

# **Intro to Deep Learning (DL)-based IP / Image Analysis / Computer Vision**

Part 1: DL: FCNNs (Later: DL:CNNs)

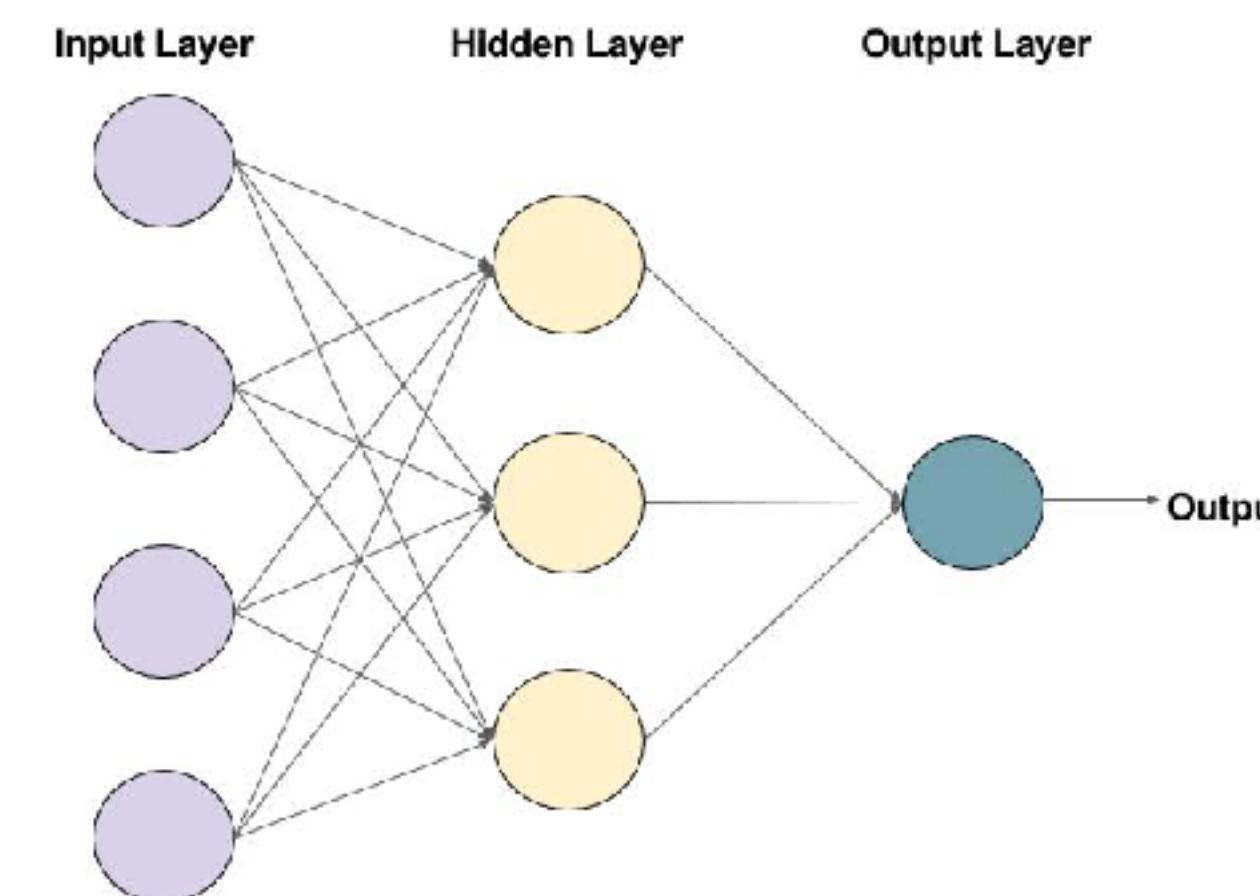
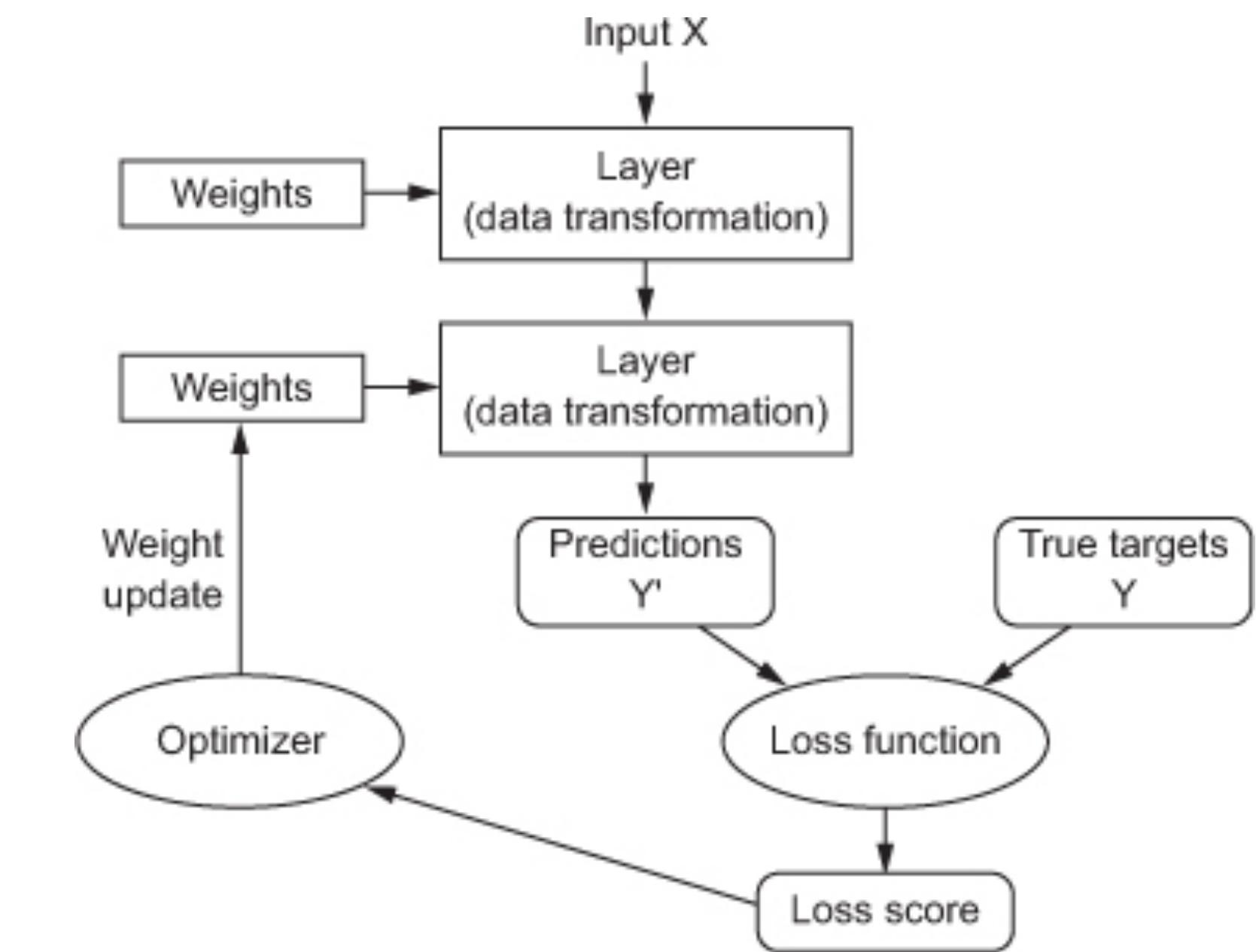
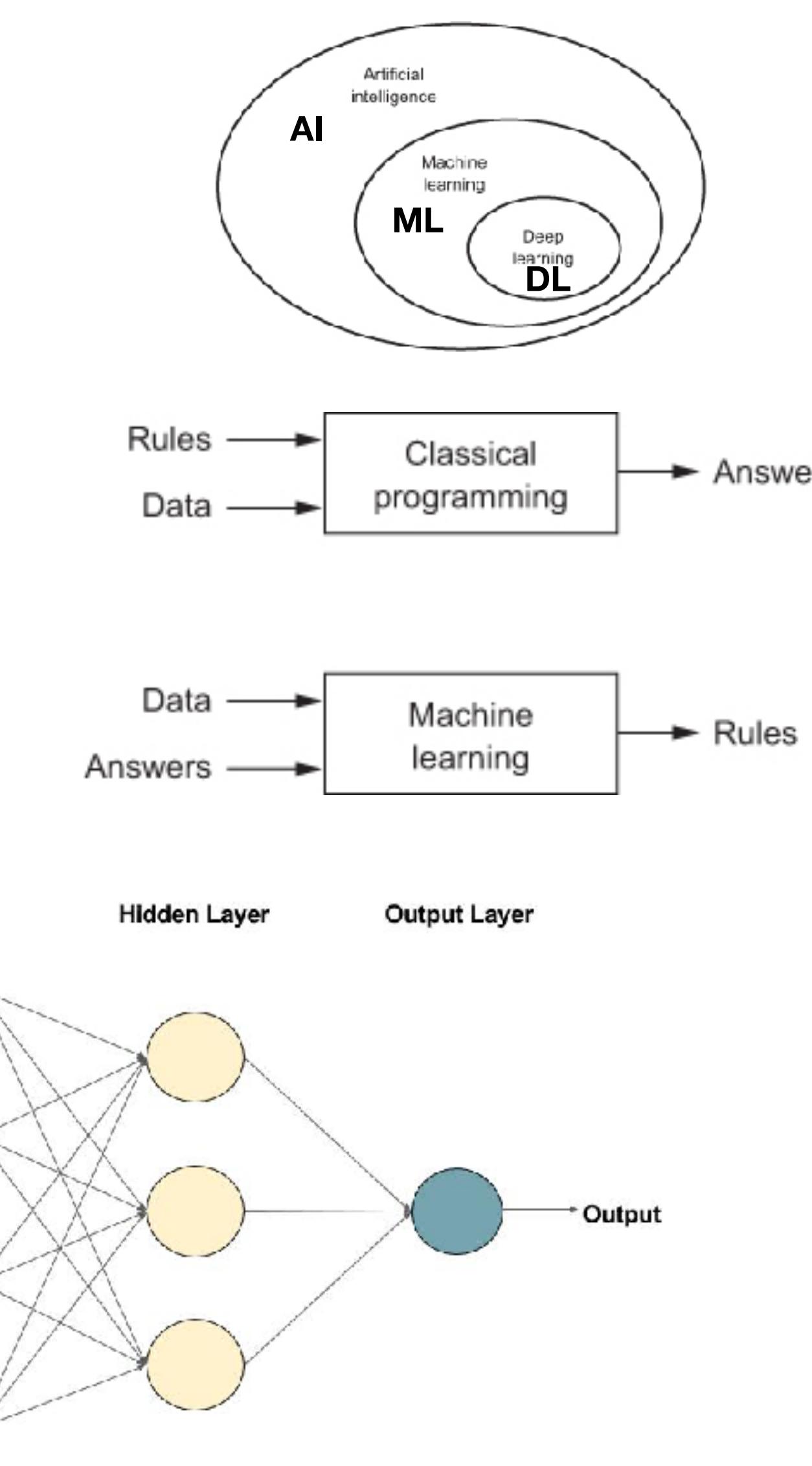
Frank Lindseth, IDI, NTNU

# DL: Overview and FCNNs



# Shallow Fully-Connected Feed-Forward ANNs

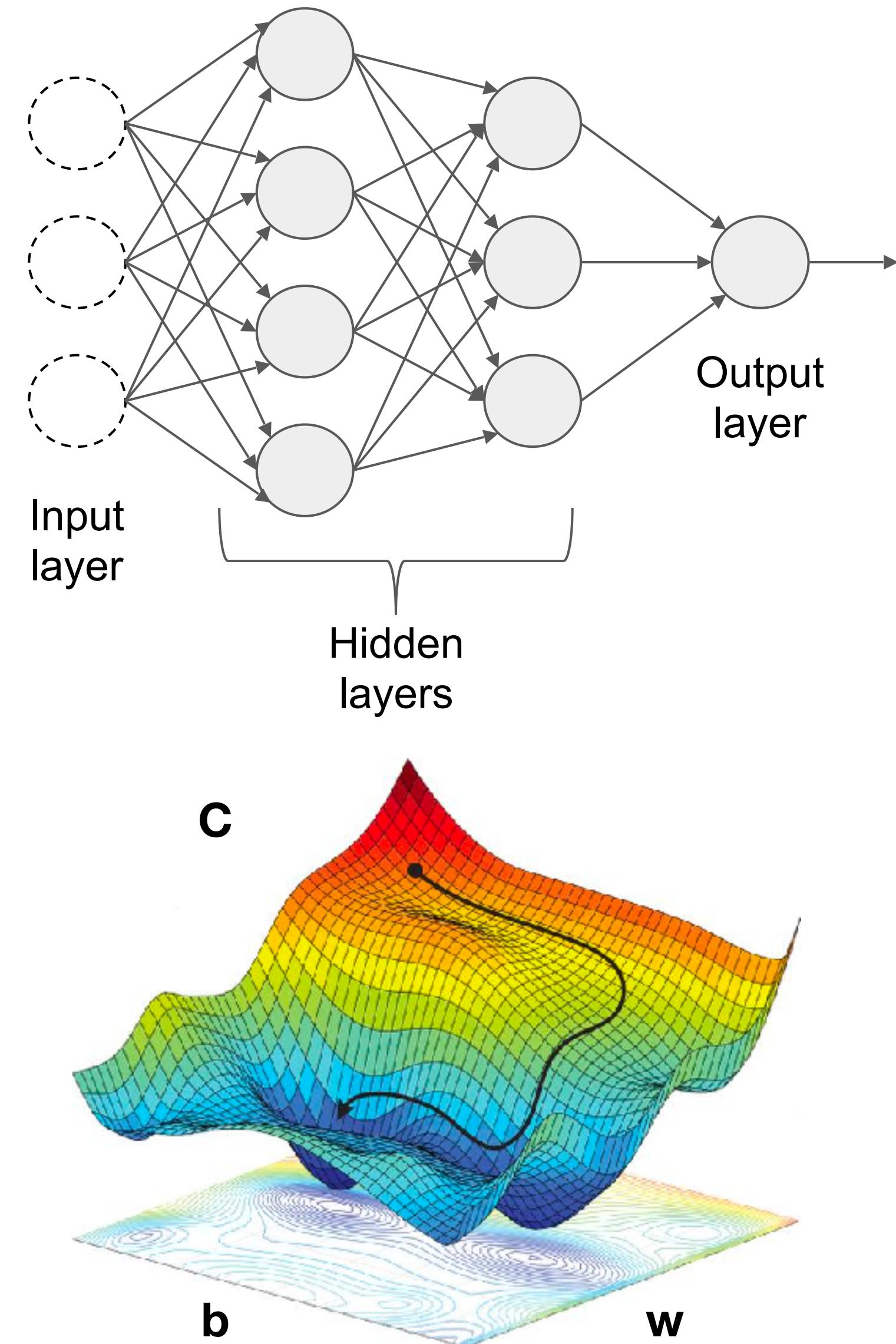
- Artificial Neural Networks (ANNs)
- Artificial Neurons (ANs)
- Activation Functions (AFs)
- Biological Neurons & NN (BNs & BNN)
- Matrix-based Notation
- Numerical Examples



The diagram shows the mathematical calculation for the output of a neuron. The input vector  $x = [x_1, x_2, \dots, x_n]$  is multiplied by the weight vector  $w = [w_1, w_2, \dots, w_n]$ , resulting in the weighted sum  $\sum_i x_i w_i$ . This sum is then added to the bias term  $b$  and passed through the activation function  $f$  to produce the final output  $f\left(b + \sum_{i=1}^n x_i w_i\right)$ .

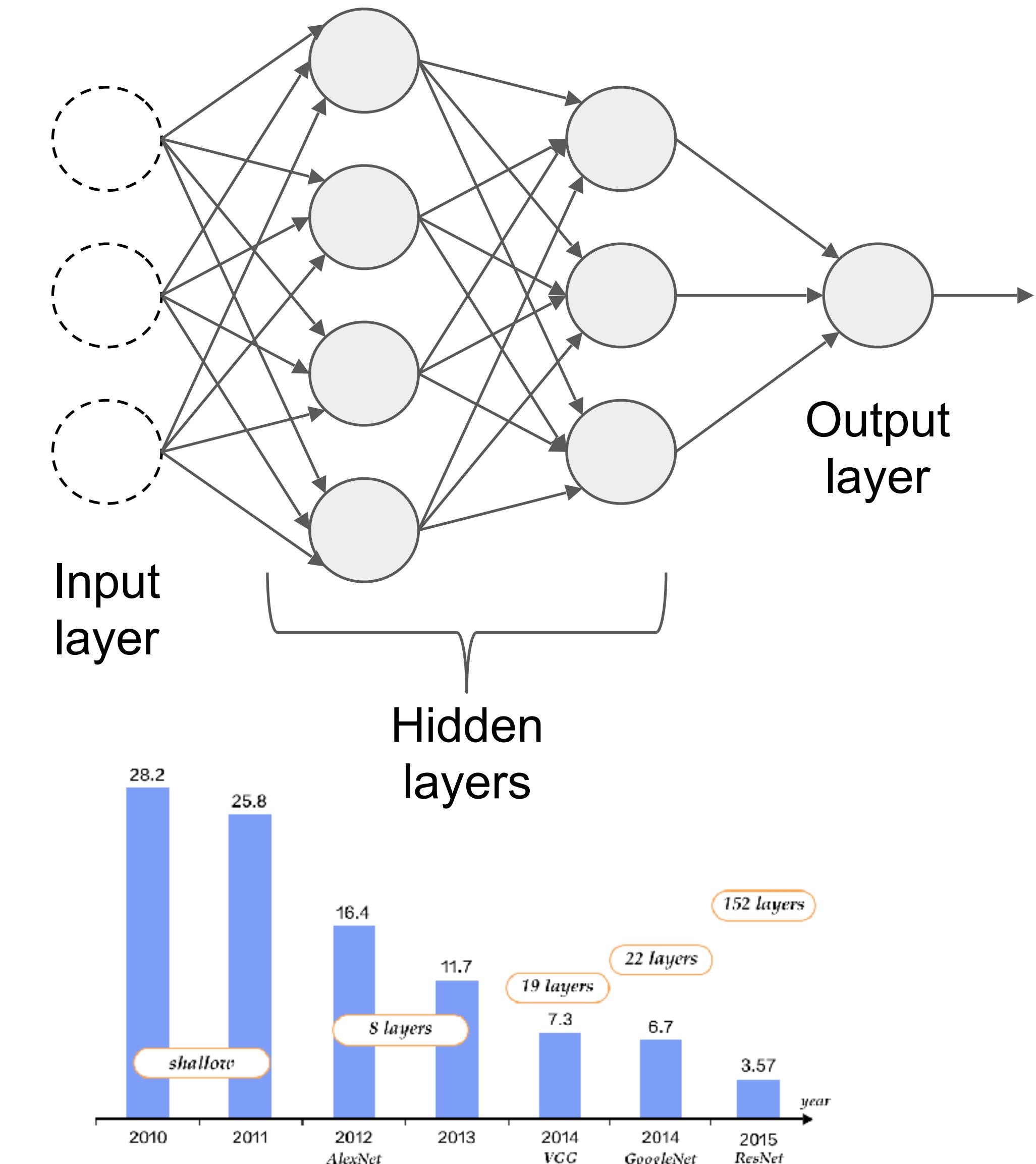
# Key concepts

- Neuron (AN) / Unit
  - Weights, weighted-input ( $z$ )
  - Activations ( $a$ ), activation functions & their derivatives
- ANN (FF, FC, shallow)
  - Layers: (input), hidden and output
  - Different architectures (FF, FC/Dense, shallow)
- Inference (use trained network), forward pass of (new unseen) input
- Learning
  - Labeled-data-split (GT): train, validation and test data
  - Cost / Error / Loss functions
  - Training, GD, update-rule, back-propagation of error, chain rule



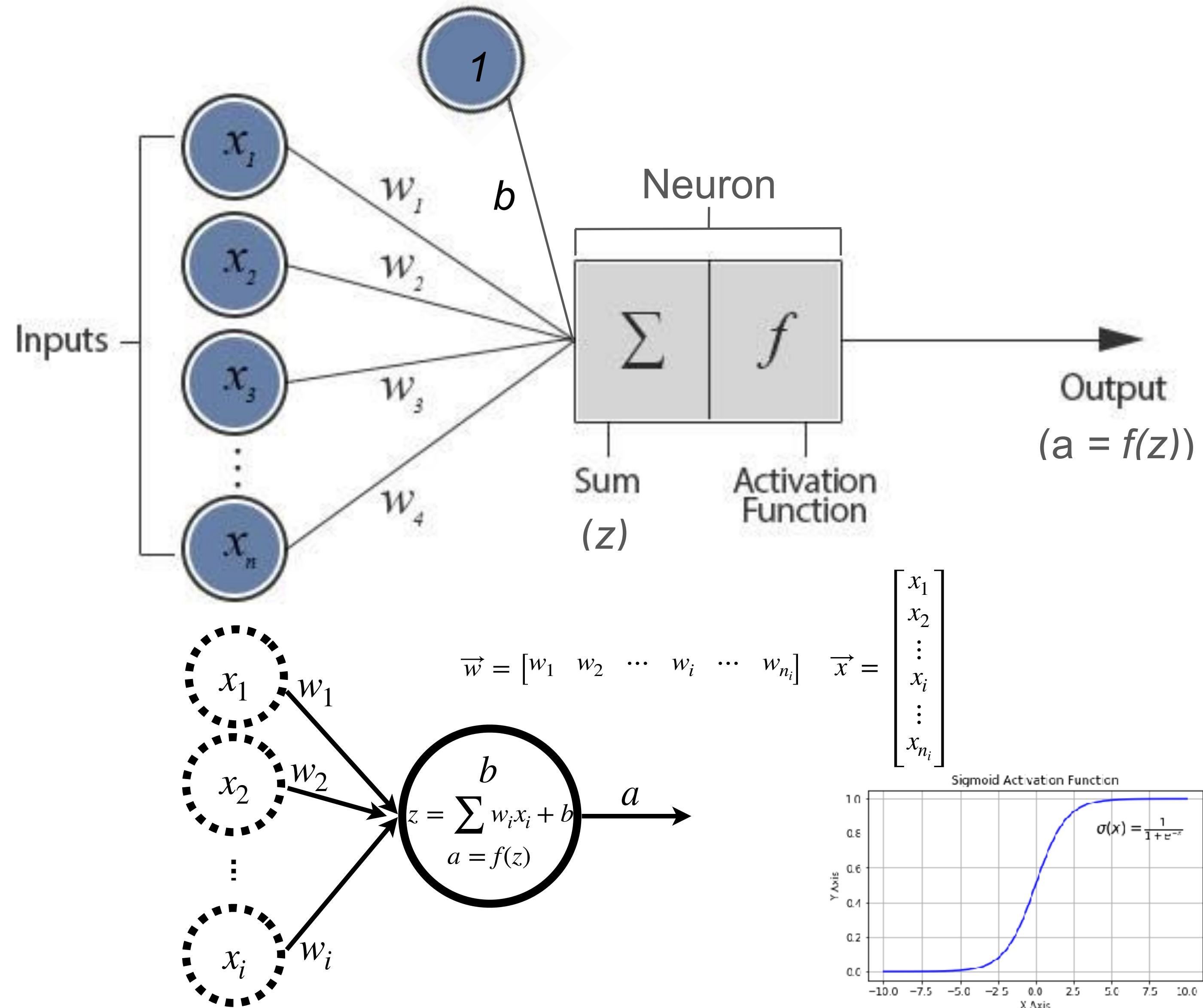
# Artificial Neural Networks (ANNs)

- Network **architecture** (model): shallow, fully-connected, feed-forward ANNs
  - An ANN consists of multiple **layers**
  - A layer consists of multiple **neurons** (units)
  - **Shallow** (vs. deep) ANNs: «a few» hidden layers
  - **Fully-connected** (vs. ConvNets) ANN: a given neuron is connected to all neurons in previous and next layer.
  - **Feed-Forward** (vs. Recurrent) ANNs: only forward links

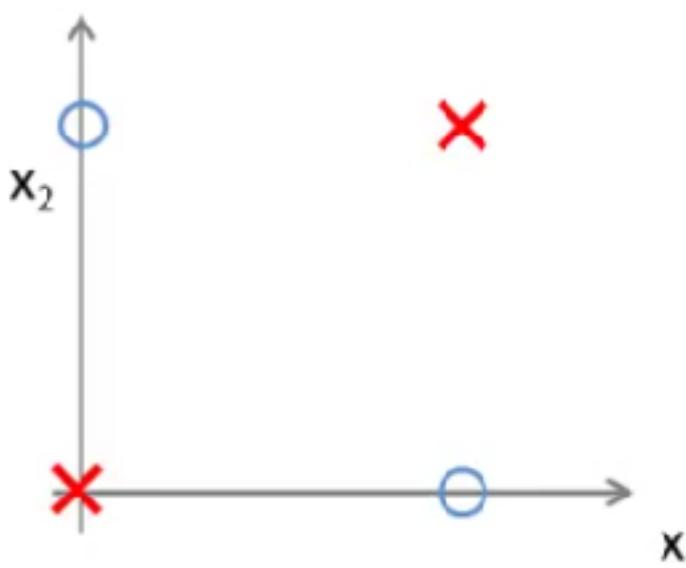


# Artificial Neurons (ANs)

- Two operations:
  - weighted input:  $z = w^*x + b$
  - activation:  $a = f(z)$
- Weights and Bias are learnable parameters.



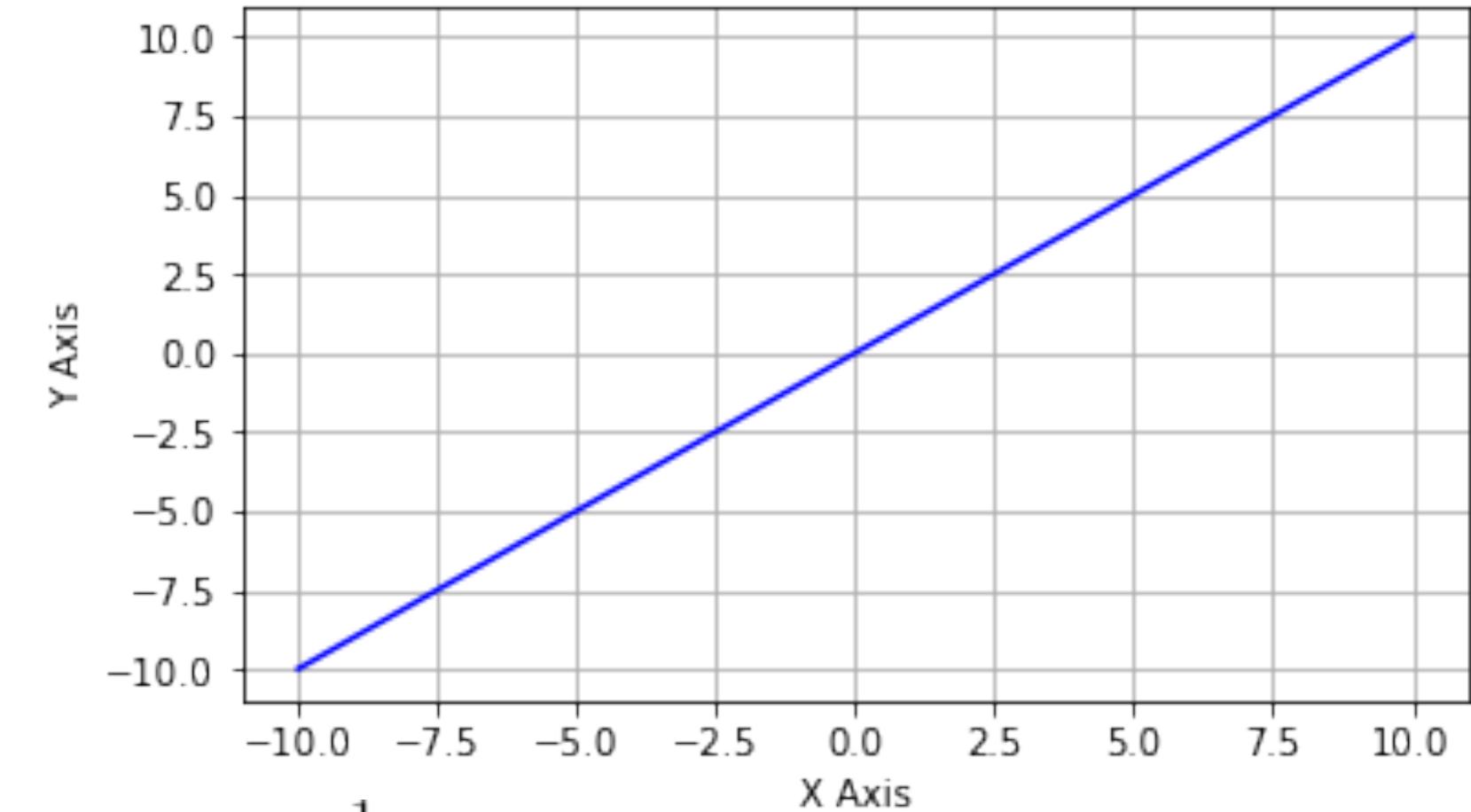
# Activation Functions (AFs)



- Types: Linear vs. Non-Linear Activation Functions
- Weights and Bias transform the input signal linearly (XOR)
- A non-linearity (TF) allows us to learn arbitrarily complex transformations between the input and the output.
  - solve problems where a non-linear decision boundary is needed to separate the data
  - also used to squash the output of the neural network to be within certain bounds.
- Still an active area of research: want TFs that makes the neural network learn better and faster (sigmoid, tanh, relu, lrelu, prelu, swish, etc.)
  - GD and Backpropagation uses **gradients** in the learning process, i.e. **partial derivatives** of the loss function w.r.t the weights/biases are used to update the weights/biases.

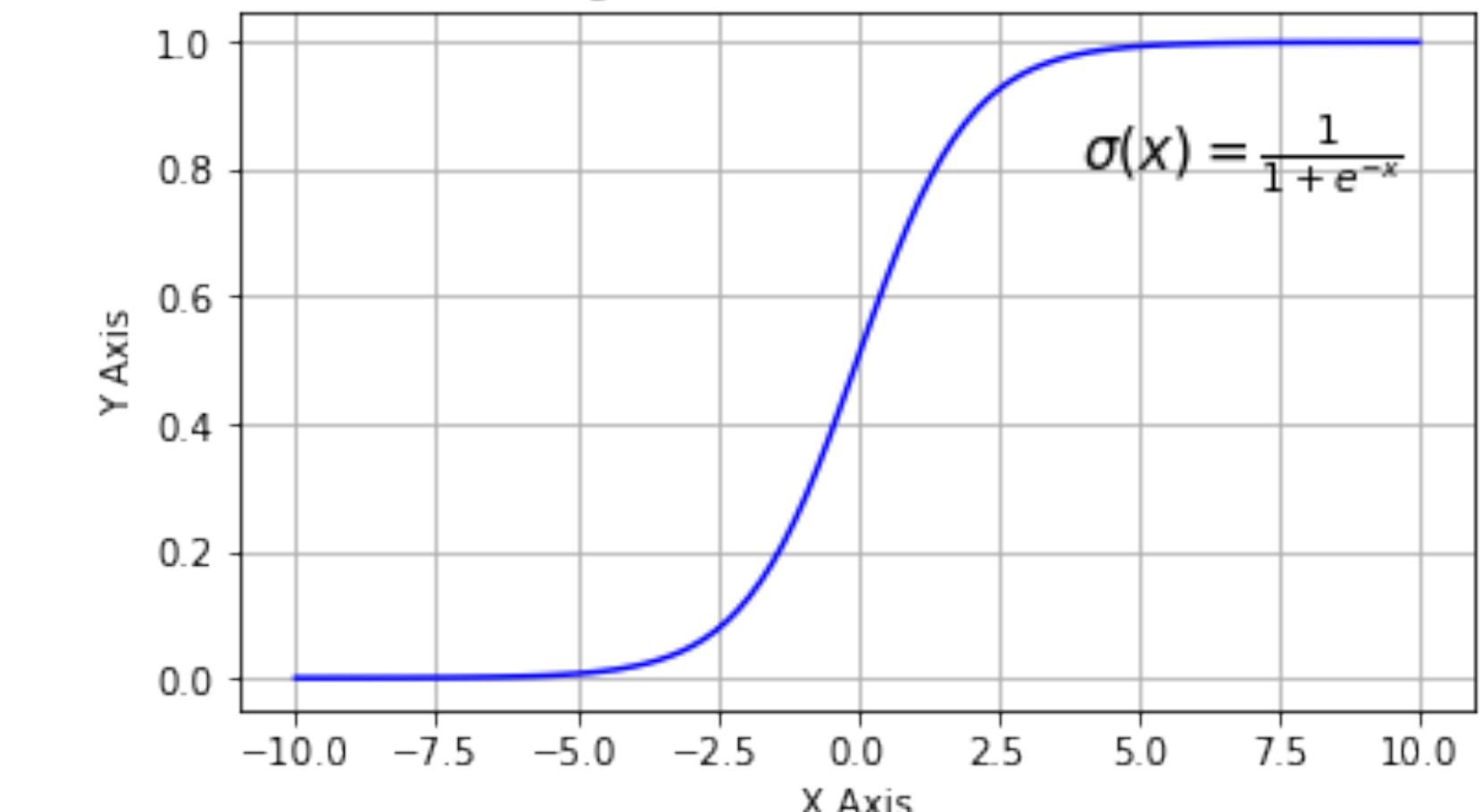
$$f(x) = x$$

Linear Activation Function



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid Activation Function

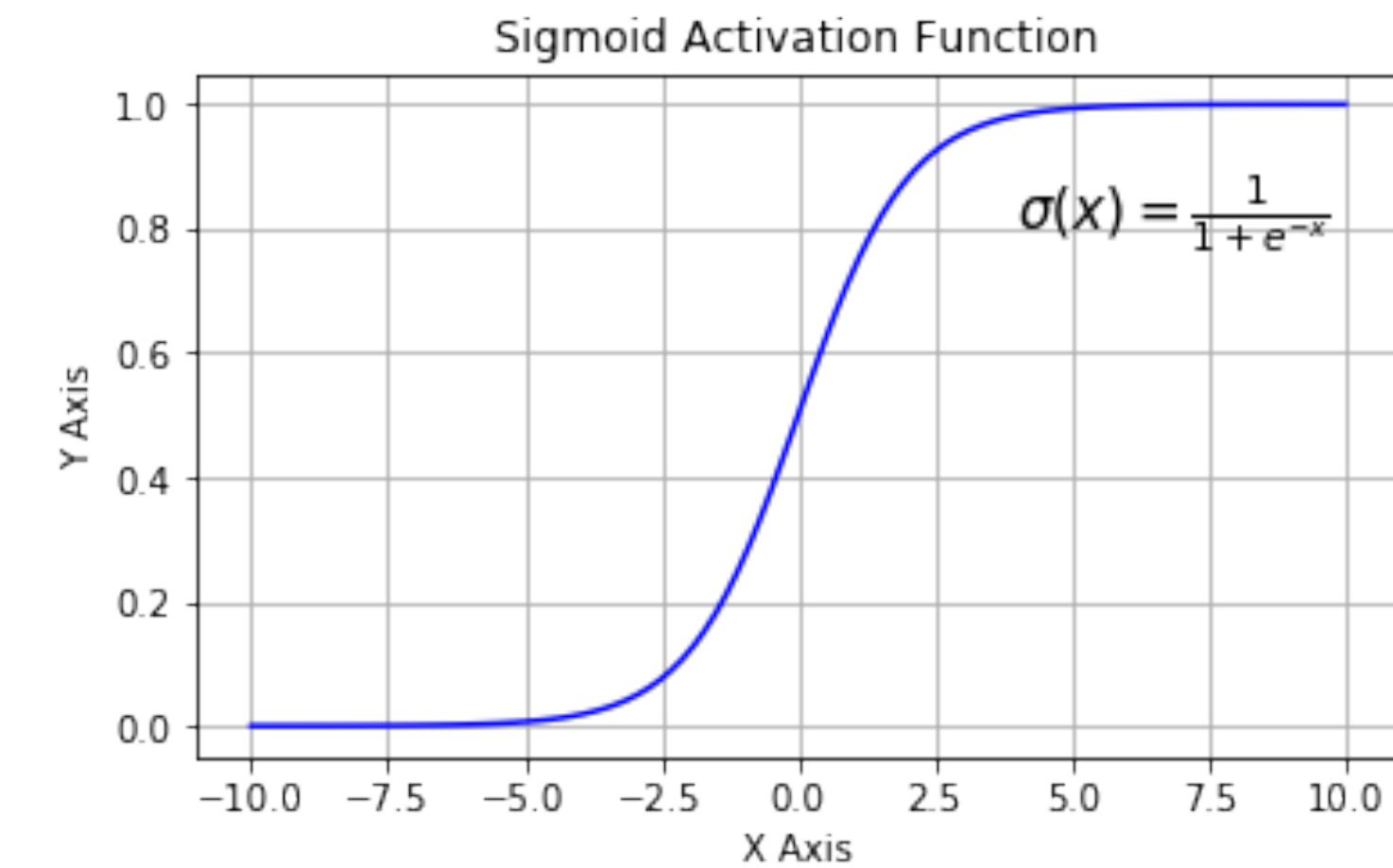


# Sigmoid (Logistic) AF

$$f(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

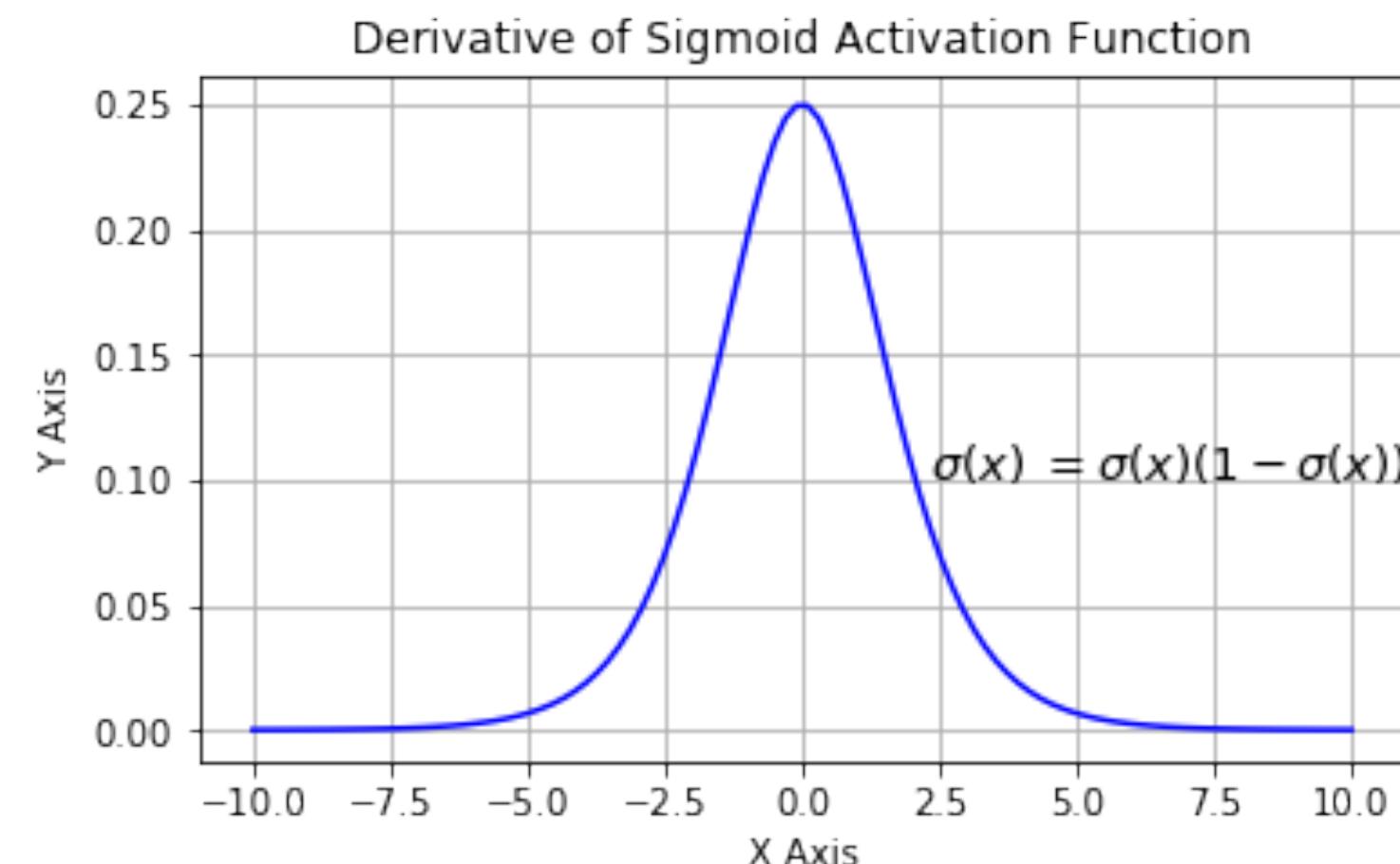
- **Squashes** input between 0 (large negative numbers) and 1 (large positive numbers).
- Can be used in the output layer where our end goal is to predict «**probability**».
- Drawbacks:
  - **Vanishing gradients:**



- Sigmoid output close to 0/1 (flat) -> gradient / derivative close to 0 -> saturated neurons -> weights of neurons (or connected neurons) do not update -> network do not backpropagate, i.e. learn.

