{bmatrix} \delta o_1^1 & \delta o_2^1 & \delta o_3^1 \\ \delta o_1^2 & \delta o_2^2 & \delta o_3^2 \\ \delta o_1^3 & \delta o_2^3 & \delta o_3^3 \\ \delta o_1^4 & \delta o_2^4 & \delta o_3^4 \end{bmatrix}$$

2 by 4 — 2: J#, 4: data#

4 by 3 — 4: data#, 3: neuron O#

$\rightarrow$ 1 slice $= j_1^1 \, \delta o_1^1 + j_1^2 \, \delta o_1^2 + j_1^3 \, \delta o_1^3 + j_1^4 \, \delta o_1^4$

- → Across all 4 training examples
- → Have to avg by training size

M = 4 training size

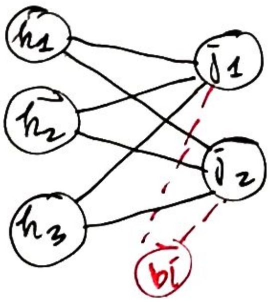
$$\Rightarrow \boxed{W_{OJ} \mathrel{+}= \frac{1}{M} * \mu * (J^T \cdot \delta_o)} = 2 \times 3$$

$$\boxed{bias_o \mathrel{+}= \frac{1}{M} * \mu * (\vec{1}_M \cdot \delta o)}$$

$[1, 1, 1, 1]$ → by 4 / M = training size

by 3

⇒ Update weights for (J)



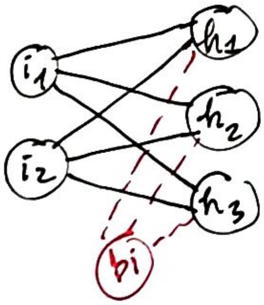
$$W_{JH} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \to \underset{H\#}{3} \times \underset{J\#}{2} \qquad \vec{bias}_J = \begin{bmatrix} b_1 & b_2 \end{bmatrix}$$

$$\delta_J = \underset{data\,\#}{4} \times \underset{J\#}{2} \qquad H = \underset{data\,\#}{4} \times \underset{H\#}{3}$$

$$\boxed{W_{JH} \mathrel{+}= \frac{1}{M} * \mu * (\underbrace{H^T \cdot \delta_J}_{3 \times 2})}$$

$$\boxed{\vec{bias}_J \mathrel{+}= \frac{1}{M} * \mu * (\vec{1}_M \cdot \delta_J)}$$

⇒ Update weights for (H)



$$W_{HI} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix} \to \underset{I\#}{2} \times \underset{H\#}{3} \qquad \vec{bias}_H = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$$

$$\delta_H = \underset{data\,\#}{4} \times \underset{H\#}{3} \qquad I = \underset{data\,\#}{4} \times \underset{I\#}{2}$$

$$\boxed{W_{HI} \mathrel{+}= \frac{1}{M} * \mu * (\underbrace{I^T \cdot \delta_H}_{2 \times 3})}$$

$$\boxed{\vec{bias}_H \mathrel{+}= \frac{1}{M} * \mu * (\vec{1}_M \cdot \delta_H)}$$