Deep Learning

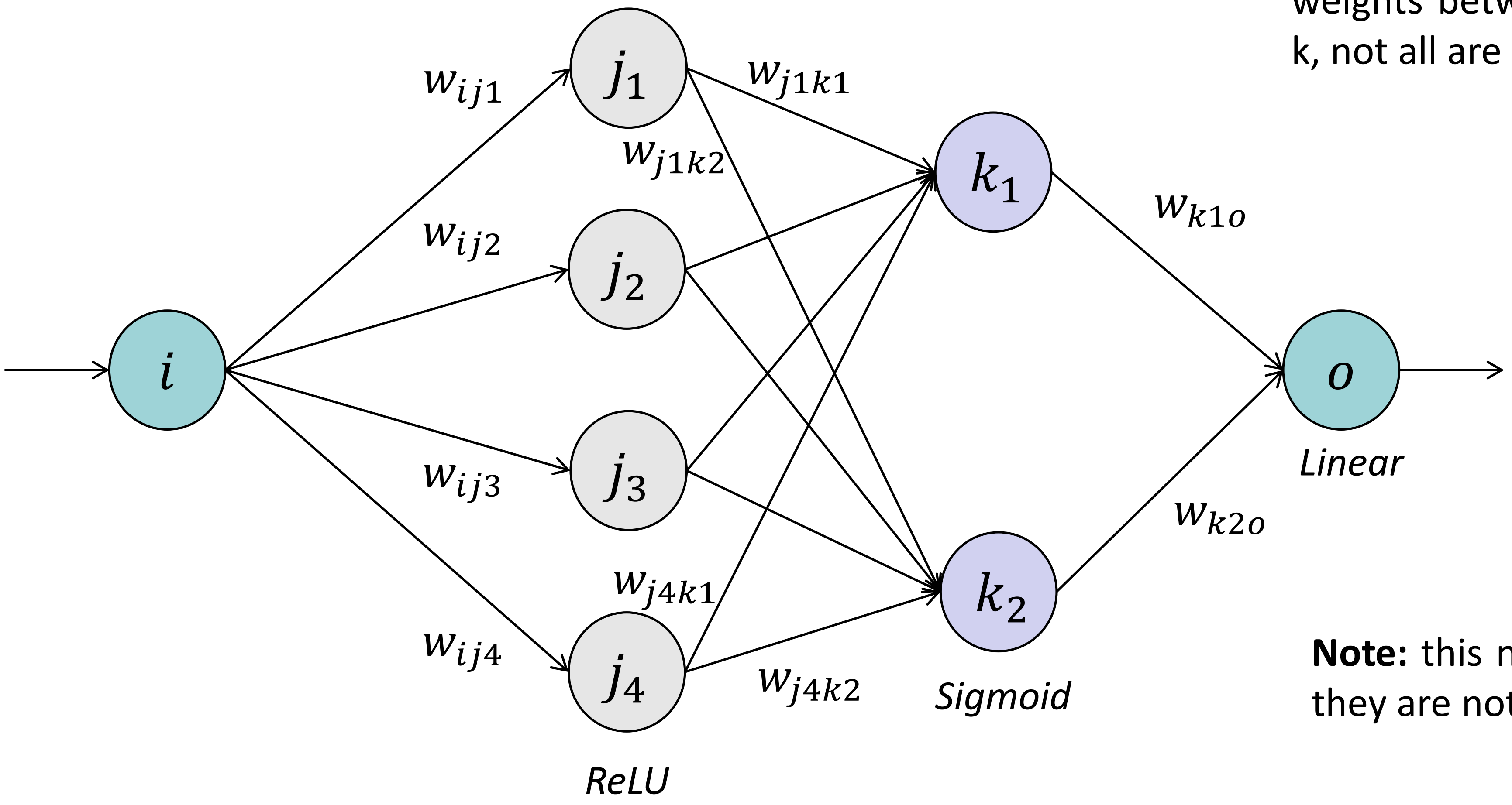
Backpropagation Example

Tuesday 28th January

Dr. Nicholas Cummins

- Network Architecture

Note: there are a total of 8 weights between layers j and k, not all are illustrated



Note: this network has biases, they are not illustrated

Backpropagation Example

- The neural network above consists of 2 hidden layers.
  - The first hidden layer uses ReLU
  - The second hidden layer uses sigmoid
  - The output layer uses linear as the activation function.
- There are a total of 21 parameters to be updated
  - 4 weights and 4 biases between the input layer and the 1st hidden layer
  - 8 weights and 2 biases between the 1st and 2nd hidden layers
  - 2 weights and 1 bias between the 2nd hidden layer and the output layer

Backpropagation Example

- Initial Values

$$input = [2.0]; output = [3.0]$$

$$W_{ij} = \begin{bmatrix} w_{ij_1} & w_{ij_2} & w_{ij_3} & w_{ij_4} \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.75 & 1.0 \end{bmatrix}$$

$$W_{jk} = \begin{bmatrix} w_{j_1k_1} & w_{j_1k_2} \\ w_{j_2k_1} & w_{j_2k_2} \\ w_{j_3k_1} & w_{j_3k_2} \\ w_{j_4k_1} & w_{j_4k_2} \end{bmatrix} = \begin{bmatrix} 1.0 & 0 \\ 0.75 & 0.25 \\ 0.5 & 0.5 \\ 0.25 & 0.75 \end{bmatrix}$$

$$W_{ko} = \begin{bmatrix} w_{k_1o} \\ w_{k_2o} \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$$

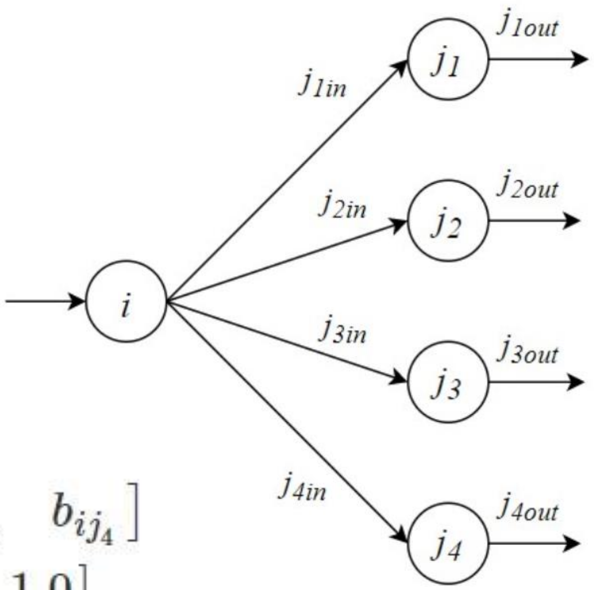
$$b_{ij} = \begin{bmatrix} b_{ij_1} & b_{ij_2} & b_{ij_3} & b_{ij_4} \end{bmatrix} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

$$b_{jk} = \begin{bmatrix} b_{jk_1} & b_{jk_2} \end{bmatrix} = \begin{bmatrix} 1.0 & 1.0 \end{bmatrix}$$

$$b_o = [1.0]$$

Backpropagation Example

- Forward pass (Input -> Hidden Layer 1)
  - Take dot product between inputs and weights
  - Add together result of dot product and bias



$$[j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}] = [input] \times [w_{ij_1} \quad w_{ij_2} \quad w_{ij_3} \quad w_{ij_4}] + [b_{ij_1} \quad b_{ij_2} \quad b_{ij_3} \quad b_{ij_4}]$$

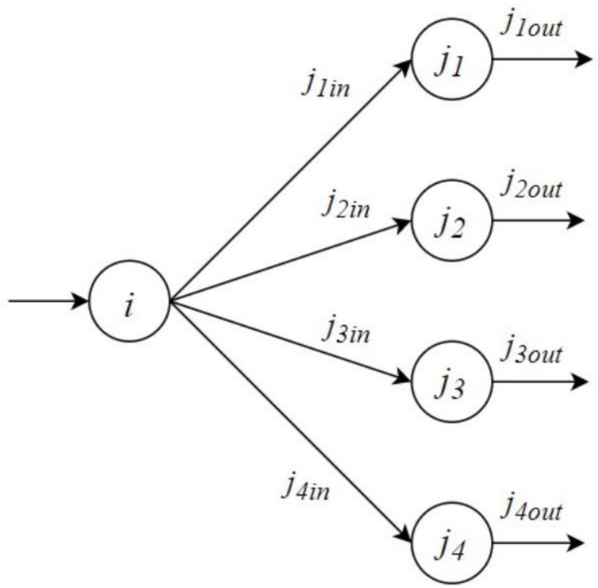
$$[j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}] = [2.0] \times [0.25 \quad 0.5 \quad 0.75 \quad 1.0] + [1.0 \quad 1.0 \quad 1.0 \quad 1.0]$$

$$[j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}] = [2.0] \times [0.25 \quad 0.5 \quad 0.75 \quad 1.0] + [1.0 \quad 1.0 \quad 1.0 \quad 1.0]$$

$$[j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}] = [0.5 \quad 1 \quad 1.5 \quad 2] + [1.0 \quad 1.0 \quad 1.0 \quad 1.0]$$

$$[j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}] = [1.5 \quad 2.0 \quad 2.5 \quad 3.0]$$