- Not zero centered:

- Computationally expensive:



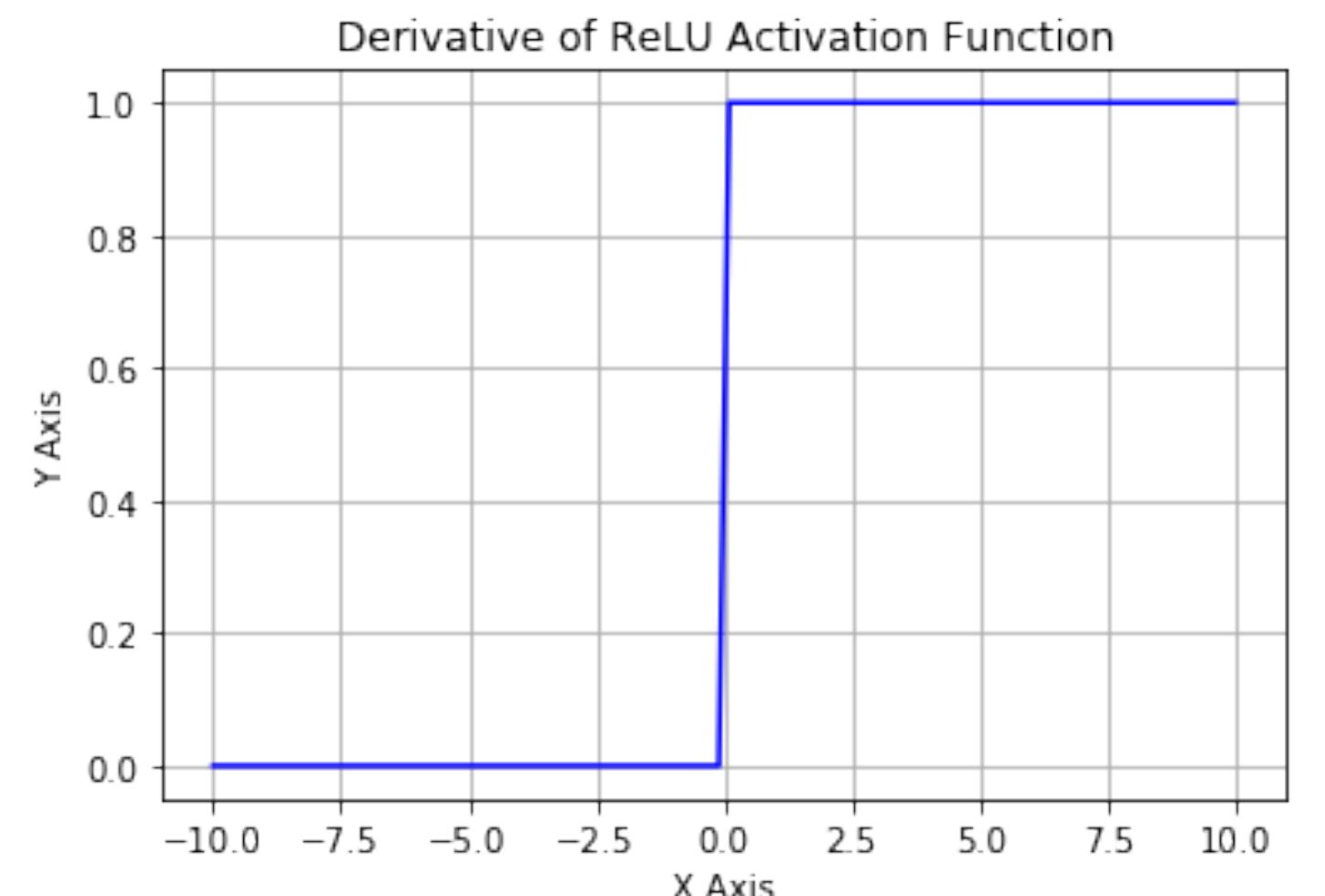
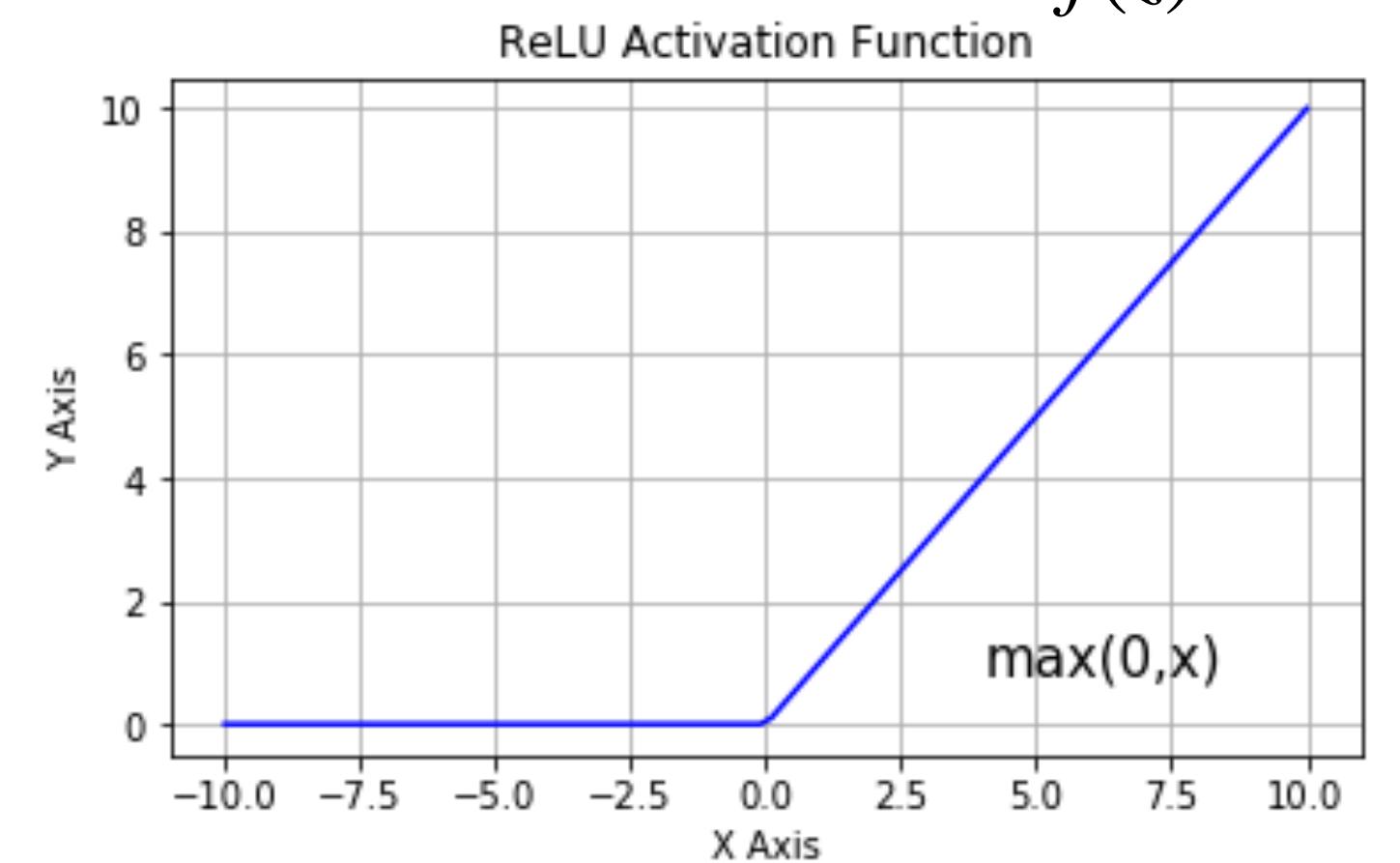
$$f'(z) = f(z)(1 - f(z))$$

# ReLU (Rectified Linear Unit) AF

- ReLU is half-rectified (when the input  $x < 0$  the output is 0 and if  $x > 0$  the output is  $x$ ).
- Network converge much faster (does not saturate, resistant to the vanishing gradient problem (positive region))
- ReLU is computationally very efficient.
- Drawbacks
  - Not zero-centered
  - if  $x < 0$  during the forward pass, the neuron remains **inactive** and it **kills** the gradient during the backward pass. Thus weights do not get updated, and the network does not learn. When  $x = 0$  the slope is undefined.

$$f(x) = \max(0, x)$$

$$f(z) = \max(0, z)$$



# Softmax AF

- We sometimes want that the output should express probabilities (e.g. image classification problems)
- Use Softmax to scale the outputs
- Want outputs between 0 and 1 and should sum to 1

$$f(z) = \begin{bmatrix} f(z_1) \\ f(z_2) \\ \vdots \\ f(z_j) \\ \vdots \\ f(z_m) \end{bmatrix}, f(z_j) = \frac{e^{z_j}}{\sum_{i=1}^m e^{z_i}}$$

node	x	$e^x$	$e^x / \text{sum}(e^x)$
1	2	7.39	0.71
2	1	2.72	0.26
3	-1	0.37	0.04
<b>total</b>		10.48	1.00

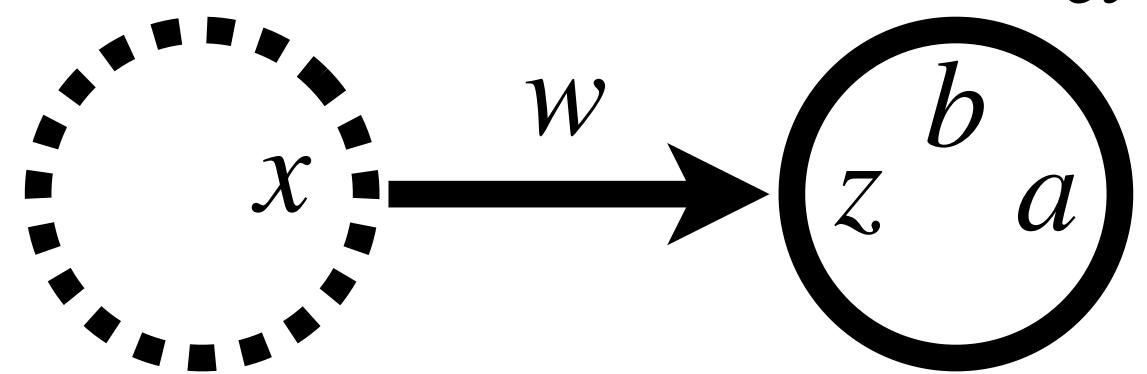


# Notation & «Simple» examples

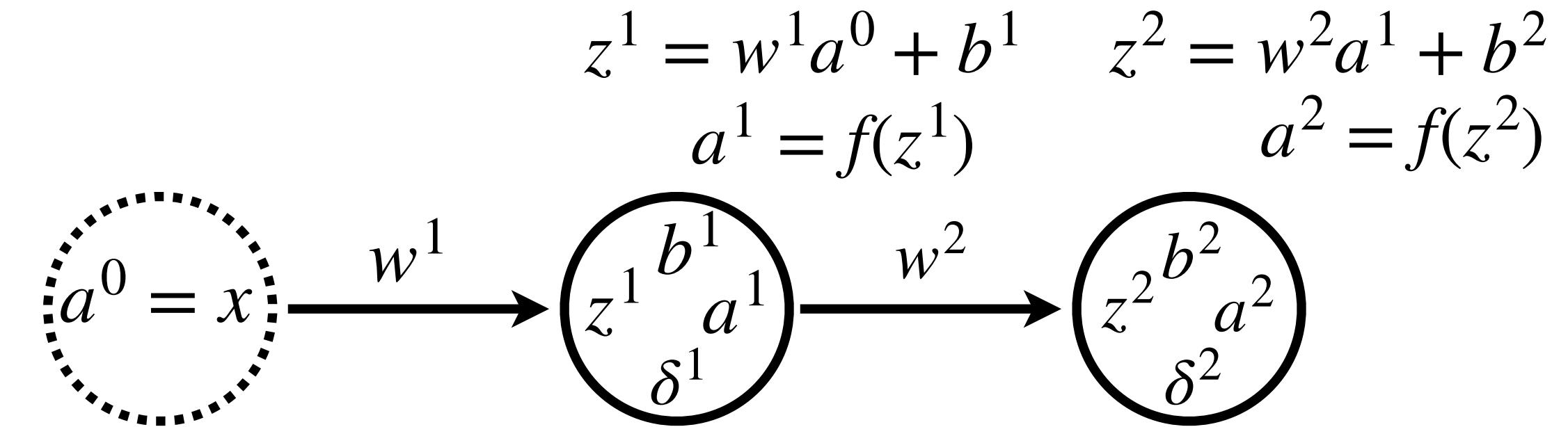
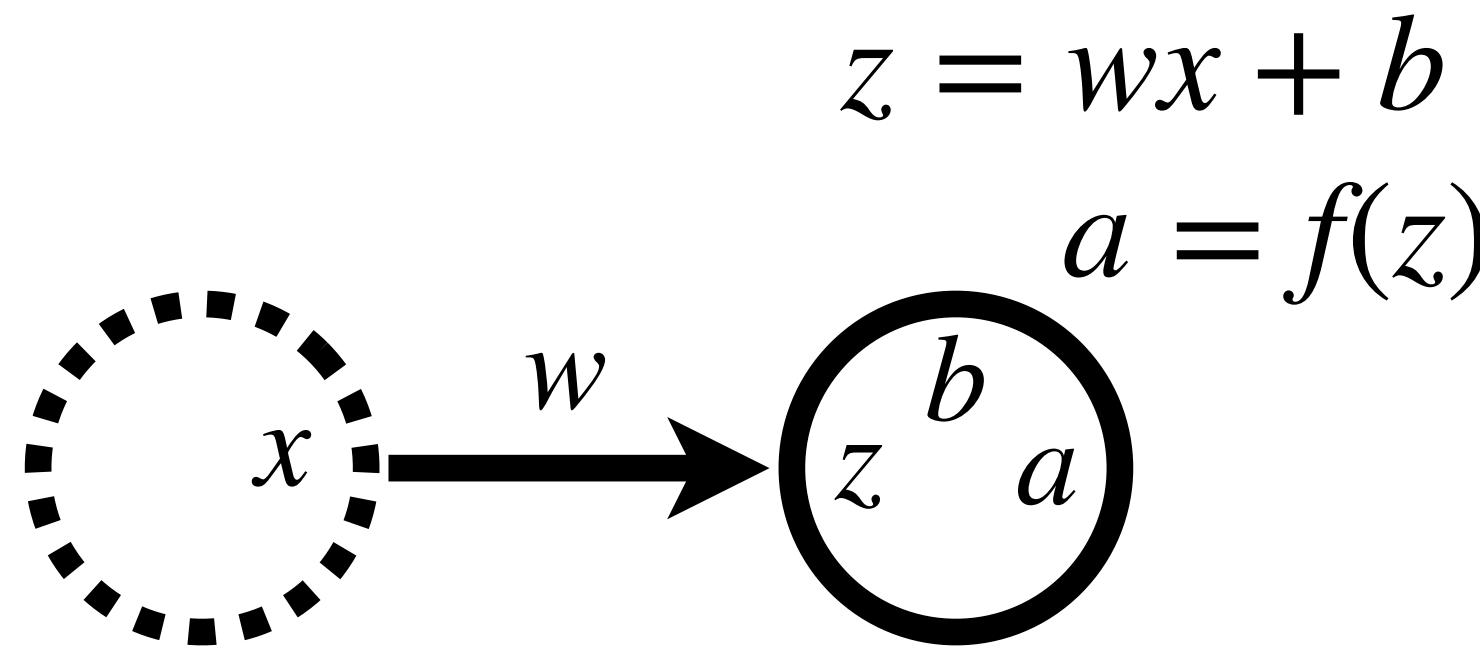
# Notation & «Simple» examples

$$z = wx + b$$

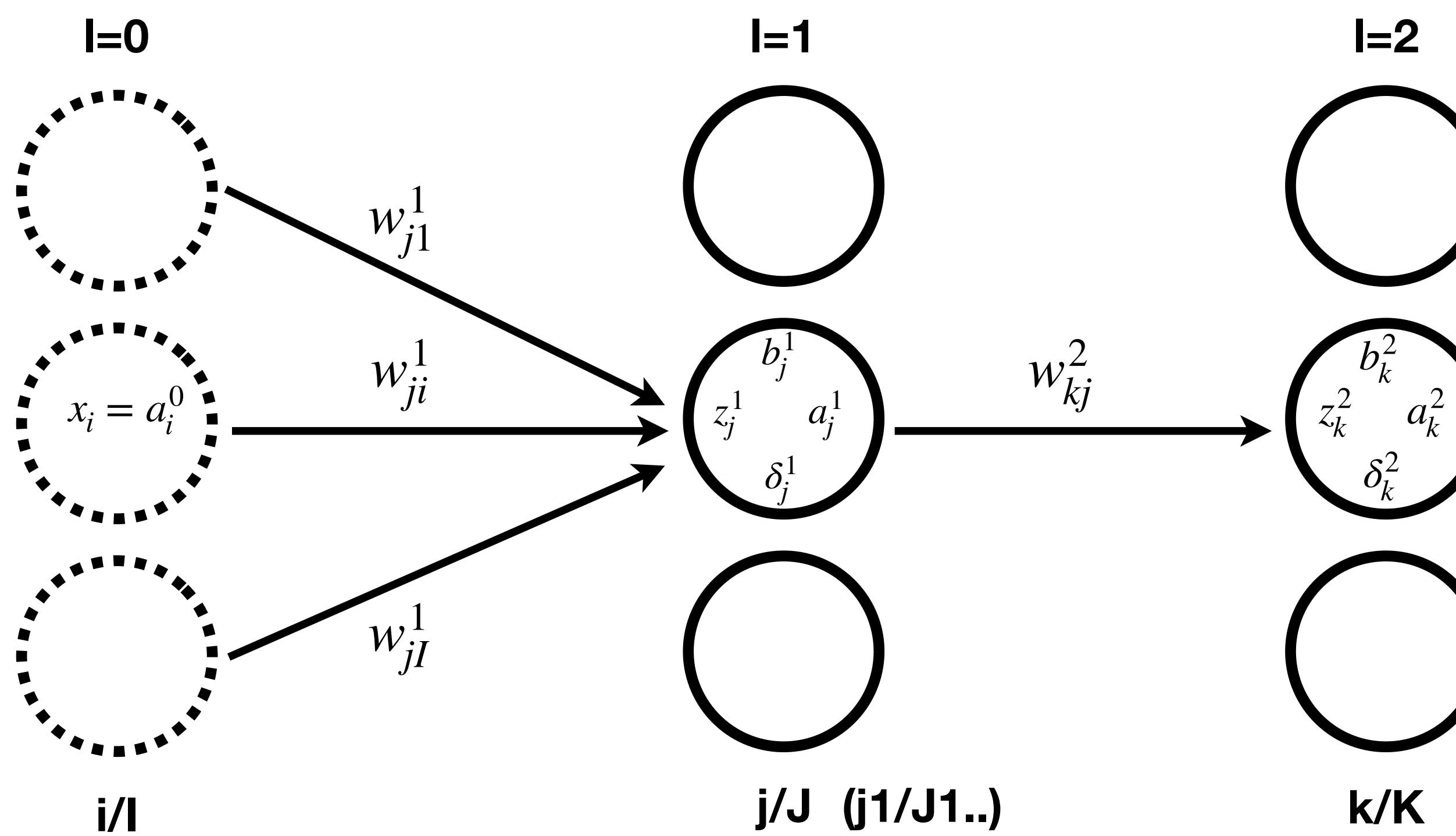
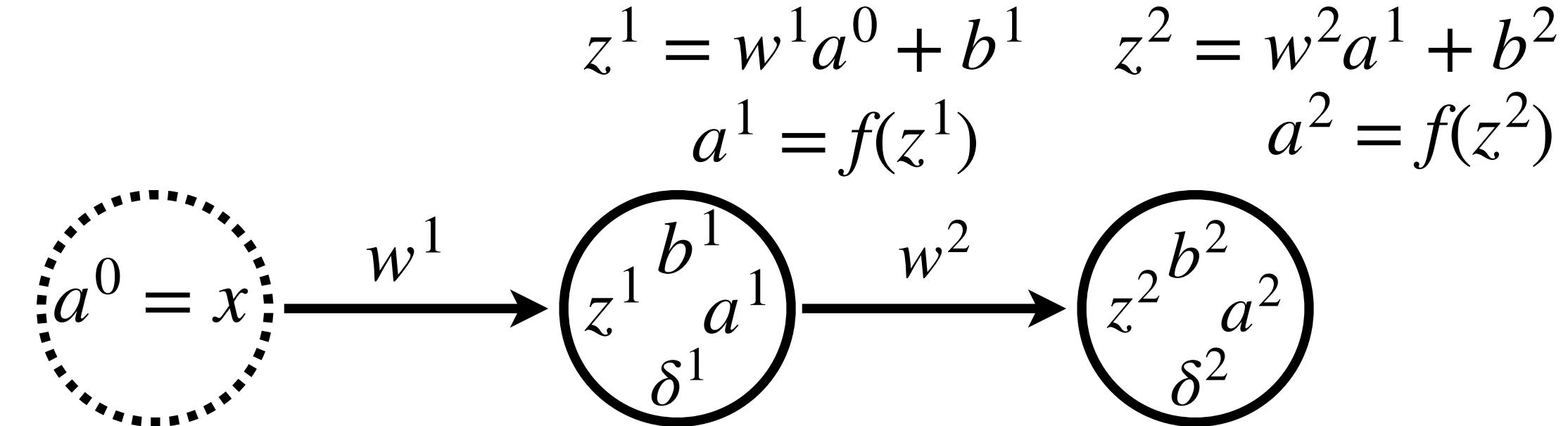
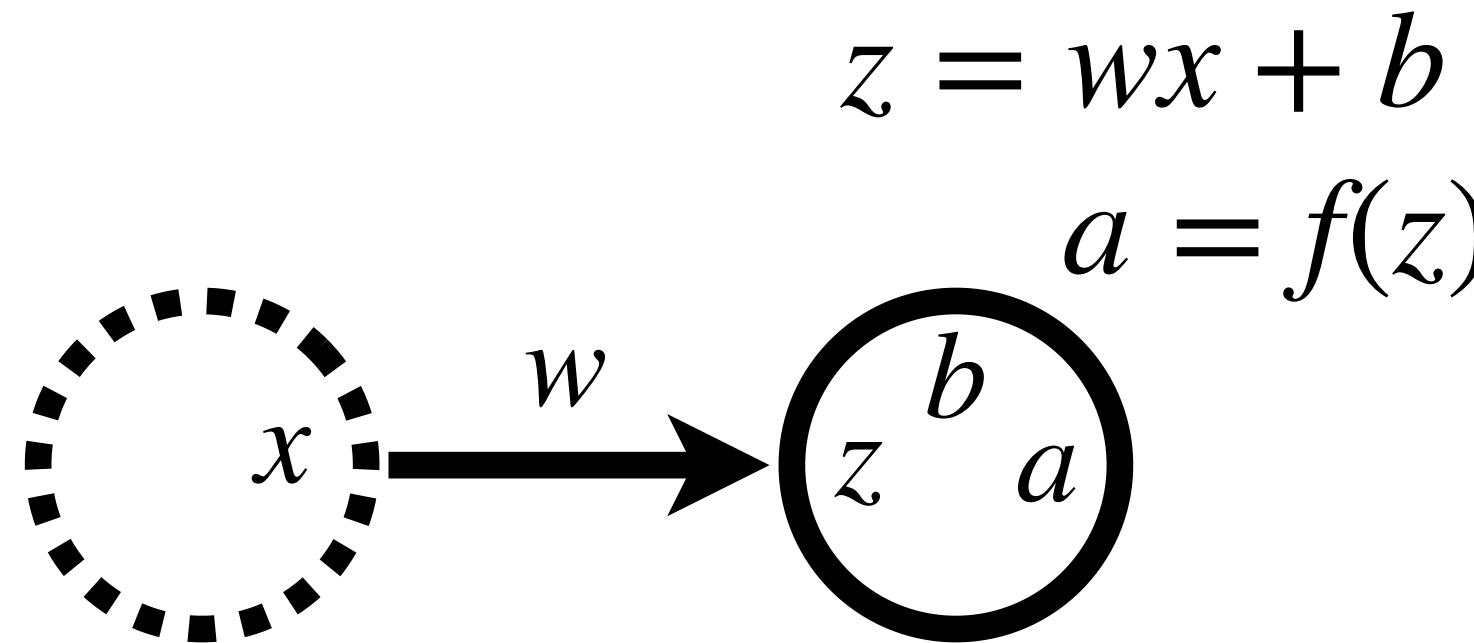
$$a = f(z)$$



# Notation & «Simple» examples



# Notation & «Simple» examples

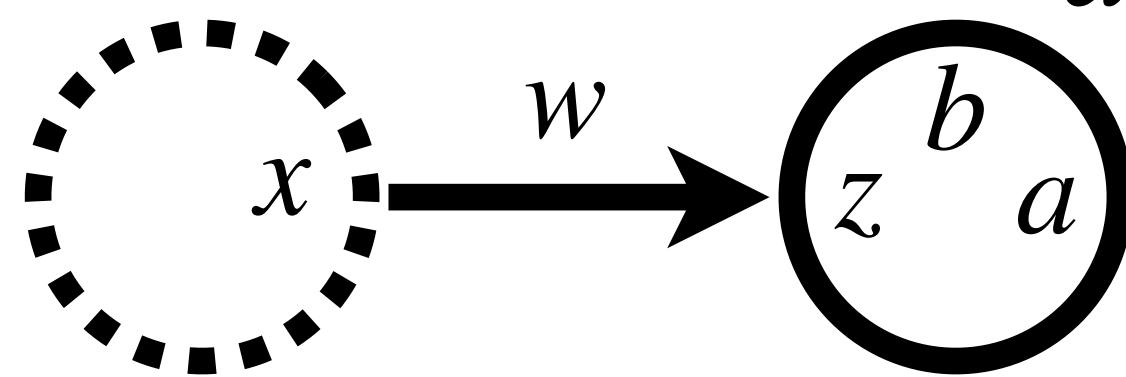


$w_{ji}^l$  is the **weight** to neuron  $j$  in layer  $l$  from neuron  $i$  in layer  $l-1$   
 $b_j^l$  is the **bias** of neuron  $j$  in layer  $l$   
 $z_j^l$  is the **weighted input** of neuron  $j$  in layer  $l$   
 $a_j^l$  is the **activation** of neuron  $j$  in layer  $l$   
 $\delta_j^l$  is the **error** of neuron  $j$  in layer  $l$

# Notation & «Simple» examples

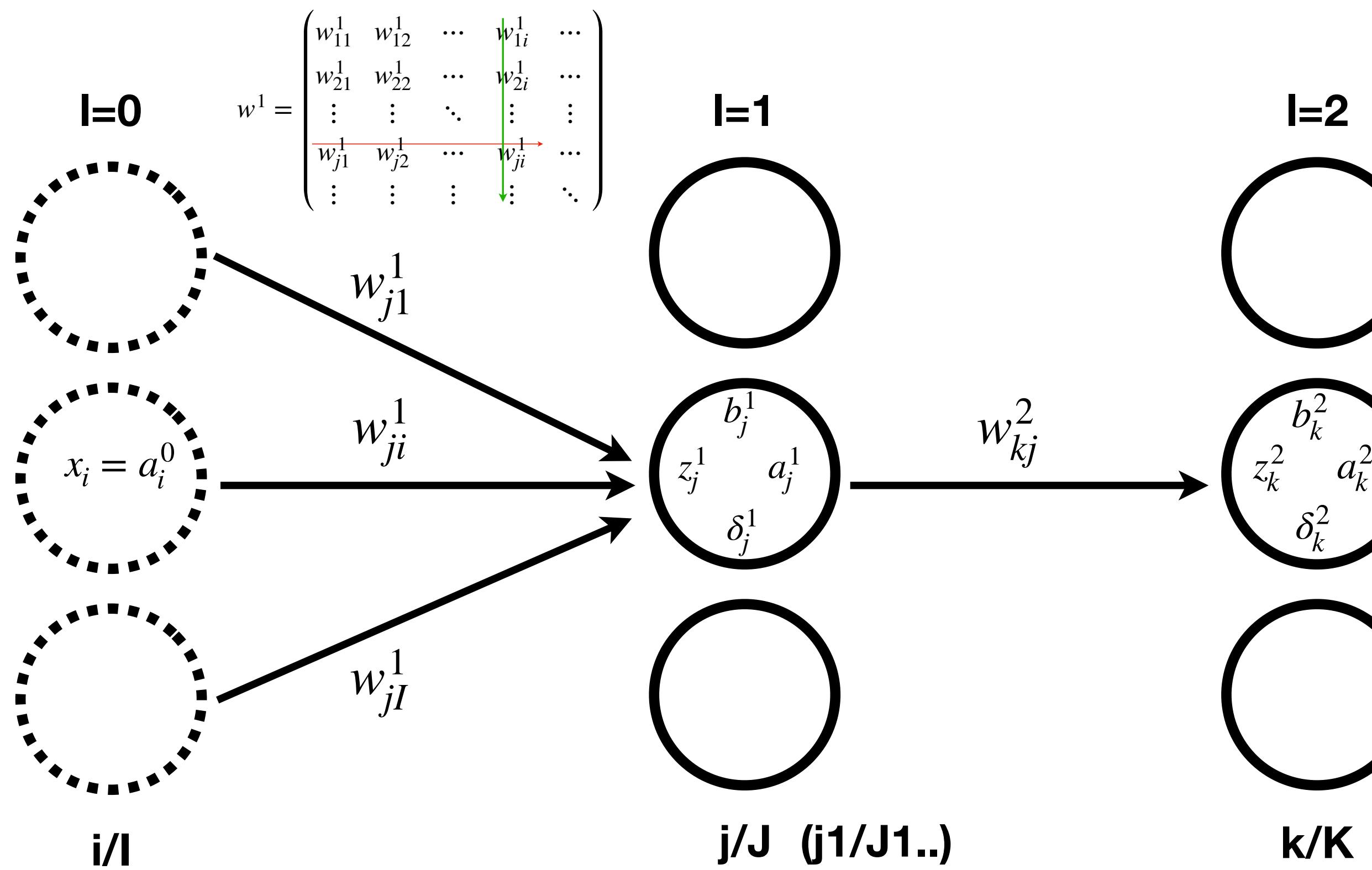
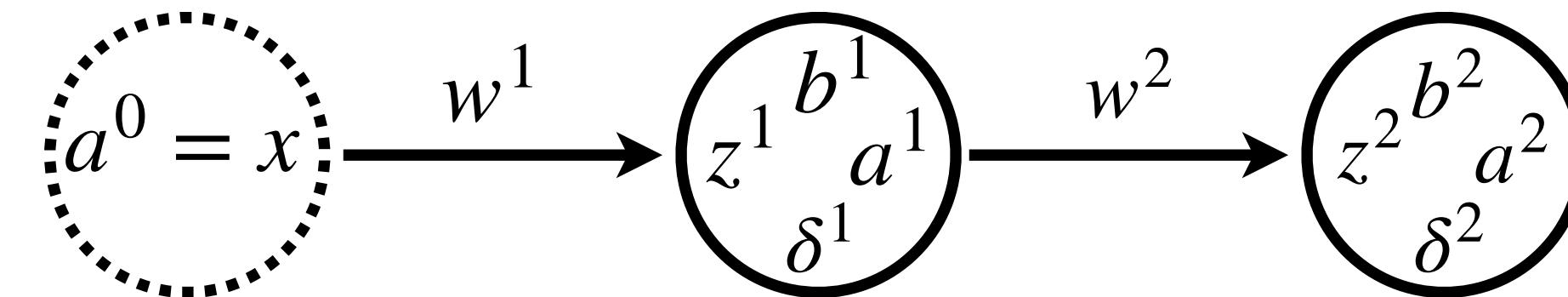
$$z = wx + b$$

$$a = f(z)$$



$$z^1 = w^1 a^0 + b^1 \quad z^2 = w^2 a^1 + b^2$$

$$a^1 = f(z^1) \quad a^2 = f(z^2)$$

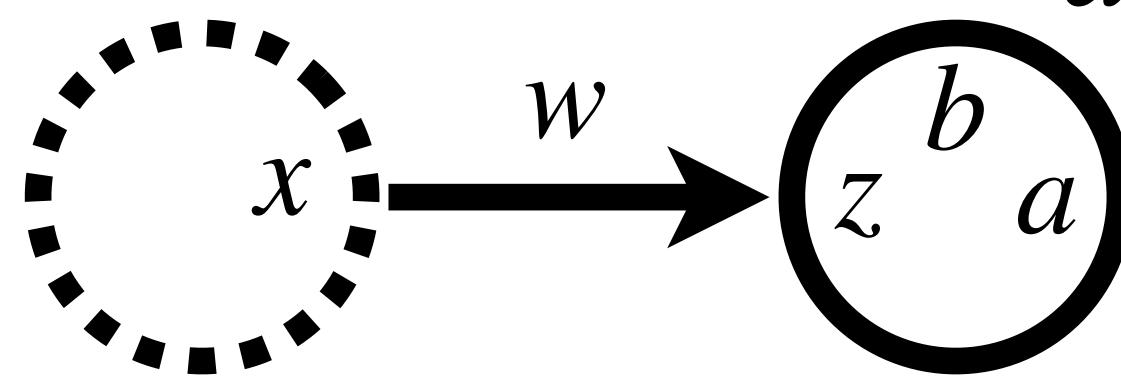


- $w_{ji}^l$  is the **weight** to neuron  $j$  in layer  $l$  from neuron  $i$  in layer  $l-1$
- $b_j^l$  is the **bias** of neuron  $j$  in layer  $l$
- $z_j^l$  is the **weighted input** of neuron  $j$  in layer  $l$
- $a_j^l$  is the **activation** of neuron  $j$  in layer  $l$
- $\delta_j^l$  is the **error** of neuron  $j$  in layer  $l$

# Notation & «Simple» examples

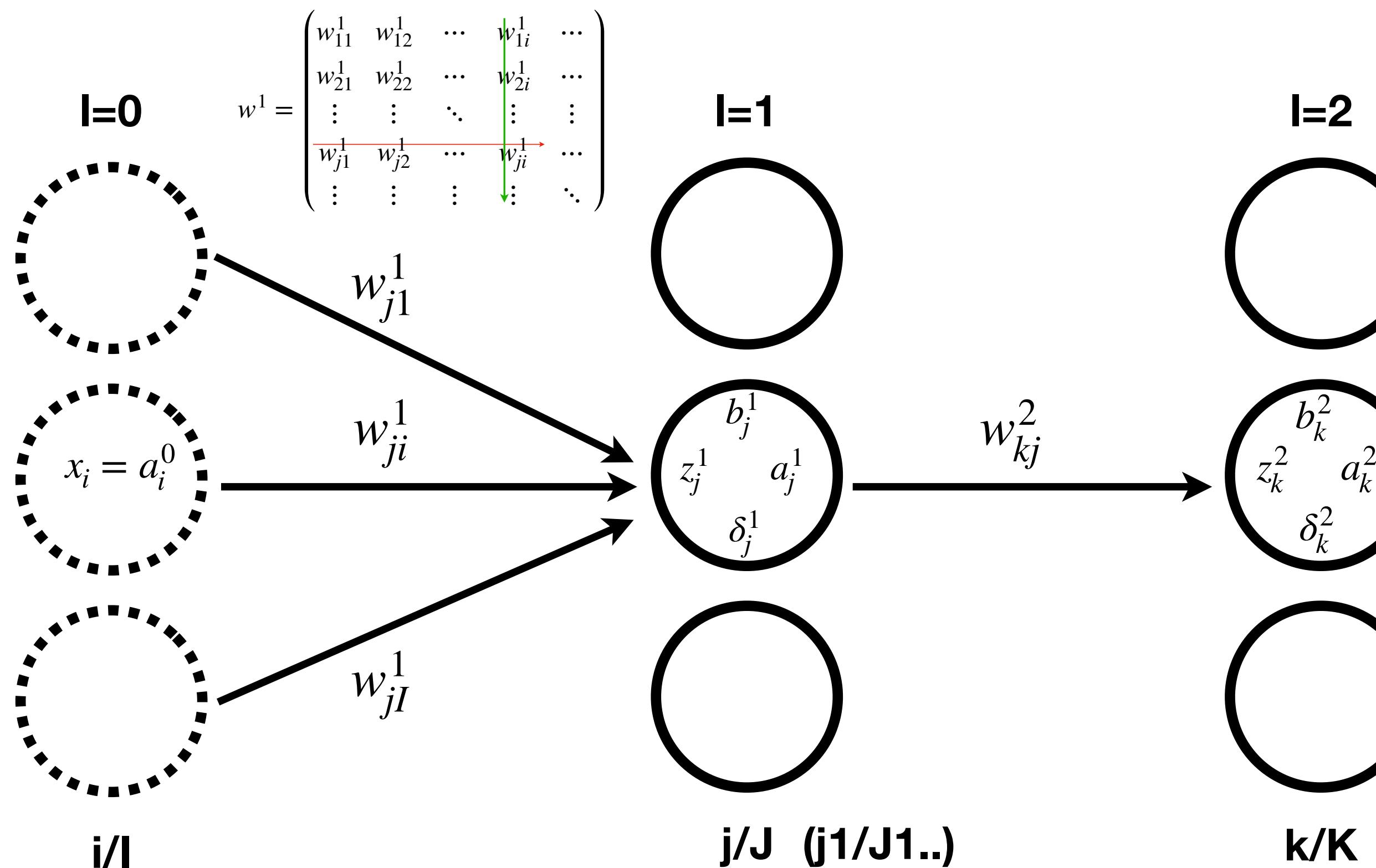
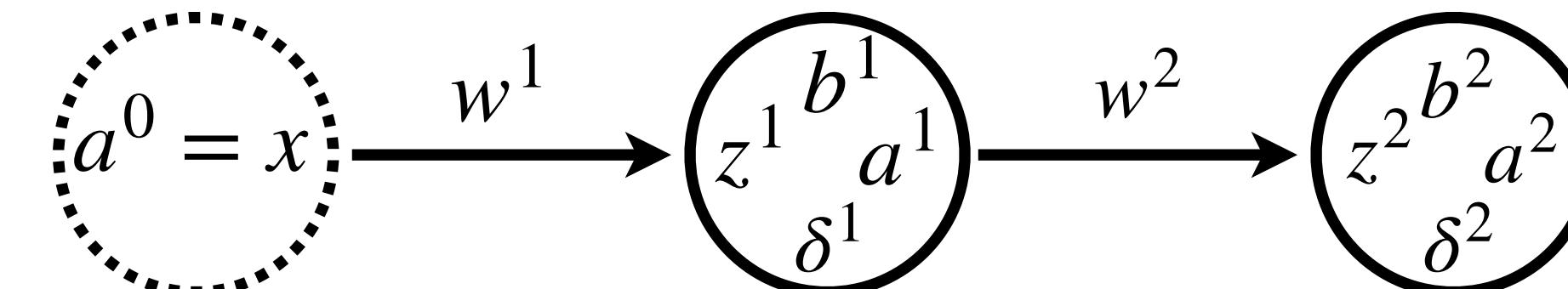
$$z = wx + b$$

$$a = f(z)$$



$$z^1 = w^1 a^0 + b^1 \quad z^2 = w^2 a^1 + b^2$$

$$a^1 = f(z^1) \quad a^2 = f(z^2)$$



$w_{ji}^l$  is the **weight** to neuron **j** in layer **l** from neuron **i** in layer **l-1**  
 $b_j^l$  is the **bias** of neuron **j** in layer **l**  
 $z_j^l$  is the **weighted input** of neuron **j** in layer **l**  
 $a_j^l$  is the **activation** of neuron **j** in layer **l**  
 $\delta_j^l$  is the **error** of neuron **j** in layer **l**

**For each training example n:**

**Input activation:**  $a^{n,0} = x^n$

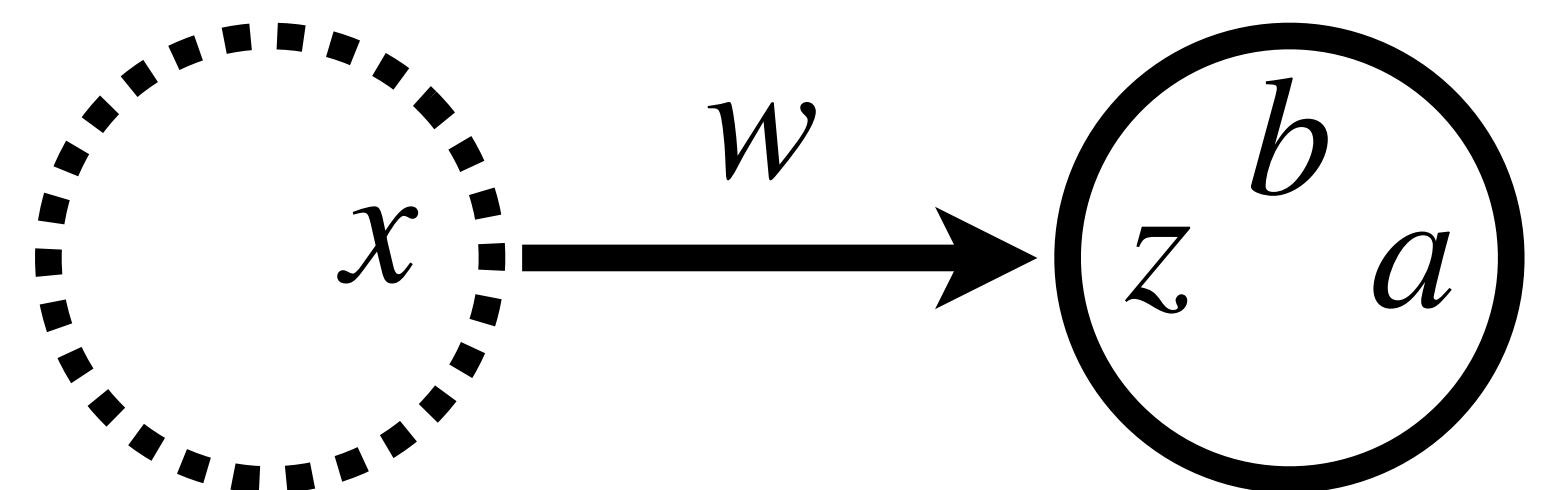
**Feed-forward:**  $z^{n,l} = w^l a^{n,l-1} + b^l$

