Introduction to Deep Learning

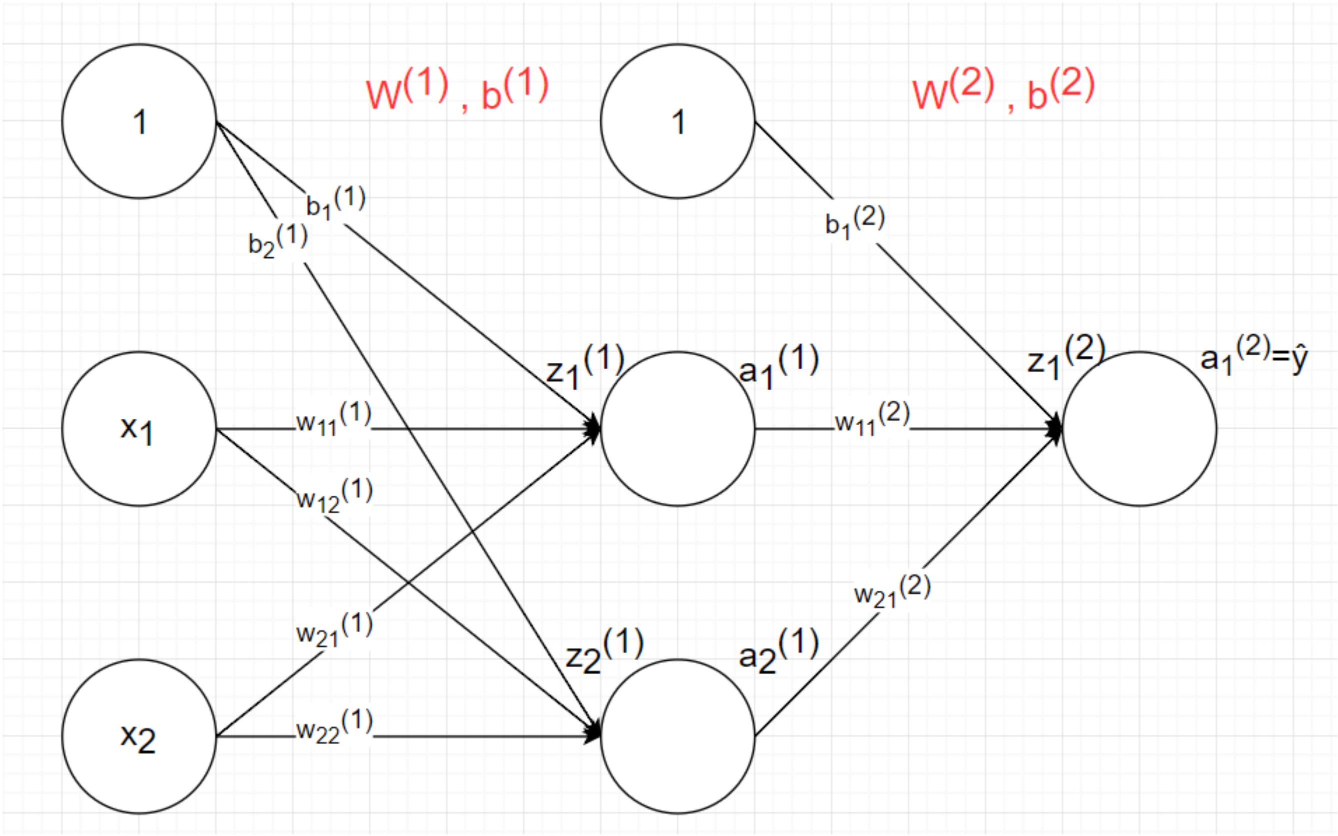
Backpropagation

XOR problem

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

x_1 XOR x_2

Neural Network Model for XOR



Neural Network Model for XOR problem

Neural Network Model for XOR

Neural Network for XOR problem:

- Model: 2-2-1: 2 nodes in input layer, 2 nodes in hidden layers, 1 node in output layer
- Nodes 1 are added to calculate bias in next layers
- Each node in hidden layers and output layer are performed two steps:
  - (1) Linear sum
  - (2) Apply activation function