- Forward pass (Input -> Hidden Layer 1)
  - Pass result from previous step through the output activation function



$$ReLU([j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}]) = [max(0, j_{1in}) \quad max(0, j_{2in}) \quad max(0, j_{3in}) \quad max(0, j_{4in})]$$

$$ReLU([j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}]) = [max(0, 1.5) \quad max(0, 2.0) \quad max(0, 2.5) \quad max(0, 3.0)]$$

$$ReLU([j_{1in} \quad j_{2in} \quad j_{3in} \quad j_{4in}]) = [1.5 \quad 2.0 \quad 2.5 \quad 3.0]$$

$$[j_{1out} \quad j_{2out} \quad j_{3out} \quad j_{4out}] = [1.5 \quad 2.0 \quad 2.5 \quad 3.0]$$

Backpropagation Example

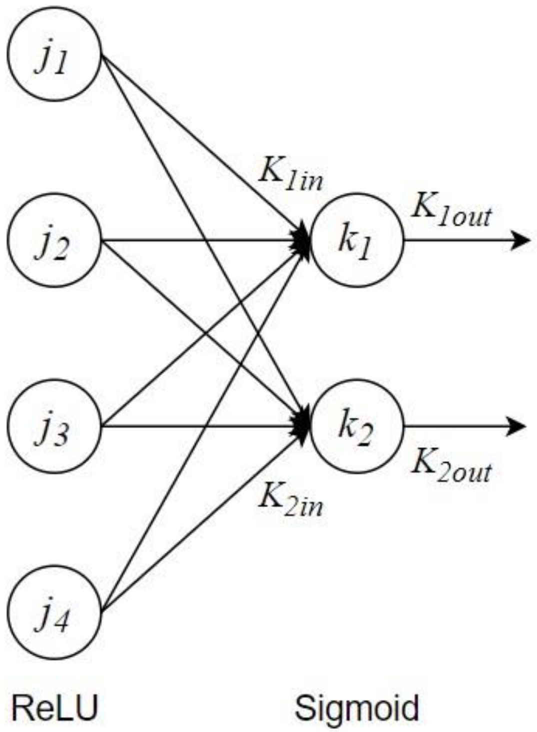
- Forward Pass (Hidden Layer 1 -> Hidden Layer 2)
  - Just as in the previous layer, the output of each neuron in the 1st hidden layer will flow to all neurons in 2nd layer

$$[k_{1in} \quad k_{2in}] = [j_{1out} \quad j_{2out} \quad j_{3out} \quad j_{4out}] \times \begin{bmatrix} w_{j_1k_1} & w_{j_1k_2} \\ w_{j_2k_1} & w_{j_2k_2} \\ w_{j_3k_1} & w_{j_3k_2} \\ w_{j_4k_1} & w_{j_4k_2} \end{bmatrix} + [b_{jk_1} \quad b_{jk_2}]$$

$$[k_{1in} \quad k_{2in}] = [1.5 \quad 2 \quad 2.5 \quad 3.0] \times \begin{bmatrix} 1.0 & 0 \\ 0.75 & 0.25 \\ 0.5 & 0.5 \\ 0.25 & 0.75 \end{bmatrix} + [1.0 \quad 1.0]$$

$$[k_{1in} \quad k_{2in}] = [5.0 \quad 4.0] + [1.0 \quad 1.0]$$

$$[k_{1in} \quad k_{2in}] = [6.0 \quad 5.0]$$



- Forward Pass (Hidden Layer 1 -> Hidden Layer 2)
  - After Sigmoid activation

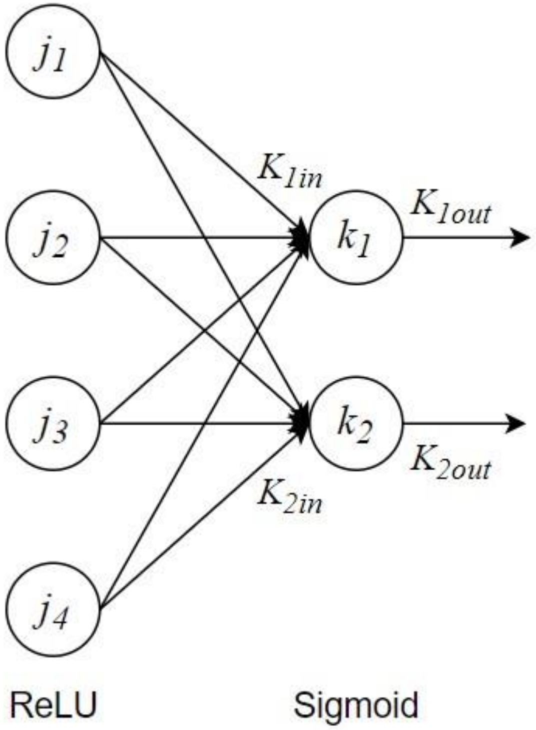
$$Sigmoid => f(x) = \frac{1}{1+e^{-x}}$$

$$Sigmoid([\,k_{1in} \quad k_{2in}\,]) = \left[\frac{1}{1+e^{-k_{1in}}} \quad \frac{1}{1+e^{-k_{2in}}}\right]$$

$$Sigmoid([\,k_{1in} \quad k_{2in}\,]) = \left[\frac{1}{1+e^{-6}} \quad \frac{1}{1+e^{-5}}\right]$$

$$Sigmoid([\,k_{1in} \quad k_{2in}\,]) = [0.9975 \quad 0.9933]$$

$$[\,k_{1out} \quad k_{2out}\,] = [0.9975 \quad 0.9933]$$



Backpropagation Example

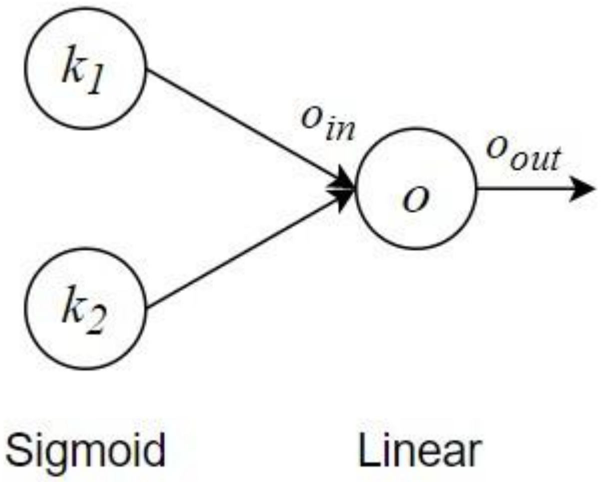
- Forward Pass (Hidden Layer 2 -> Output)
  - Same concept as for previous layers
  - Note linear activation function

$$[o_{in}] = [k_{1out} \quad k_{2out}] \times \begin{bmatrix} w_{k_1 o} \\ w_{k_2 o} \end{bmatrix} + [b_o]$$

$$[o_{in}] = [0.9975 \quad 0.9933] \times \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} + [b_o]$$

$$[o_{in}] = [1.494] + [1.0]$$

$$[o_{in}] = [2.494]$$

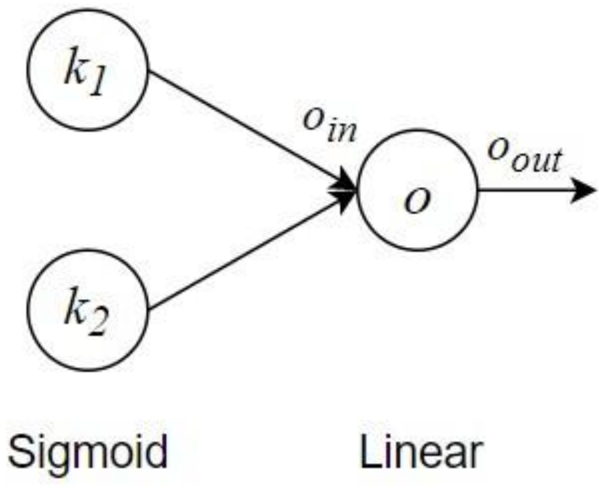


$$Linear => f(x) = x$$

$$Linear([o_{in}]) = [2.494]$$

$$[o_{out}] = [2.494]$$

- Loss function calculation
  - How well did the network perform?



$$Loss = \frac{1}{2}(Prediction - Target)^2$$
$$Loss = \frac{1}{2}(o_{out} - output)^2$$
$$Loss = \frac{1}{2}(2.494 - 3)^2$$
$$Loss = \frac{1}{2}(-0.506)^2$$
$$Loss = \frac{1}{2}(0.256)$$
$$Loss = 0.128$$

$$Linear => f(x) = x$$
$$Linear([o_{in}]) = [2.494]$$
$$[o_{out}] = [2.494]$$

- Important derivatives

ReLU

$$y = max(0, x)$$

$$\frac{\partial y}{\partial x} = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Sigmoid

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

$$\frac{\partial y}{\partial x} = \frac{1}{1 + e^{-x}} \times (1 - \frac{1}{1 + e^{-x}})$$