$a^{n,l} = f(z^{n,l}) \quad l = 1, 2, \dots, L$

**Output error:**  $\delta^{n,L} = \nabla_a C_n \odot f'(z^{n,L})$

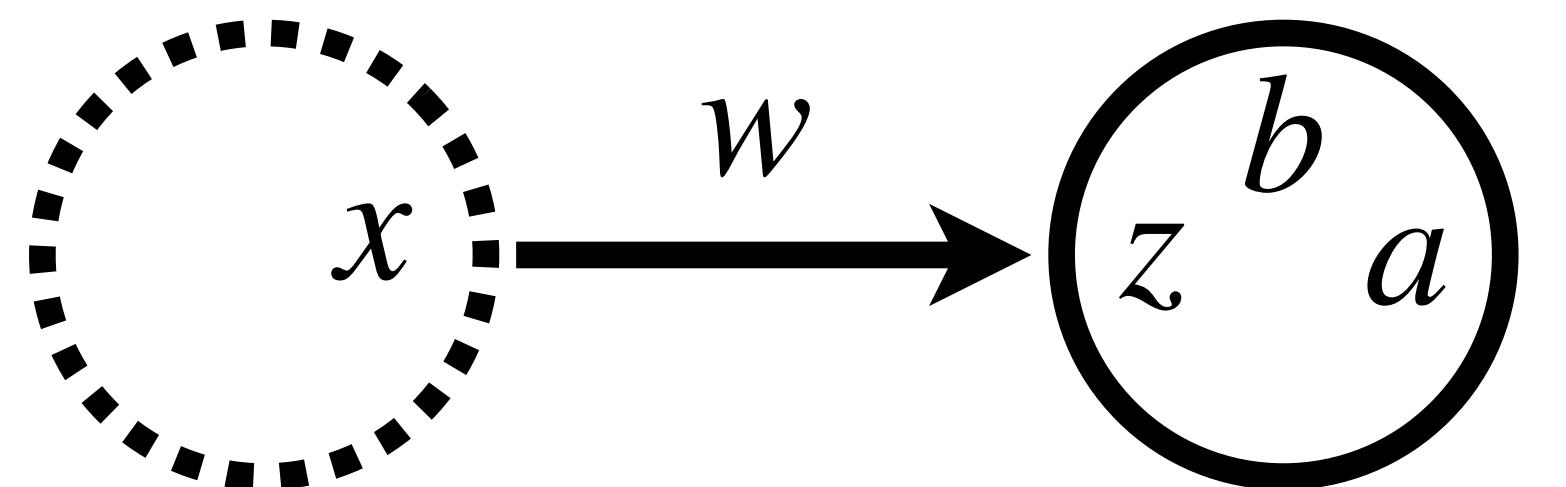
**Back-propagate:**  $\delta^{n,l} = ((w^{l+1})^T \delta^{n,l+1} \odot f'(z^{n,l}))$

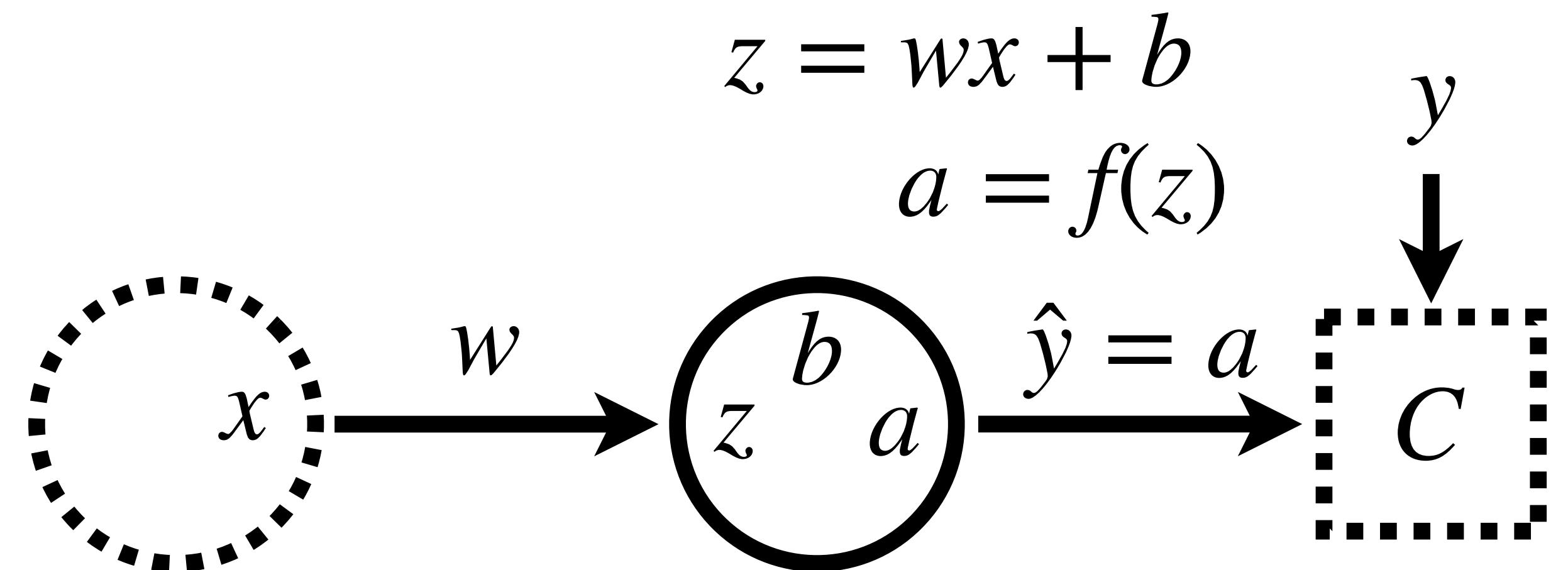
$l = L - 1, L - 2, \dots, 1$

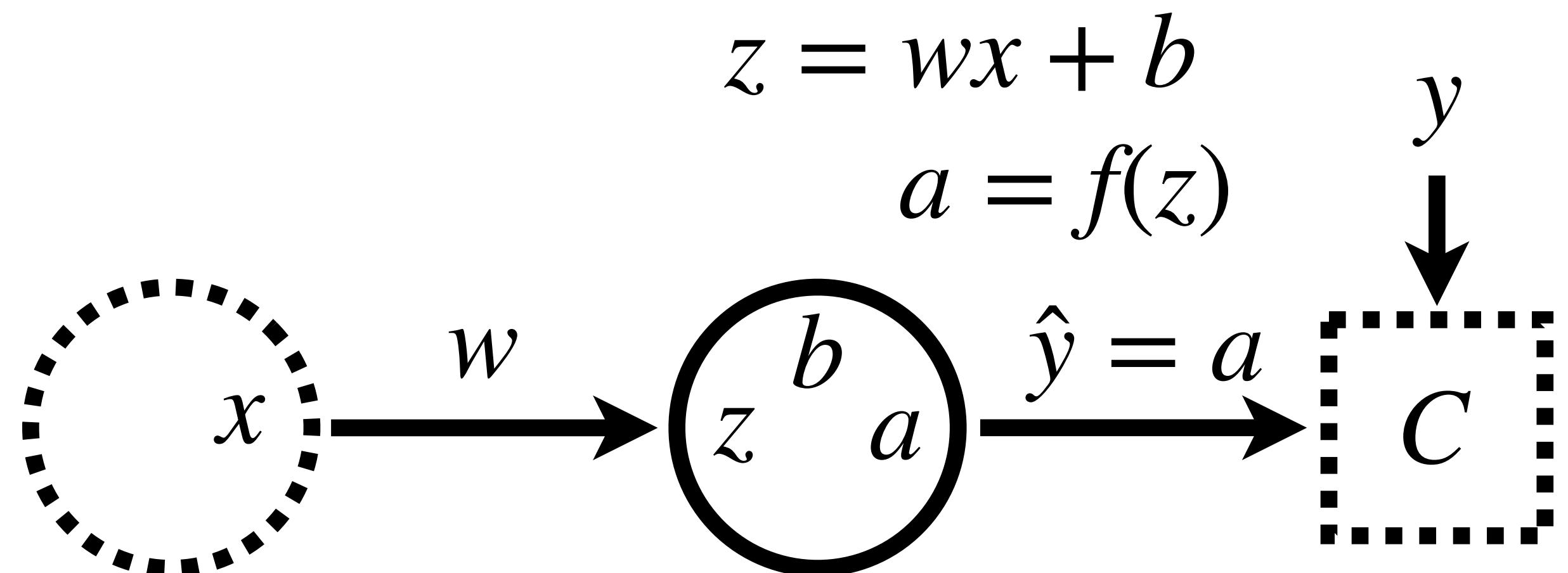


$$z = wx + b$$

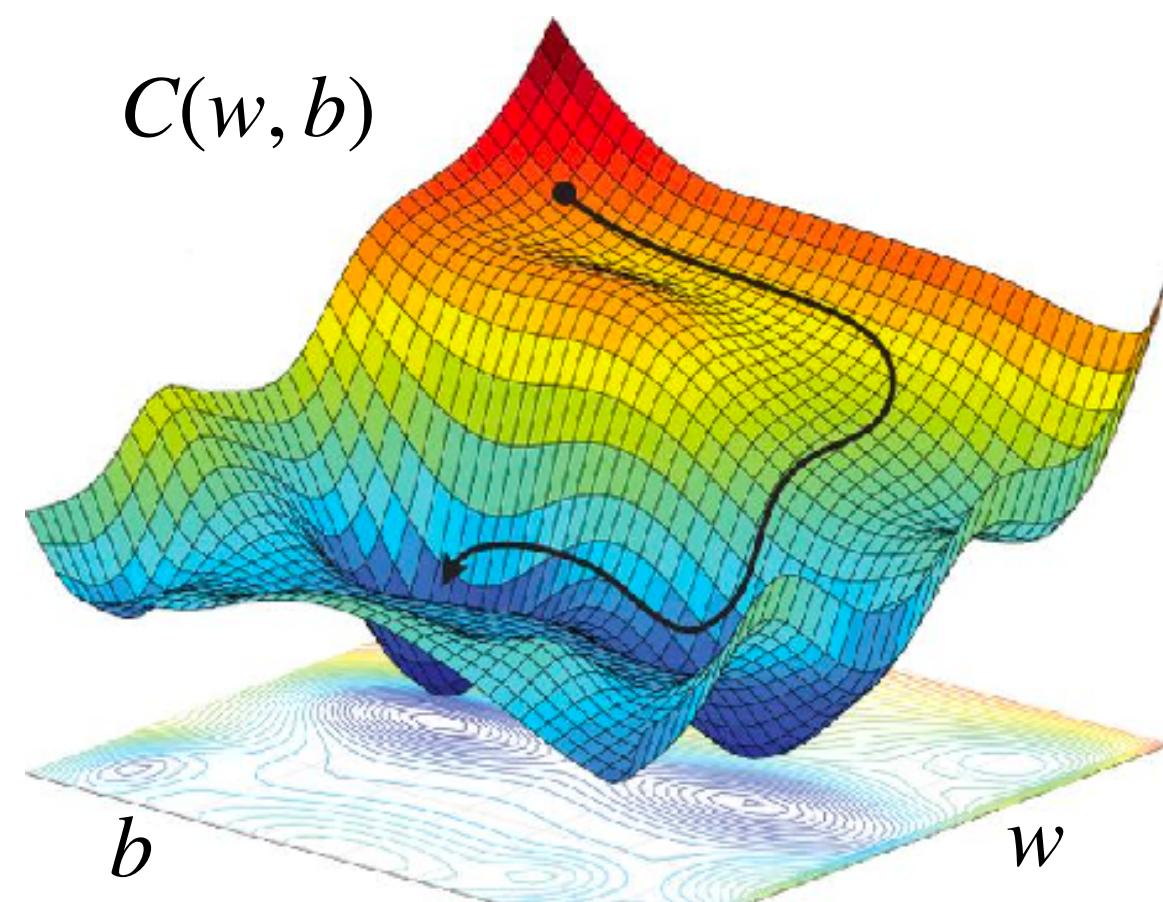
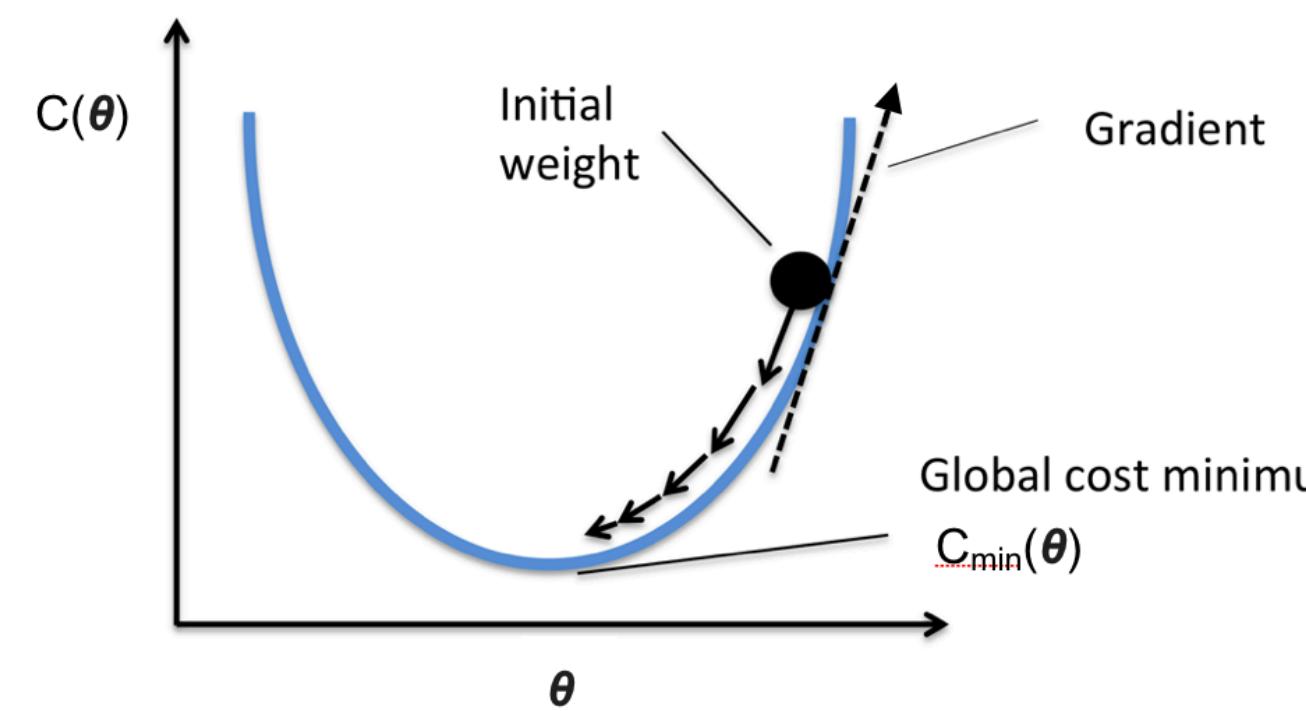
$$a = f(z)$$

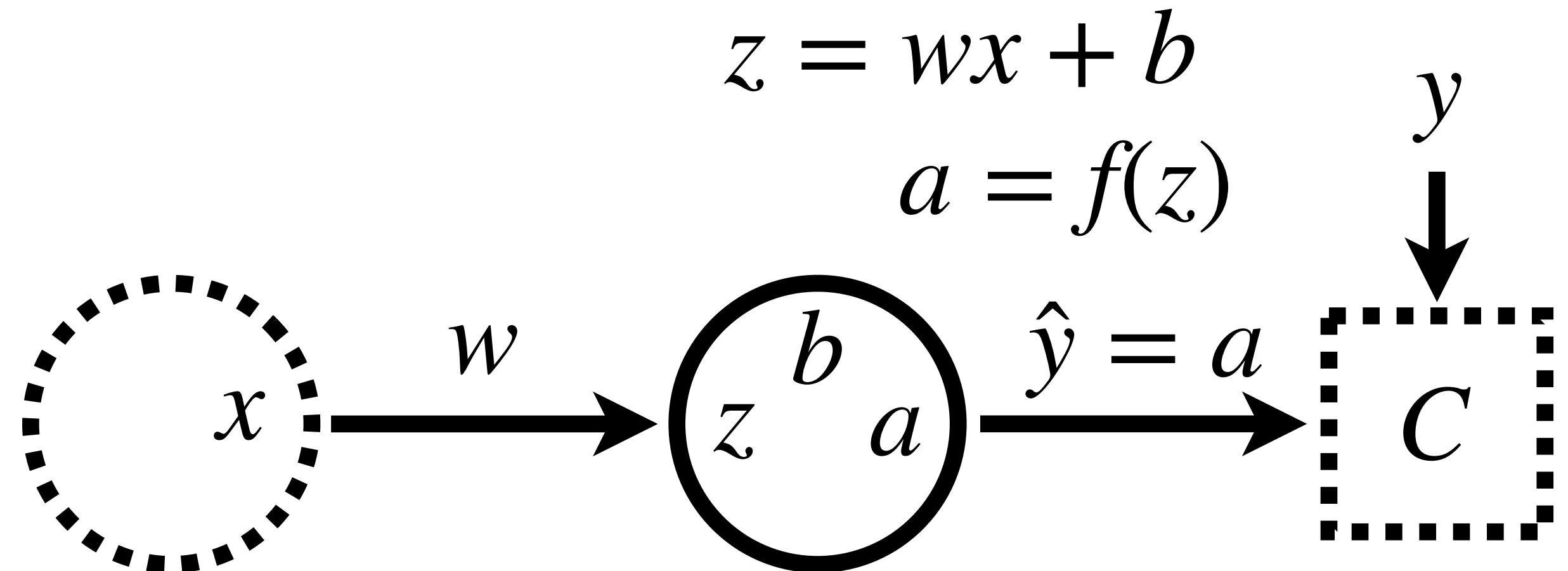






$$C(a(z(\theta))) = C(a(z(w, b)))$$

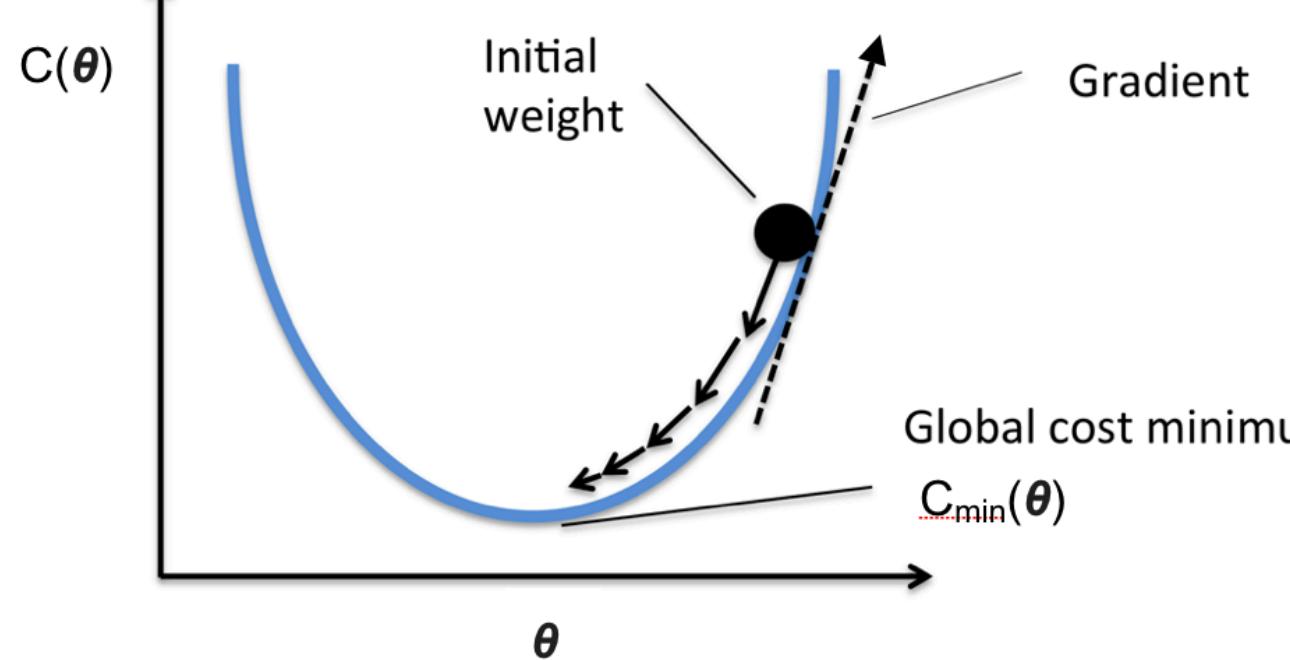




$$C(a(z(\theta))) = C(a(z(w, b)))$$

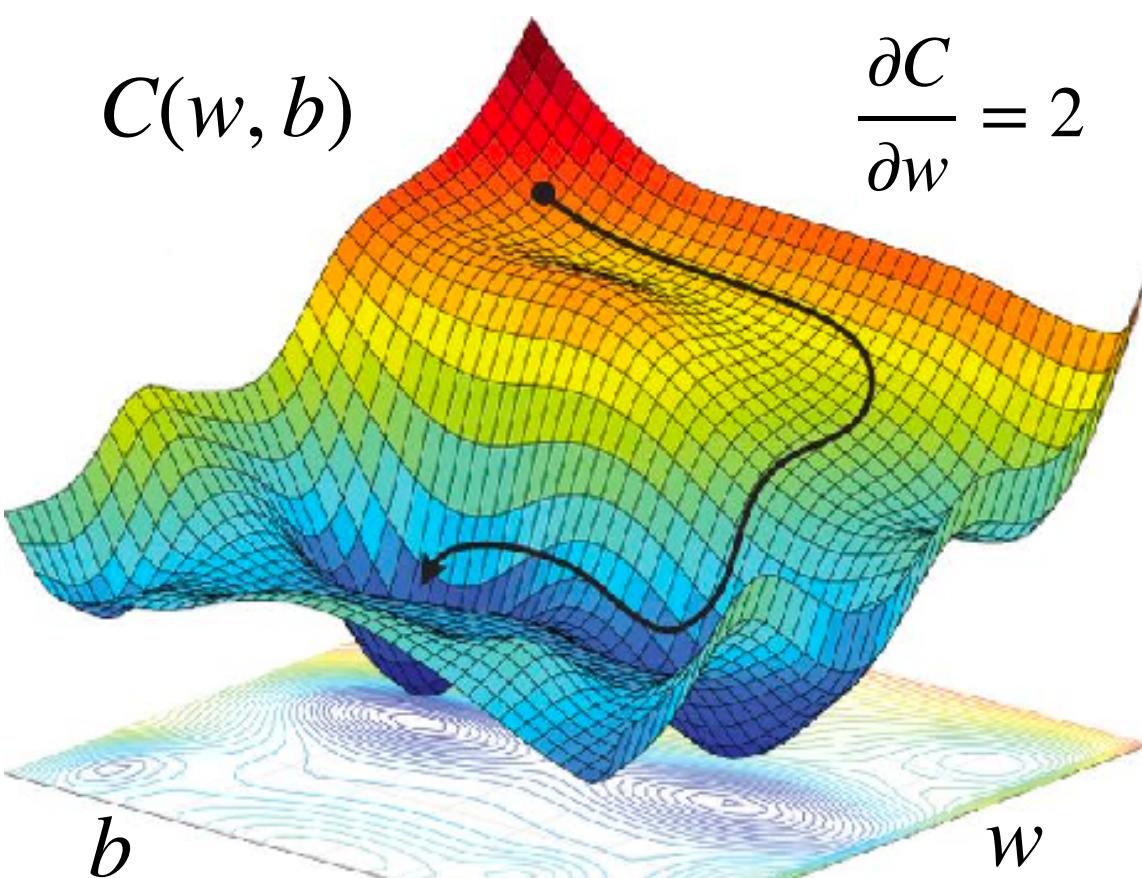
$$\theta_{new} = \theta_{old} - \alpha \nabla C$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_i} \\ \vdots \\ \frac{\partial C}{\partial \theta_n} \end{bmatrix}$$

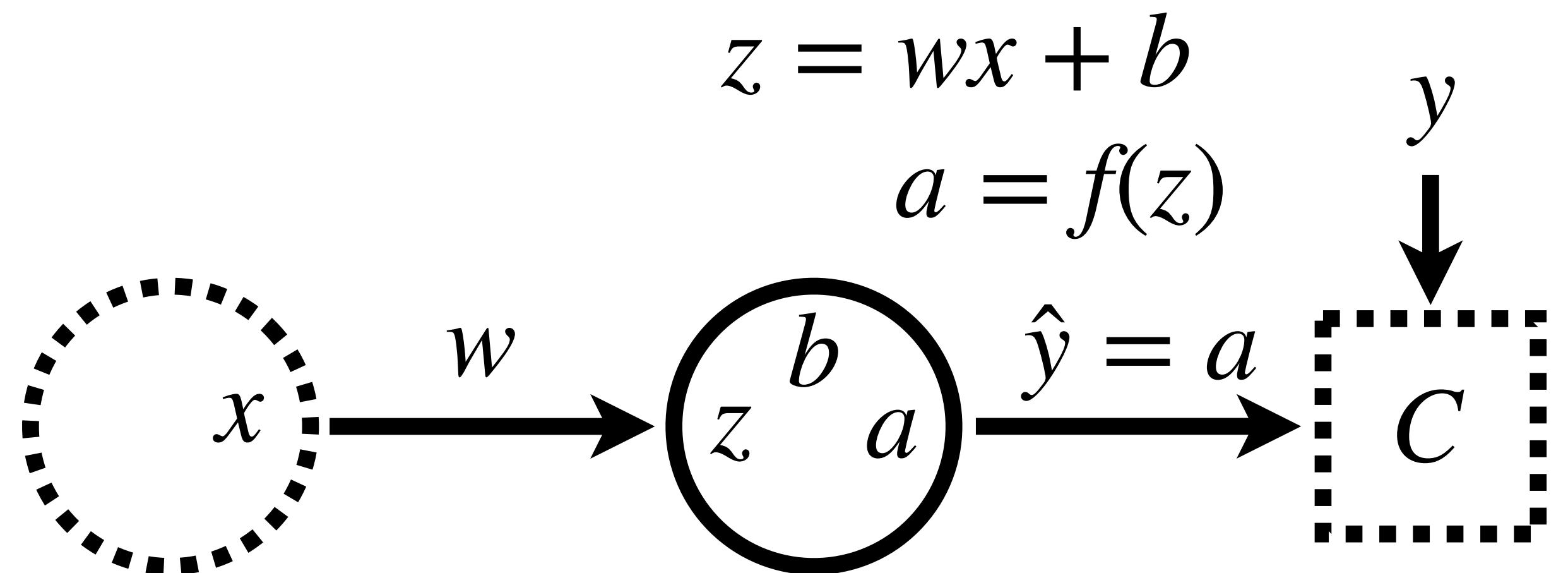


$$C(w, b) \quad \frac{\partial C}{\partial w} = 2$$

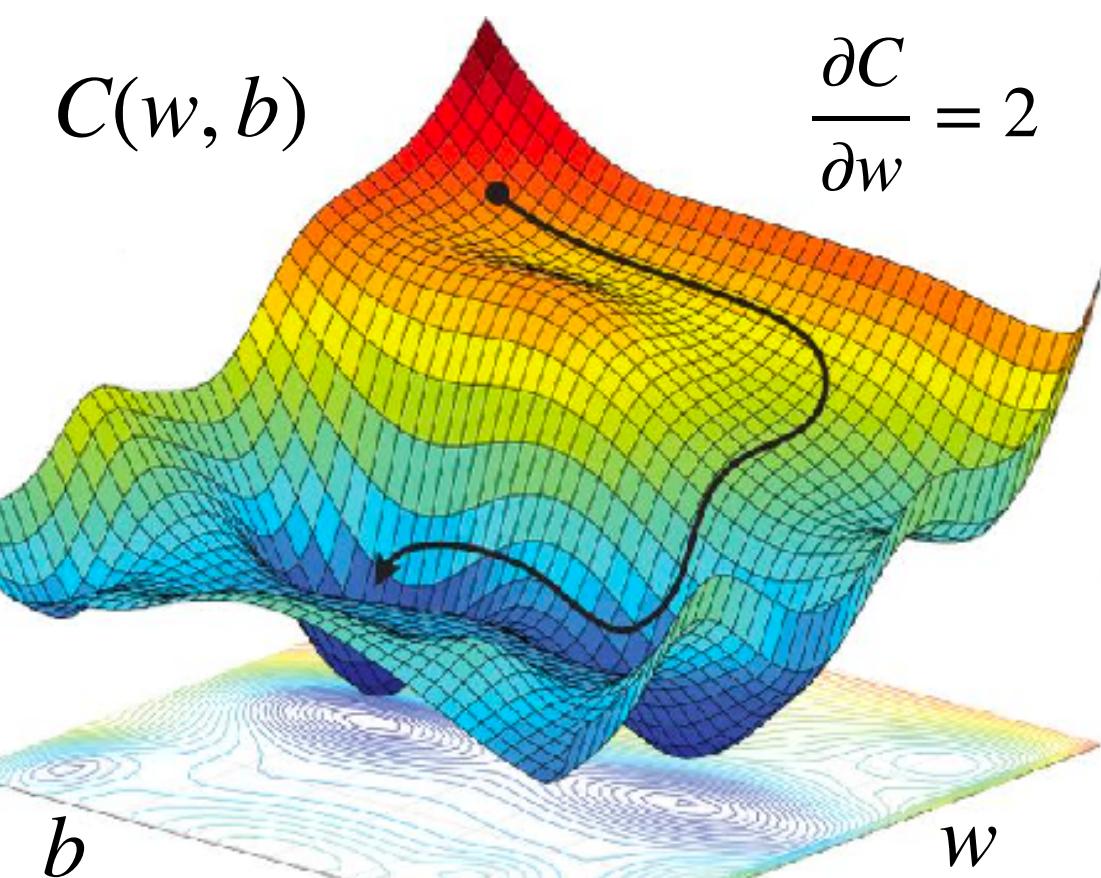
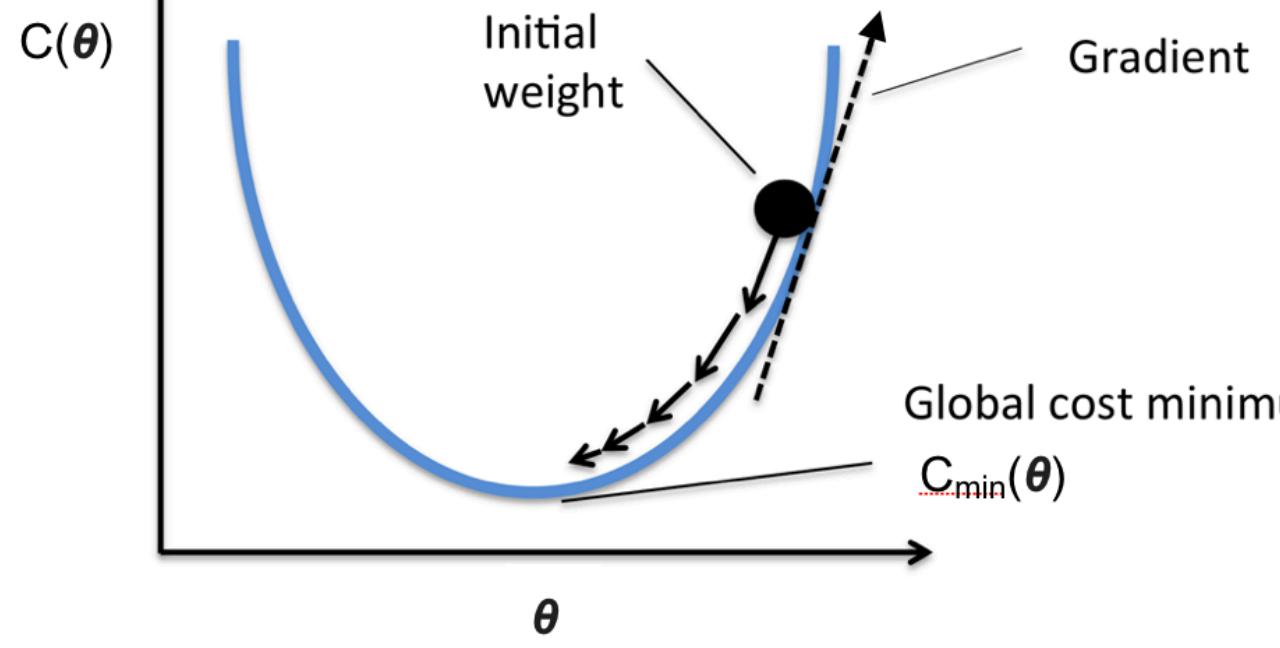
$$\begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \\ \theta'_n \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_n} \end{bmatrix}$$



$$\begin{bmatrix} w' \\ b' \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$



$$C(a(z(\theta))) = C(a(z(w, b)))$$

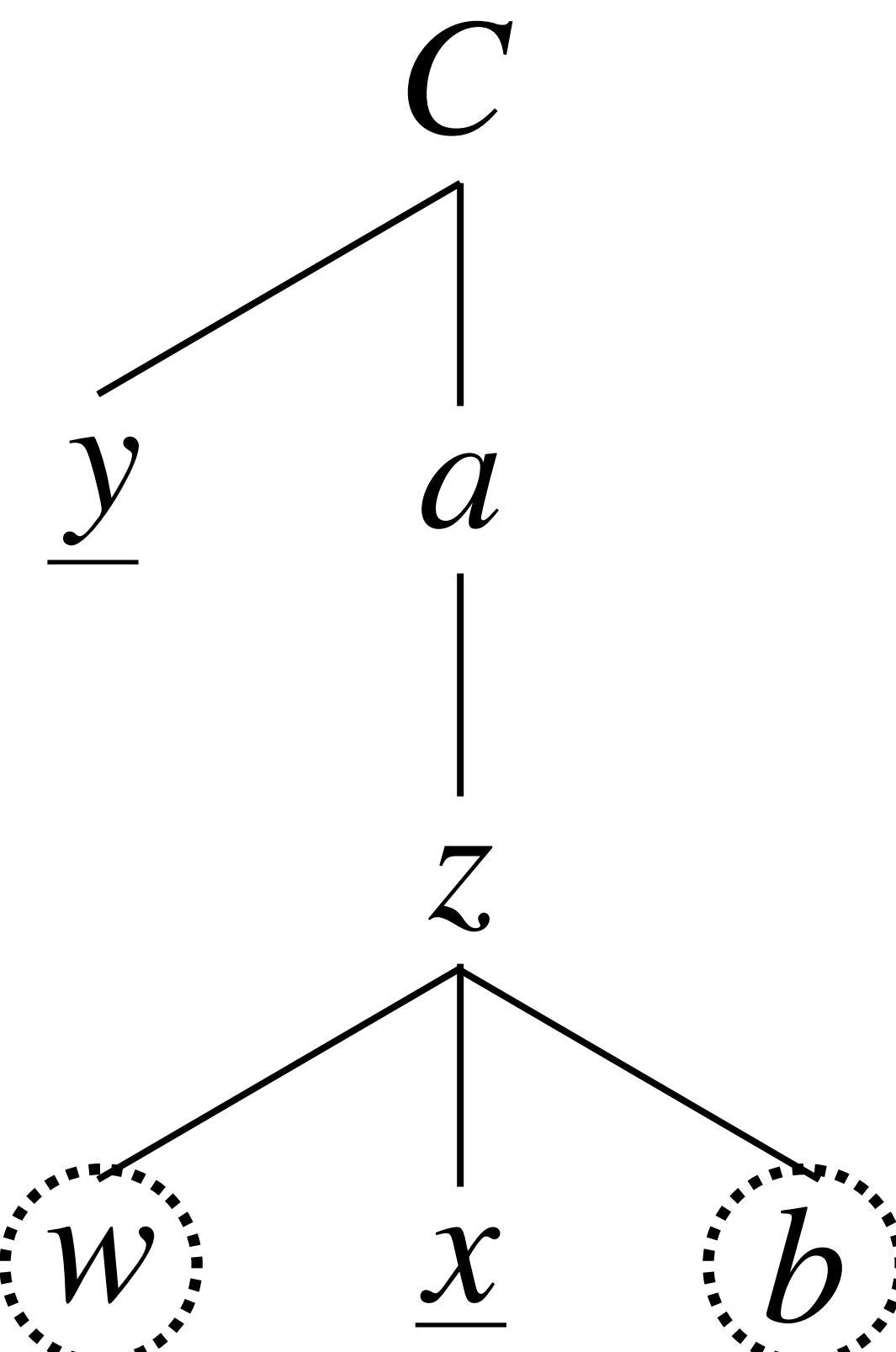


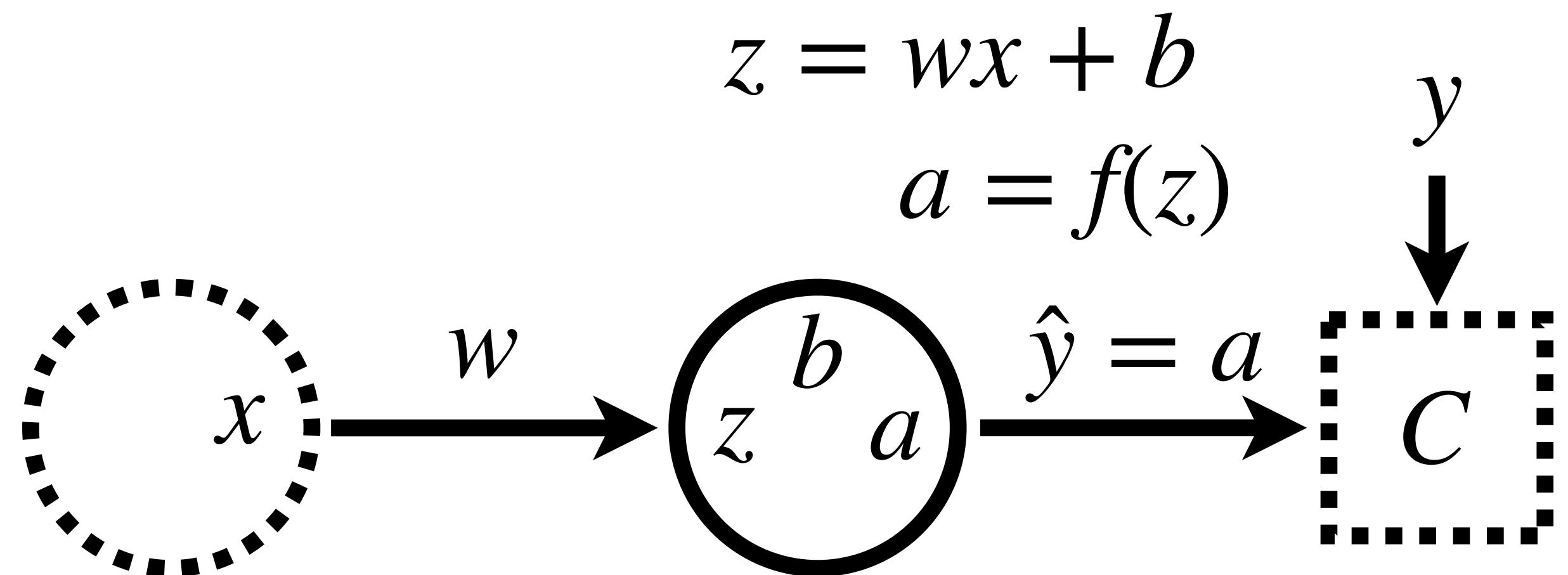
$$\theta_{new} = \theta_{old} - \alpha \nabla C$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_i} \\ \vdots \\ \frac{\partial C}{\partial \theta_{n_i}} \end{bmatrix}$$