Feedforward

$$z_1^{(1)} = b_1^{(1)} + x_1 * w_{11}^{(1)} + x_2 * w_{21}^{(1)}$$
$$a_1^{(1)} = \sigma(z_1^{(1)})$$
$$z_2^{(1)} = b_2^{(1)} + x_1 * w_{12}^{(1)} + x_2 * w_{22}^{(1)}$$
$$a_2^{(1)} = \sigma(z_2^{(1)})$$
$$z_1^{(2)} = b_1^{(2)} + a_1^{(1)} * w_{11}^{(2)} + a_2^{(1)} * w_{21}^{(2)}$$
$$\hat{y} = a_1^{(2)} = \sigma(z_1^{(2)})$$

In Matrix form

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

In Matrix form

$$Z^{(1)} = X * W^{(1)} + b^{(1)}$$
$$A^{(1)} = \sigma(Z^{(1)})$$
$$Z^{(2)} = A^{(1)} * W^{(2)} + b^{(2)}$$
$$\hat{Y} = A^{(2)} = \sigma(Z^{(2)})$$

Loss Function

For each data point ($x^{[i]}$, y_i), the loss function L is defined as follows:

$$L = -(y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - \hat{y}_i))$$

In which: y_i is the actual value of data, $\hat{y}_i$ is the value predicted by the model:

$$\hat{y}_i = a_1^{(2)} = \sigma(a_1^{(1)} * w_{11}^{(2)} + a_2^{(1)} * w_{21}^{(2)} + b_1^{(2)})$$

Loss Function

For all data points, the loss function J is defined as follows:

$$J = -\sum_{i=1}^{N}(y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - \hat{y}_i))$$

Gradient Descent

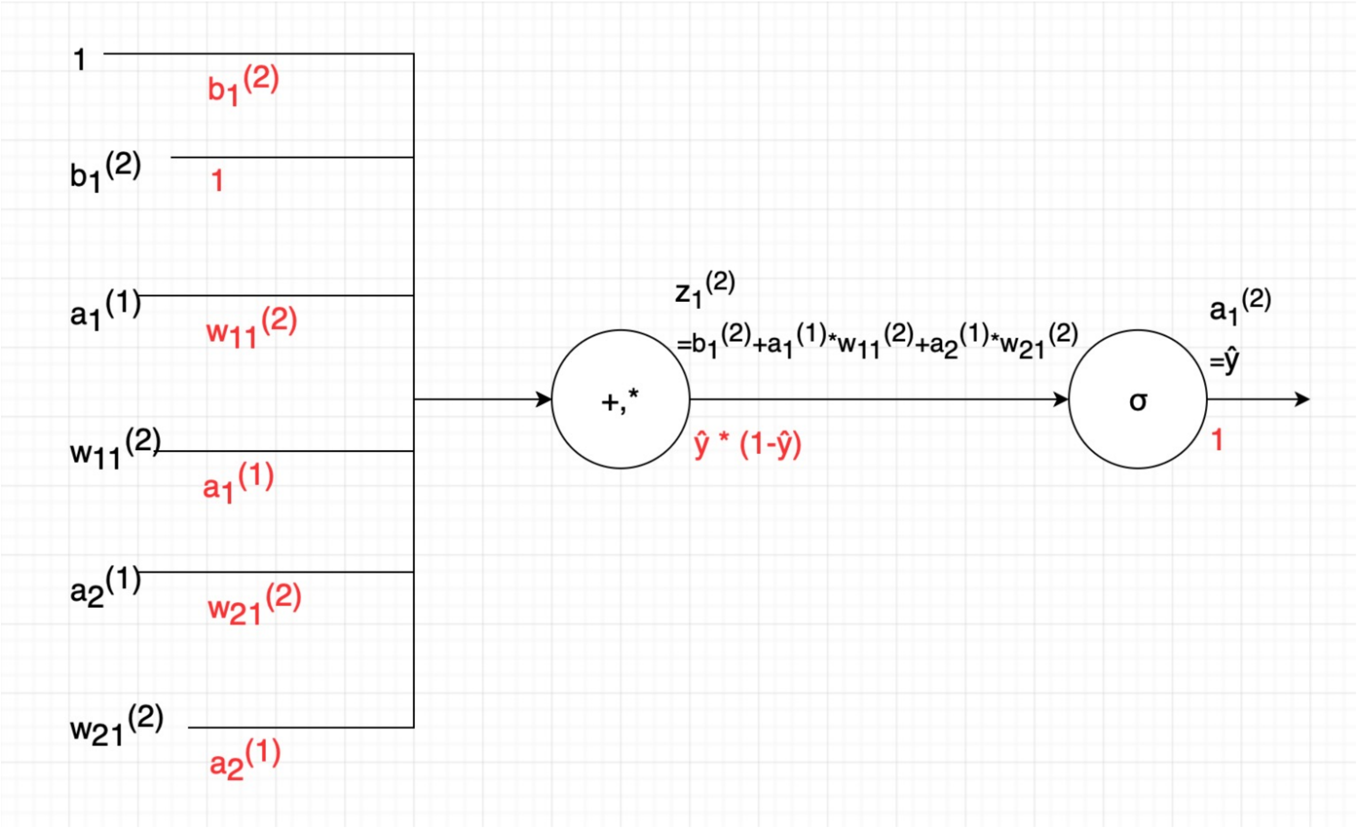
- To apply gradient descent, we need to calculate the derivative of the coefficient W and bias b of the loss function:
- Step 1, calculate L' with $W^{(2)}$, $b^{(2)}$,we have:

$$\frac{\partial L}{\partial b_1^{(2)}} = \frac{dL}{d\hat{y}_i} * \frac{\partial \hat{y}_i}{\partial b_1^{(2)}}$$

In which:

$$\frac{\partial L}{\partial \hat{y}_i} = -\frac{\partial (y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - \hat{y}_i))}{\partial \hat{y}_i} = -(\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{(1 - \hat{y})})$$

Gradient Descent



Chain rule for node 1 layer 2

Gradient Descent

From the chain rule, we have:

$$\frac{\partial \hat{y}_i}{\partial b_1^{(2)}} = \hat{y}_i * (1 - \hat{y}_i)$$

$$\frac{\partial \hat{y}_i}{\partial w_{11}^{(2)}} = a_1^{(1)} * \hat{y}_i * (1 - \hat{y}_i)$$

$$\frac{\partial \hat{y}_i}{\partial w_{21}^{(2)}} = a_2^{(1)} * \hat{y}_i * (1 - \hat{y}_i)$$

$$\frac{\partial \hat{y}_i}{\partial a_1^{(1)}} = w_{11}^{(2)} * \hat{y}_i * (1 - \hat{y}_i)$$

$$\frac{\partial \hat{y}_i}{\partial a_2^{(1)}} = w_{21}^{(2)} * \hat{y}_i * (1 - \hat{y}_i)$$

Gradient Descent

From the chain rule, we have:

$$\frac{\partial L}{\partial b_1^{(2)}} = \frac{\partial L}{\partial \hat{y}_i} * \frac{\partial \hat{y}_i}{\partial b_1^{(2)}} = -(\frac{y_i}{\hat{y}_i} - \frac{1-y_i}{(1-\hat{y}_i)}) * \hat{y}_i * (1-\hat{y}_i) = -(y_i * (1-\hat{y}_i) - (1-y_i) * \hat{y}_i)) = \hat{y}_i - y_i$$