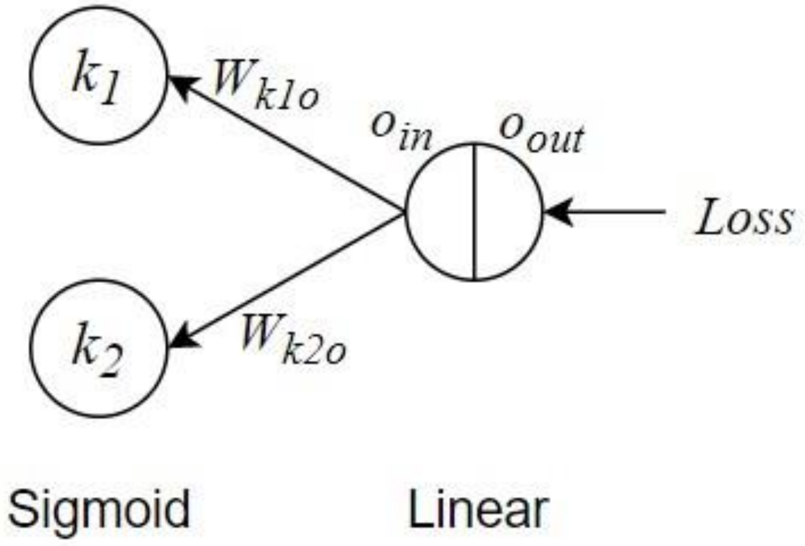
Sigmoid

$$y = x$$

$$\frac{\partial y}{\partial x} = 1$$

- Backward Pass (Output -> Hidden Layer 2)
  - We use the chain rule to calculate the derivatives for each weight update
  - E.g. for W_{k10}

$$\frac{\partial Loss}{\partial w_{k_1 o}} = \frac{\partial Loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k_1 o}}$$



- Backward Pass (Output -> Hidden Layer 2)
  - First find the partial derivative of the loss function to output

$$Loss = \frac{1}{2}(output - o_{out})^2$$

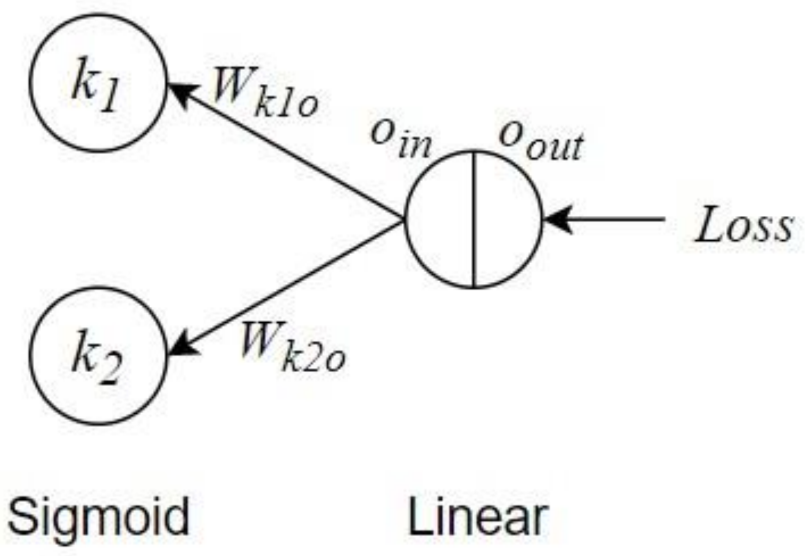
$$\frac{\partial Loss}{\partial o_{out}} = \frac{\partial(\frac{1}{2}(output - o_{out})^2)}{\partial o_{out}}$$

$$\frac{\partial Loss}{\partial o_{out}} = -1 \times 2 \times \frac{1}{2}(output - o_{out})$$

$$\frac{\partial Loss}{\partial o_{out}} = o_{out} - output$$

$$\frac{\partial Loss}{\partial o_{out}} = 2.494 - 3$$

$$\frac{\partial Loss}{\partial o_{out}} = -0.506$$

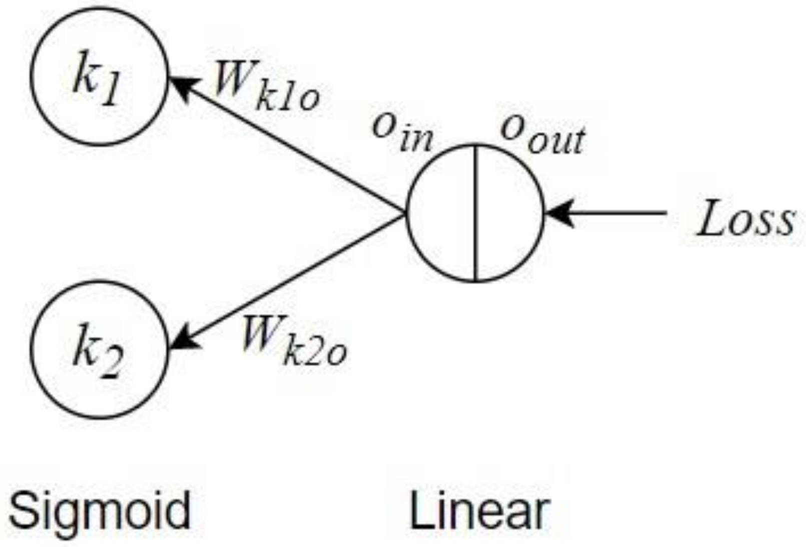


- Backward Pass (Output -> Hidden Layer 2)
  - Next calculate the gradient from O_{out} to O_{in} .

$$o_{out} = o_{in}$$

$$\frac{\partial o_{out}}{\partial o_{in}} = \frac{\partial (o_{in})}{\partial o_{in}}$$

$$\frac{\partial o_{out}}{\partial o_{in}} = 1$$



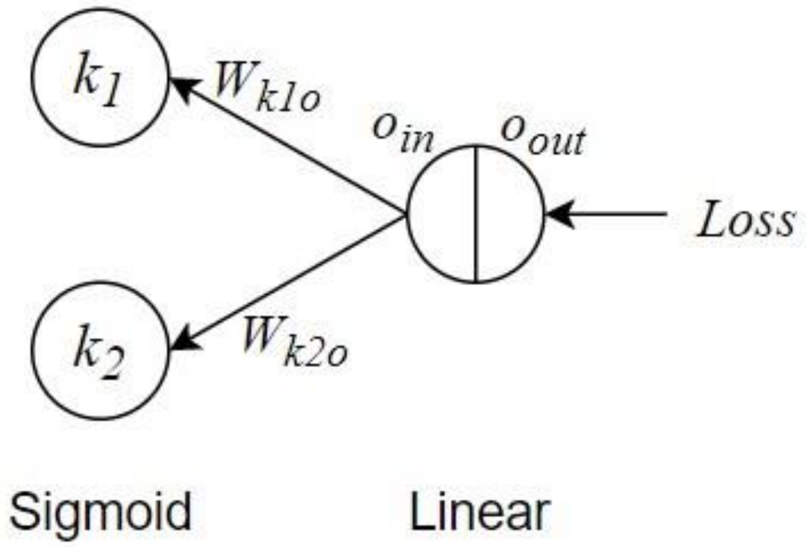
- Backward Pass (Output -> Hidden Layer 2)
  - Then find the gradient from O_{in} to W_{k1o}, *Wk2o and bias* (b_o)

$$o_{in} = w_{k_1o}k_{1out} + w_{k_2o}k_{2out} + b_o$$

$$\frac{\partial o_{in}}{\partial w_{k_1o}} = \frac{\partial(w_{k_1o}k_{1out} + w_{k_2o}k_{2out} + b_o)}{\partial w_{k_1o}}$$

$$\begin{bmatrix} \frac{\partial o_{in}}{\partial w_{k_1o}} \\ \frac{\partial o_{in}}{\partial w_{k_2o}} \end{bmatrix} = \begin{bmatrix} k_{1out} \\ k_{2out} \end{bmatrix} = \begin{bmatrix} 0.9975 \\ 0.9933 \end{bmatrix}$$

$$\left[\frac{\partial o_{in}}{\partial b_o} \right] = [1]$$



- Backward Pass (Output -> Hidden Layer 2)
  - Finally, we will apply the chain rule to find the gradient loss for weight and bias.

$$\begin{bmatrix} \frac{\partial Loss}{\partial w_{k_1o}} \\ \frac{\partial Loss}{\partial w_{k_2o}} \end{bmatrix} = \begin{bmatrix} \frac{\partial Loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k_1o}} \\ \frac{\partial Loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k_2o}} \end{bmatrix}$$