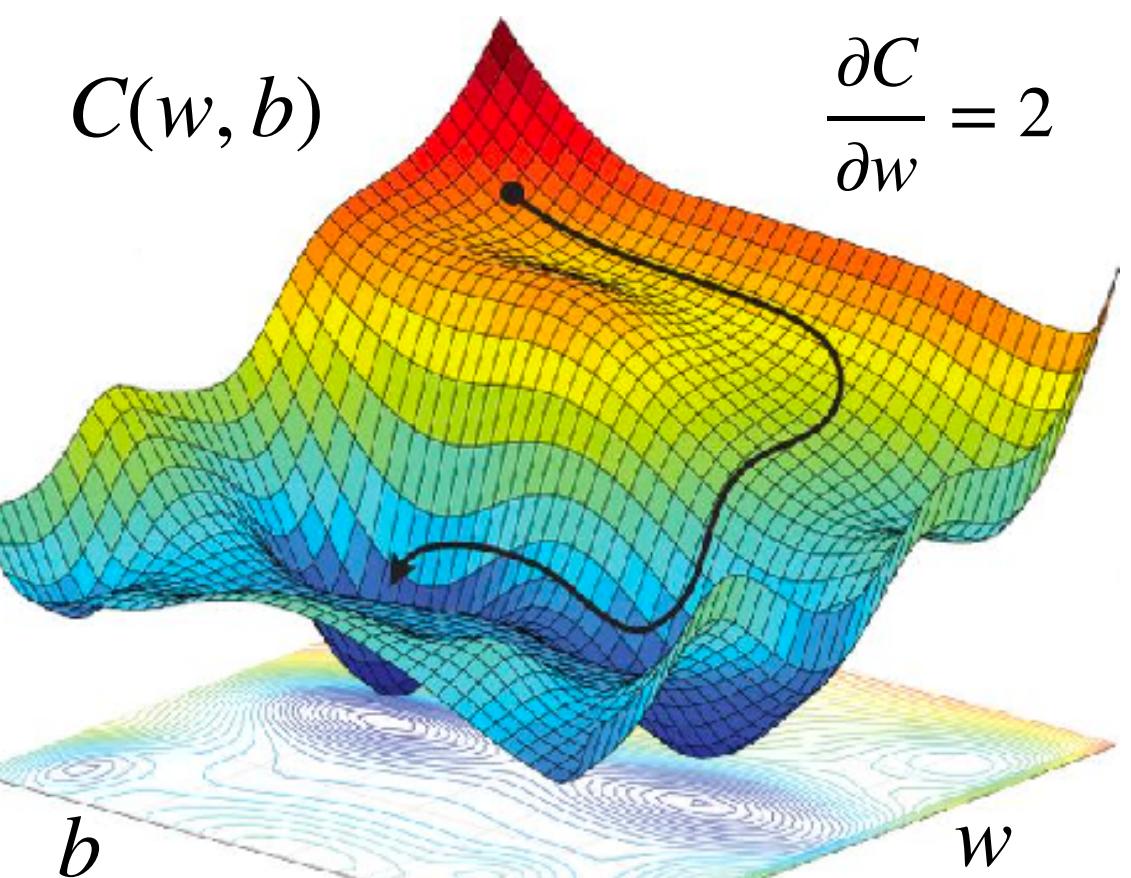
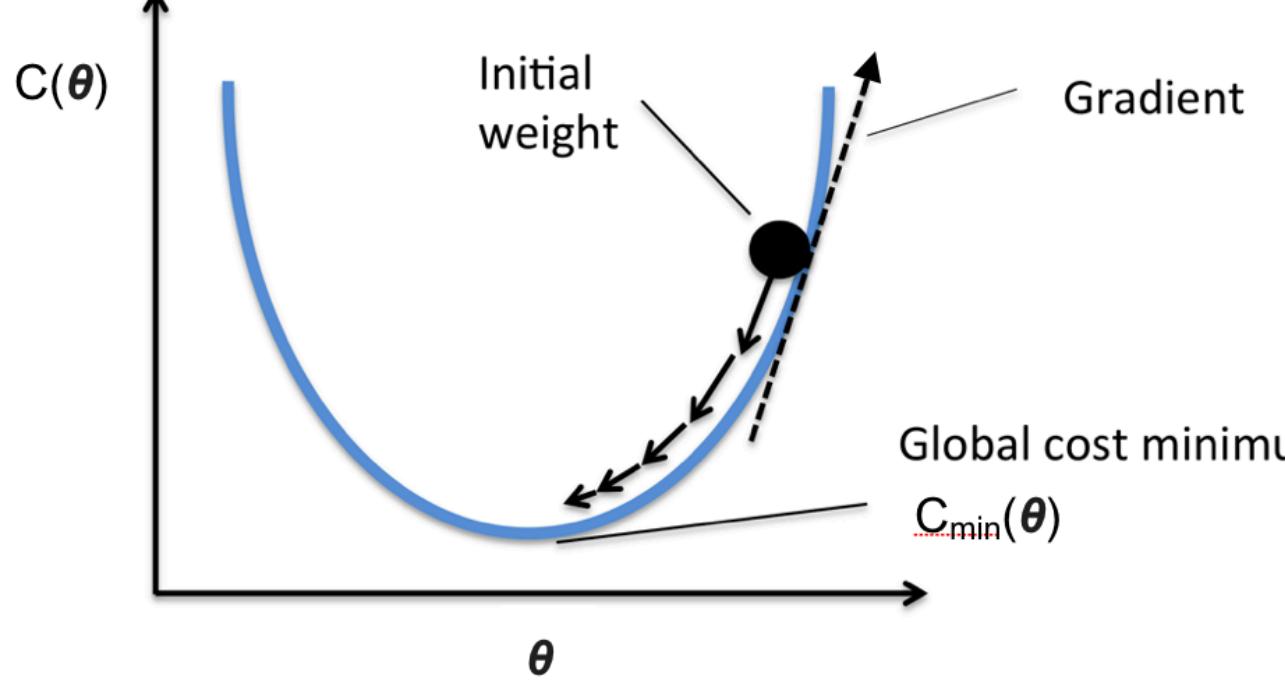
$$\begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_{n_i}} \end{bmatrix}$$

$$\begin{bmatrix} w' \\ b' \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$





$$C(a(z(\theta))) = C(a(z(w, b)))$$

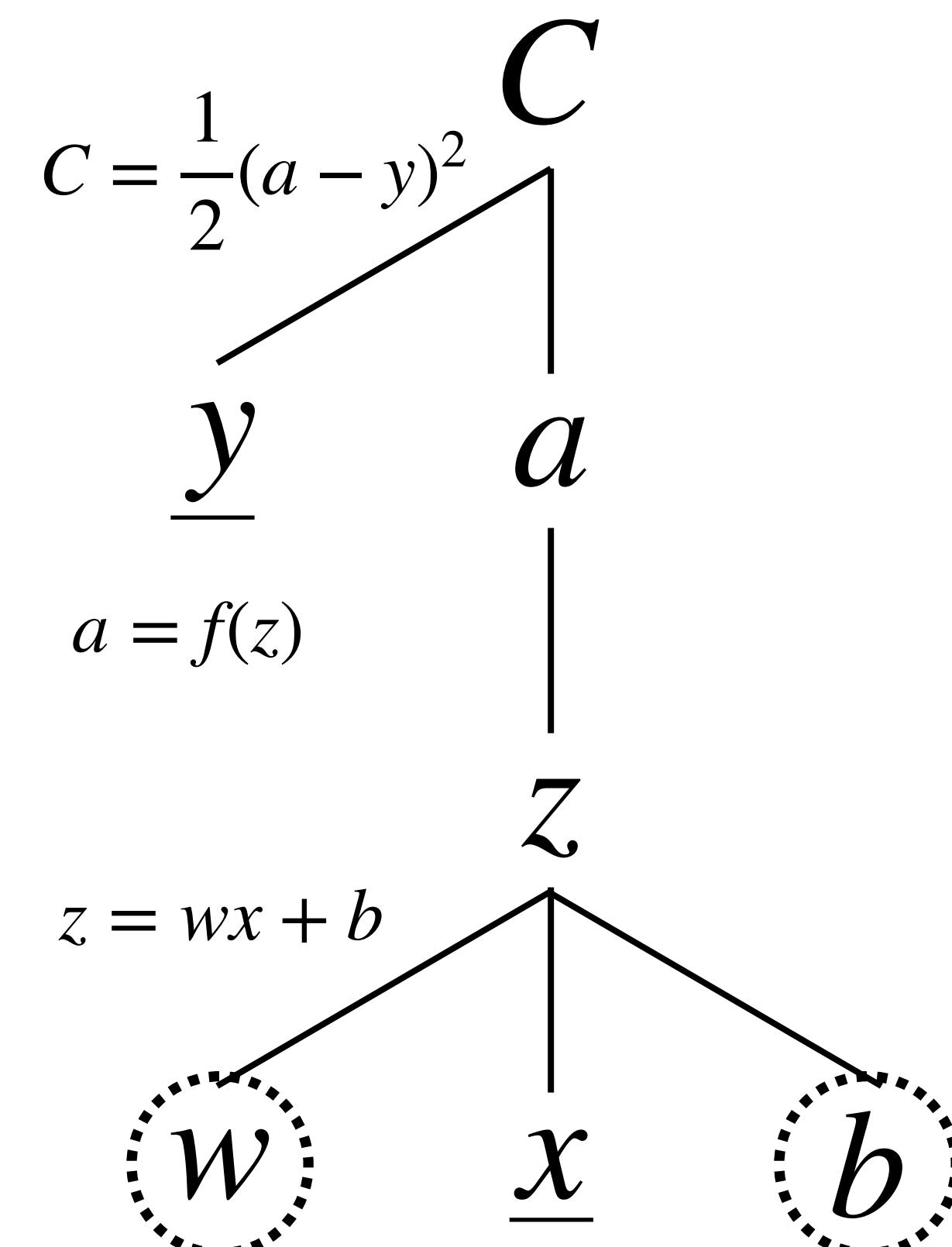


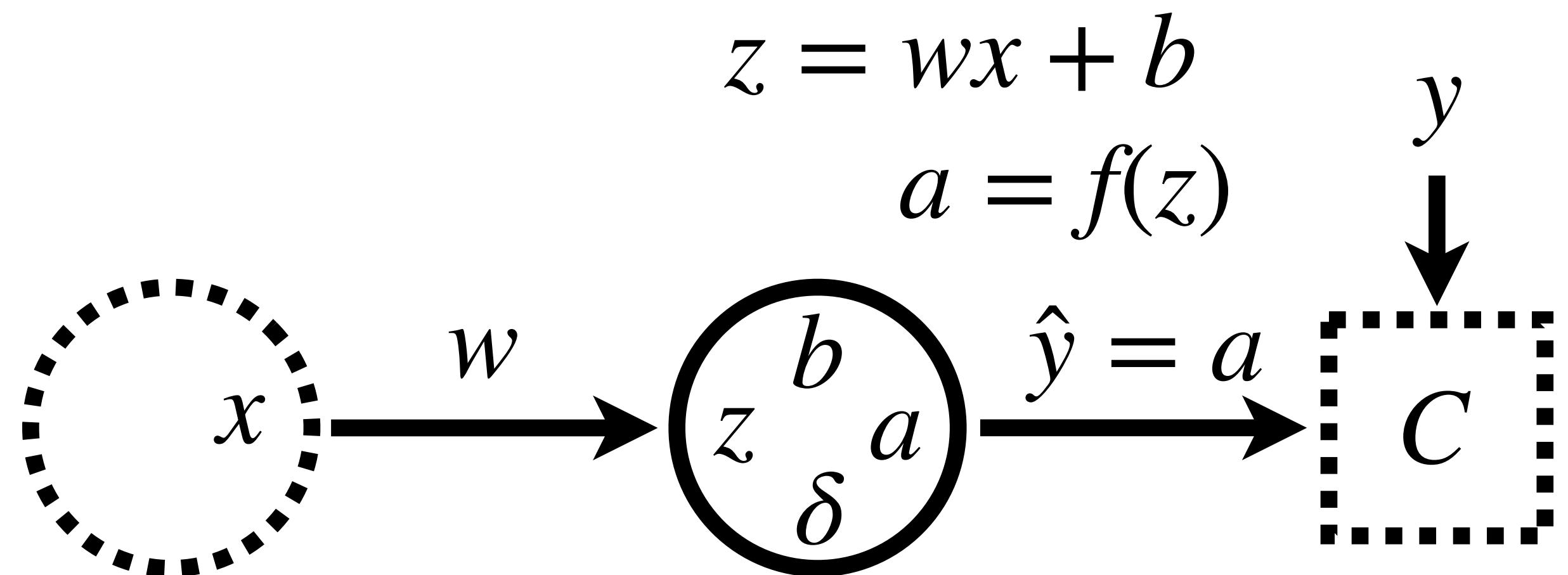
$$\theta_{new} = \theta_{old} - \alpha \nabla C$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_i} \\ \vdots \end{bmatrix}$$

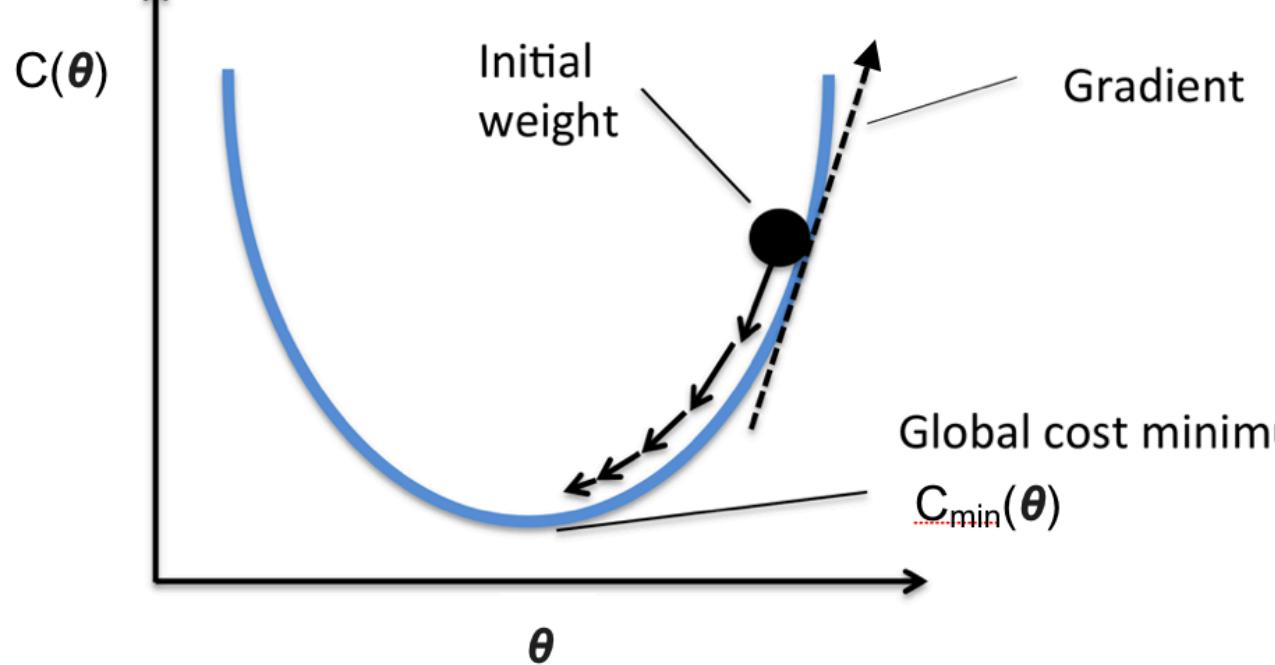
$$\begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} w' \\ b' \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$

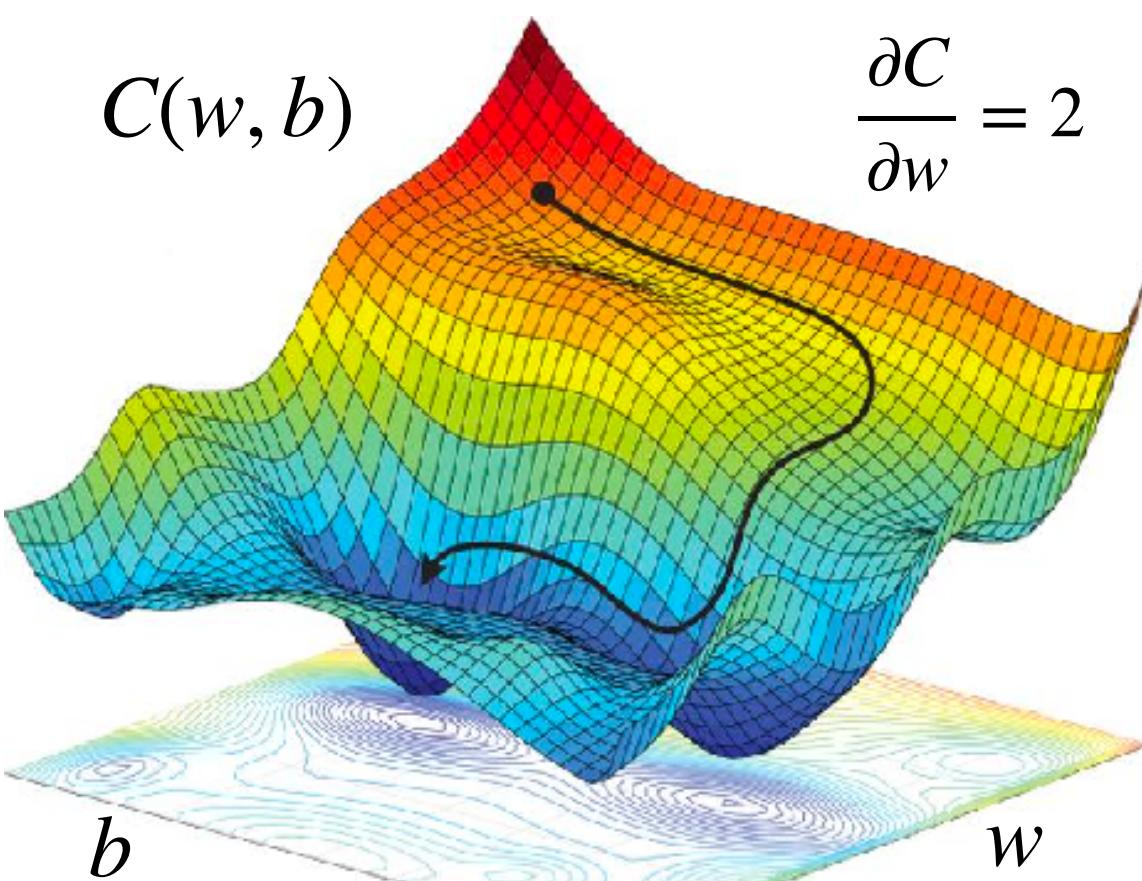




$$C(a(z(\theta))) = C(a(z(w, b)))$$



$$C(w, b) \quad \frac{\partial C}{\partial w} = 2$$

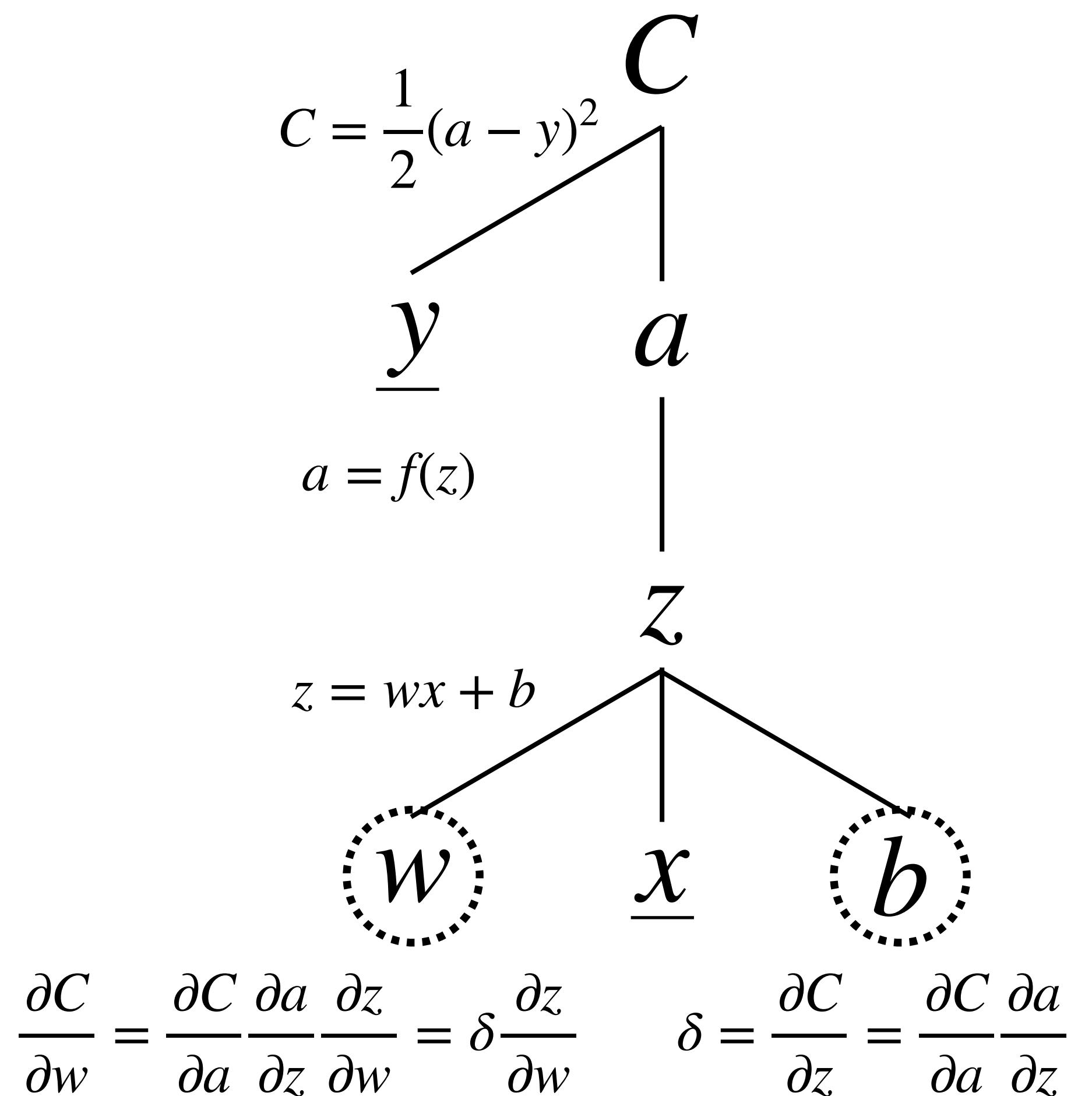


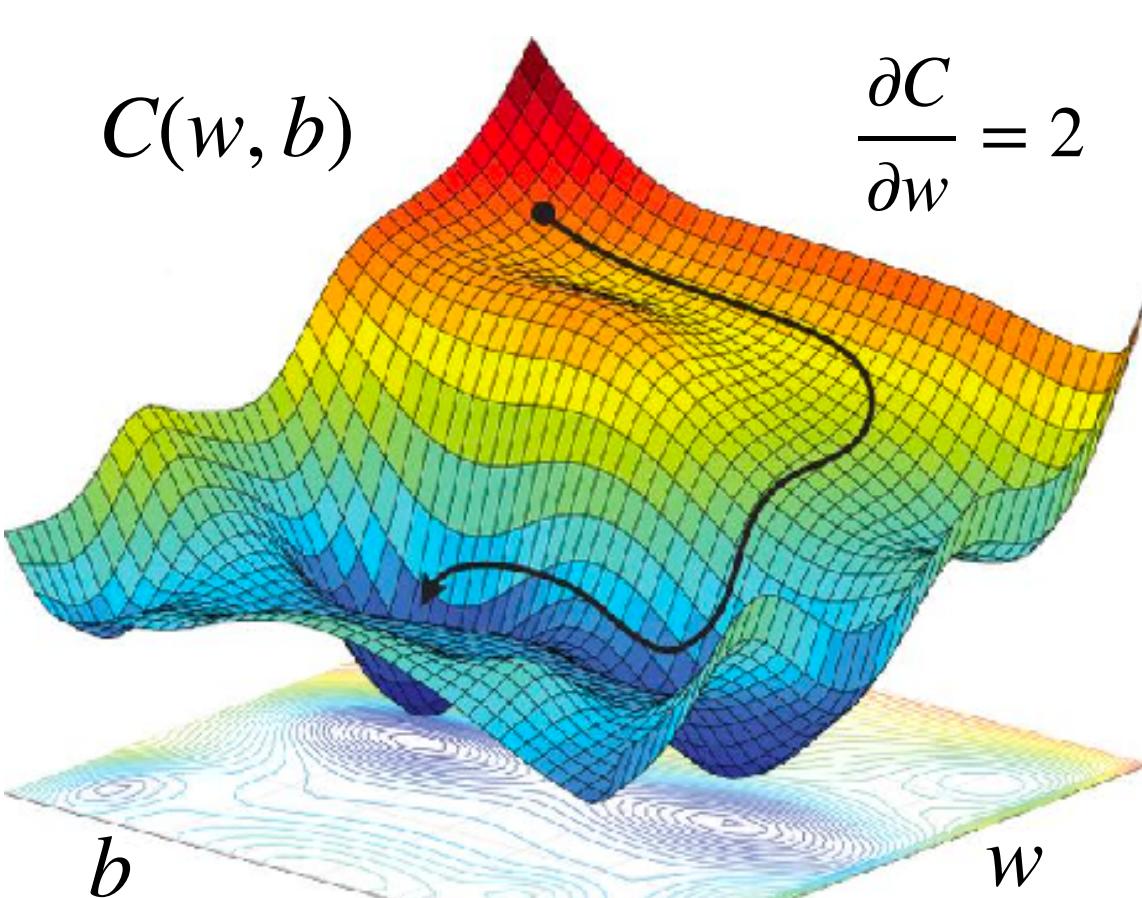
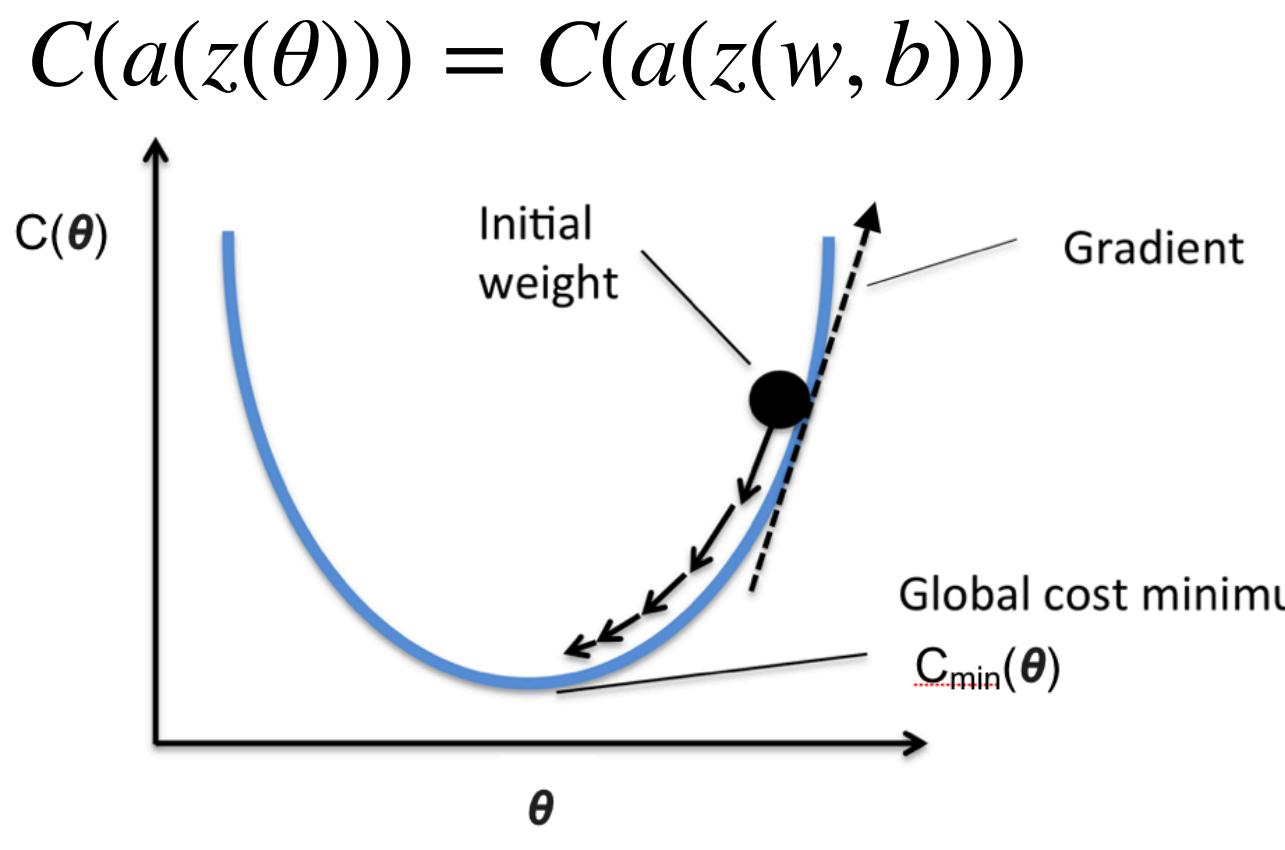
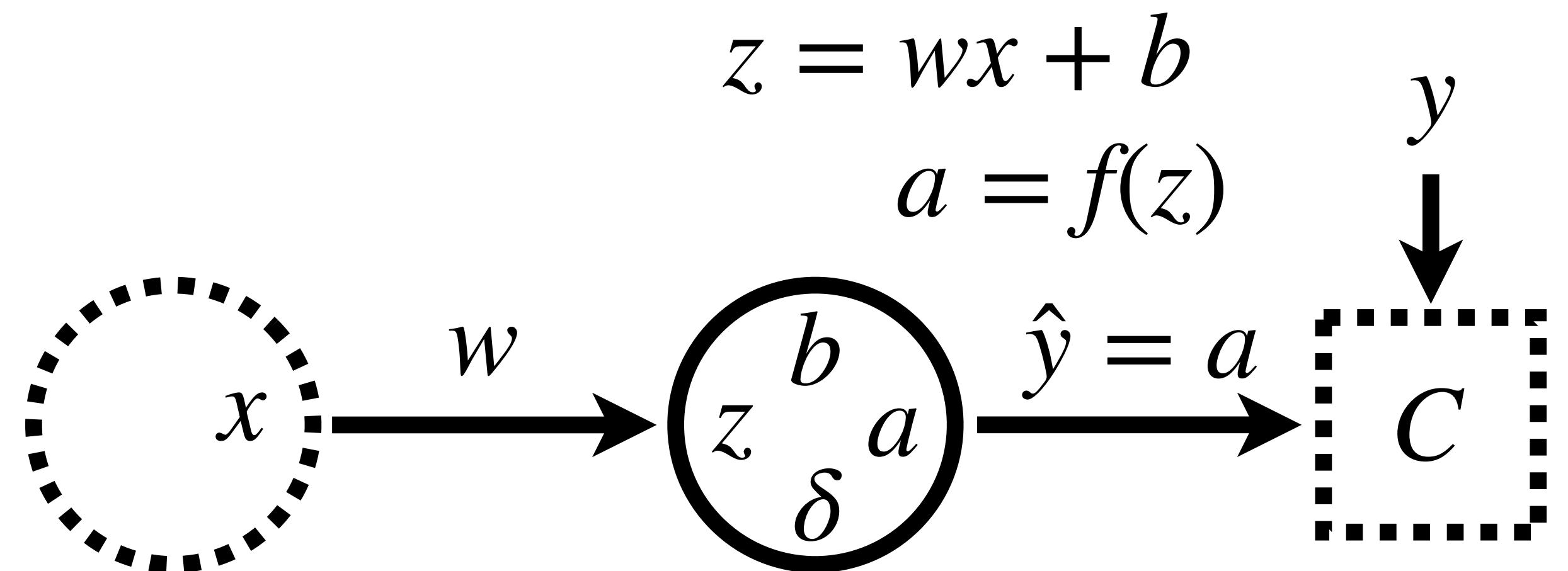
$$\theta_{new} = \theta_{old} - \alpha \nabla C$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_i} \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} w' \\ b' \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$



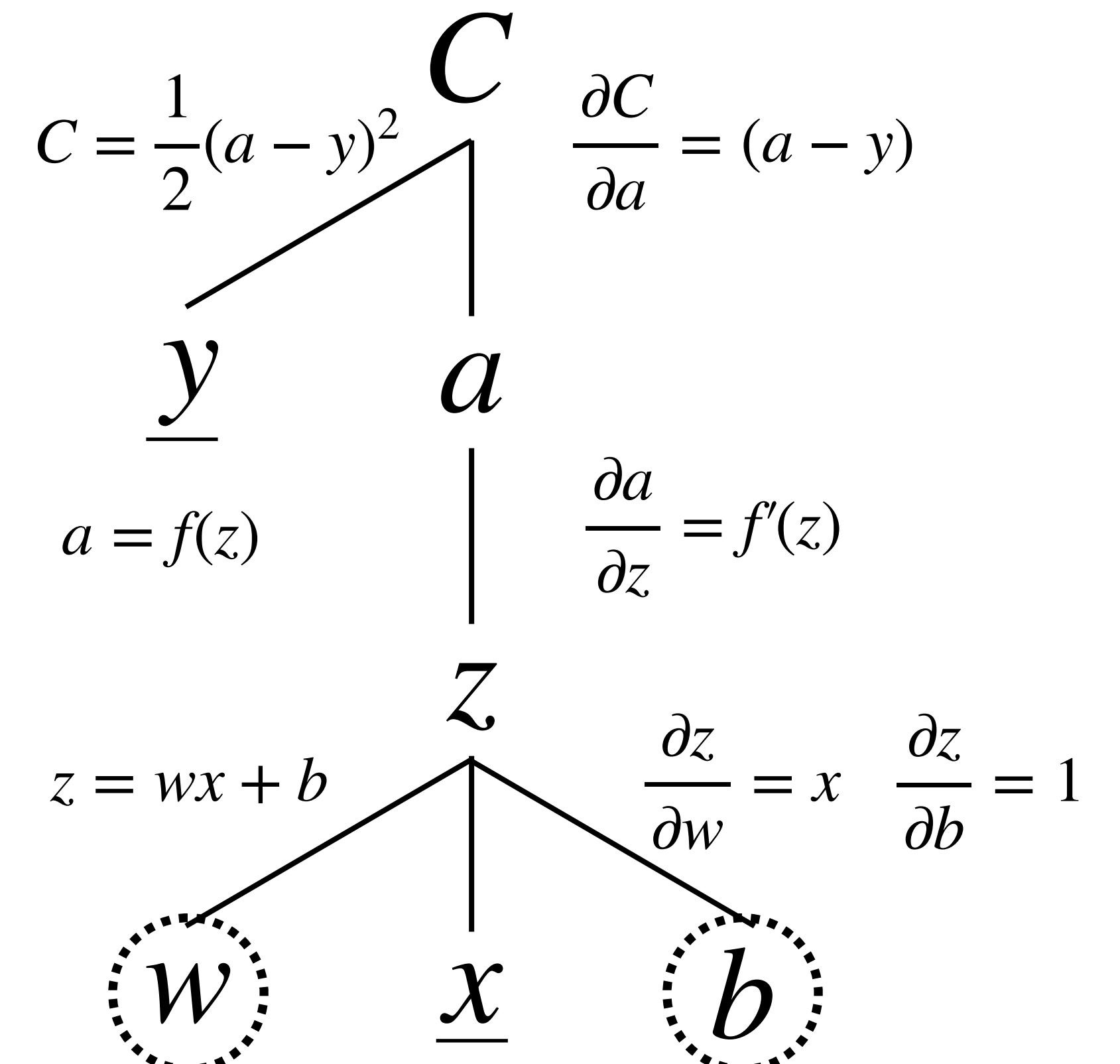


$$\theta_{new} = \theta_{old} - \alpha \nabla C$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_i} \\ \vdots \end{bmatrix}$$

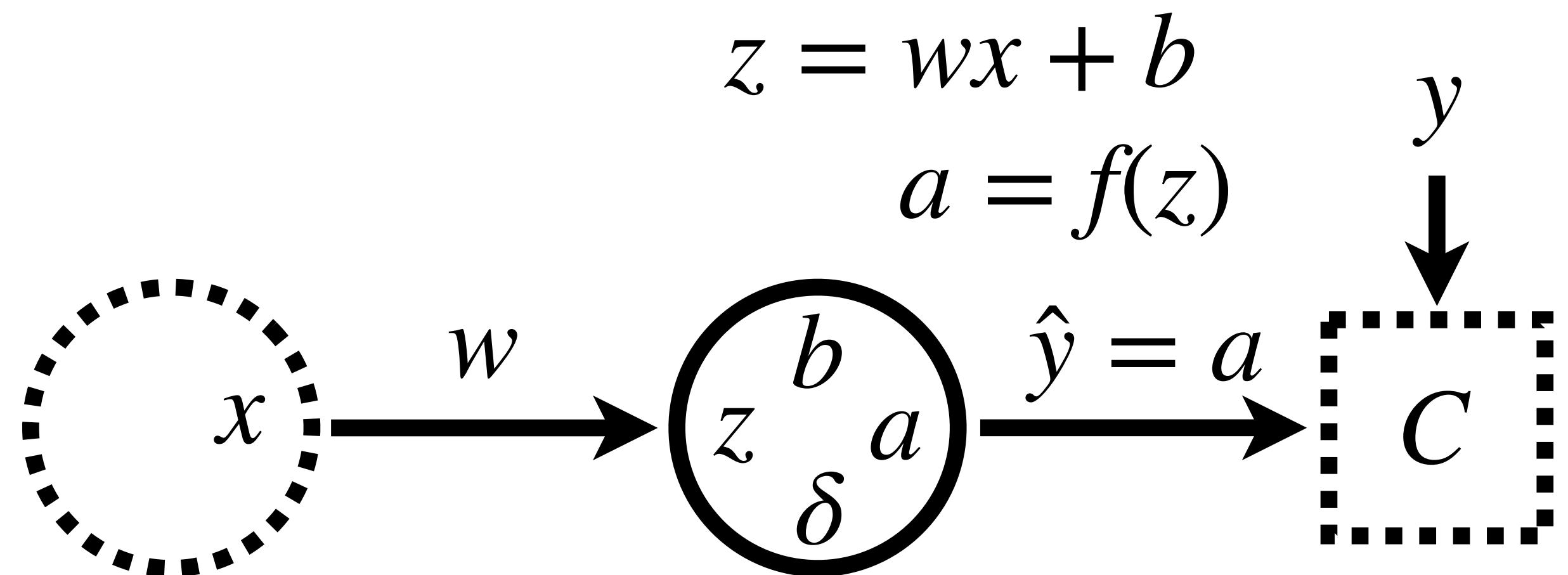
$$\begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_n} \end{bmatrix}$$

$$\begin{bmatrix} w' \\ b' \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$

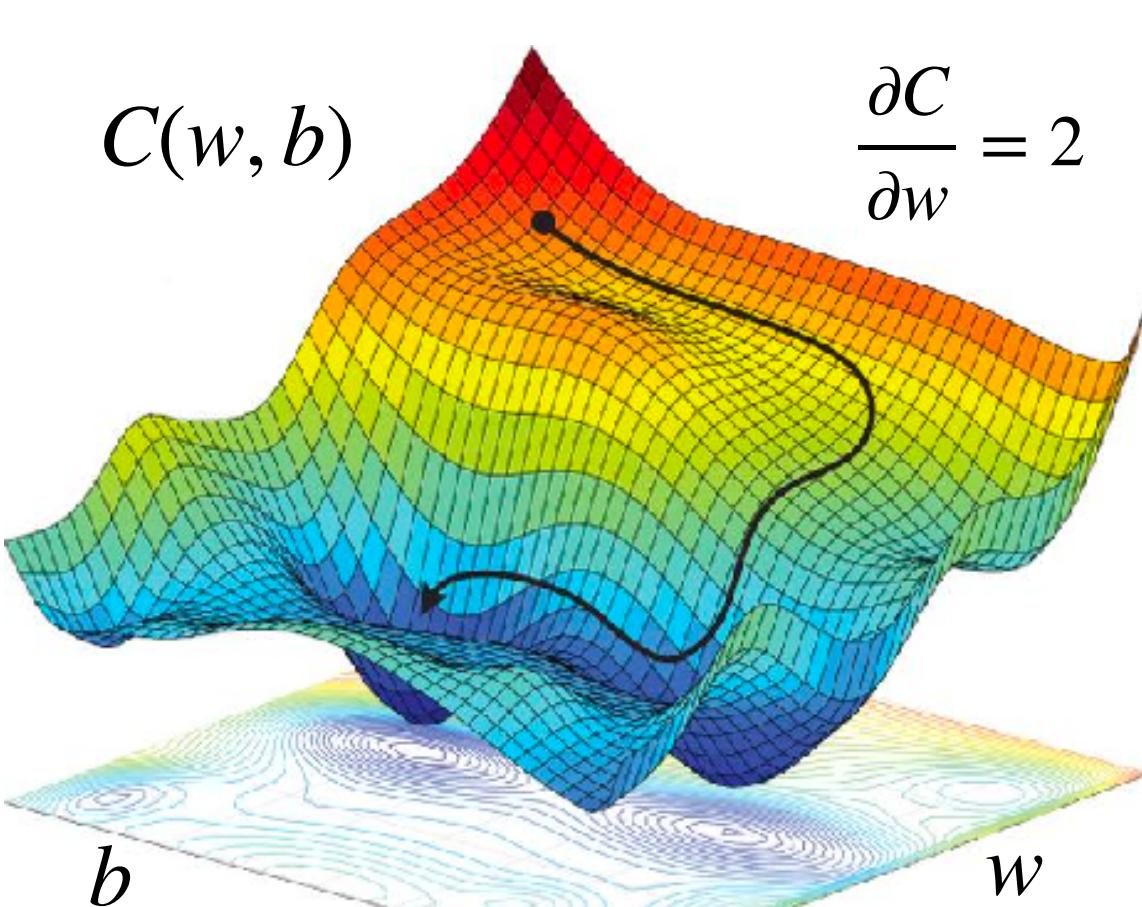
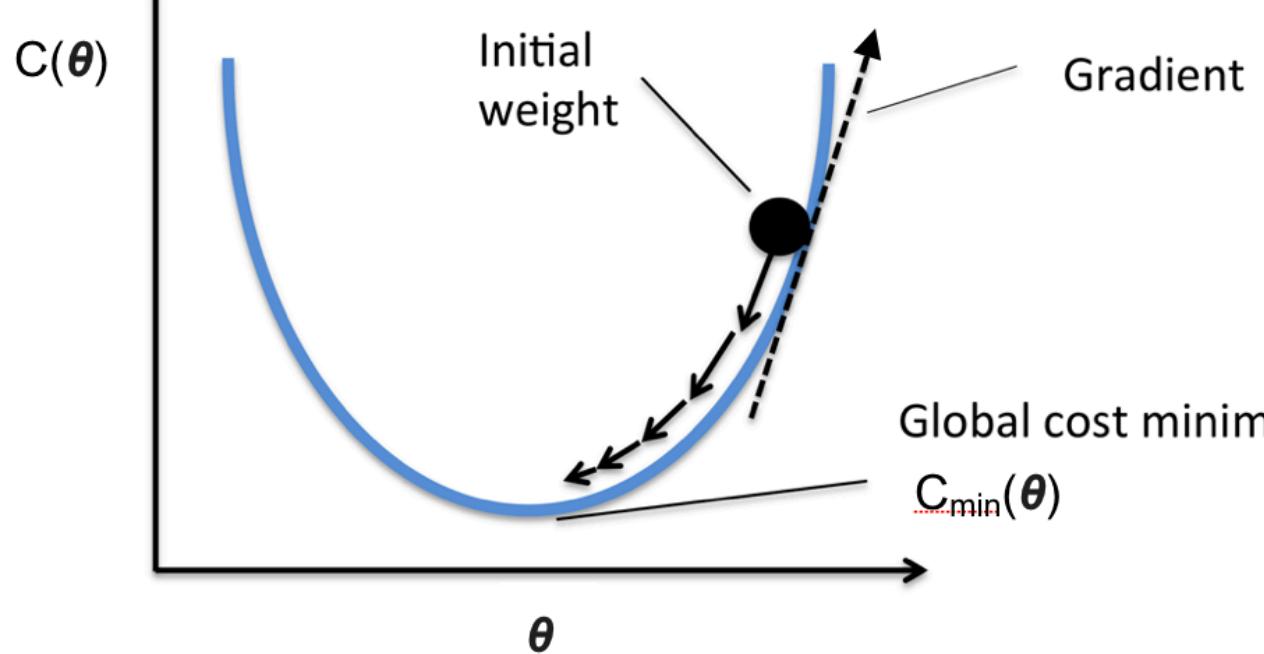