Gradient Descent

Similarly, we have:

$$\frac{\partial L}{\partial w_{11}^{(2)}} = a_1^{(1)} * (\hat{y}_i - y_i)$$

$$\frac{\partial L}{\partial w_{21}^{(2)}} = a_2^{(1)} * (\hat{y}_i - y_i)$$

$$\frac{\partial L}{\partial a_1^{(1)}} = w_{11}^{(2)} * (\hat{y}_i - y_i)$$

$$\frac{\partial L}{\partial a_2^{(1)}} = w_{21}^{(2)} * (\hat{y}_i - y_i)$$

Gradient Descent

Step 2, calculate L' with $W^{(1)}$, $b^{(1)}$, since:

$$a_1^{(1)} = \sigma(b_1^{(1)} + x_1 * w_{11}^{(1)} + x_2 * w_{21}^{(1)})$$

Apply chain rule, we have:

$$\frac{\partial L}{\partial b_1^{(1)}} = \frac{\partial L}{\partial a_1^{(1)}} * \frac{\partial a_1^{(1)}}{\partial b_1^{(1)}}$$

Gradient Descent

We have:

$$\frac{\partial a_1^{(1)}}{\partial b_1^{(1)}} = \frac{\partial a_1^{(1)}}{z_1^{(1)}} * \frac{z_1^{(1)}}{\partial b_1^{(1)}} = a_1^{(1)} * (1 - a_1^{(1)})$$

Therefore:

$$\frac{\partial L}{\partial b_1^{(1)}} = a_1^{(1)} * (1 - a_1^{(1)}) * w_{11}^{(2)} * (\hat{y}_i - y_i)$$

Gradient Descent

We have:

$$\frac{\partial a_1^{(1)}}{\partial b_1^{(1)}} = \frac{\partial a_1^{(1)}}{z_1^{(1)}} * \frac{z_1^{(1)}}{\partial b_1^{(1)}} = a_1^{(1)} * (1 - a_1^{(1)})$$

Therefore:

$$\frac{\partial L}{\partial b_1^{(1)}} = a_1^{(1)} * (1 - a_1^{(1)}) * w_{11}^{(2)} * (\hat{y}_i - y_i)$$

Gradient Descent

Similarly:

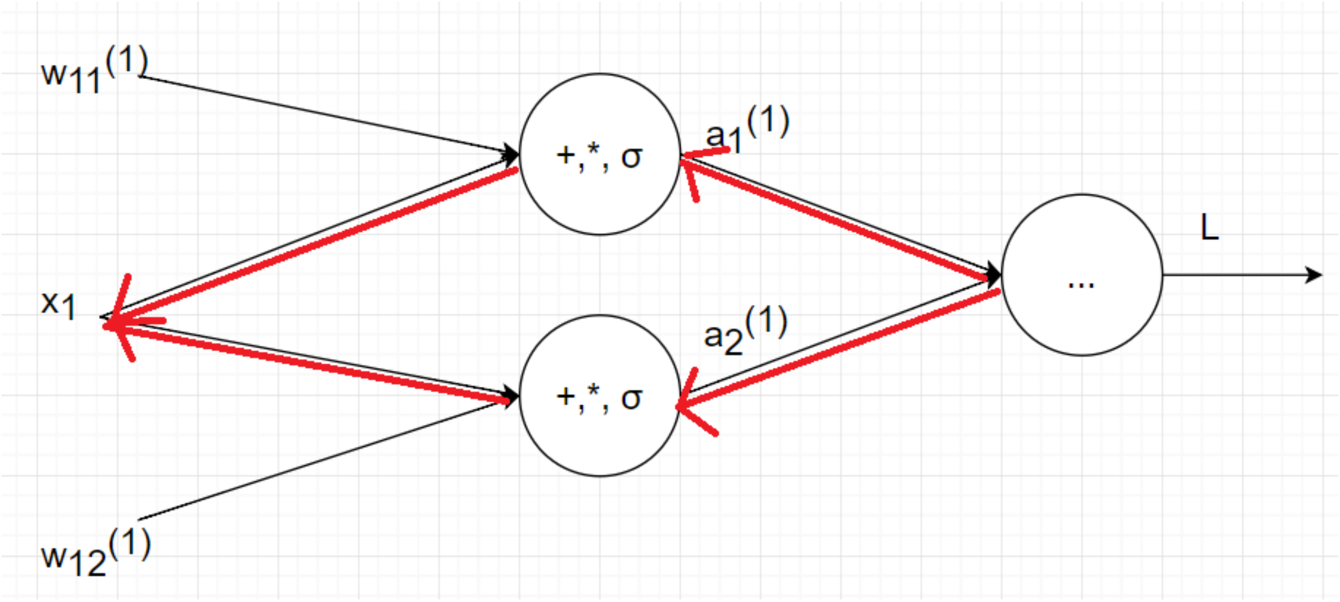
$$\frac{\partial L}{\partial w_{11}^{(1)}} = x_1 * a_1^{(1)} * (1 - a_1^{(1)}) * w_{11}^{(2)} * (\hat{y}_i - y_i)$$

