Finally we can compute the partial derivative of the error with respect to the weight w_{lj} as:

$$\frac{\partial J}{\partial w_{lj}} = -\delta_l u_{lj}$$

Update Rule for Output Units

$$w_{lj} \leftarrow w_{lj} + \eta \delta_l u_{lj}$$

where $\delta_l = (y_l - o_l) o_l (1 - o_l)$.

- *Question:* What is u_{lj} for the l^{th} output node?
- u_{lj} is the j^{th} input to l^{th} output node, which will be the output coming from the j^{th} hidden node.

Observation 3
net_j for a **hidden node** is connected to J through all output nodes

$$\frac{\partial J}{\partial net_j} = \sum_{l=1}^{k} \frac{\partial J}{\partial net_l} \frac{\partial net_l}{\partial net_j}$$

Remember that we have already computed the first term on the right hand side for output nodes:

$$\frac{\partial J}{\partial net_l} = -\delta_l$$

where $\delta_l = (y_l - o_l) o_l (1 - o_l)$. This result gives us:

$$\begin{aligned}
\frac{\partial J}{\partial net_j} &= \sum_{l=1}^{k} -\delta_l \frac{\partial net_l}{\partial net_j} \\
&= \sum_{l=1}^{k} -\delta_l \frac{\partial net_l}{\partial o_j} \frac{\partial o_j}{\partial net_j} \\
&= \sum_{l=1}^{k} -\delta_l w_{lj} \frac{\partial o_j}{\partial net_j} \\
&= \sum_{l=1}^{k} -\delta_l w_{lj} o_j (1 - o_j) \\
&= -o_j (1 - o_j) \sum_{l=1}^{k} \delta_l w_{lj}
\end{aligned}$$

Thus, the gradient becomes:

$$\begin{aligned}\frac{\partial J}{\partial w_{jp}} &= \frac{\partial J}{\partial net_j} u_{jp} \\ &= -o_j(1-o_j)(\sum_{l=1}^{k} \delta_l w_{lj}) u_{jp} \\ &= -\delta_j u_{jp}\end{aligned}$$

Update Rule for Hidden Units

$$w_{jp} \leftarrow w_{jp} + \eta \delta_j u_{jp}$$

$$\delta_j = o_j(1-o_j) \sum_{l=1}^{k} \delta_l w_{lj}$$

$$\delta_l = (y_l - o_l) o_l (1-o_l)$$

- *Question:* What is u_{jp} for the j^{th} hidden node?
- u_{jp} is the p^{th} input to j^{th} hidden node, which will be p^{th} attribute value for the input, i.e., x_p.

5 Final Algorithm

- While not converged:
  - *Move forward* to compute outputs at hidden and output nodes
  - *Move backward* to propagate errors back
    - Compute δ errors at output nodes (δ_l)
    - Compute δ errors at hidden nodes (δ_j)
  - Update all weights according to weight update equations