$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial z} \frac{\partial z}{\partial w} = \delta \frac{\partial z}{\partial w}$$

$$\delta = \frac{\partial C}{\partial z} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial z}$$



$$C(a(z(\theta))) = C(a(z(w, b)))$$

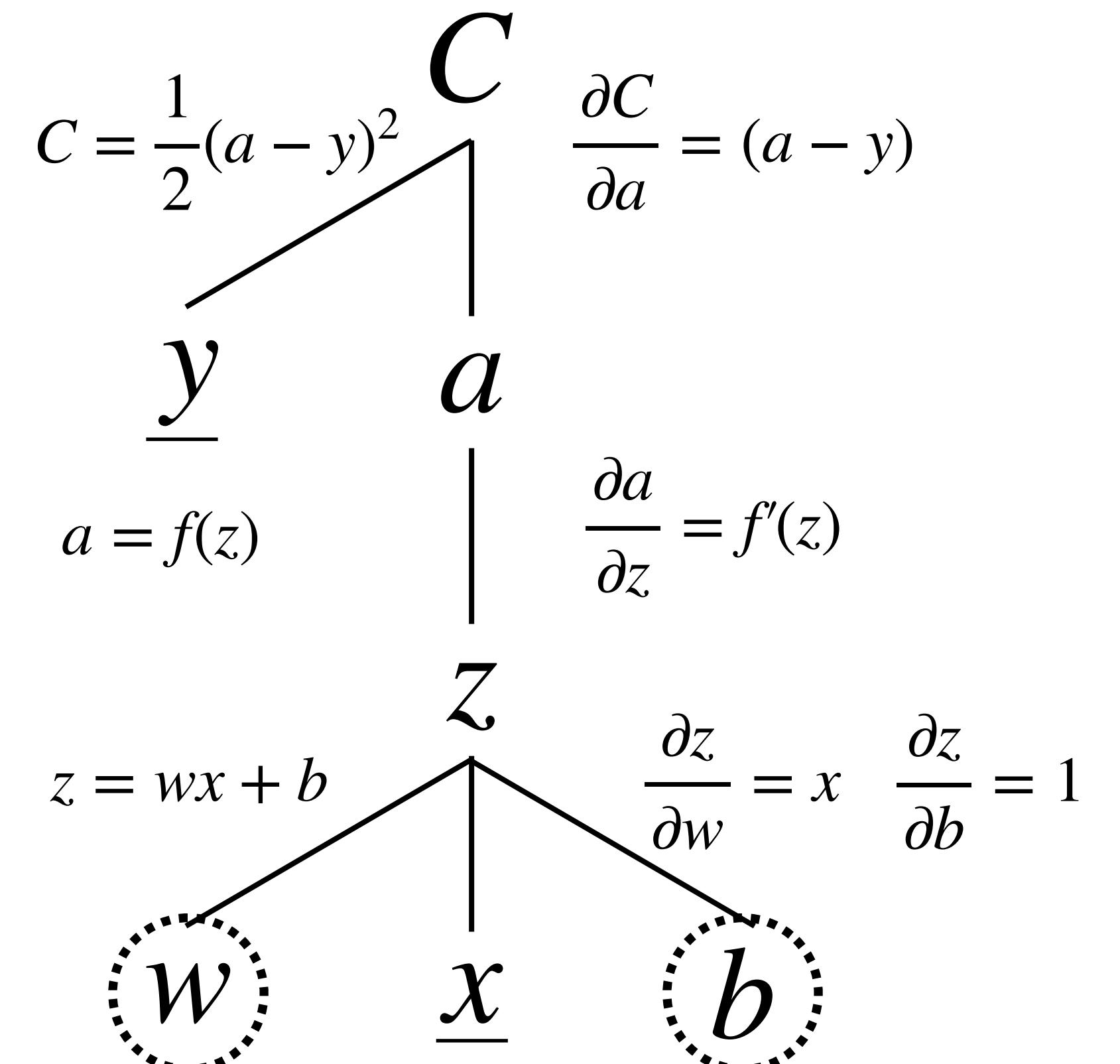


$$\theta_{new} = \theta_{old} - \alpha \nabla C$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \\ \frac{\partial C}{\partial \theta_i} \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial \theta_1} \\ \frac{\partial C}{\partial \theta_2} \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} w' \\ b' \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$



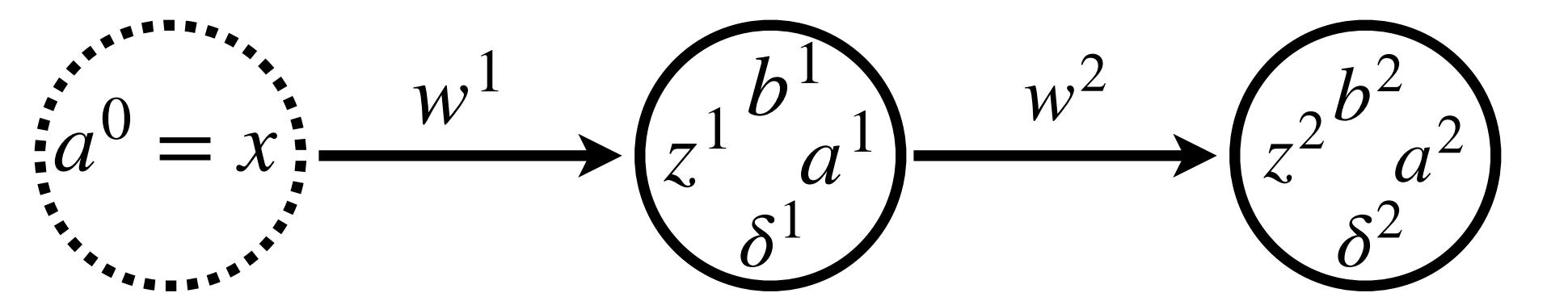
$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial z} \frac{\partial z}{\partial w} = \delta \frac{\partial z}{\partial w}$$

$$\frac{\partial C}{\partial w} = (a - y)f'(z)x = \delta_x$$

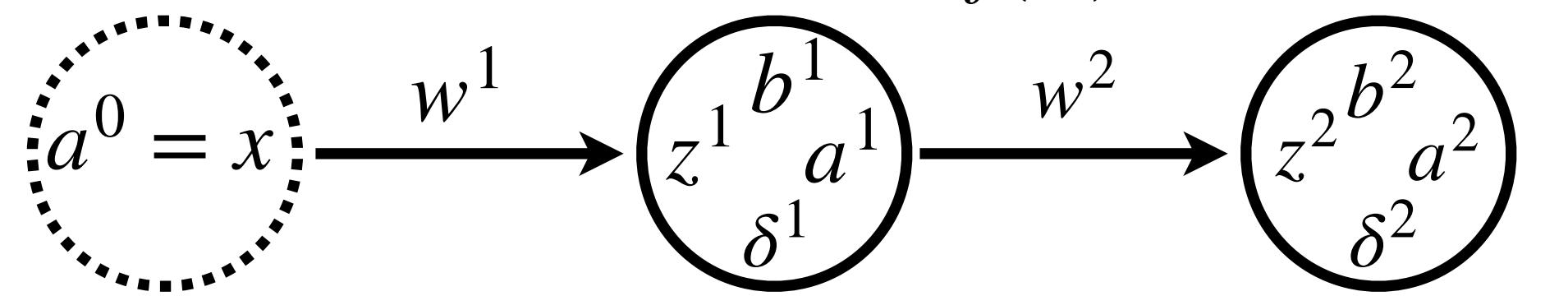
$$\frac{\partial C}{\partial b} = (a - y)f'(z)1 = \delta$$

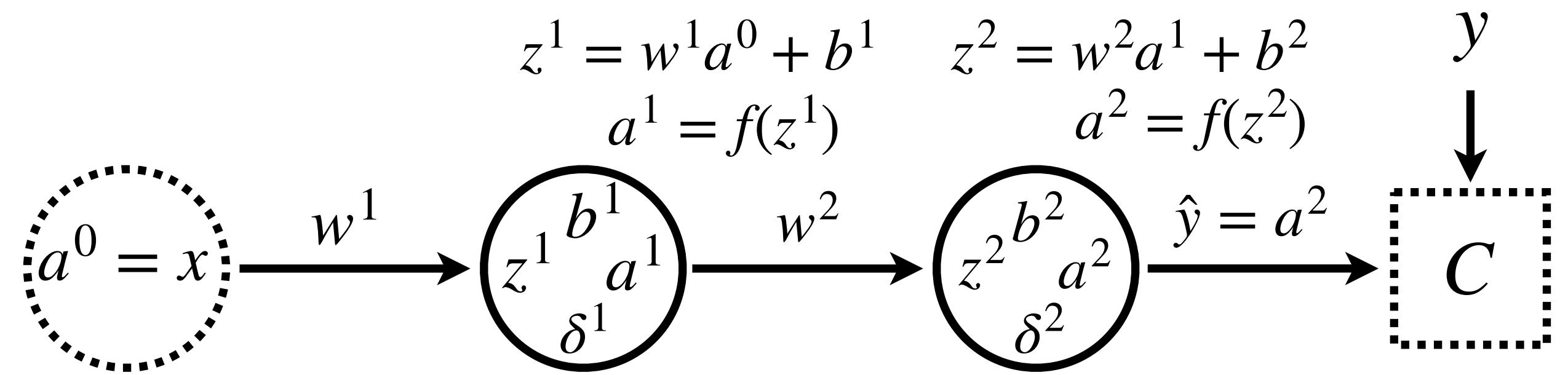
$$\delta = \frac{\partial C}{\partial z} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial z}$$

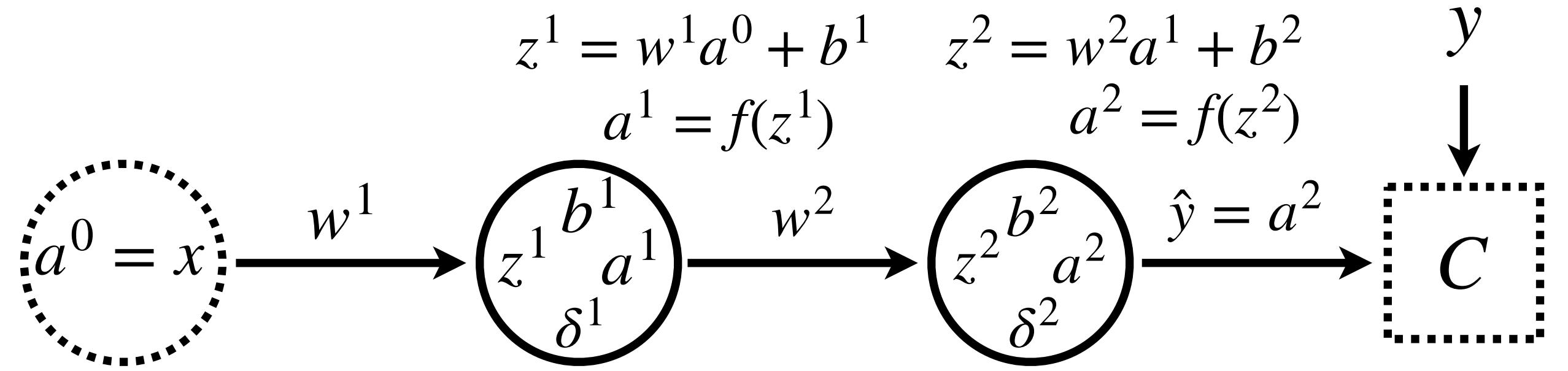
$$\delta = (a - y)f'(z)$$



$$\begin{array}{ll} z^1 = w^1 a^0 + b^1 & z^2 = w^2 a^1 + b^2 \\ a^1 = f(z^1) & a^2 = f(z^2) \end{array}$$





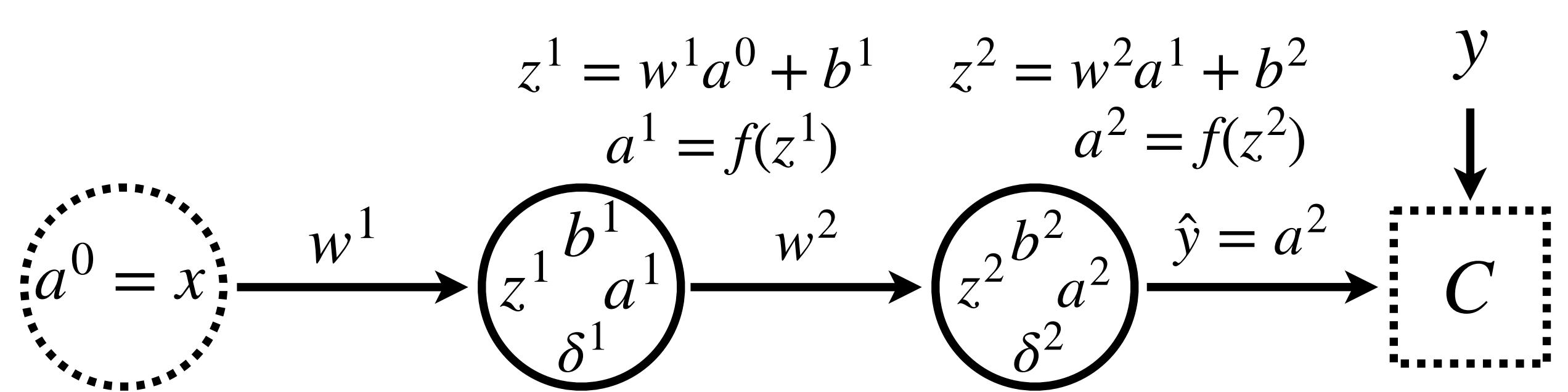


$$C(a^2(z^2(w^2, b^2)))$$

$$C(a^2(z^2(a^1(z^1(w^1, b^1)))))$$

$$\theta' = \theta - \alpha \nabla C$$

$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$

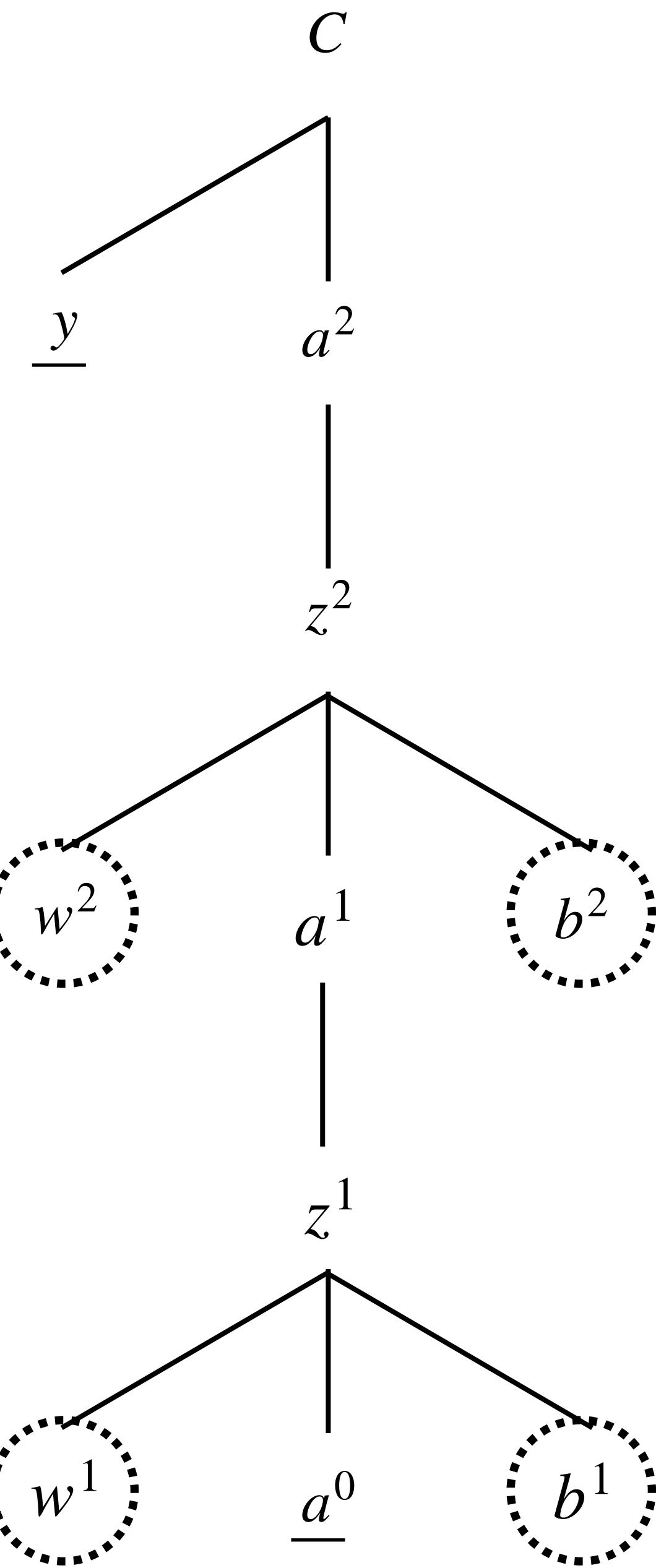


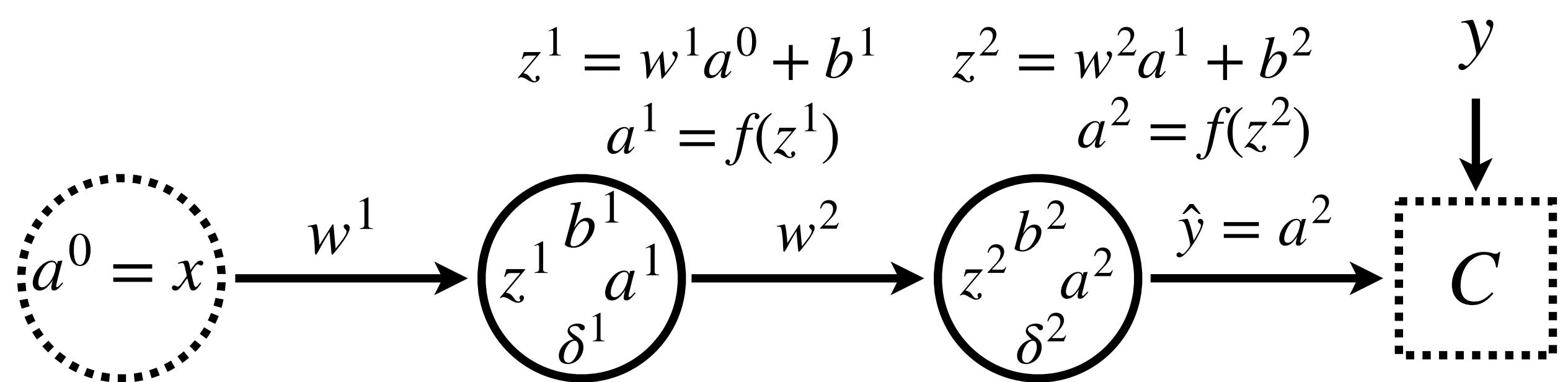
$$C(a^2(z^2(w^2, b^2)))$$

$$C(a^2(z^2(a^1(z^1(w^1, b^1))))))$$

$$\theta' = \theta - \alpha \nabla C$$

$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$



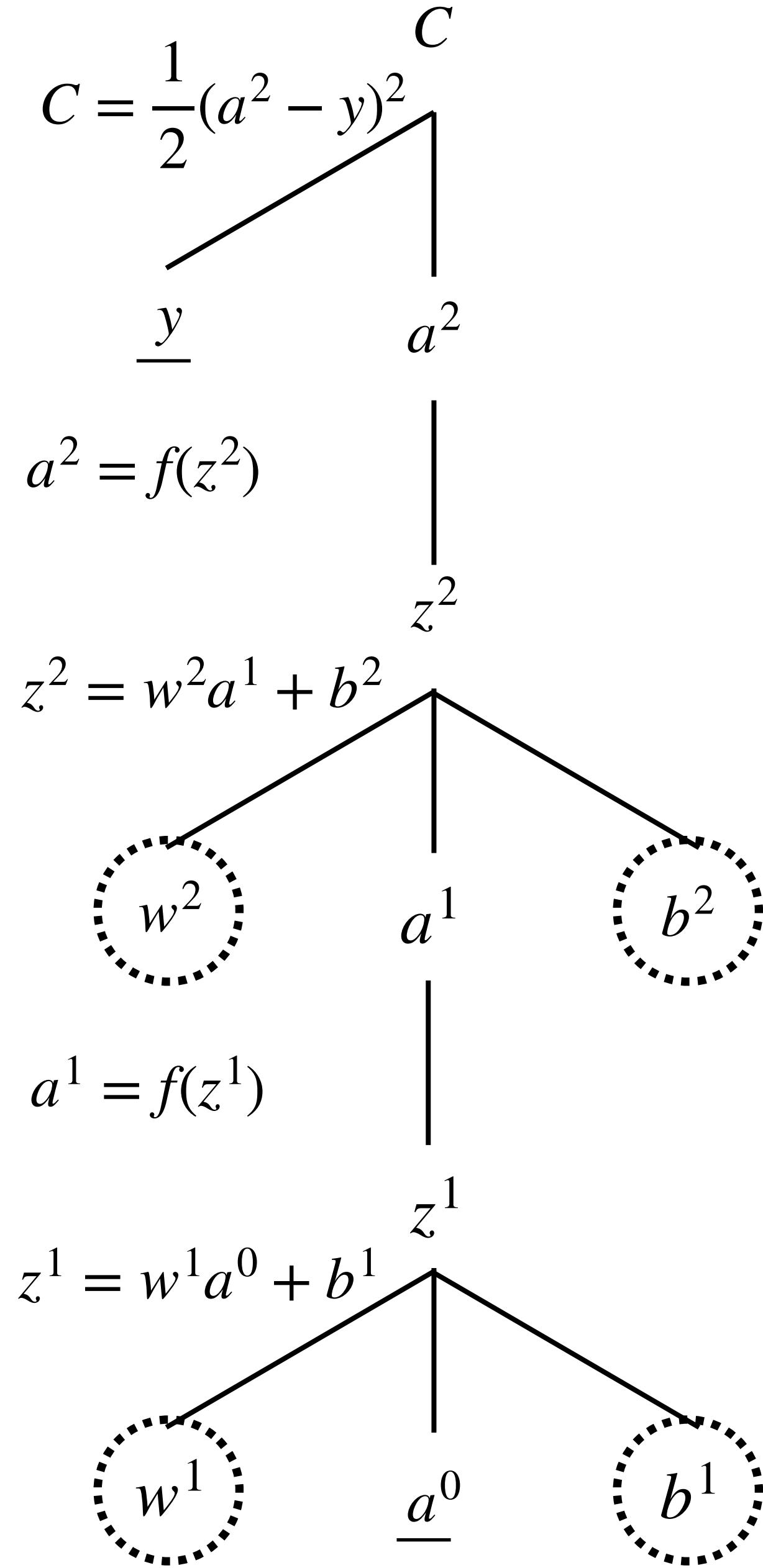


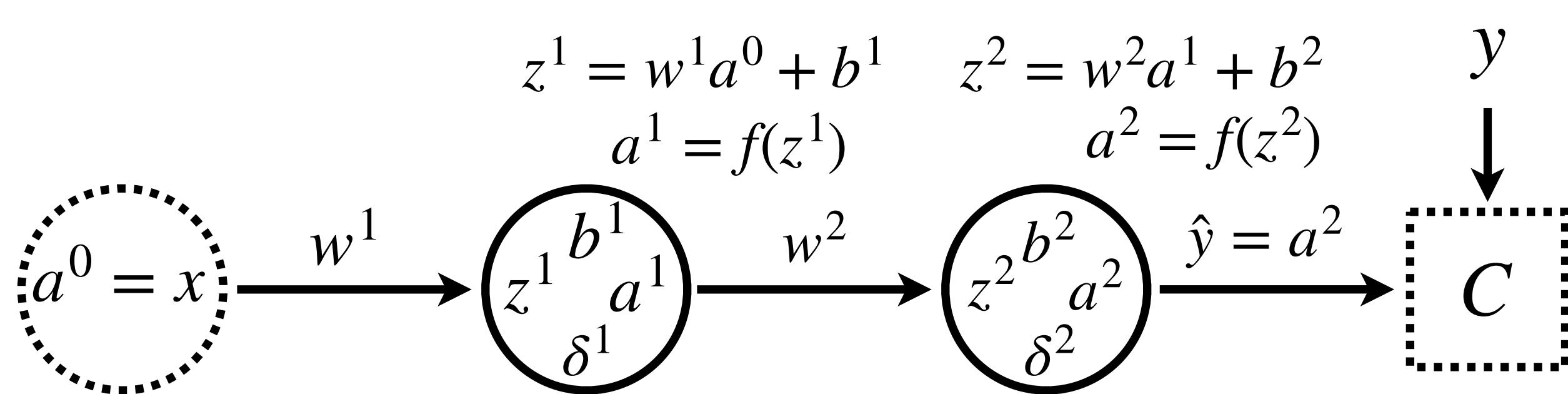
$$C(a^2(z^2(w^2, b^2))))$$

$$C(a^2(z^2(a^1(z^1(w^1, b^1))))))$$

$$\theta' = \theta - \alpha \nabla C$$

$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$





$$C(a^2(z^2(w^2, b^2)))$$

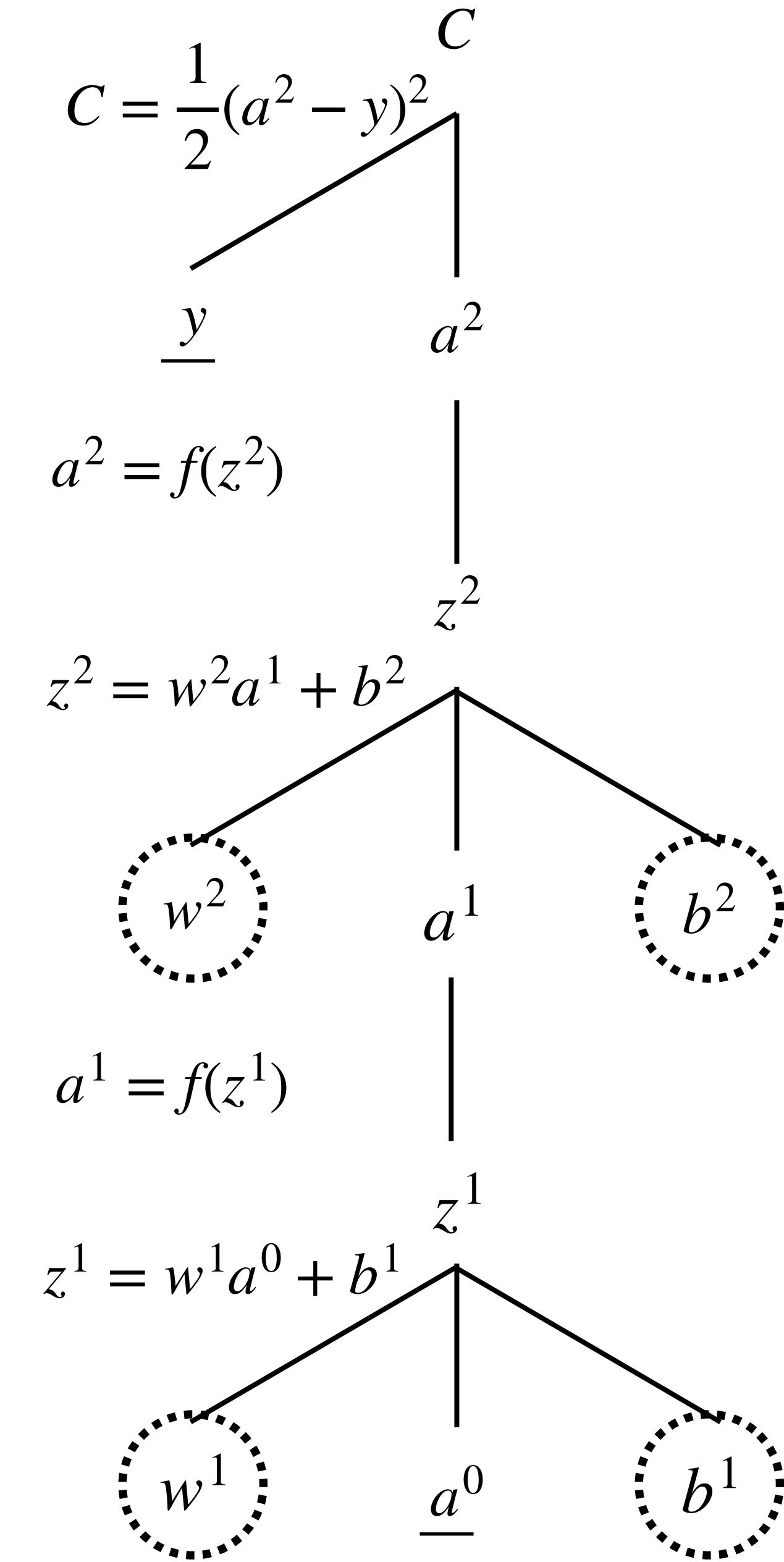
$$C(a^2(z^2(a^1(z^1(w^1, b^1))))))$$

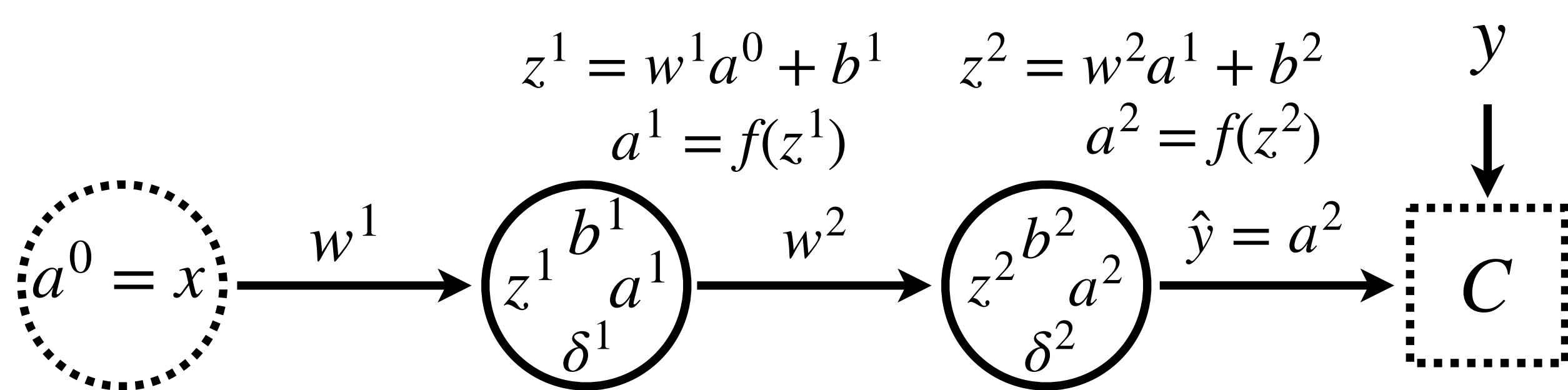
$$\theta' = \theta - \alpha \nabla C$$

$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$

$$\frac{\partial C}{\partial w^2} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial w^2} = \delta^2 \frac{\partial z^2}{\partial w^2} = \delta^2 a^1$$

$$\frac{\partial C}{\partial b^2} = \delta^2 \frac{\partial z^2}{\partial b^2} = \delta^2$$





$$C(a^2(z^2(w^2, b^2)))$$

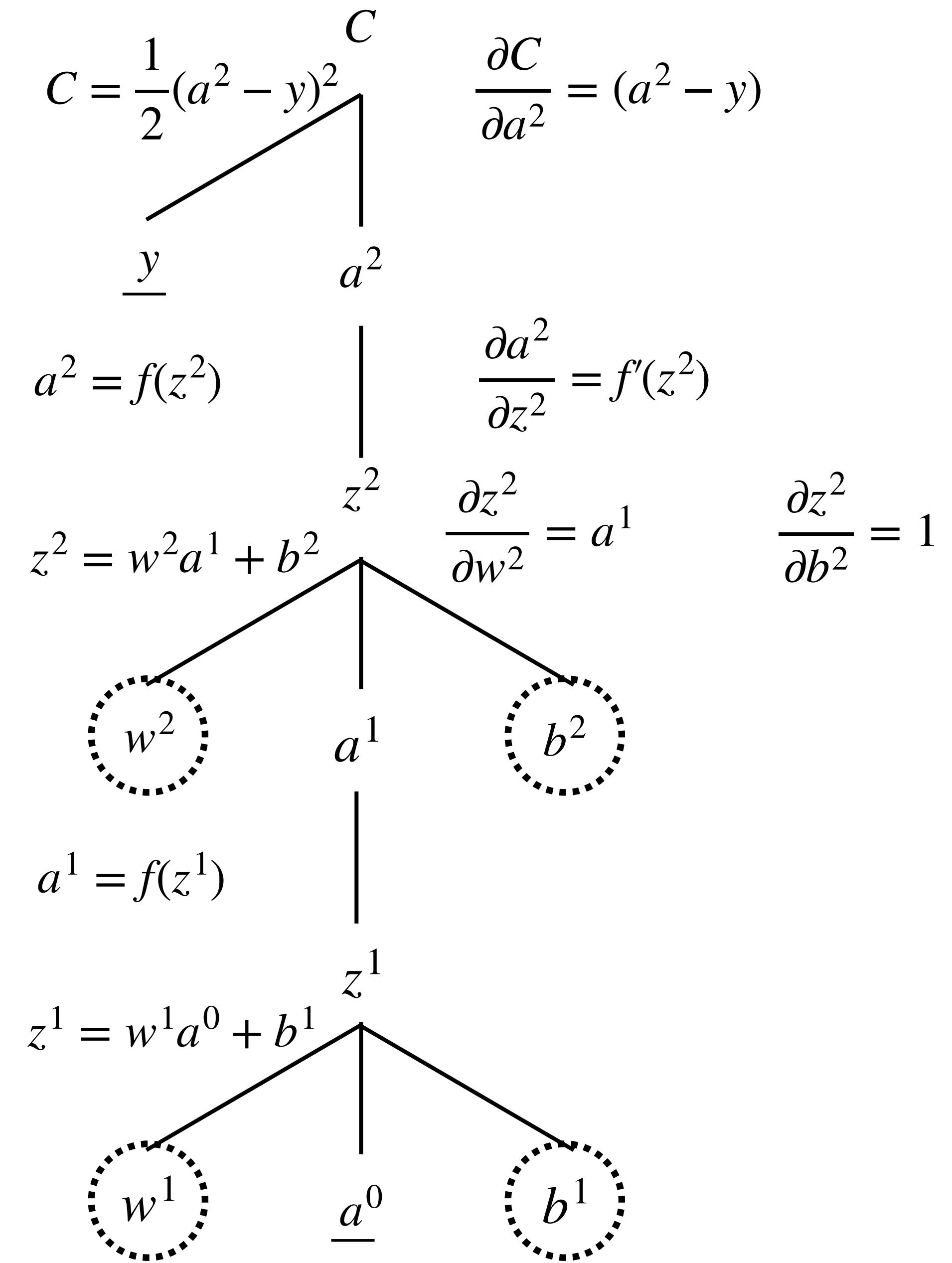
$$C(a^2(z^2(a^1(z^1(w^1, b^1))))))$$

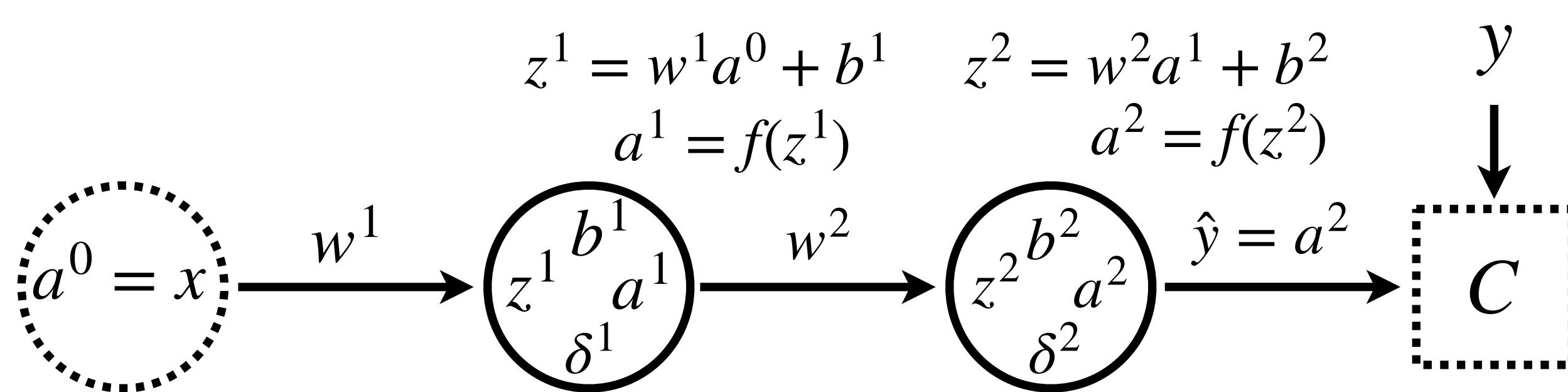
$$\theta' = \theta - \alpha \nabla C$$

$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$

$$\frac{\partial C}{\partial w^2} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial w^2} = \delta^2 \frac{\partial z^2}{\partial w^2} = \delta^2 a^1$$

$$\frac{\partial C}{\partial b^2} = \delta^2 \frac{\partial z^2}{\partial b^2} = \delta^2$$





$$C(a^2(z^2(w^2, b^2)))$$

$$C(a^2(z^2(a^1(z^1(w^1, b^1)))))$$

$$\theta' = \theta - \alpha \nabla C$$

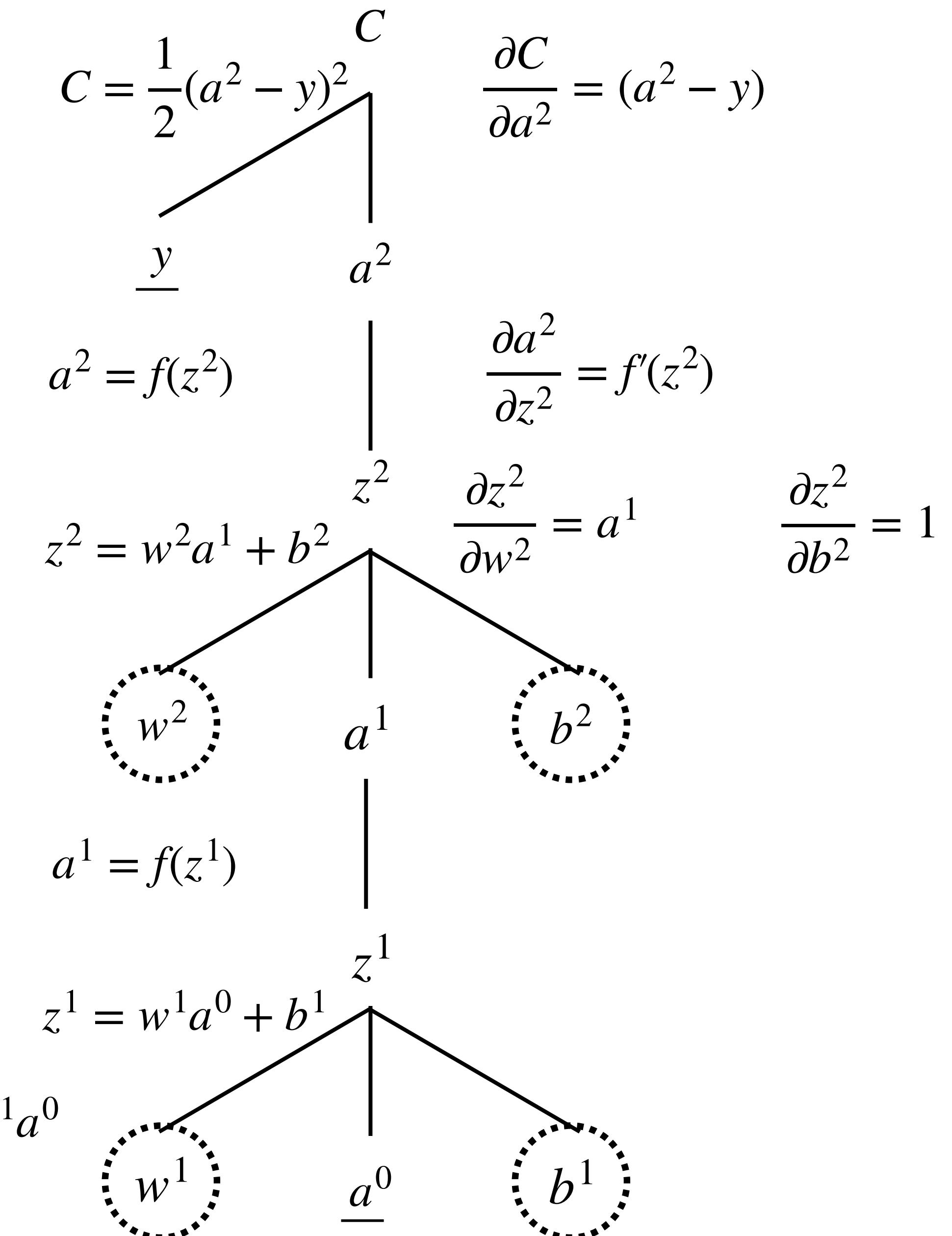
$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$