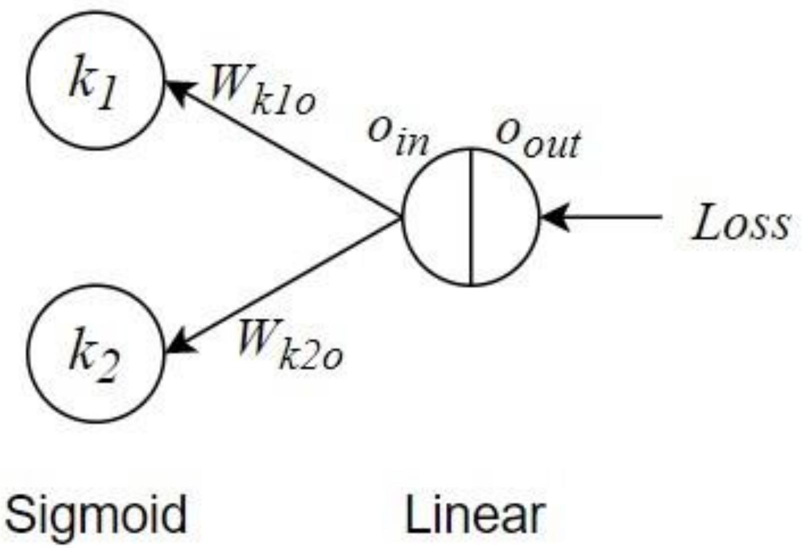
$$\begin{bmatrix} \frac{\partial Loss}{\partial w_{k_1o}} \\ \frac{\partial Loss}{\partial w_{k_2o}} \end{bmatrix} = \begin{bmatrix} -0.506 \times 1 \times 0.9975 \\ -0.506 \times 1 \times 0.9933 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial Loss}{\partial w_{k_1o}} \\ \frac{\partial Loss}{\partial w_{k_2o}} \end{bmatrix} = \begin{bmatrix} -0.50474 \\ -0.50261 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial Loss}{\partial b_o} \end{bmatrix} = \begin{bmatrix} \frac{\partial Loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial b_o} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial Loss}{\partial b_o} \end{bmatrix} = \begin{bmatrix} -0.506 \times 1 \times 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial Loss}{\partial b_o} \end{bmatrix} = \begin{bmatrix} -0.506 \end{bmatrix}$$

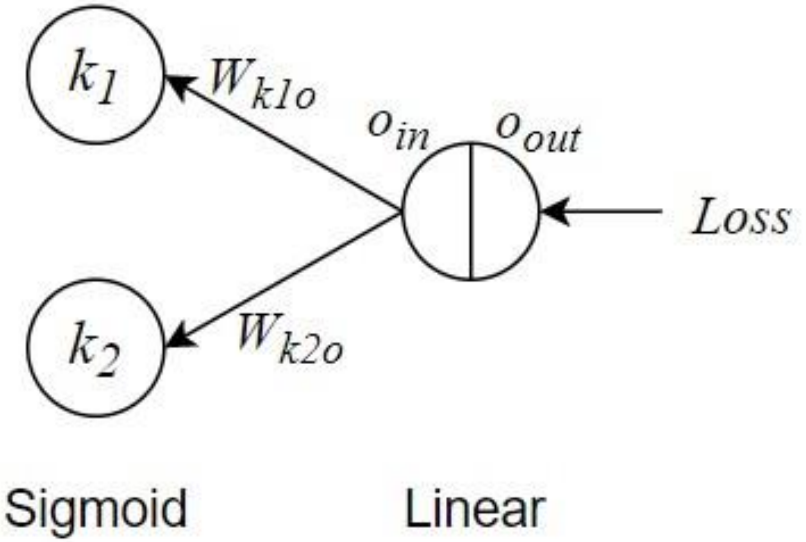


- Update the weights and bias
  - Using a learning rate of 0.25

$$w'_{k_1o} = w_{k_1o} - \propto \left(\frac{\partial Loss}{\partial w_{k_1o}}\right) = 1 - 0.25(-0.50474) = 1.1262$$

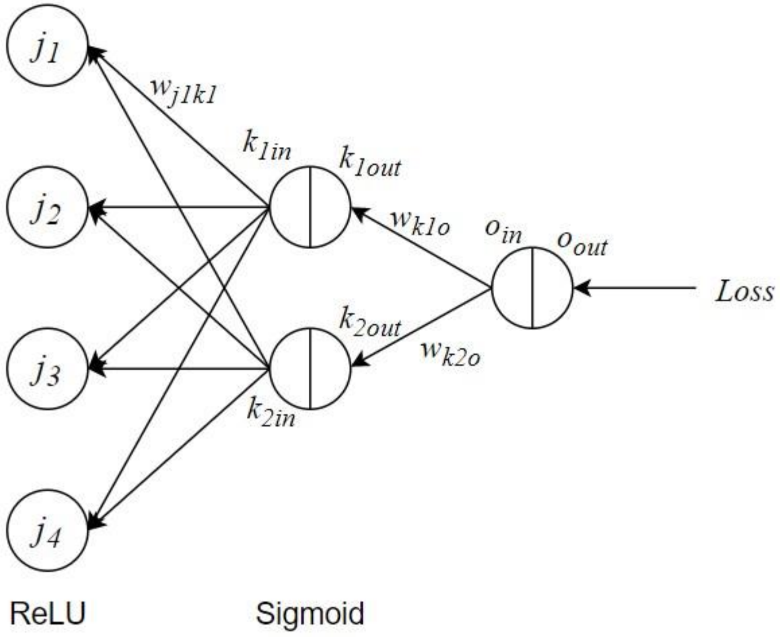
$$w'_{k_2o} = w_{k_2o} - \propto \left(\frac{\partial Loss}{\partial w_{k_2o}}\right) = 0.5 - 0.25(-0.50261) = 0.6256$$

$$b'_o = b_o - \propto \left(\frac{\partial Loss}{\partial b_o}\right) = 1 - 0.25(-0.506) = 1.1265$$



- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - Again use the chain rule to pass the gradient back
  - E.g., for W_{j1k1}

$$\frac{\partial Loss}{\partial w_{j_1 k_1}} = \frac{\partial Loss}{\partial k_{1out}} \times \frac{\partial k_{1out}}{\partial k_{1in}} \times \frac{\partial k_{1in}}{\partial w_{j_1 k_1}}$$



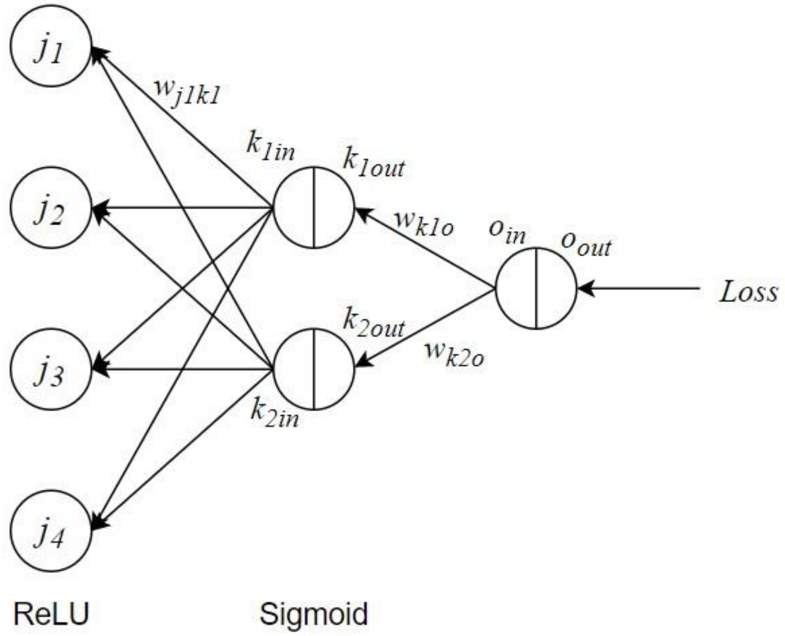
- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - First calculate gradient loss for K_{1out}
    - Use original weight values

$$\frac{\partial Loss}{\partial k_{1out}} = \frac{\partial Loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k_1 o}} \times \frac{\partial w_{k_1 o}}{\partial k_{1out}}$$

$$\frac{\partial Loss}{\partial k_{1out}} = -0.506 \times 1 \times 0.9975 \times w_{k_1 o(Lama)}$$

$$\frac{\partial Loss}{\partial k_{1out}} = -0.506 \times 1 \times 0.9975 \times 1.0$$

$$\begin{bmatrix} \frac{\partial Loss}{\partial k_{1out}} & \frac{\partial Loss}{\partial k_{1out}} \end{bmatrix} = \begin{bmatrix} -0.50474 & -0.25130 \end{bmatrix}$$



- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - Then find gradient K_{1out} against K_{1in} .
    - Use derivative of the sigmoid

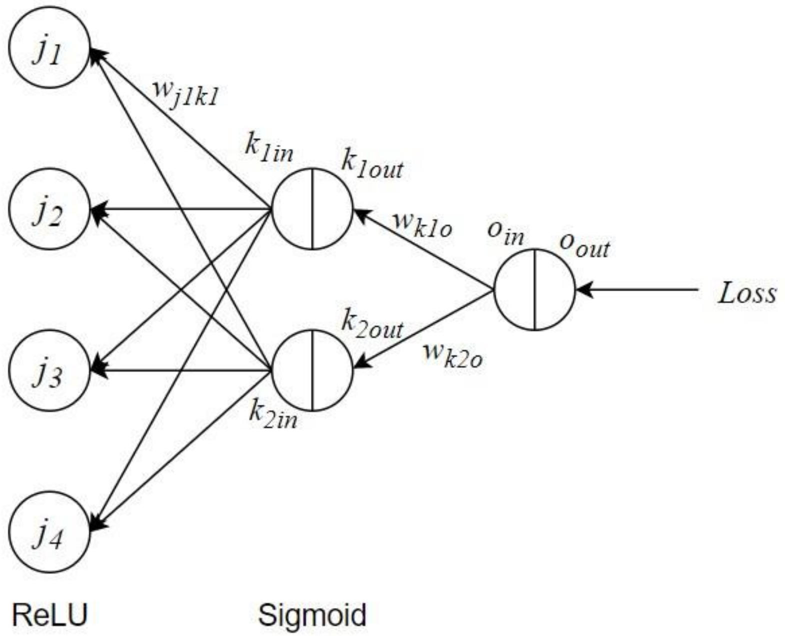
$$k_{1out} = \frac{1}{1+e^{-k_{1in}}}$$

$$\frac{\partial k_{1out}}{\partial k_{1in}} = \frac{\partial(\frac{1}{1+e^{-k_{1in}}})}{\partial k_{1in}}$$

$$\frac{\partial k_{1out}}{\partial k_{1in}} = \frac{1}{1+e^{-k_{1in}}} \times (1 - \frac{1}{1+e^{-k_{1in}}})$$

$$\frac{\partial k_{1out}}{\partial k_{1in}} = \frac{1}{1+e^{-6}} \times (1 - \frac{1}{1+e^{-6}})$$

$$\begin{bmatrix} \frac{\partial k_{1out}}{\partial k_{1in}} \\ \frac{\partial k_{2out}}{\partial k_{2in}} \end{bmatrix} = \begin{bmatrix} 0.00249 \\ 0.00665 \end{bmatrix}$$



- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - Next the gradient from K_{1in} to W_{j1k1}

$$k_{1in} = w_{j_1k_1} j_{1out} + w_{j_2k_1} j_{2out} + w_{j_3k_1} j_{3out} + w_{j_4k_1} j_{4out} + b_{jk_1}$$

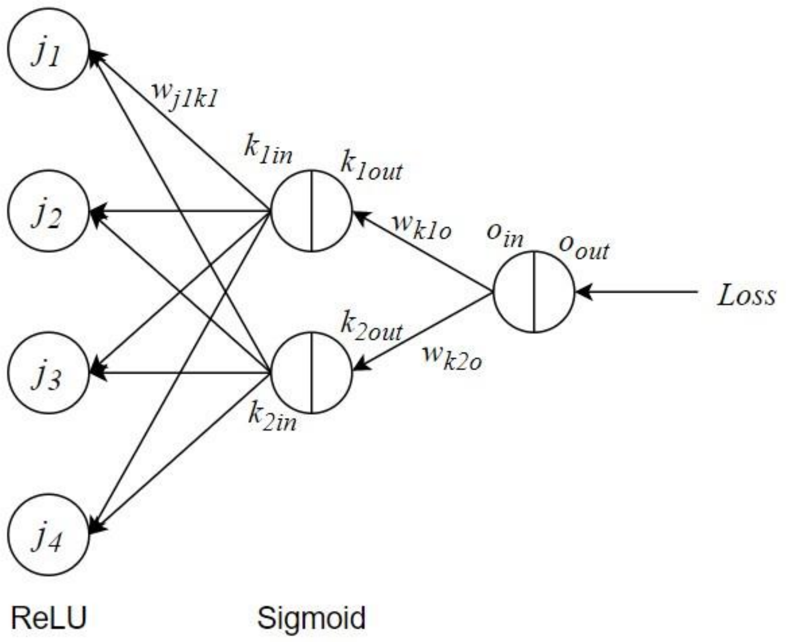
$$\frac{\partial k_{1in}}{\partial w_{j_1k_1}} = \frac{\partial (w_{j_1k_1} j_{1out} + w_{j_2k_1} j_{2out} + w_{j_3k_1} j_{3out} + w_{j_4k_1} j_{4out} + b_{jk_1})}{\partial w_{j_1k_1}}$$

$$\begin{bmatrix} \frac{\partial k_{1in}}{\partial w_{j_1k_1}} & \frac{\partial k_{1in}}{\partial w_{j_2k_1}} & \frac{\partial k_{1in}}{\partial w_{j_3k_1}} & \frac{\partial k_{1in}}{\partial w_{j_4k_1}} \end{bmatrix} = \begin{bmatrix} j_{1out} & j_{2out} & j_{3out} & j_{4out} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial k_{2in}}{\partial w_{j_1k_2}} & \frac{\partial k_{2in}}{\partial w_{j_2k_2}} & \frac{\partial k_{2in}}{\partial w_{j_3k_2}} & \frac{\partial k_{2in}}{\partial w_{j_4k_2}} \end{bmatrix} = \begin{bmatrix} j_{1out} & j_{2out} & j_{3out} & j_{4out} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial k_{1in}}{\partial w_{j_1k_1}} & \frac{\partial k_{1in}}{\partial w_{j_2k_1}} & \frac{\partial k_{1in}}{\partial w_{j_3k_1}} & \frac{\partial k_{1in}}{\partial w_{j_4k_1}} \end{bmatrix} = \begin{bmatrix} 1.5 & 2.0 & 2.5 & 3.0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial k_{2in}}{\partial w_{j_1k_2}} & \frac{\partial k_{2in}}{\partial w_{j_2k_2}} & \frac{\partial k_{2in}}{\partial w_{j_3k_2}} & \frac{\partial k_{2in}}{\partial w_{j_4k_2}} \end{bmatrix} = \begin{bmatrix} 1.5 & 2.0 & 2.5 & 3.0 \end{bmatrix}$$

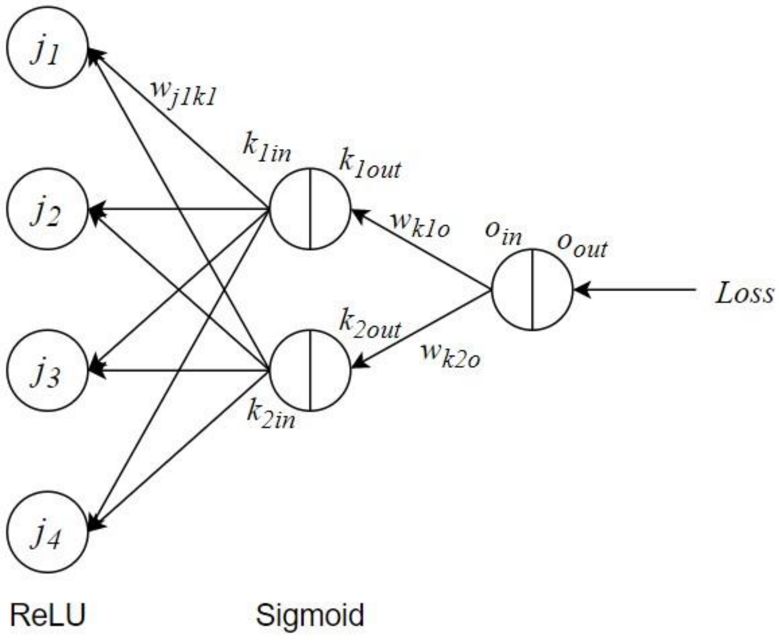


- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - Now we can calculate the gradient loss to W_{j1k1} by applying the chain rule

$$\frac{\partial Loss}{\partial w_{j_1 k_1}} = \frac{\partial Loss}{\partial k_{1out}} \times \frac{\partial k_{1out}}{\partial k_{1in}} \times \frac{\partial k_{1in}}{\partial w_{j_1 k_1}}$$

$$\frac{\partial Loss}{\partial w_{j_1 k_1}} = -0.50474 \times 0.00249 \times 1.5$$

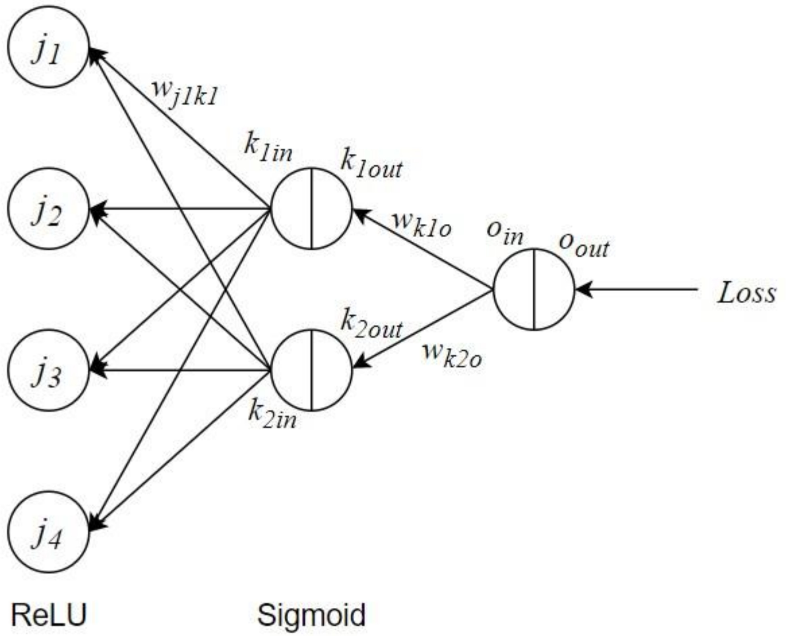
Vanishing Gradient Issues



- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - Apply the same strategy across all parameters

$$\begin{bmatrix} \frac{\partial Loss}{\partial w_{j_1k_1}} & \frac{\partial Loss}{\partial w_{j_1k_2}} \\ \frac{\partial Loss}{\partial w_{j_2k_1}} & \frac{\partial Loss}{\partial w_{j_2k_2}} \\ \frac{\partial Loss}{\partial w_{j_3k_1}} & \frac{\partial Loss}{\partial w_{j_3k_2}} \\ \frac{\partial Loss}{\partial w_{j_4k_1}} & \frac{\partial Loss}{\partial w_{j_4k_2}} \end{bmatrix} = \begin{bmatrix} -0.00188 & -0.00252 \\ -0.00251 & -0.00334 \\ -0.00314 & -0.00417 \\ -0.00377 & -0.00501 \end{bmatrix}$$