$$\frac{\partial L}{\partial w_{12}^{(1)}} = x_1 * a_2^{(1)} * (1 - a_2^{(1)}) * w_{11}^{(2)} * (\hat{y}_i - y_i)$$

$$\frac{\partial L}{\partial w_{21}^{(1)}} = x_2 * a_1^{(1)} * (1 - a_1^{(1)}) * w_{21}^{(2)} * (\hat{y}_i - y_i)$$

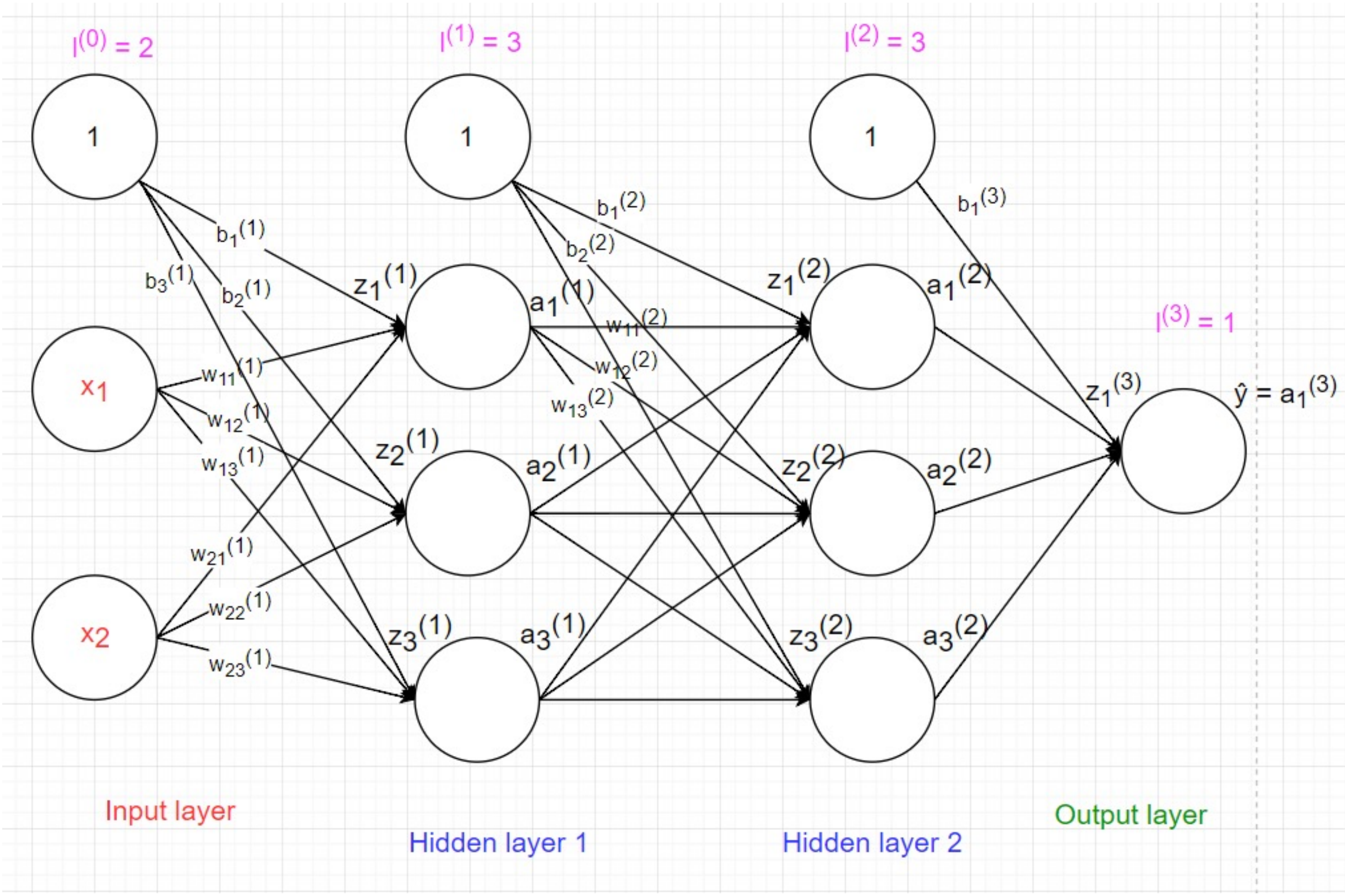
$$\frac{\partial L}{\partial w_{22}^{(1)}} = x_2 * a_2^{(1)} * (1 - a_2^{(1)}) * w_{21}^{(2)} * (\hat{y}_i - y_i)$$