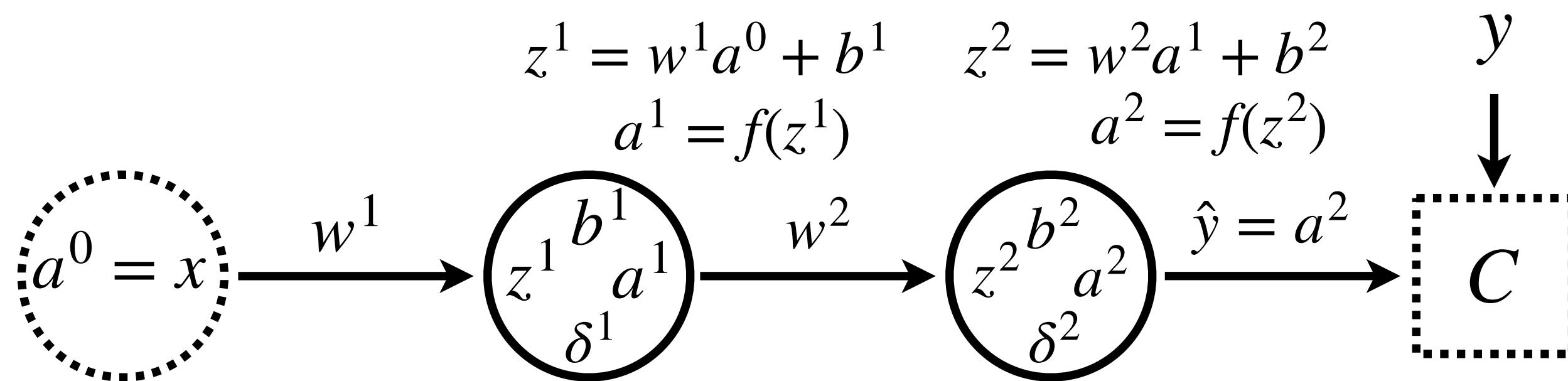
$$\frac{\partial C}{\partial w^2} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial w^2} = \delta^2 \frac{\partial z^2}{\partial w^2} = \delta^2 a^1$$

$$\frac{\partial C}{\partial b^2} = \delta^2 \frac{\partial z^2}{\partial b^2} = \delta^2$$

$$\frac{\partial C}{\partial w^1} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial a^1} \frac{\partial a^1}{\partial z^1} \frac{\partial z^1}{\partial w^1} = \delta^2 w^2 f'(z^1) a^0 = \delta^1 a^0$$

$$\frac{\partial C}{\partial b^1} = \delta^2 w^2 f'(z^1) = w^2 \delta^2 f'(z^1) = \delta^1$$





$$C(a^2(z^2(w^2, b^2)))$$

$$C(a^2(z^2(a^1(z^1(w^1, b^1))))))$$

$$\theta' = \theta - \alpha \nabla C$$

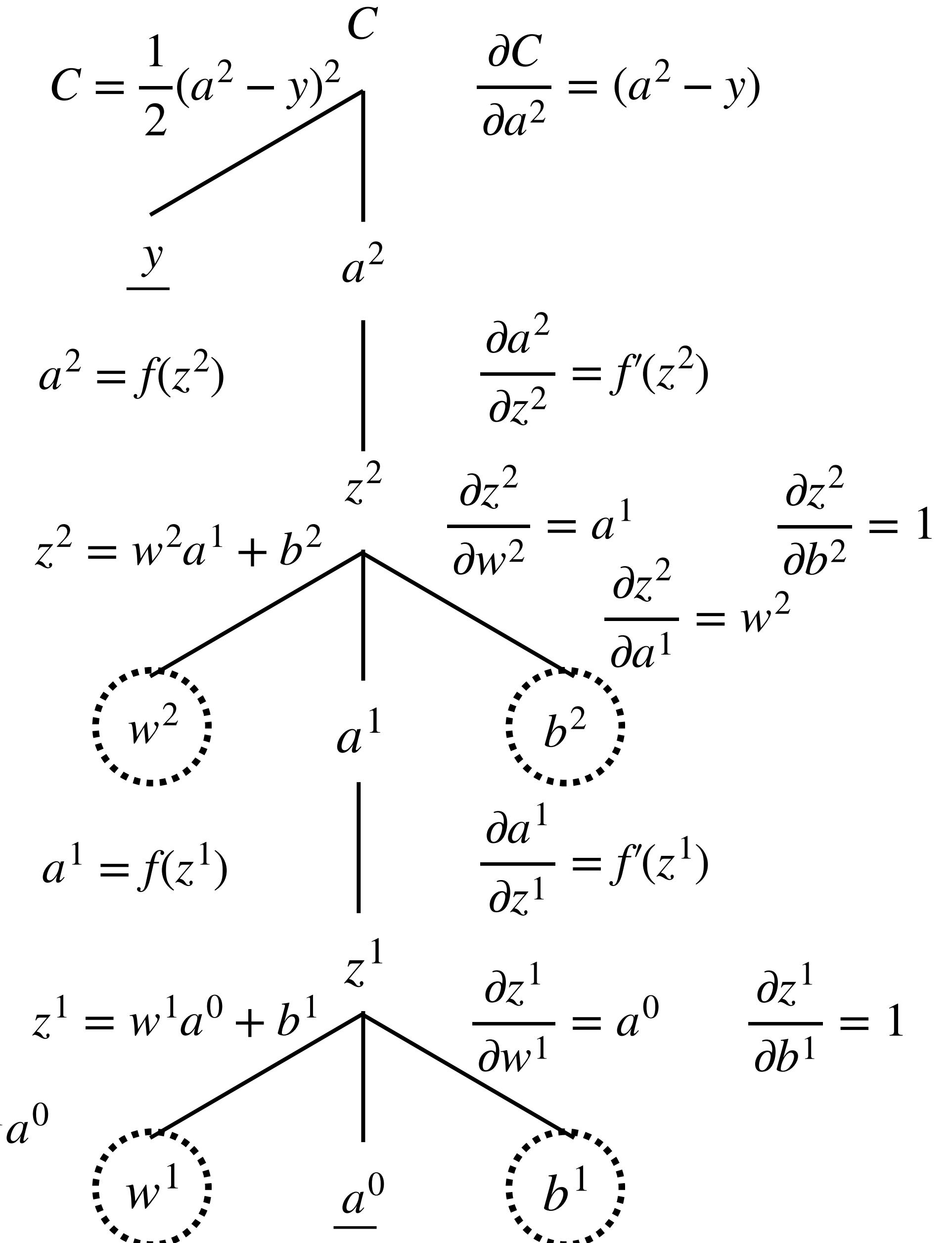
$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$

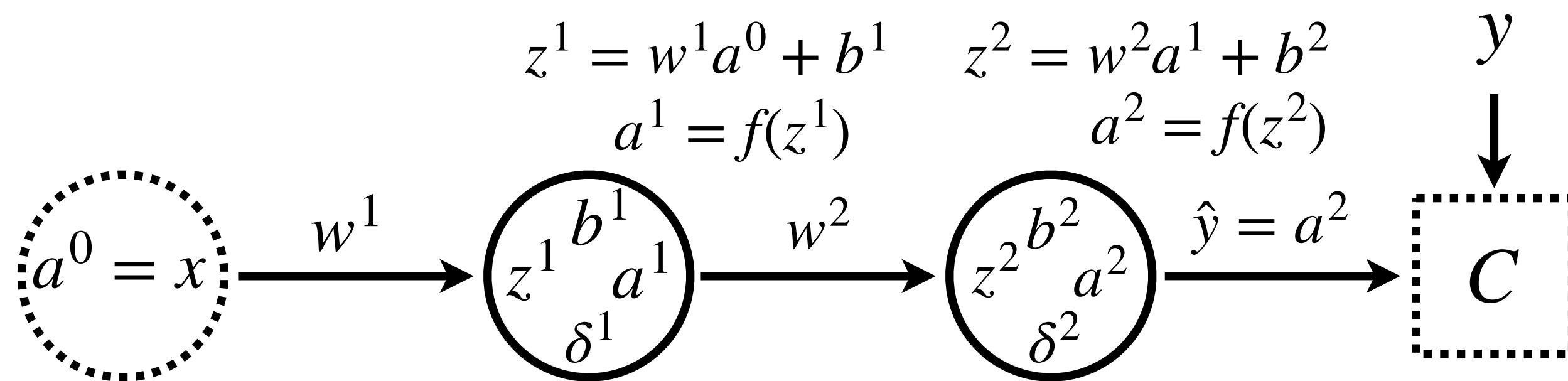
$$\frac{\partial C}{\partial w^2} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial w^2} = \delta^2 \frac{\partial z^2}{\partial w^2} = \delta^2 a^1$$

$$\frac{\partial C}{\partial b^2} = \delta^2 \frac{\partial z^2}{\partial b^2} = \delta^2$$

$$\frac{\partial C}{\partial w^1} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial a^1} \frac{\partial a^1}{\partial z^1} \frac{\partial z^1}{\partial w^1} = \delta^2 w^2 f'(z^1) a^0 = \delta^1 a^0$$

$$\frac{\partial C}{\partial b^1} = \delta^2 w^2 f'(z^1) = w^2 \delta^2 f'(z^1) = \delta^1$$





$$C(a^2(z^2(w^2, b^2)))$$

$$C(a^2(z^2(a^1(z^1(w^1, b^1)))))$$

$$\theta' = \theta - \alpha \nabla C$$

$$\begin{bmatrix} w^{1'} \\ b^{1'} \\ w^{2'} \\ b^{2'} \end{bmatrix} = \begin{bmatrix} w^1 \\ b^1 \\ w^2 \\ b^2 \end{bmatrix} - \alpha \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \frac{\partial C}{\partial w^2} \\ \frac{\partial C}{\partial b^2} \end{bmatrix}$$

$$\frac{\partial C}{\partial w^2} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial w^2} = \delta^2 \frac{\partial z^2}{\partial w^2} = \delta^2 a^1$$

$$\frac{\partial C}{\partial b^2} = \delta^2 \frac{\partial z^2}{\partial b^2} = \delta^2$$

$$\frac{\partial C}{\partial w^1} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} \frac{\partial z^2}{\partial a^1} \frac{\partial a^1}{\partial z^1} \frac{\partial z^1}{\partial w^1} = \delta^2 w^2 f'(z^1) a^0 = \delta^1 a^0$$

$$\frac{\partial C}{\partial b^1} = \delta^2 w^2 f'(z^1) = w^2 \delta^2 f'(z^1) = \delta^1$$

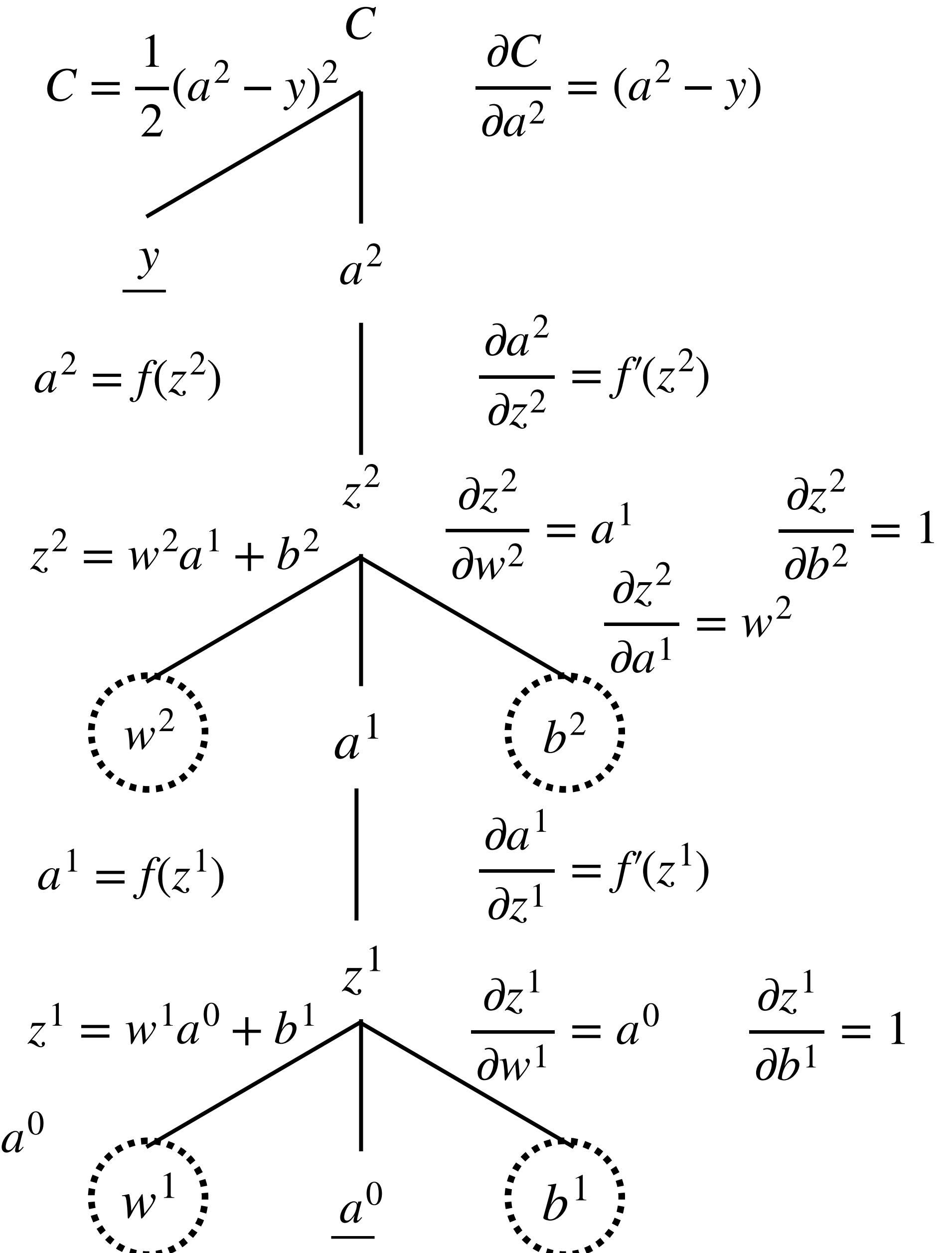
**For each training example n:**  $x^n$

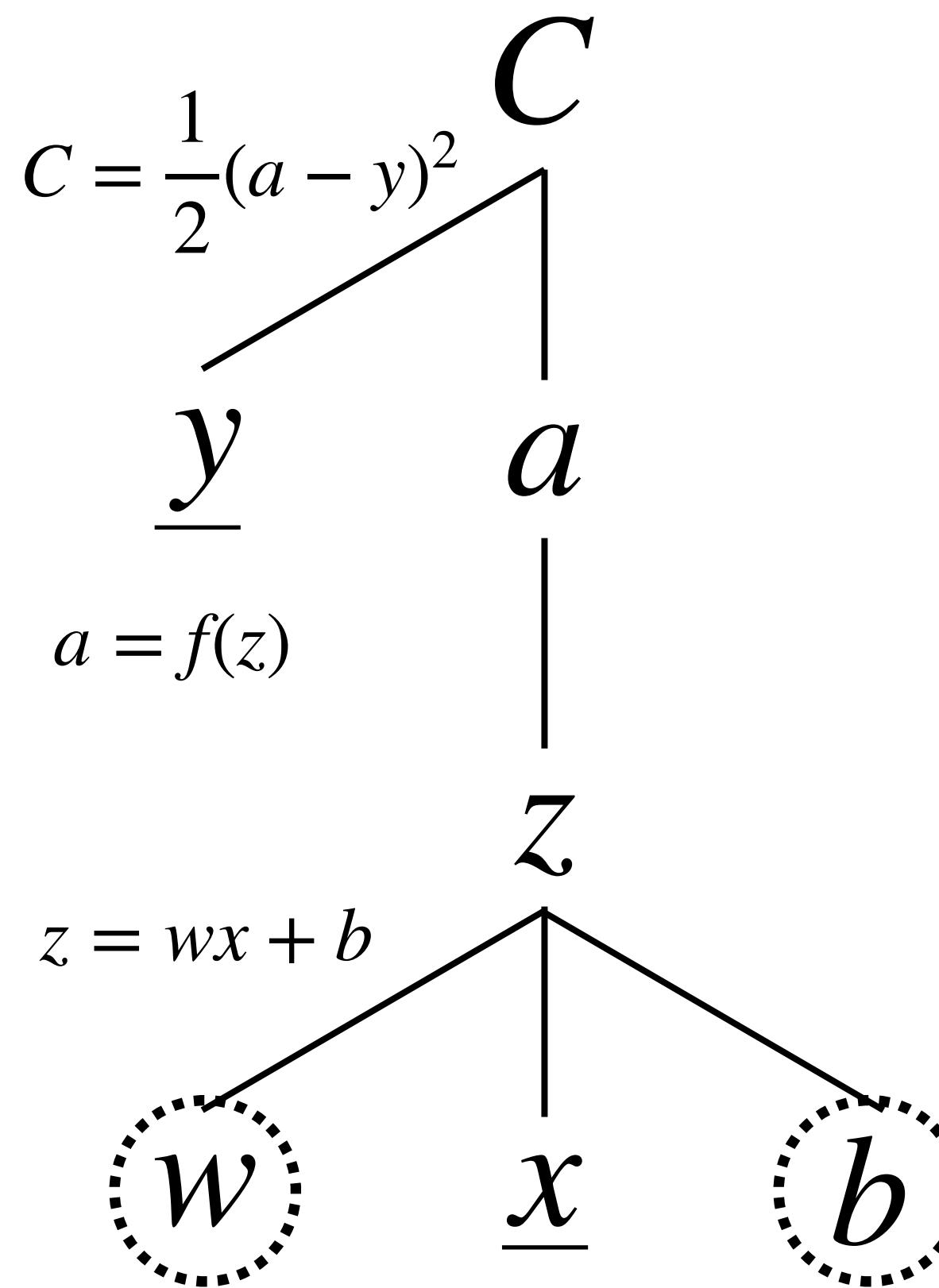
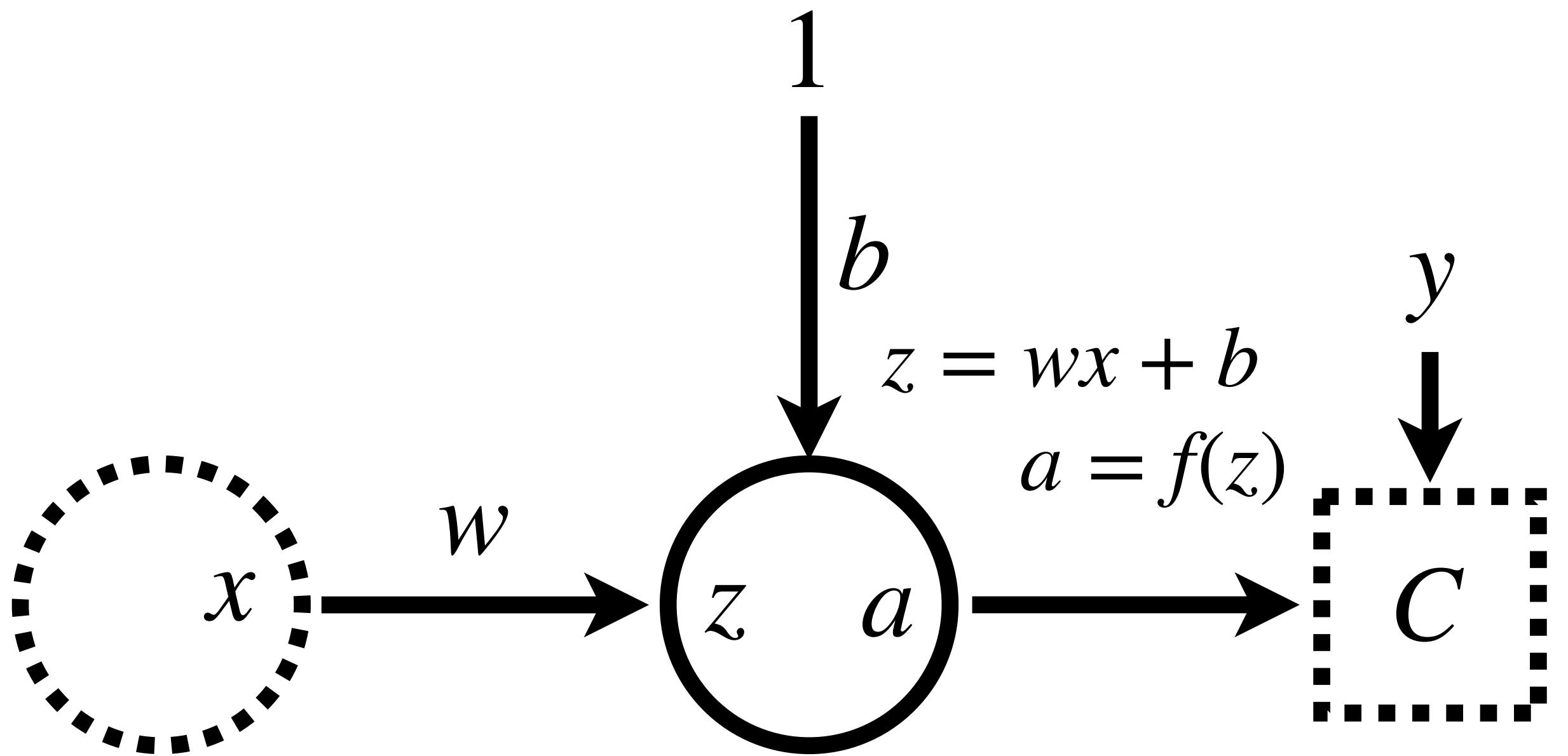
**Input activation:**  $a^{n,0} = x^n$

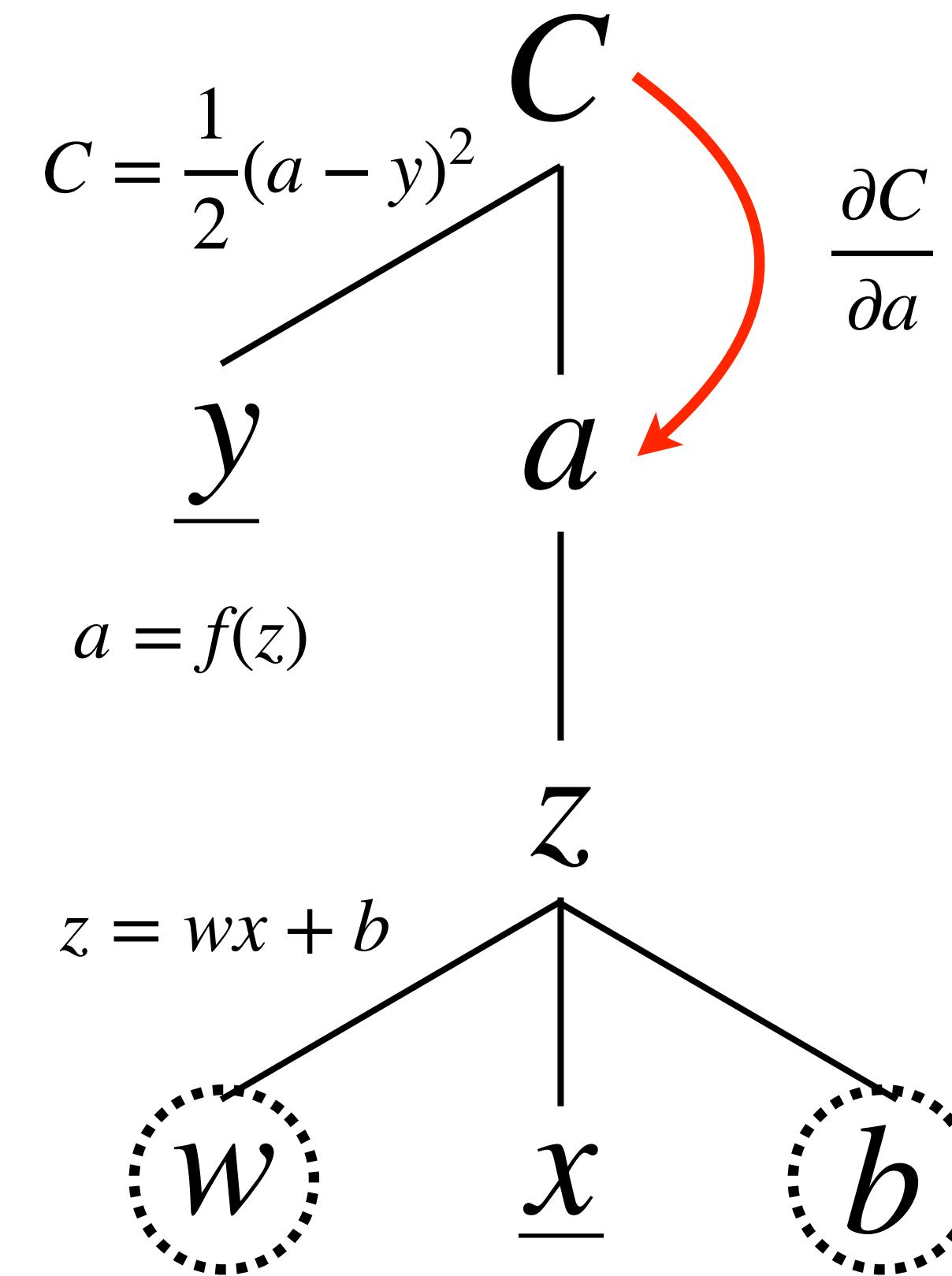
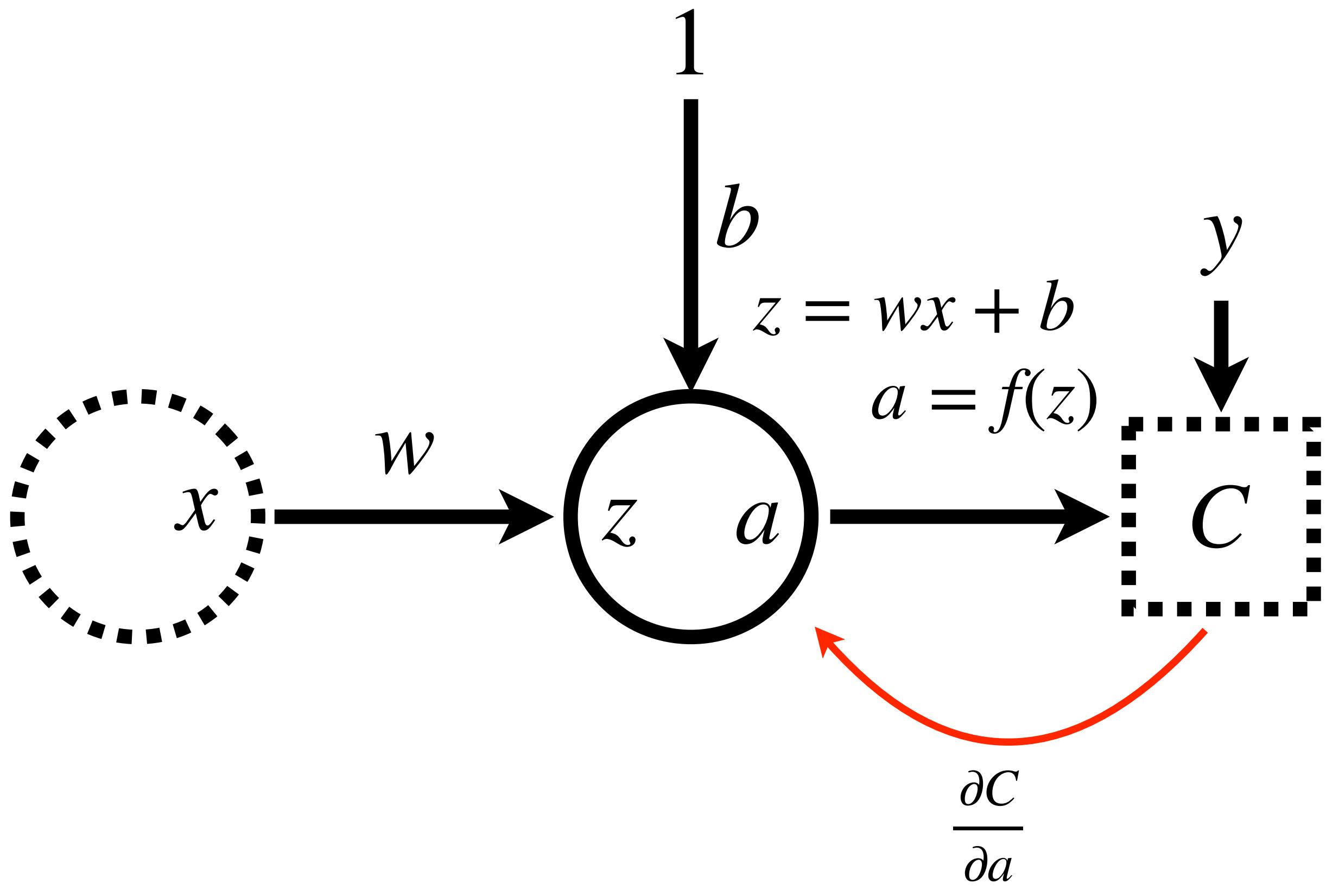
**Feed-forward:**  $z^{n,l} = w^l a^{n,l-1} + b^l$   
 $a^{n,l} = f(z^{n,l}) \quad l = 1, 2, \dots, L$

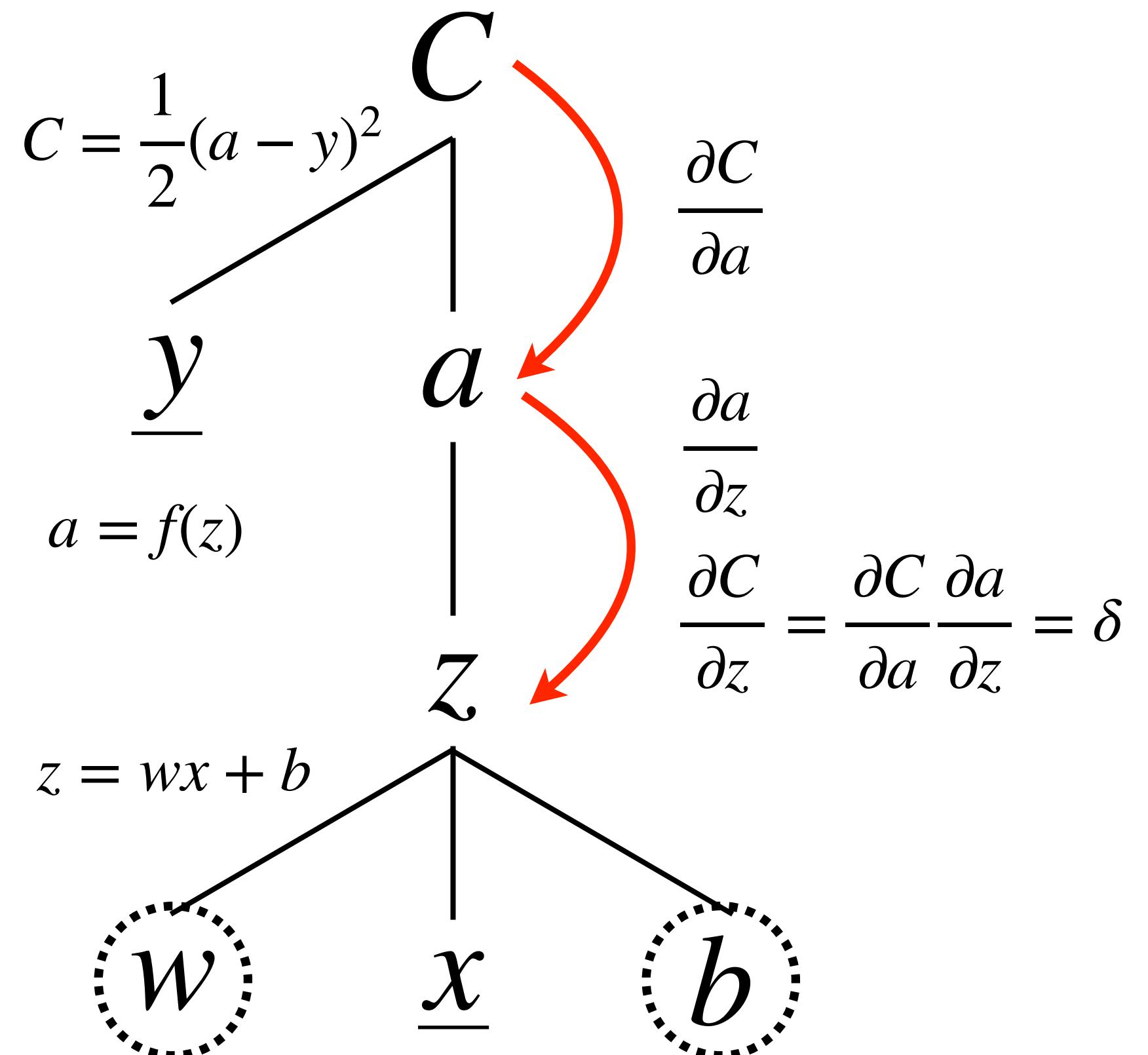
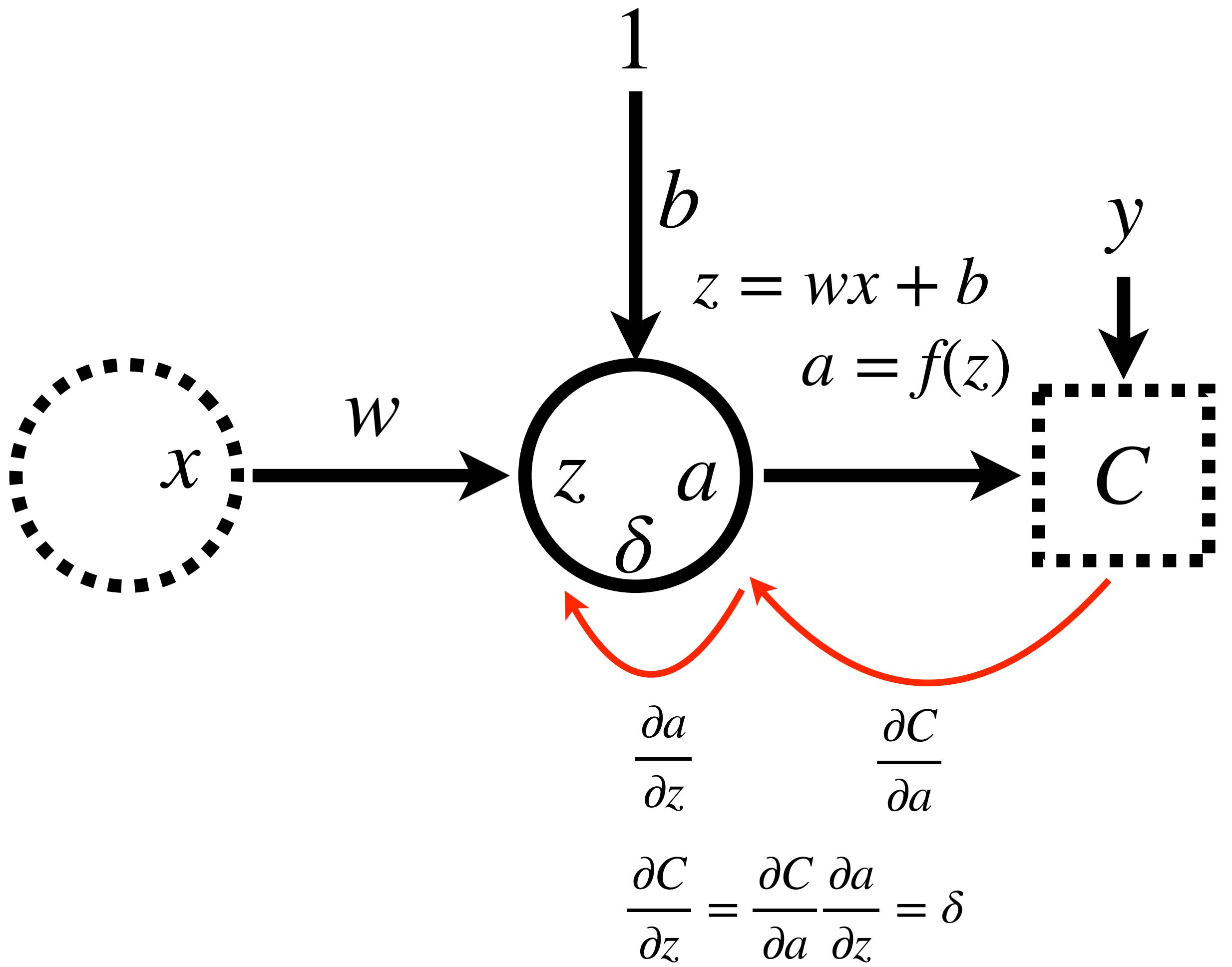
**Output error:**  $\delta^{n,L} = \nabla_a C_n \odot f'(z^{n,L})$