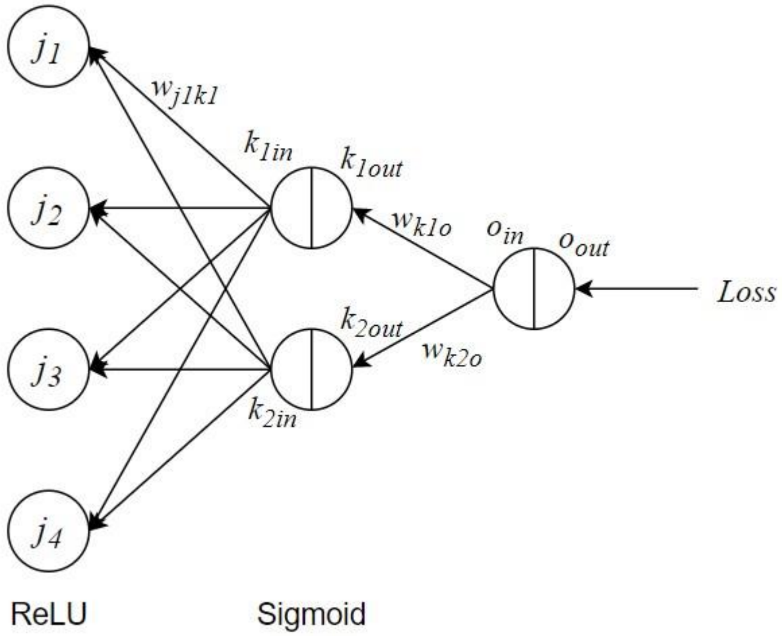
$$\begin{bmatrix} \frac{\partial Loss}{\partial b_{jk_1}} & \frac{\partial Loss}{\partial b_{jk_2}} \end{bmatrix} = \begin{bmatrix} -0.00125 & -0.00167 \end{bmatrix}$$



- Backward Pass (Hidden Layer 2 -> Hidden Layer 1)
  - Update the weights

$$\begin{bmatrix} w'_{j_1k_1} & w'_{j_1k_2} \\ w'_{j_2k_1} & w'_{j_2k_2} \\ w'_{j_3k_1} & w'_{j_3k_2} \\ w'_{j_4k_1} & w'_{j_4k_2} \end{bmatrix} = \begin{bmatrix} w_{j_1k_1} - \alpha(\frac{\partial Loss}{\partial w_{j_1k_1}}) & w_{j_1k_2} - \alpha(\frac{\partial Loss}{\partial w_{j_1k_2}}) \\ w_{j_2k_1} - \alpha(\frac{\partial Loss}{\partial w_{j_2k_1}}) & w_{j_2k_2} - \alpha(\frac{\partial Loss}{\partial w_{j_2k_2}}) \\ w_{j_3k_1} - \alpha(\frac{\partial Loss}{\partial w_{j_3k_1}}) & w_{j_3k_2} - \alpha(\frac{\partial Loss}{\partial w_{j_3k_2}}) \\ w_{j_4k_1} - \alpha(\frac{\partial Loss}{\partial w_{j_4k_1}}) & w_{j_4k_2} - \alpha(\frac{\partial Loss}{\partial w_{j_4k_2}}) \end{bmatrix} = \begin{bmatrix} 1.00047 & 0.00062 \\ 0.75062 & 0.25083 \\ 0.50078 & 0.50104 \\ 0.25094 & 0.75125 \end{bmatrix}$$

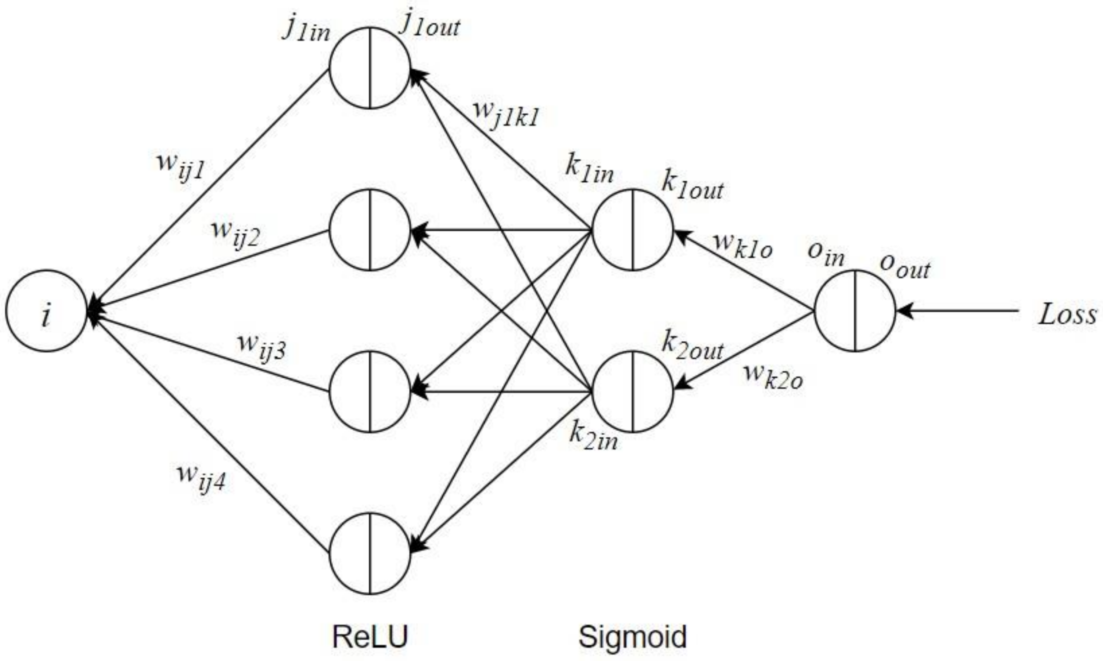
$$\begin{bmatrix} b'_{jk_1} & b'_{jk_2} \end{bmatrix} = \begin{bmatrix} b_{jk_1} - \alpha(\frac{\partial Loss}{\partial b_{jk_1}}) & b_{jk_2} - \alpha(\frac{\partial Loss}{\partial b_{jk_2}}) \end{bmatrix} = \begin{bmatrix} 1.00031 & 1.00042 \end{bmatrix}$$



Backpropagation Example

- Backward Pass (Hidden Layer 1 -> Input Layer)
  - Again use the chain rule to calculate gradients
  - E.g., for w_{ij1}

$$\frac{\partial Loss}{\partial w_{ij_1}} = \frac{\partial Loss}{\partial j_{1out}} \times \frac{\partial j_{1out}}{\partial j_{1in}} \times \frac{\partial j_{1in}}{\partial w_{ij_1}}$$



Backpropagation Example

- Backward Pass (Hidden Layer 1 -> Input Layer)
  - First calculate gradient loss with respect to J_{1out}

$$\frac{\partial Loss}{\partial j_{1out}} = \frac{\partial Loss}{\partial k_{out}} \times \frac{\partial k_{out}}{\partial k_{in}} \times \frac{\partial k_{in}}{\partial w_{j_1 k}} \times \frac{\partial w_{j_1 k}}{\partial j_{1out}}$$

  - This time it's more complicated than K_{1out} calculation
    - J_{1out} is influenced by a gradient that comes from K_2
    - So we have to look at Layer K as a whole

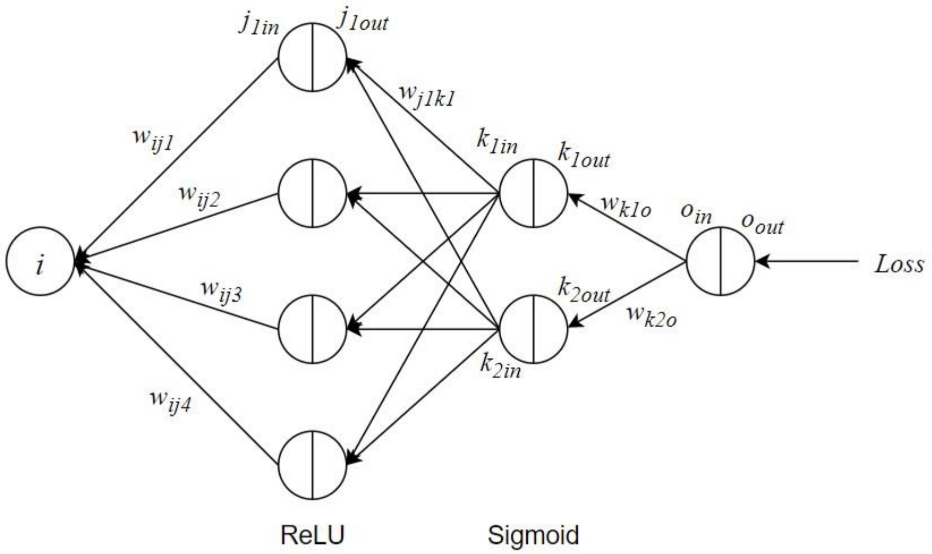
- Backward Pass (Hidden Layer 1 -> Input Layer)
  - First calculate gradient loss with respect to J_{1out}

$$\frac{\partial Loss}{\partial k_{out}} = \frac{\partial Loss}{\partial k_{1out}} + \frac{\partial Loss}{\partial k_{2out}} = -0.50474 + -0.25130 = -0.75604$$

$$\frac{\partial k_{out}}{\partial k_{in}} = \frac{\partial k_{1out}}{\partial k_{in}} + \frac{\partial k_{2out}}{\partial k_{in}} = 0.00249 + 0.00665 = 0.00914$$

$$\frac{\partial k_{in}}{\partial w_{j_1k}} = \frac{\partial k_{1in}}{\partial w_{j_1k_1}} + \frac{\partial k_{2in}}{\partial w_{j_1k_2}} = 1.5 + 1.5 = 3.0$$

$$\frac{\partial w_{j_1k}}{\partial j_{1out}} = \frac{\partial w_{j_1k_1}}{\partial j_{1out}} + \frac{\partial w_{j_1k_2}}{\partial j_{1out}} = w_{j_1k_1} + w_{j_1k_2} = 1.0 + 0 = 1.0$$