Gradient Descent

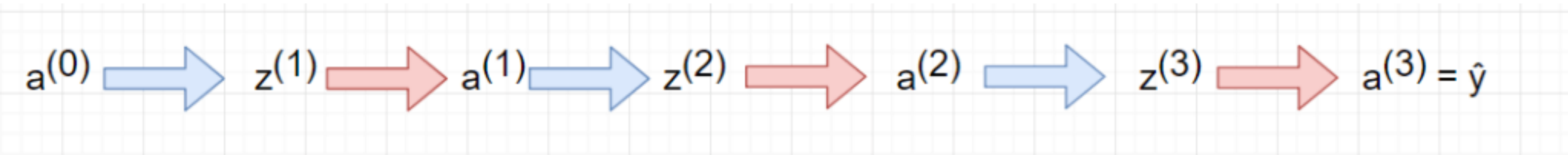


$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial a_1^{(1)}} * \frac{\partial a_1^{(1)}}{\partial x_1} + \frac{\partial L}{\partial a_2^{(1)}} * \frac{\partial a_2^{(1)}}{\partial x_1} = w_{11}^{(1)} * a_1^{(1)} * (1 - a_1^{(1)}) * w_{11}^{(2)} * (y_i - \hat{y}_i) + w_{12}^{(1)} * a_2^{(1)} * (1 - a_2^{(1)}) * w_{21}^{(2)} * (y_i - \hat{y}_i)$$

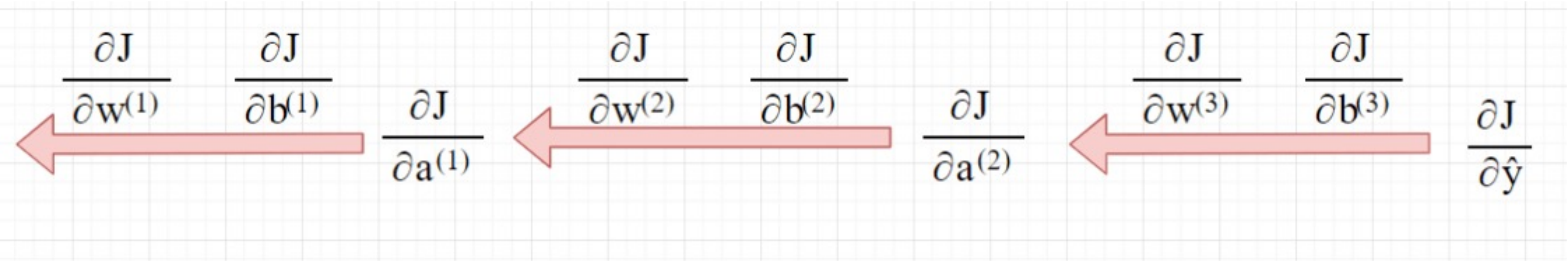
General Model



General Model



Feedforward process



Backpropagation process