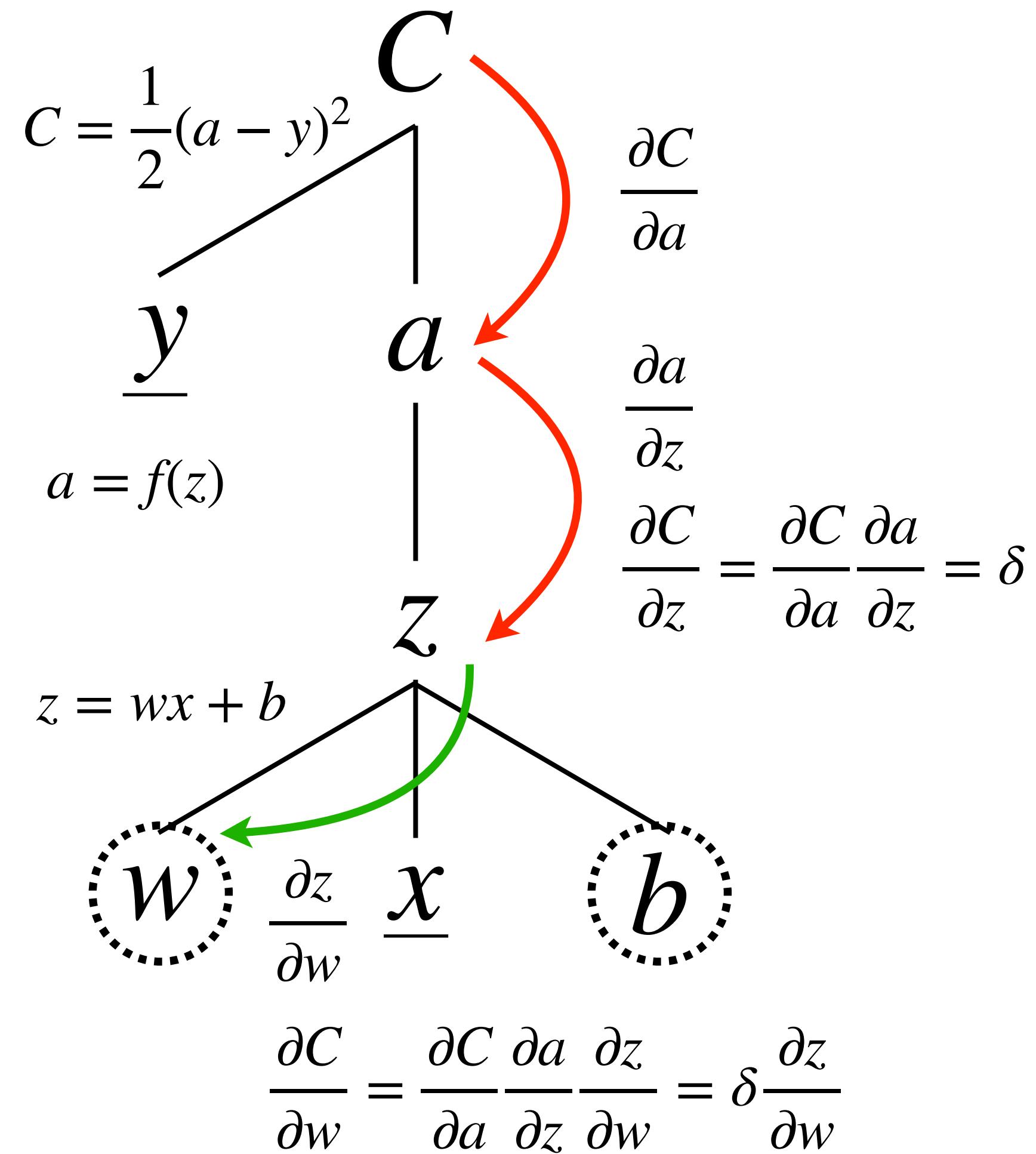
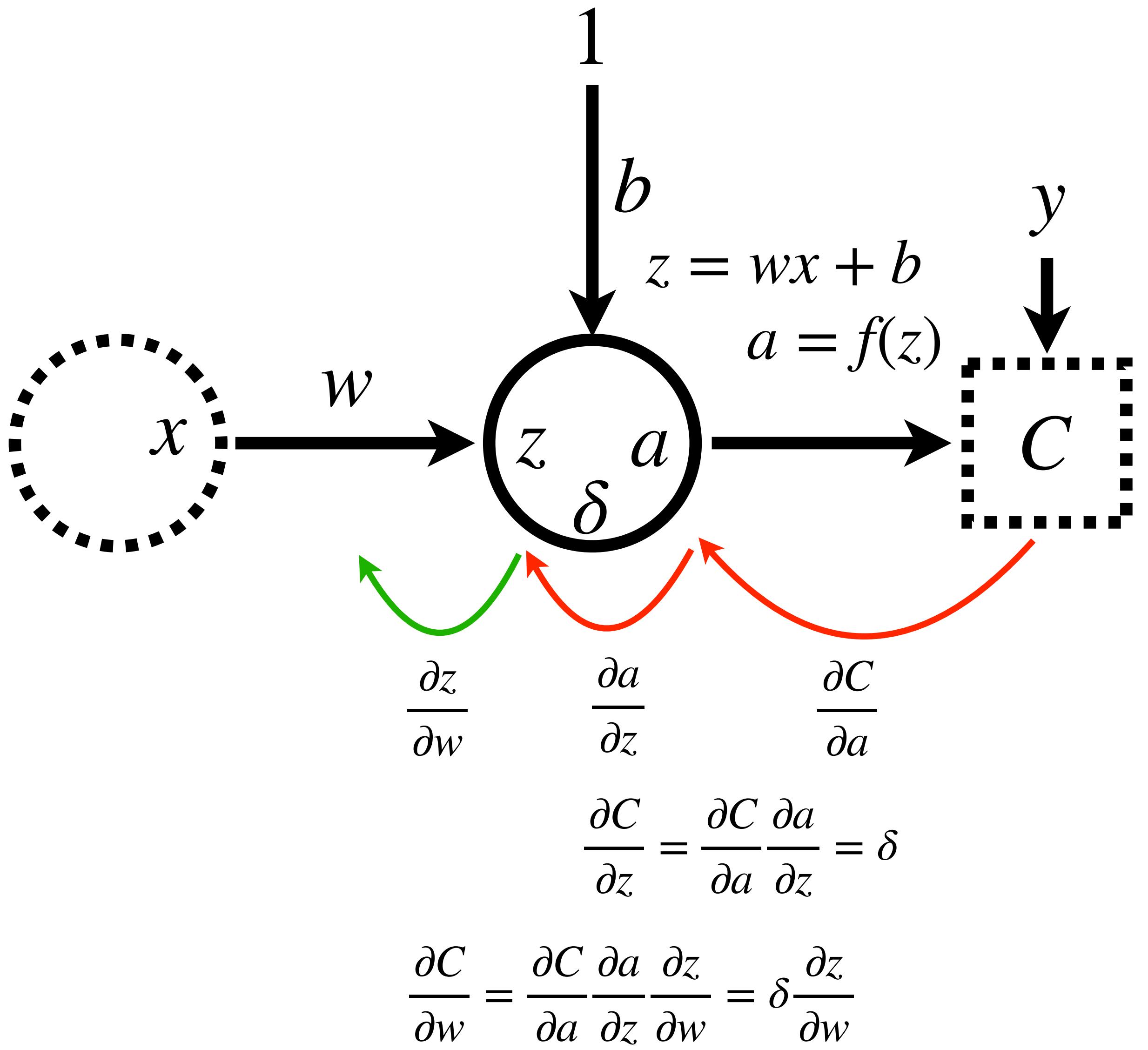
**Back-propagate:**  $\delta^{n,l} = ((w^{l+1})^T \delta^{n,l+1} \odot f'(z^{n,l}))$   
 $l = L-1, L-2, \dots, 1$

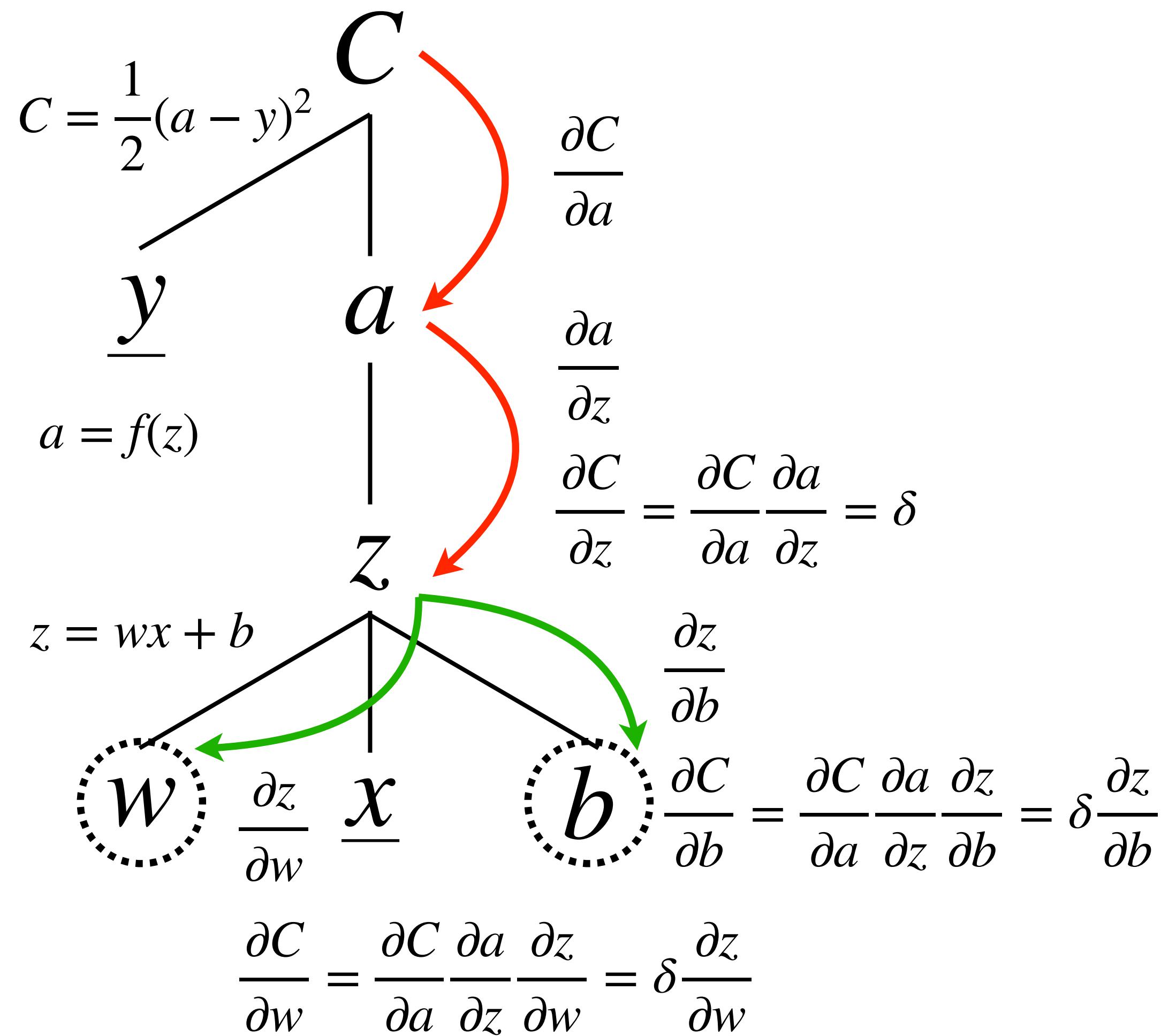
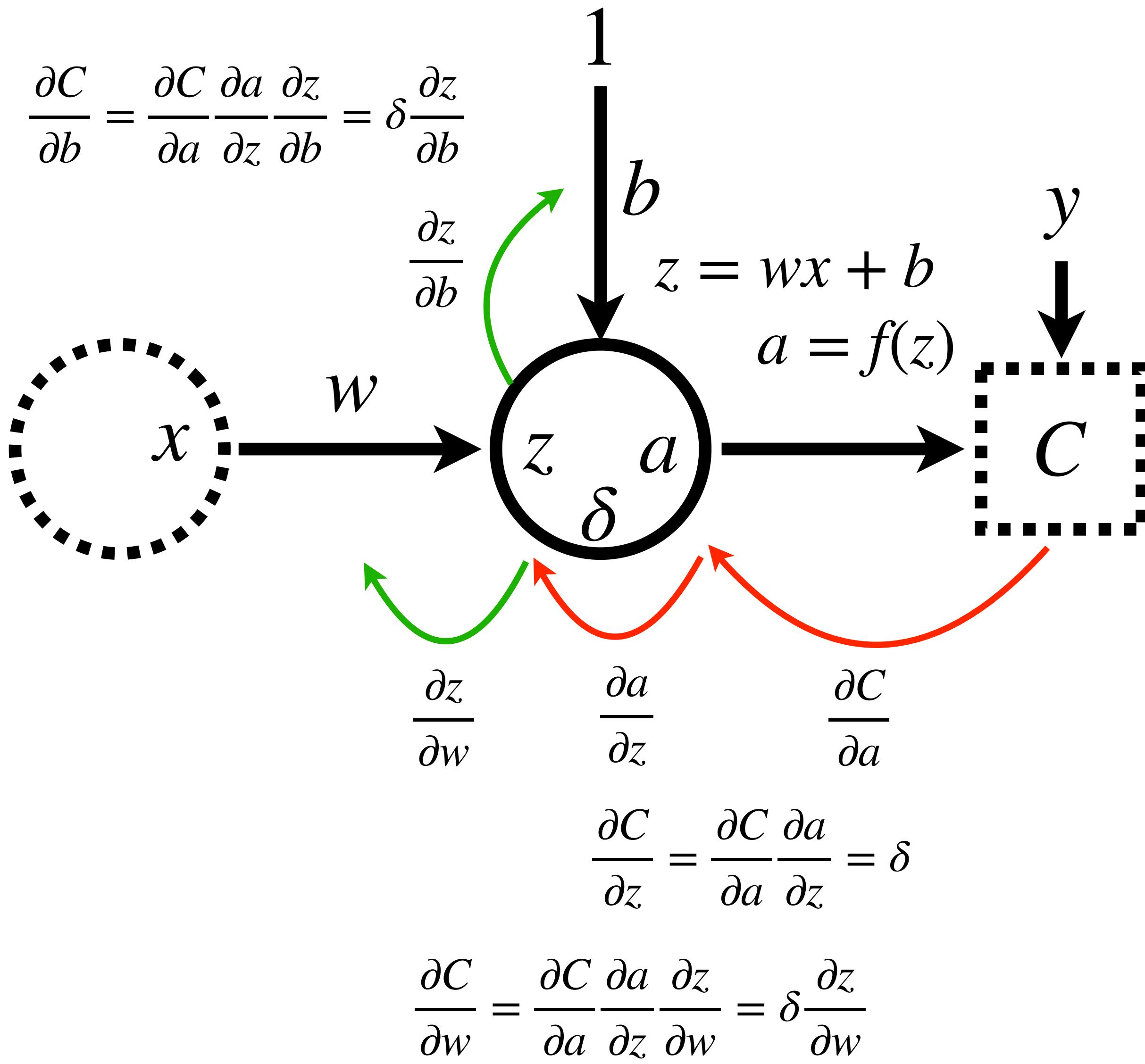


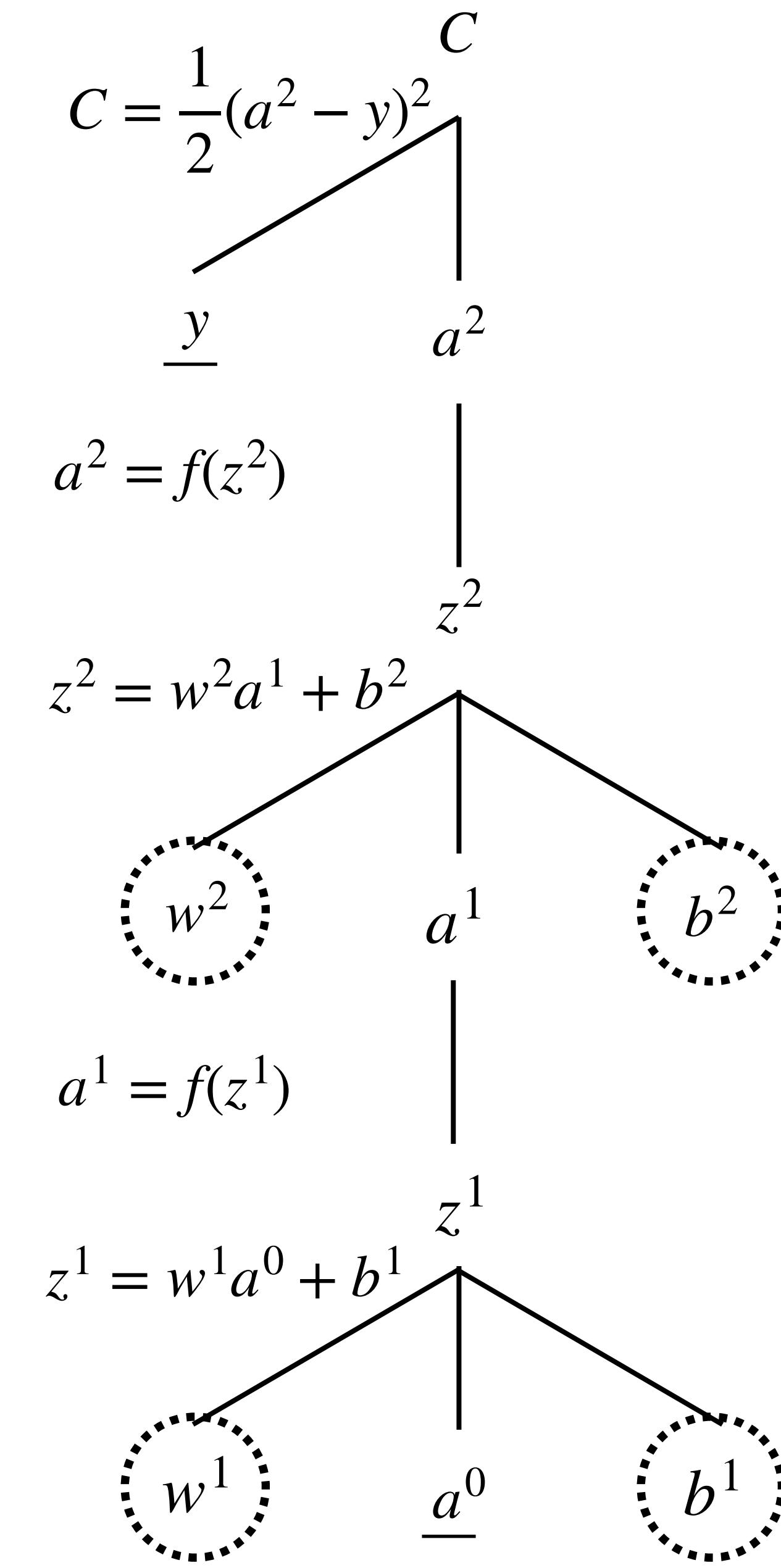
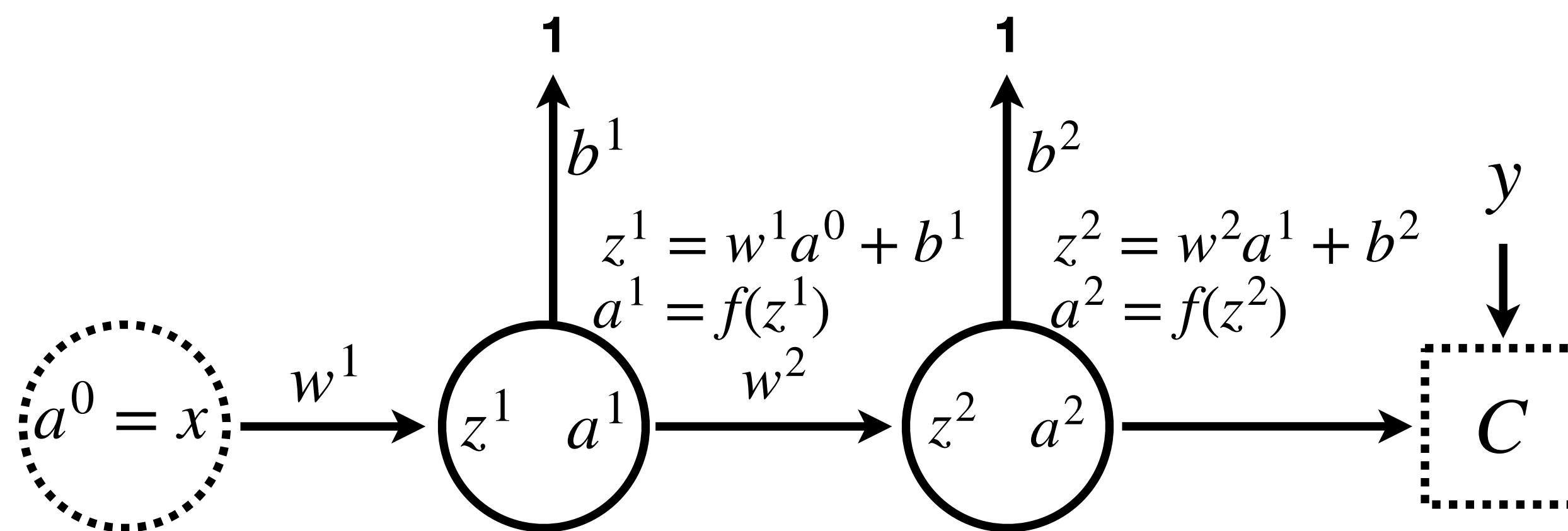


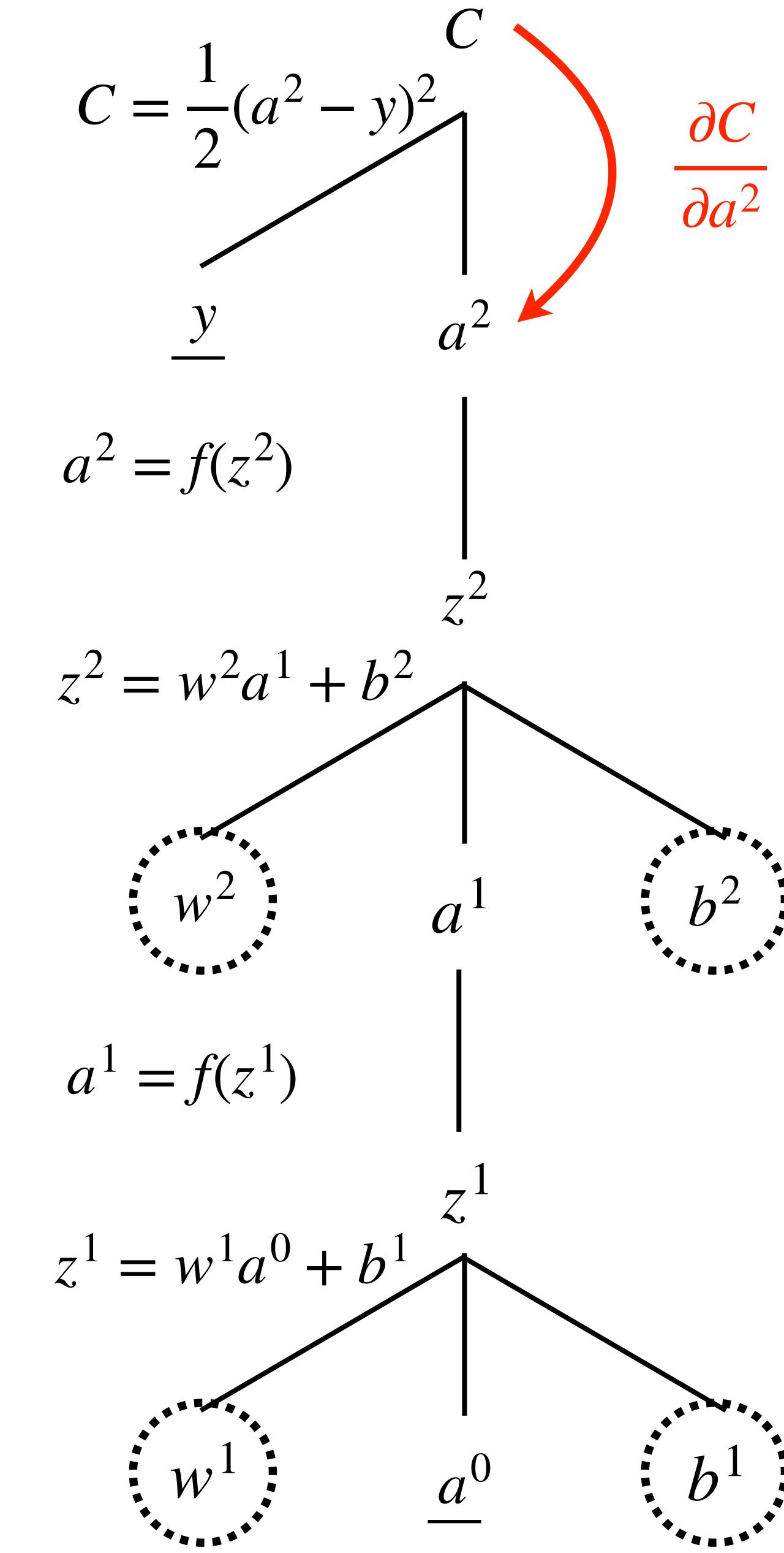
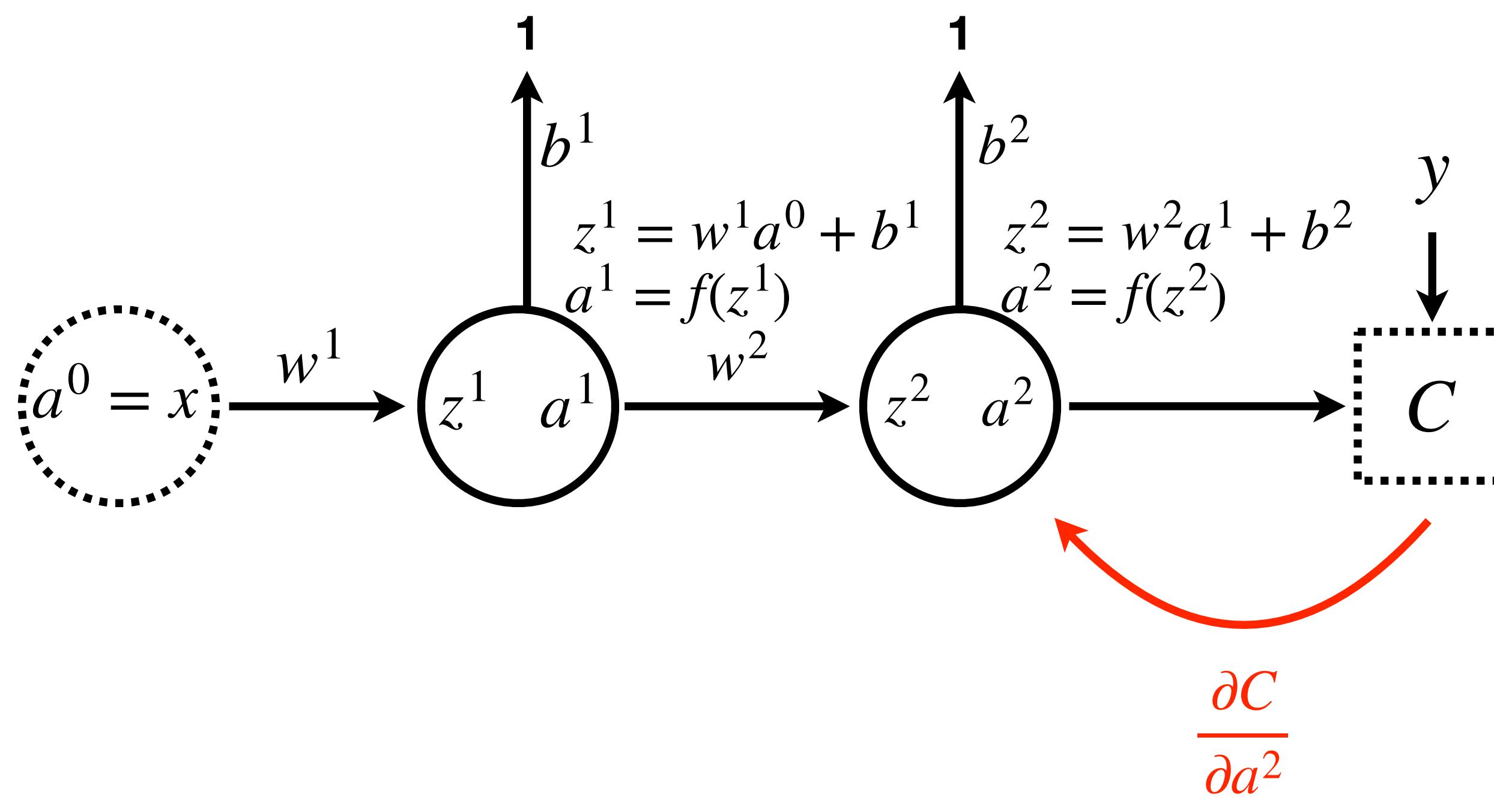


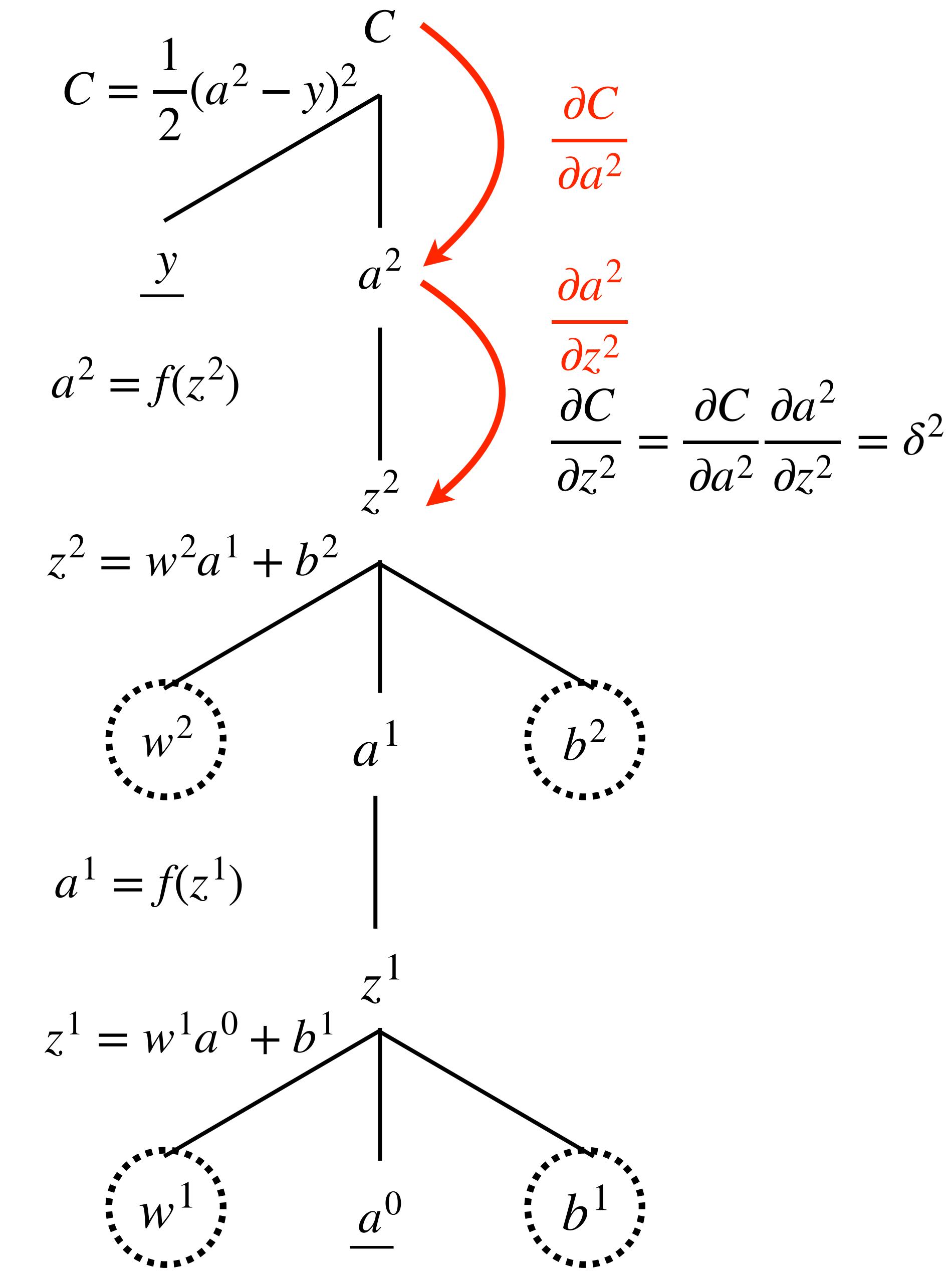
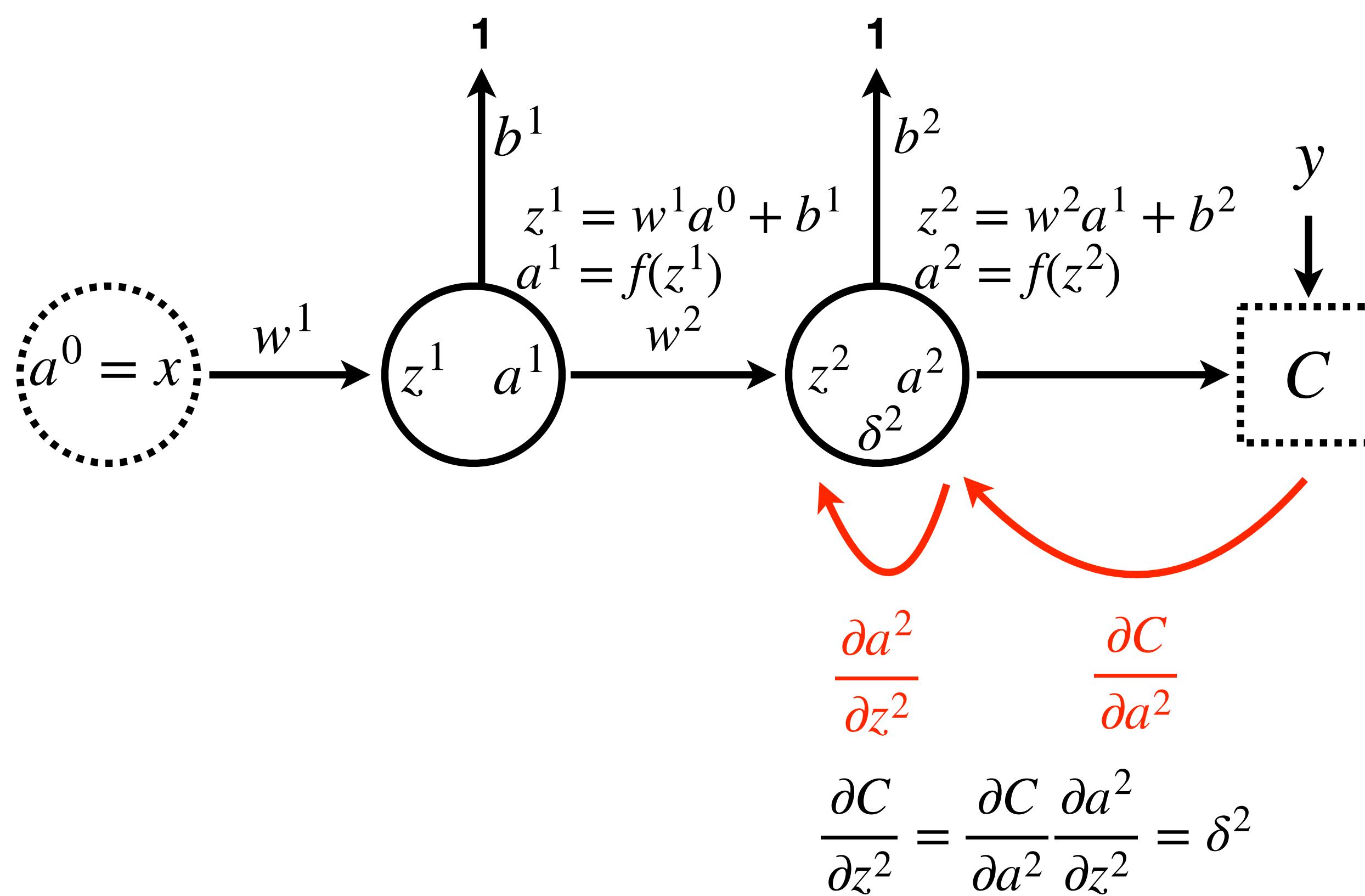


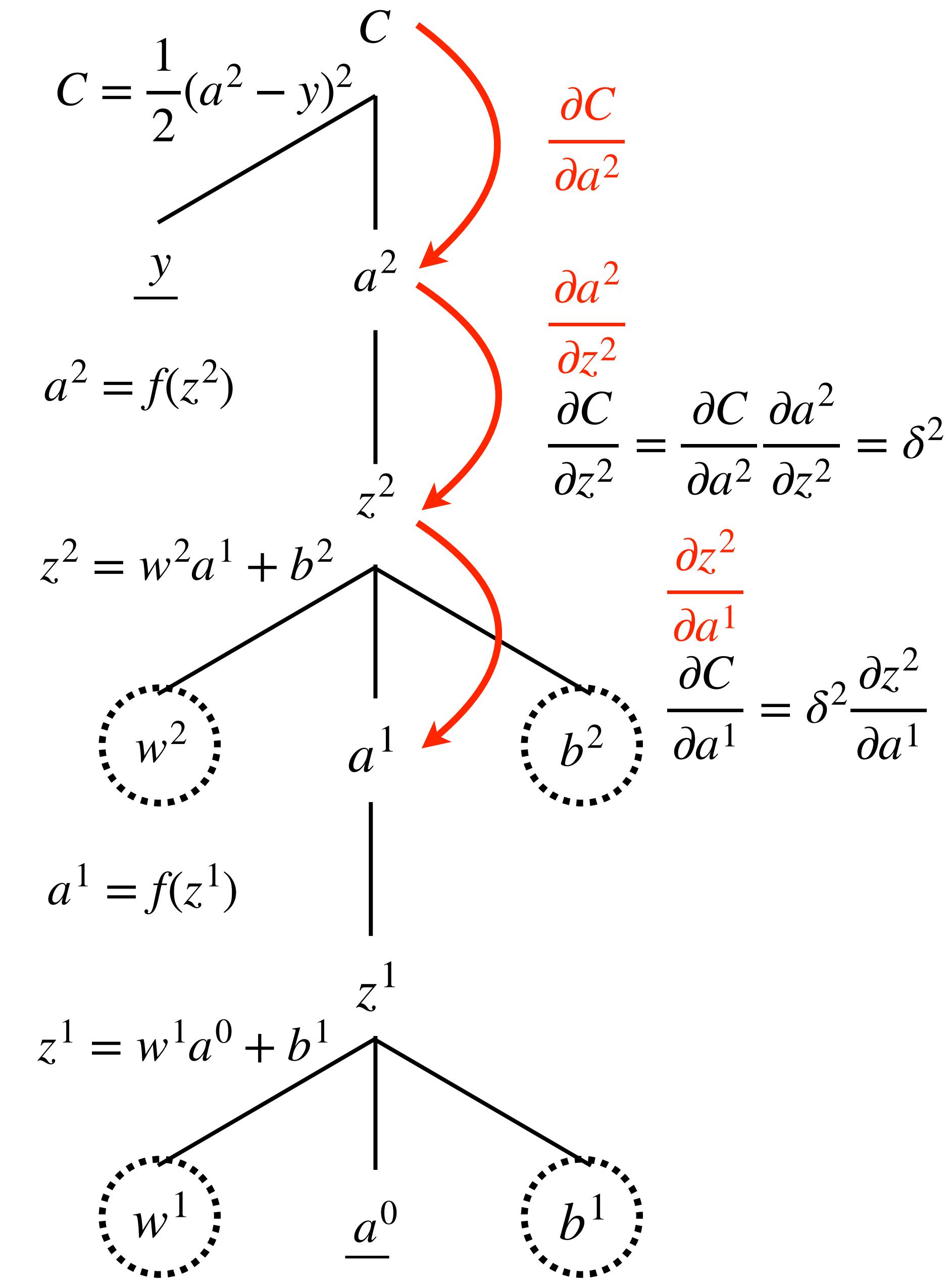
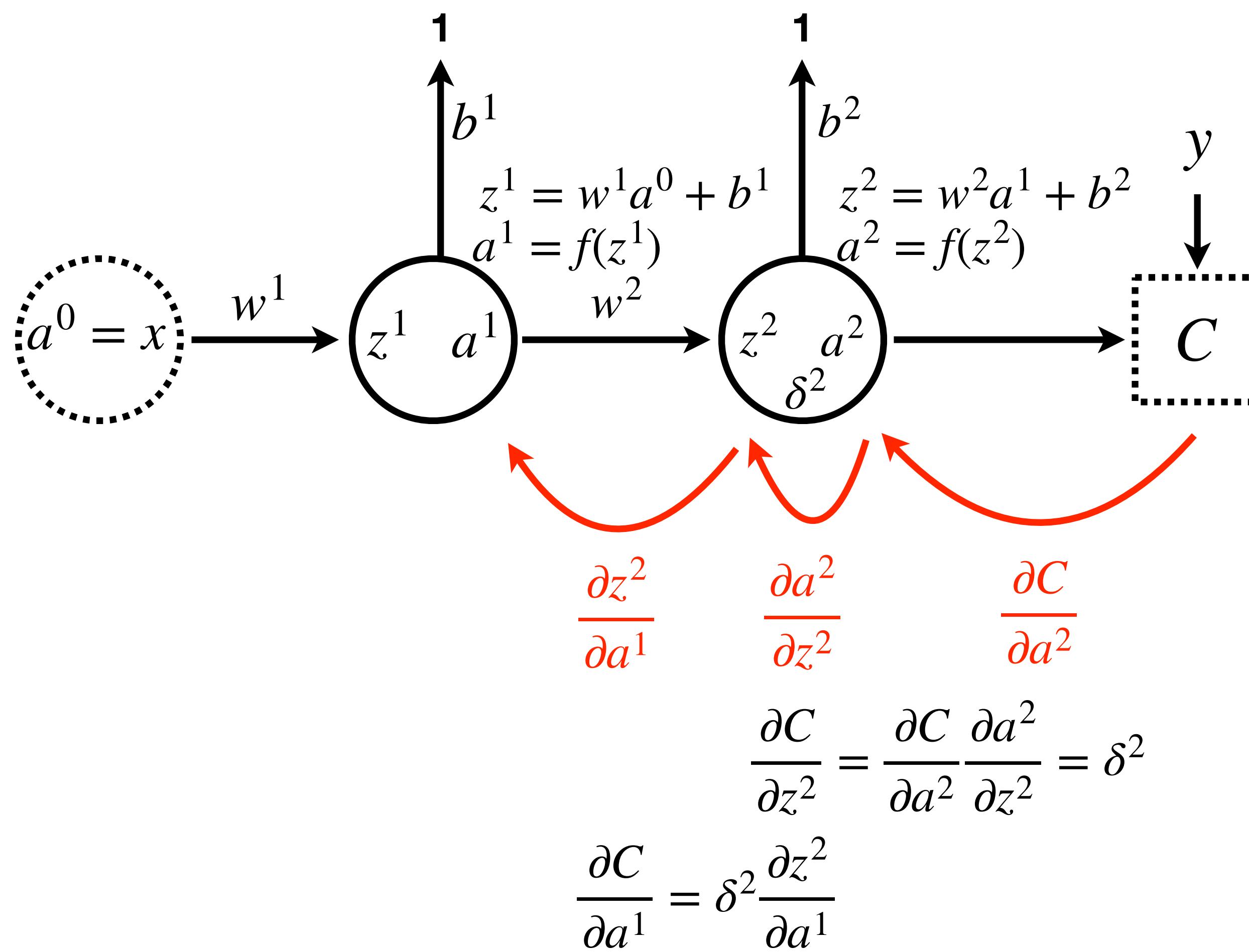