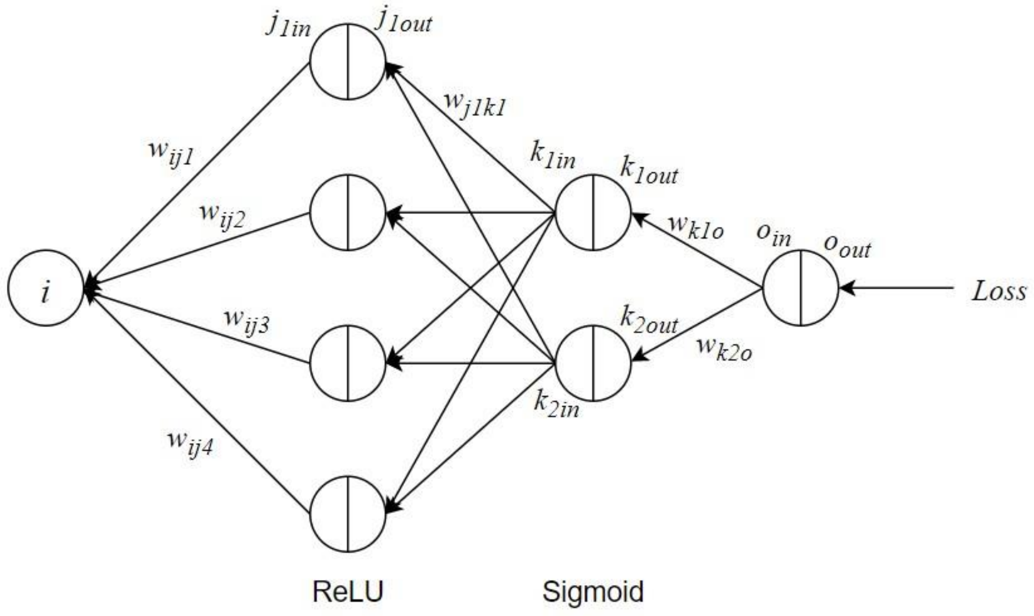


- Backward Pass (Hidden Layer 1 -> Input Layer)
  - First calculate gradient loss with respect to J_{1out}

$$\frac{\partial Loss}{\partial j_{1out}} = \frac{\partial Loss}{\partial k_{out}} \times \frac{\partial k_{out}}{\partial k_{in}} \times \frac{\partial k_{in}}{\partial w_{j_1 k}} \times \frac{\partial w_{j_1 k}}{\partial j_{1out}}$$

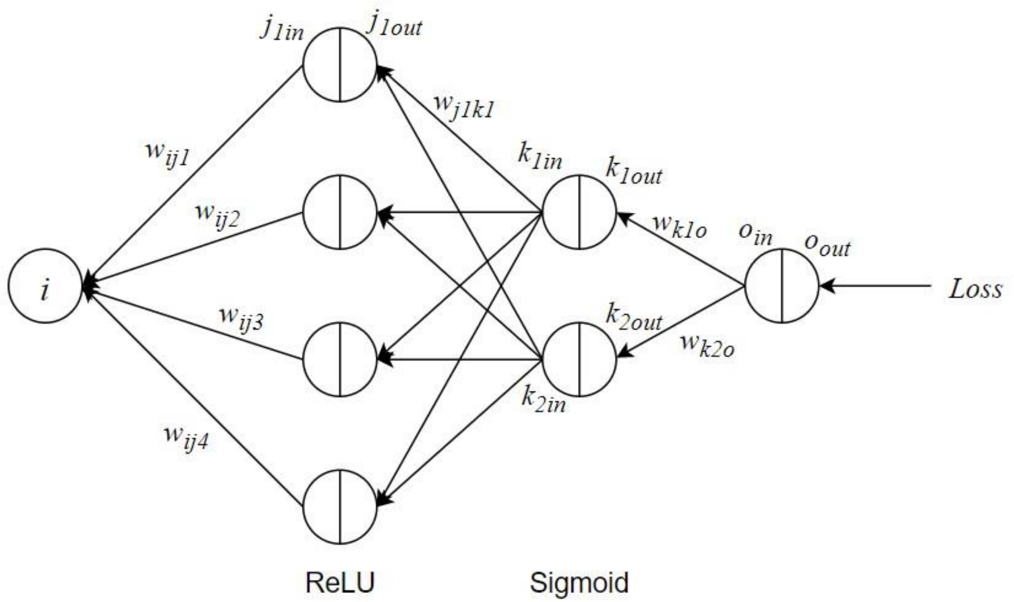
$$\frac{\partial Loss}{\partial j_{1out}} = -0.75604 \times 0.00914 \times 3.0 \times 1.0$$

$$\frac{\partial Loss}{\partial j_{1out}} = -0.02073$$



- Backward Pass (Hidden Layer 1 -> Input Layer)
  - Continue with the J_{1out} gradient towards J_{1in}

$$j_{1out} = max(0, j_{1in})$$
$$j_{1out} = max(0, 1.5)$$
$$\frac{\partial j_{1out}}{\partial j_{1in}} = \frac{\partial (ReLU)}{\partial j_{1in}} = \begin{cases} 1 & j_{1in} > 0 \\ 0 & j_{1in} = 0 \end{cases}$$
$$\frac{\partial j_{1out}}{\partial j_{1in}} = 1$$



Backpropagation Example

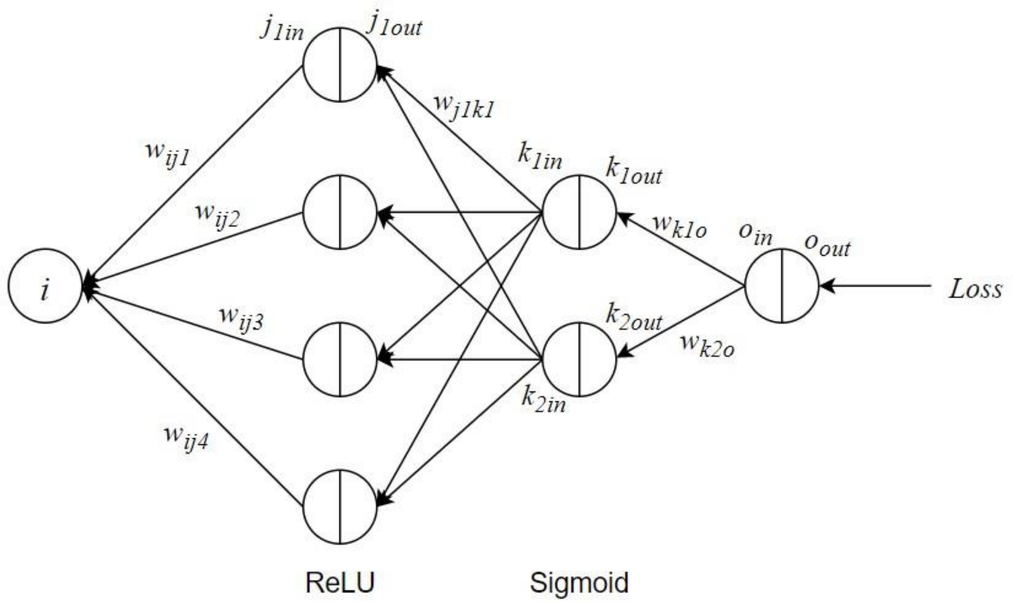
- Backward Pass (Hidden Layer 1 -> Input Layer)
  - Next find the J_{1in} gradient towards W_{ij1}

$$j_{1in} = w_{ij_1} i + b_{ij_1}$$

$$\frac{\partial j_{1in}}{\partial w_{ij_1}} = \frac{\partial (w_{ij_1} i + b_{ij_1})}{\partial w_{ij_1}}$$

$$\frac{\partial j_{1in}}{\partial w_{ij_1}} = i$$

$$\frac{\partial j_{1in}}{\partial w_{ij_1}} = 2.0$$

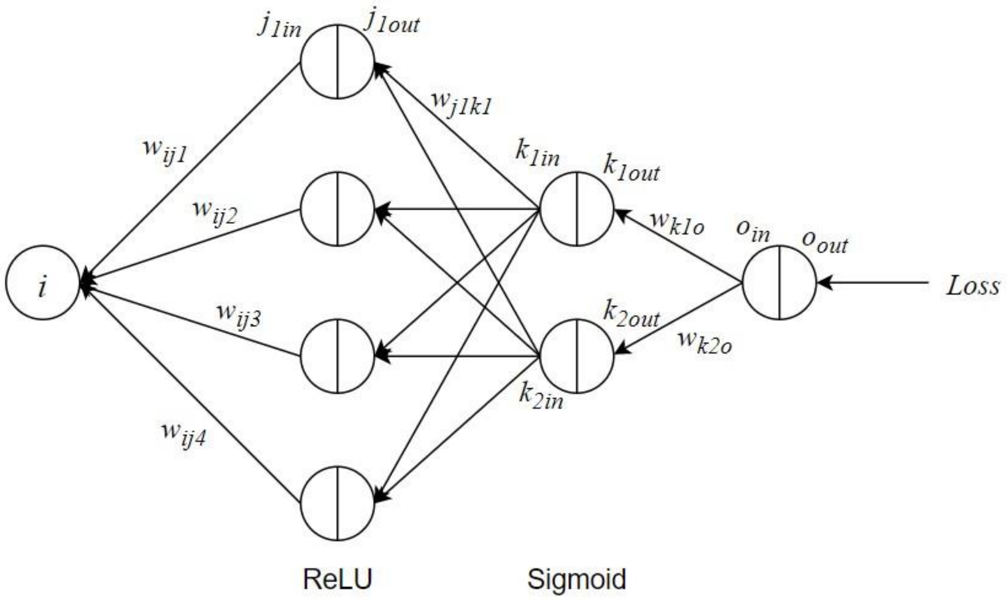


- Backward Pass (Hidden Layer 1 -> Input Layer)
  - Now, use the chain rule to calculate gradient

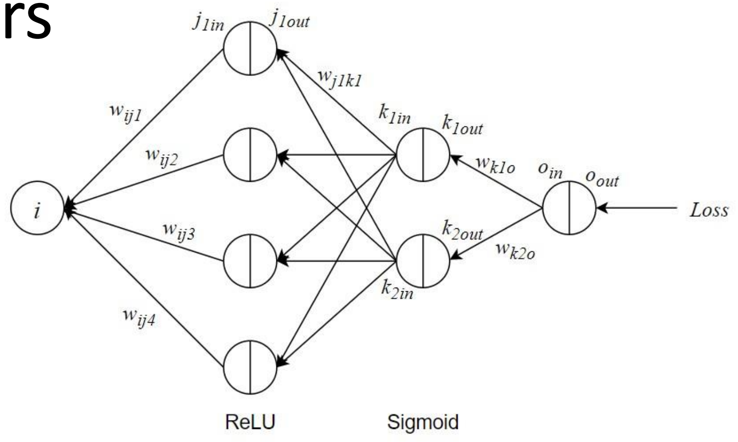
$$\frac{\partial Loss}{\partial w_{ij_1}} = \frac{\partial Loss}{\partial j_{1out}} \times \frac{\partial j_{1out}}{\partial j_{1in}} \times \frac{\partial j_{1in}}{\partial w_{ij_1}}$$

$$\frac{\partial Loss}{\partial w_{ij_1}} = -0.02073 \times 1 \times 2$$

$$\frac{\partial Loss}{\partial w_{ij_1}} = -0.04146$$



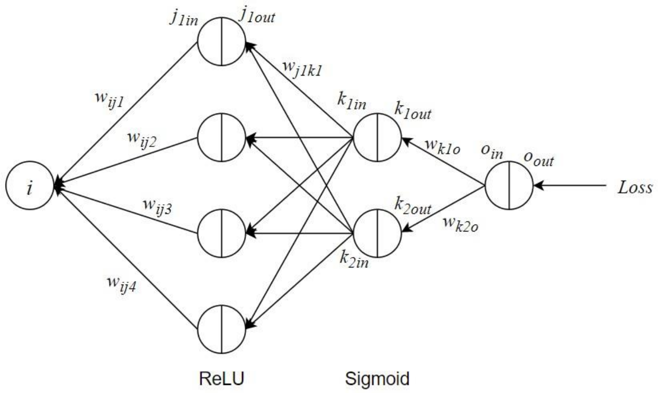
- Backward Pass (Hidden Layer 1 -> Input Layer)
  - Apply same calculations across all parameters



$$\begin{bmatrix} \frac{\partial Loss}{\partial w_{ij_1}} & \frac{\partial Loss}{\partial w_{ij_2}} & \frac{\partial Loss}{\partial w_{ij_3}} & \frac{\partial Loss}{\partial w_{ij_4}} \end{bmatrix} = \begin{bmatrix} -0.04146 & -0.05528 & -0.06910 & -0.08292 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial Loss}{\partial b_{ij_1}} & \frac{\partial Loss}{\partial b_{ij_2}} & \frac{\partial Loss}{\partial b_{ij_3}} & \frac{\partial Loss}{\partial b_{ij_4}} \end{bmatrix} = \begin{bmatrix} -0.02073 & -0.02764 & -0.03455 & -0.04146 \end{bmatrix}$$

- Backward Pass (Hidden Layer 1 -> Input Layer)
  - Calculate Weight updates



$$\begin{bmatrix} w'_{ij_1} & w'_{ij_2} & w'_{ij_3} & w'_{ij_4} \end{bmatrix} = \begin{bmatrix} w_{ij_1} - \alpha(\frac{\partial Loss}{\partial w_{ij_1}}) & w_{ij_2} - \alpha(\frac{\partial Loss}{\partial w_{ij_2}}) & w_{ij_3} - \alpha(\frac{\partial Loss}{\partial w_{ij_3}}) & w_{ij_4} - \alpha(\frac{\partial Loss}{\partial w_{ij_4}}) \end{bmatrix} = [0.26037 \quad 0.51382 \quad 0.76728 \quad 1.02073]$$

$$\begin{bmatrix} b'_{ij_1} & b'_{ij_2} & b'_{ij_3} & b'_{ij_4} \end{bmatrix} = \begin{bmatrix} b_{ij_1} - \alpha(\frac{\partial Loss}{\partial b_{ij_1}}) & b_{ij_2} - \alpha(\frac{\partial Loss}{\partial b_{ij_2}}) & b_{ij_3} - \alpha(\frac{\partial Loss}{\partial b_{ij_3}}) & b_{ij_4} - \alpha(\frac{\partial Loss}{\partial b_{ij_4}}) \end{bmatrix} = [1.02073 \quad 1.02764 \quad 1.03455 \quad 1.04146]$$

Universität Augsburg University

- Old vs New parameter values

$$W_{ij} = \begin{bmatrix} w_{ij_1} & w_{ij_2} & w_{ij_3} & w_{ij_4} \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.75 & 1.0 \end{bmatrix}$$

$$W_{jk} = \begin{bmatrix} w_{j_1k_1} & w_{j_1k_2} \\ w_{j_2k_1} & w_{j_2k_2} \\ w_{j_3k_1} & w_{j_3k_2} \\ w_{j_4k_1} & w_{j_4k_2} \end{bmatrix} = \begin{bmatrix} 1.0 & 0 \\ 0.75 & 0.25 \\ 0.5 & 0.5 \\ 0.25 & 0.75 \end{bmatrix}$$

$$W_{ko} = \begin{bmatrix} w_{k_1o} \\ w_{k_2o} \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$$

$$b_{ij} = \begin{bmatrix} b_{ij_1} & b_{ij_2} & b_{ij_3} & b_{ij_4} \end{bmatrix} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

$$b_{jk} = \begin{bmatrix} b_{jk_1} & b_{jk_2} \end{bmatrix} = \begin{bmatrix} 1.0 & 1.0 \end{bmatrix}$$

$$b_o = \begin{bmatrix} 1.0 \end{bmatrix}$$

$$W'_{ij} = \begin{bmatrix} w_{ij_1} & w_{ij_2} & w_{ij_3} & w_{ij_4} \end{bmatrix} = \begin{bmatrix} 0.26037 & 0.51382 & 0.76728 & 1.02073 \end{bmatrix}$$

$$W'_{jk} = \begin{bmatrix} w_{j_1k_1} & w_{j_1k_2} \\ w_{j_2k_1} & w_{j_2k_2} \\ w_{j_3k_1} & w_{j_3k_2} \\ w_{j_4k_1} & w_{j_4k_2} \end{bmatrix} = \begin{bmatrix} 1.00047 & 0.00062 \\ 0.75062 & 0.25083 \\ 0.50078 & 0.50104 \\ 0.25094 & 0.75125 \end{bmatrix}$$

$$W'_{ko} = \begin{bmatrix} w_{k_1o} \\ w_{k_2o} \end{bmatrix} = \begin{bmatrix} 1.1262 \\ 0.6256 \end{bmatrix}$$

$$b'_{ij} = \begin{bmatrix} b_{ij_1} & b_{ij_2} & b_{ij_3} & b_{ij_4} \end{bmatrix} = \begin{bmatrix} 1.02073 & 1.02674 & 1.03455 & 1.04146 \end{bmatrix}$$

$$b'_{jk} = \begin{bmatrix} b_{jk_1} & b_{jk_2} \end{bmatrix} = \begin{bmatrix} 1.00031 & 1.00042 \end{bmatrix}$$

$$b'_o = \begin{bmatrix} 1.1265 \end{bmatrix}$$