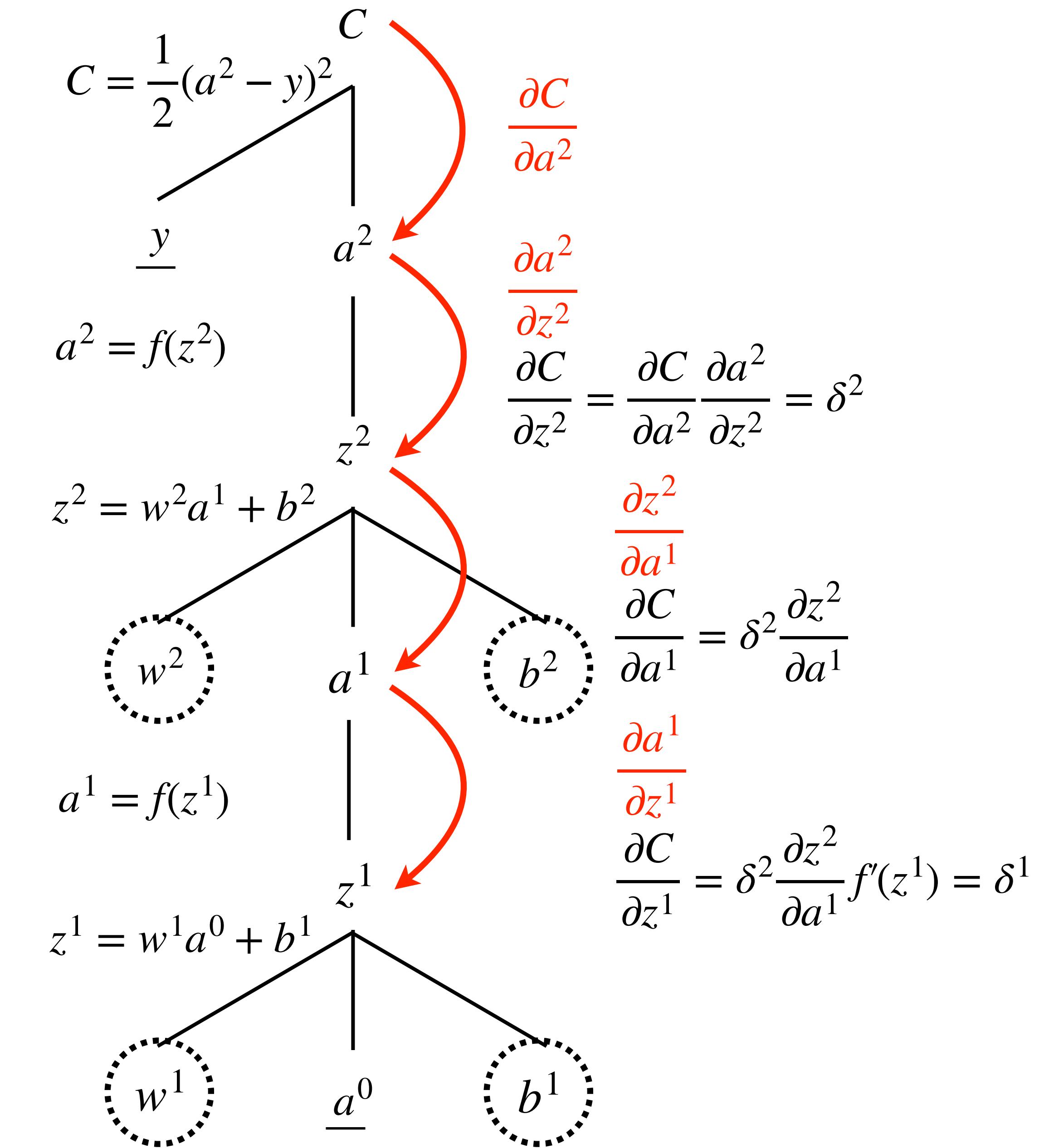
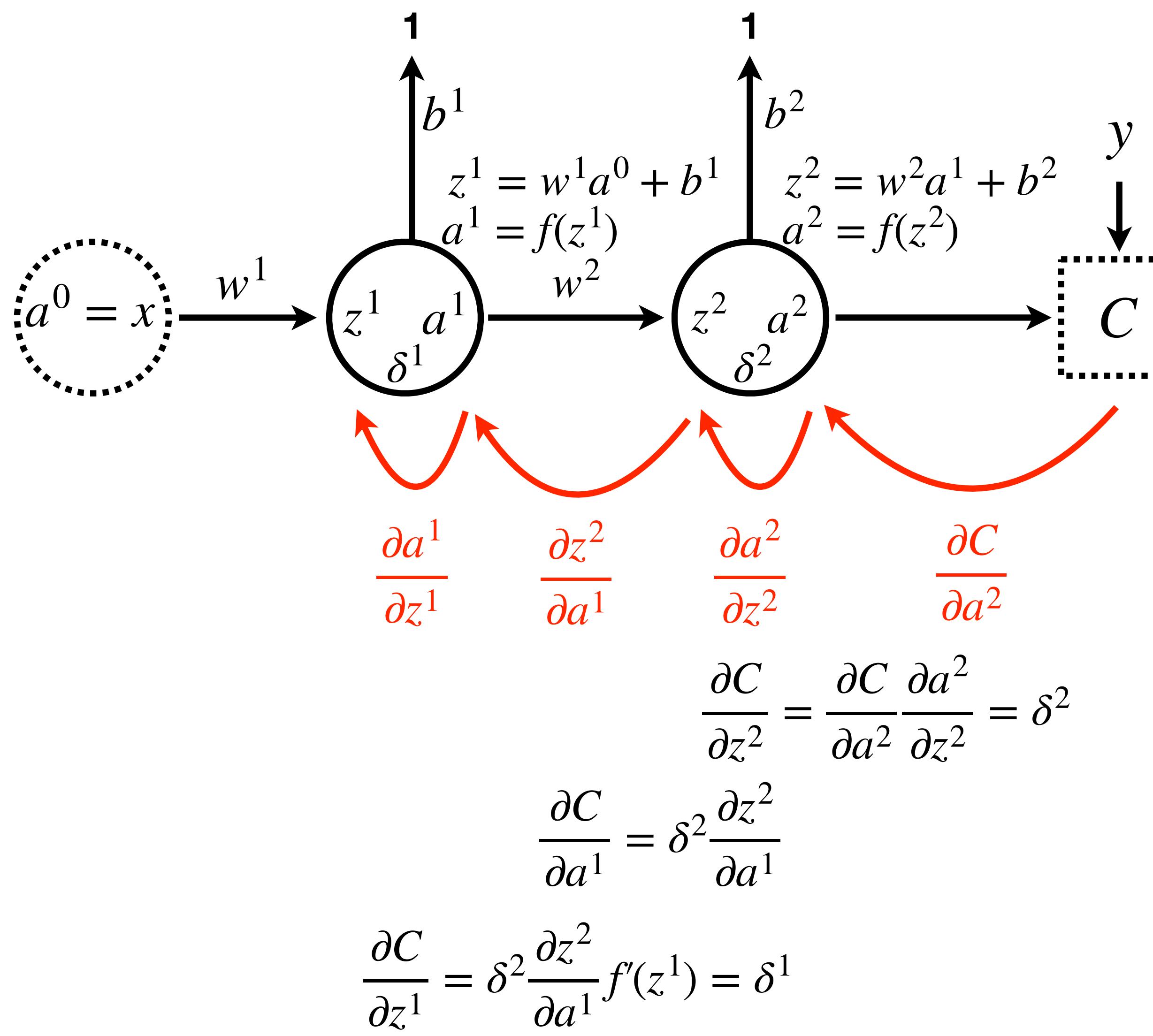


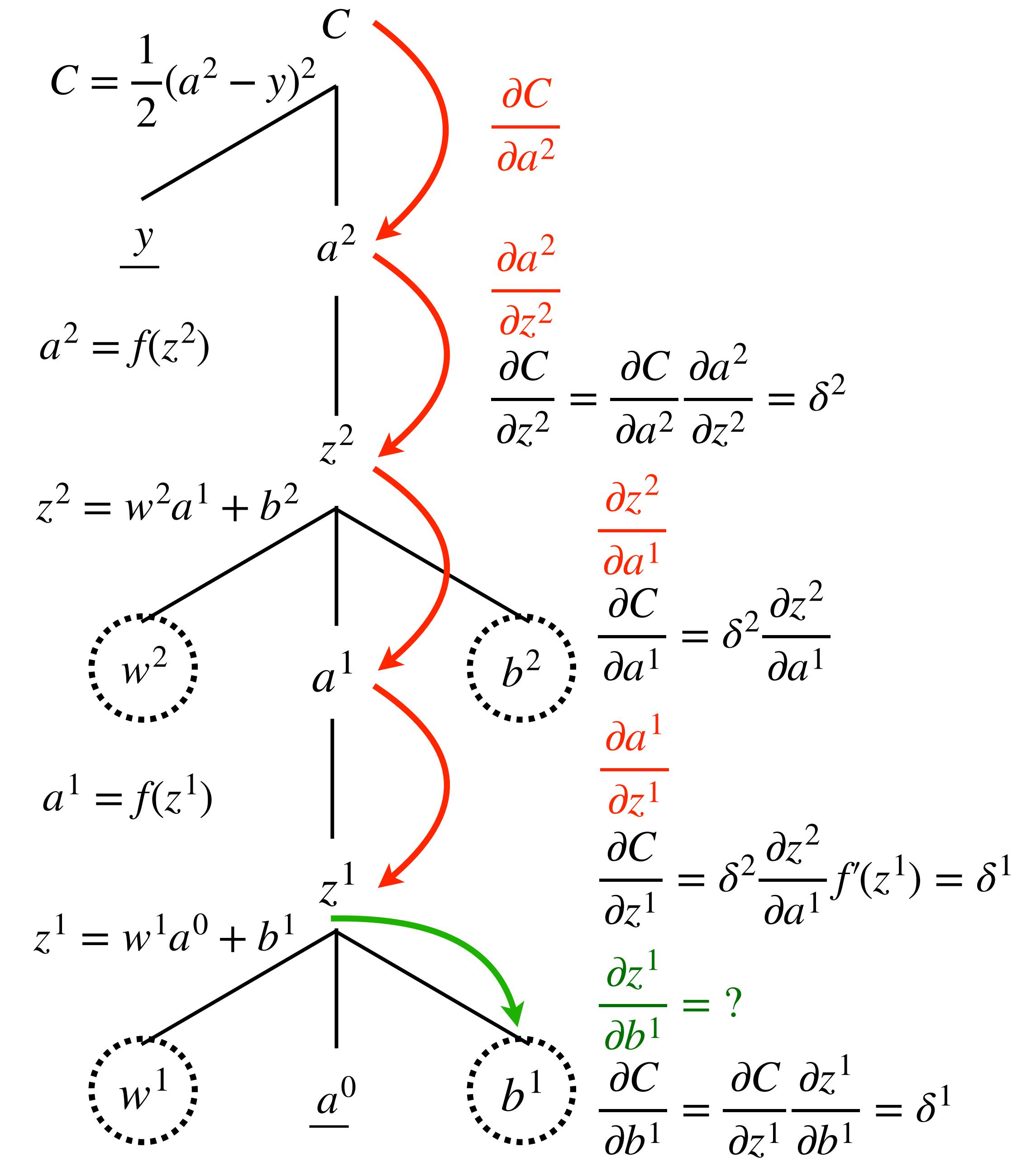
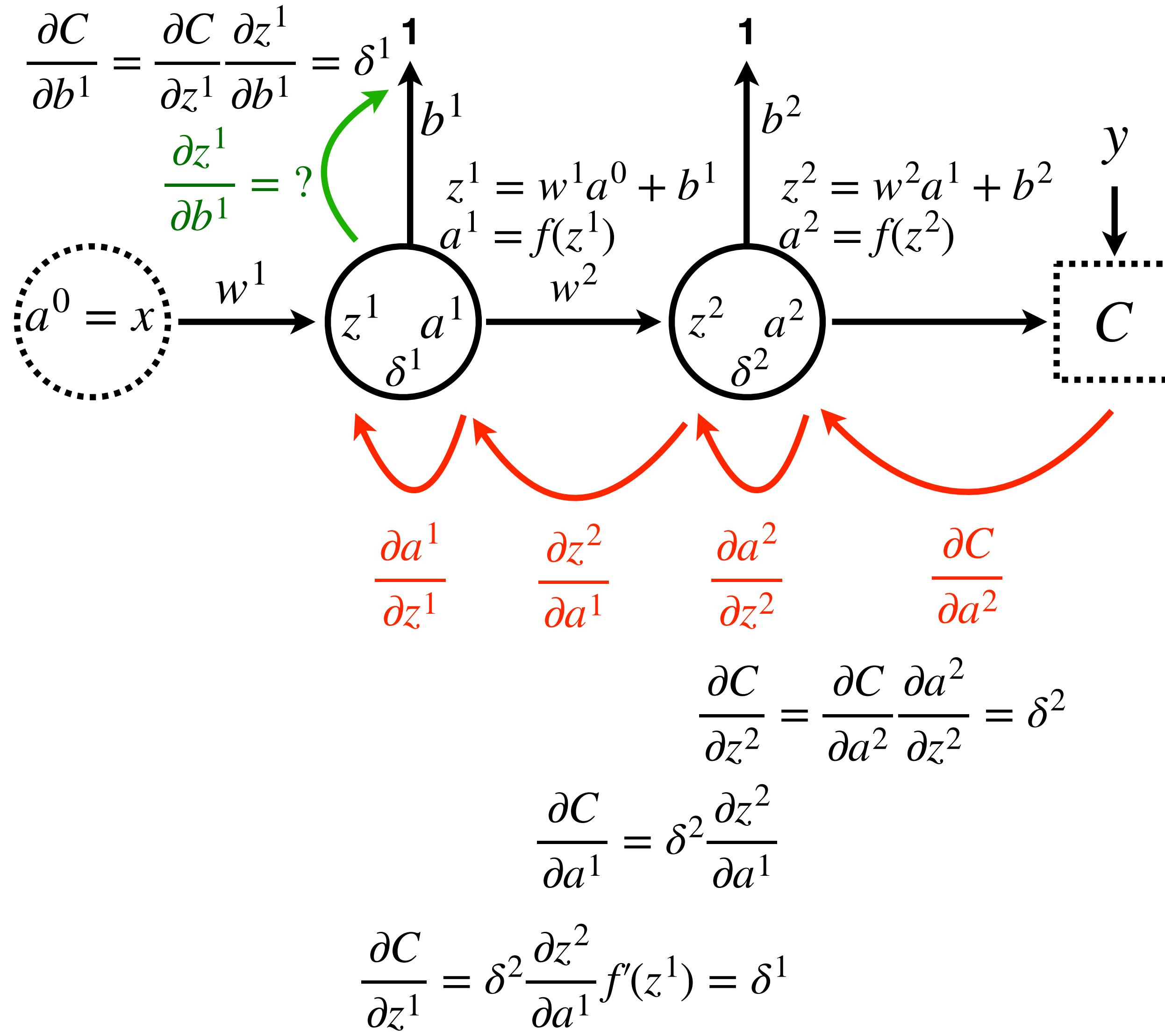


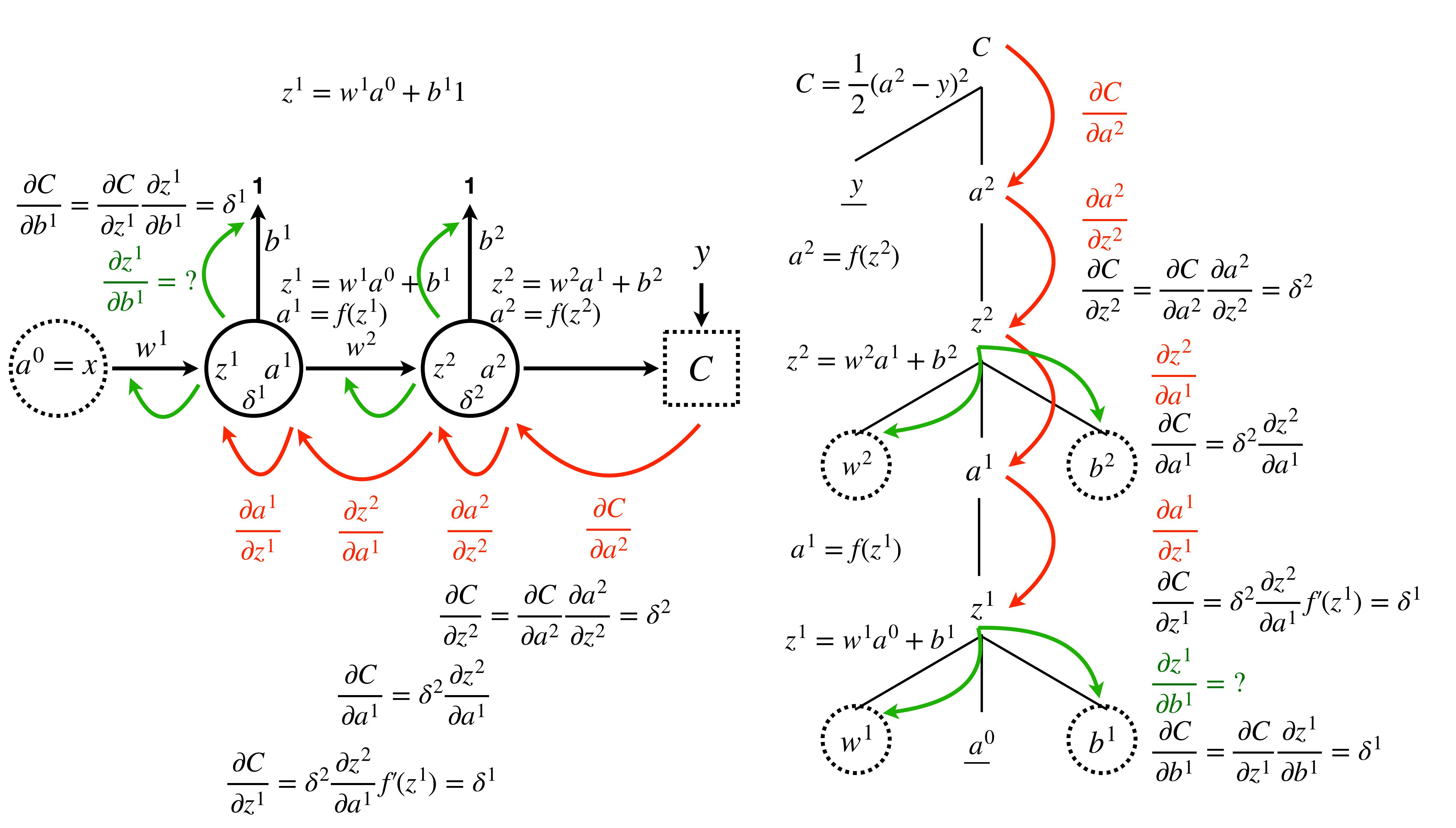


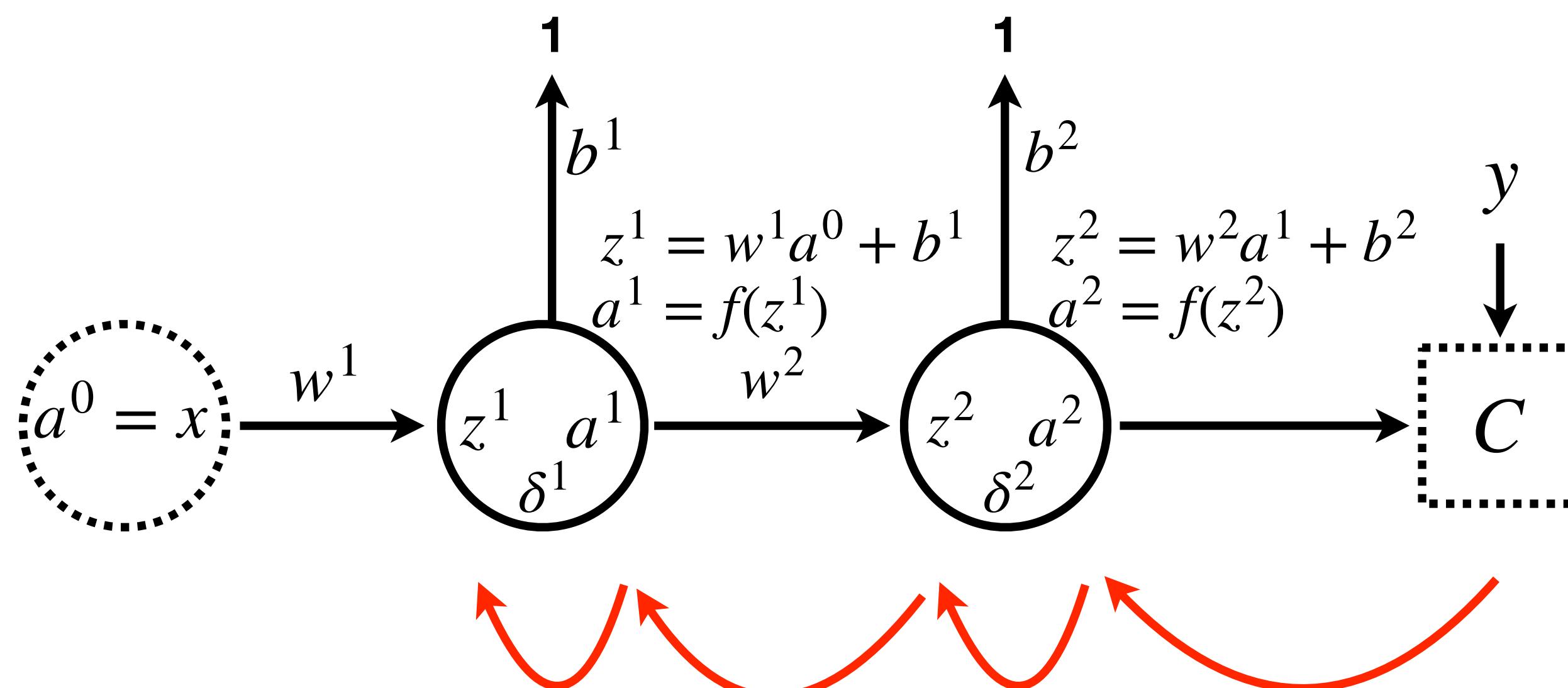








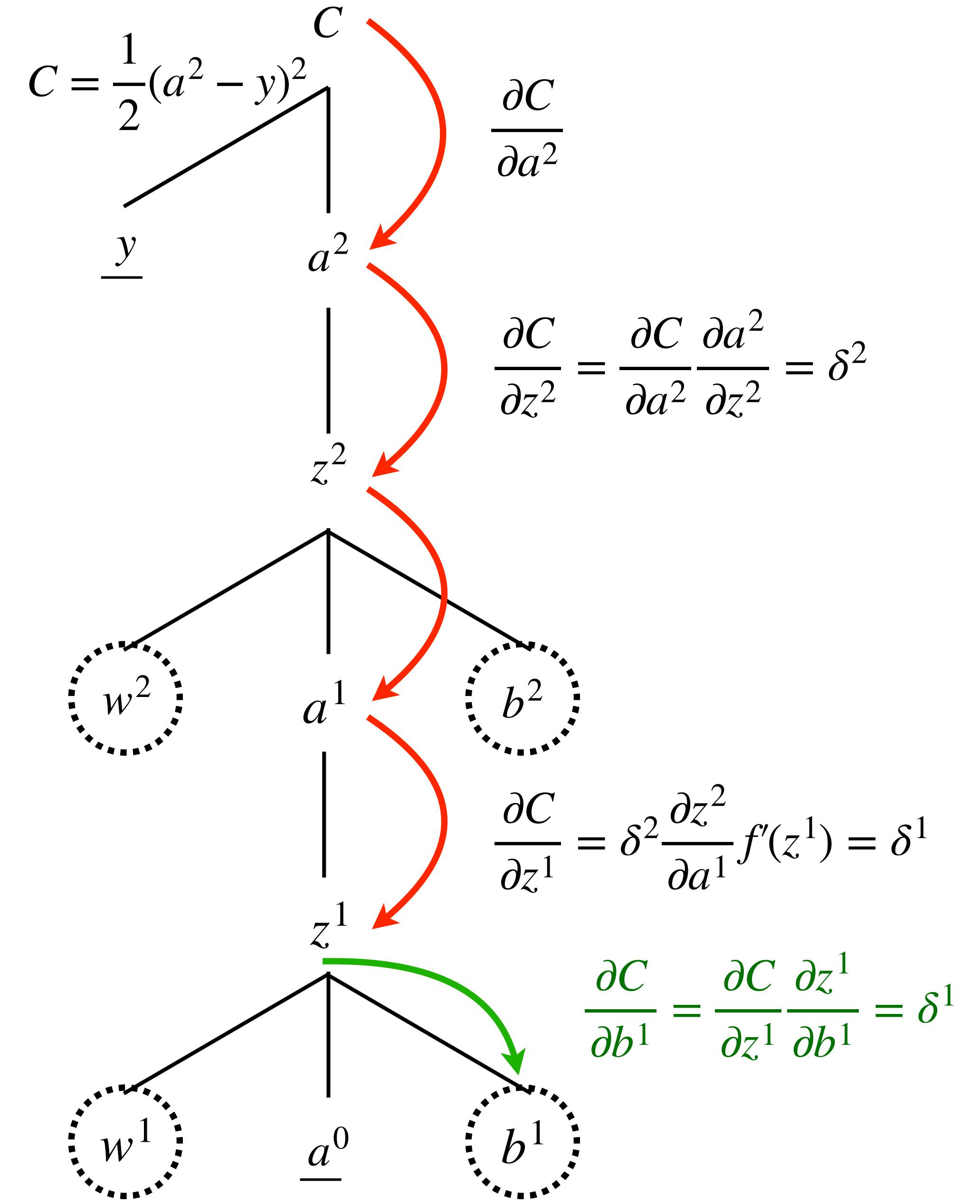
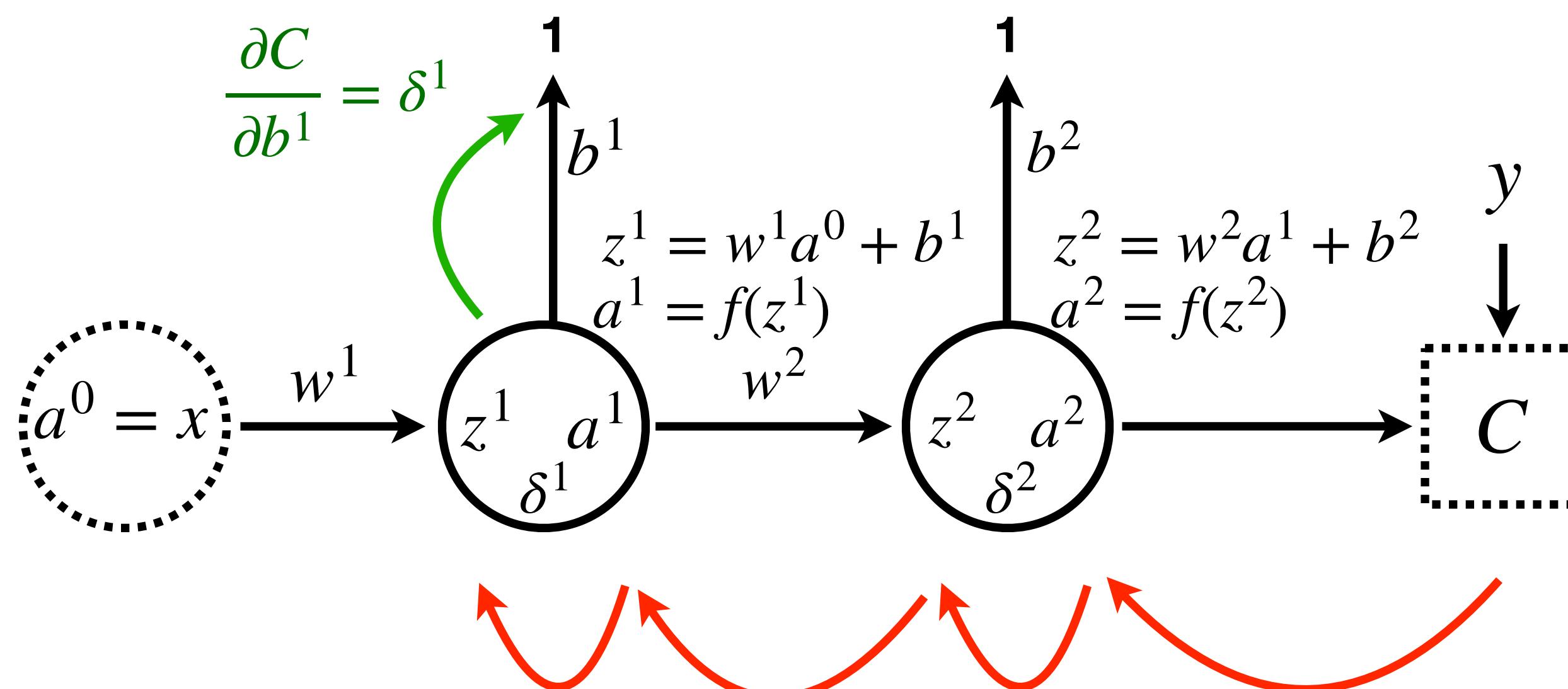


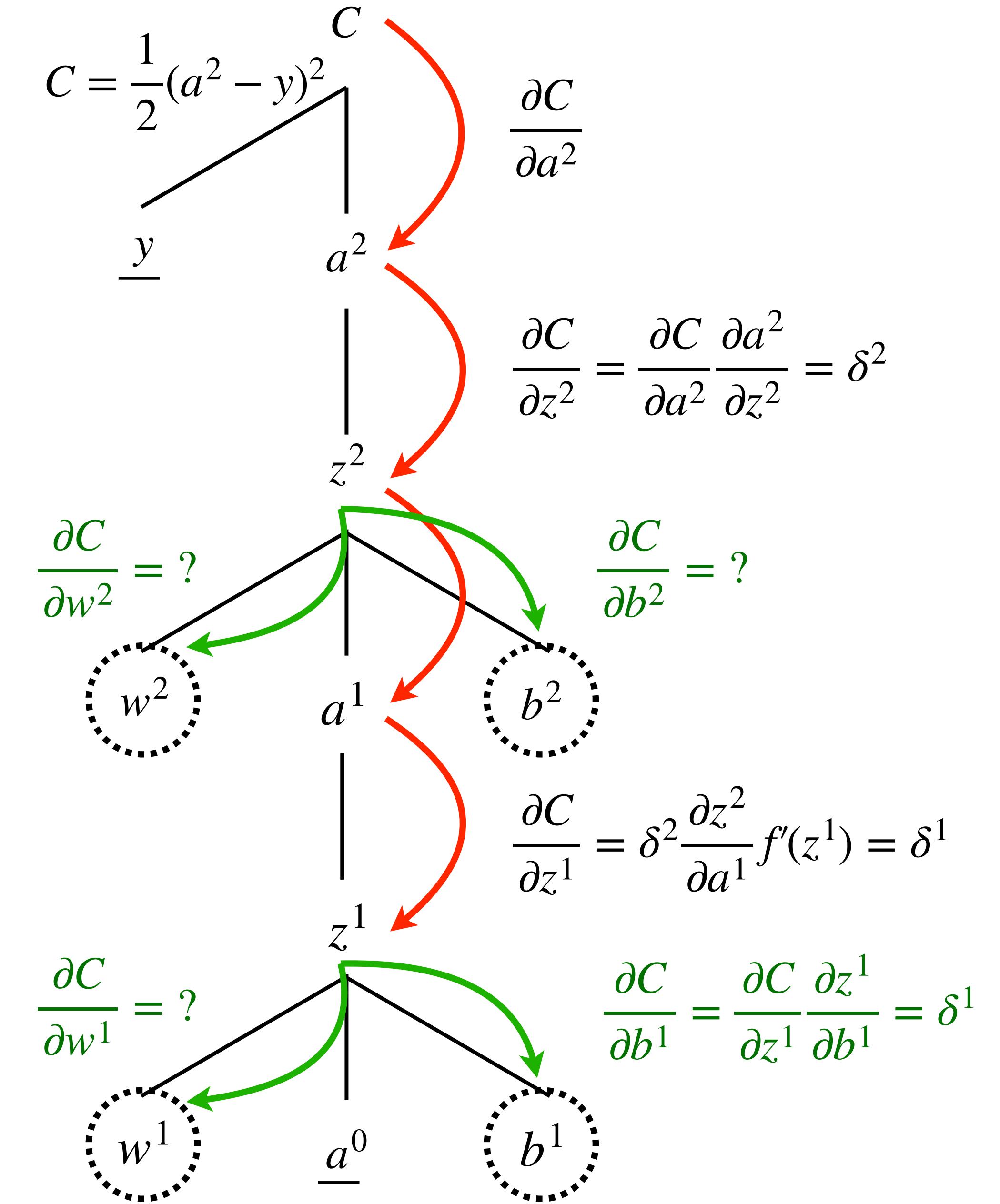
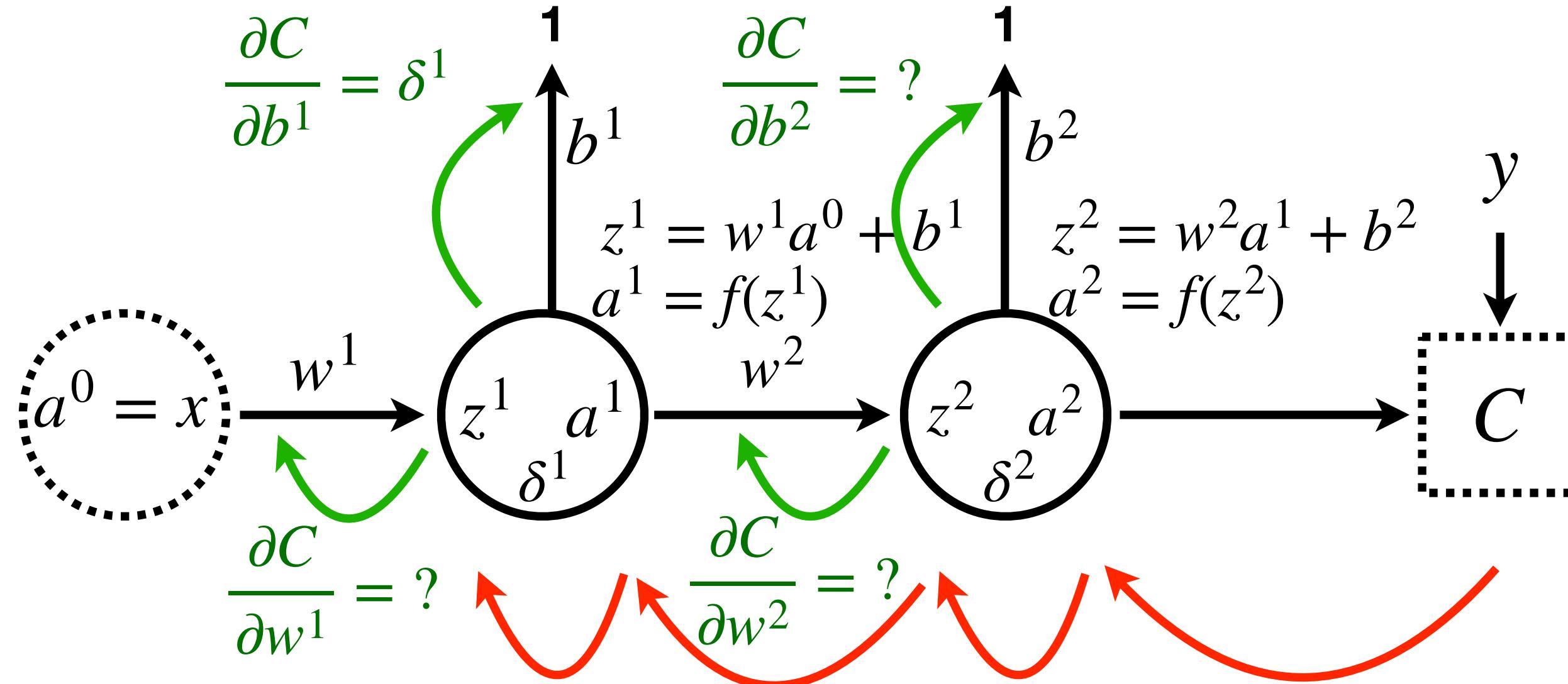


Diagrams illustrating the calculation of partial derivatives of the cost function  $C$ :

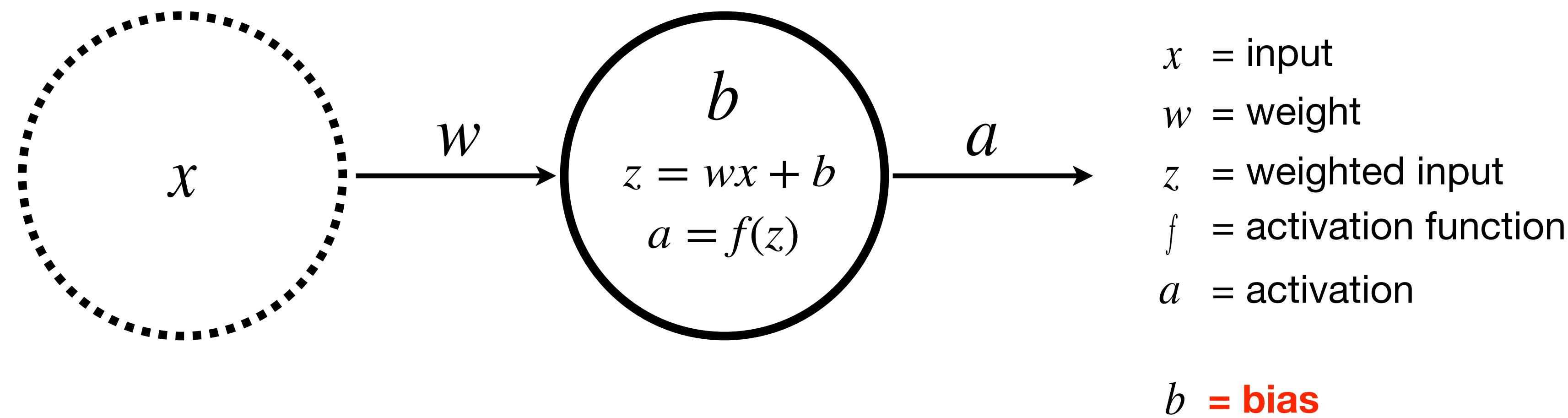
- $C = \frac{1}{2}(a^2 - y)^2$
- $\frac{\partial C}{\partial a^2}$
- $\frac{\partial C}{\partial z^2} = \frac{\partial C}{\partial a^2} \frac{\partial a^2}{\partial z^2} = \delta^2$
- $\frac{\partial C}{\partial z^1} = \delta^2 \frac{\partial z^2}{\partial a^1} f'(z^1) = \delta^1$

The diagrams show the flow of gradients (red arrows) from the cost function  $C$  down through the layers to the input nodes  $w^2, w^1, a^0, b^2, b^1$ .

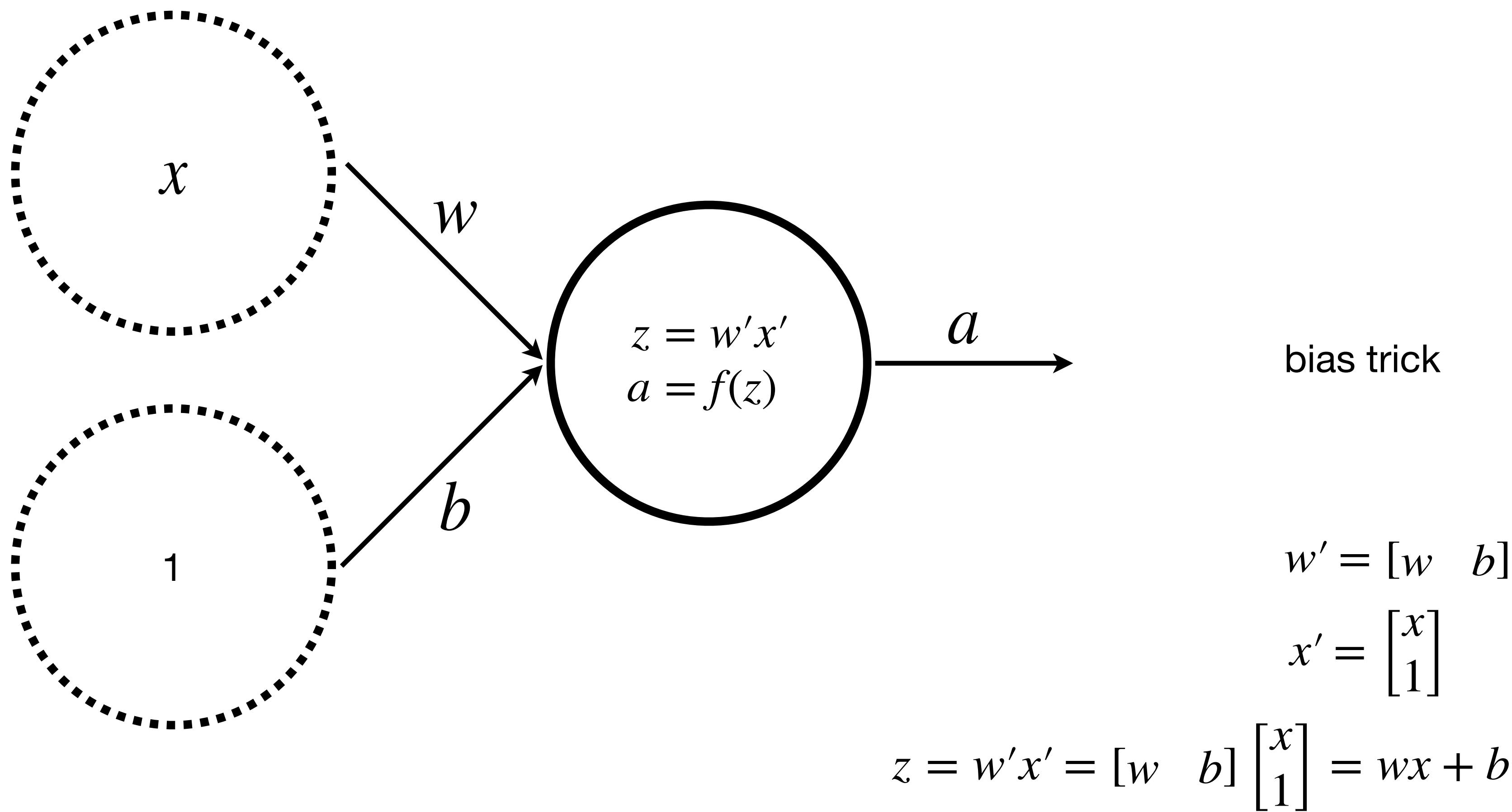




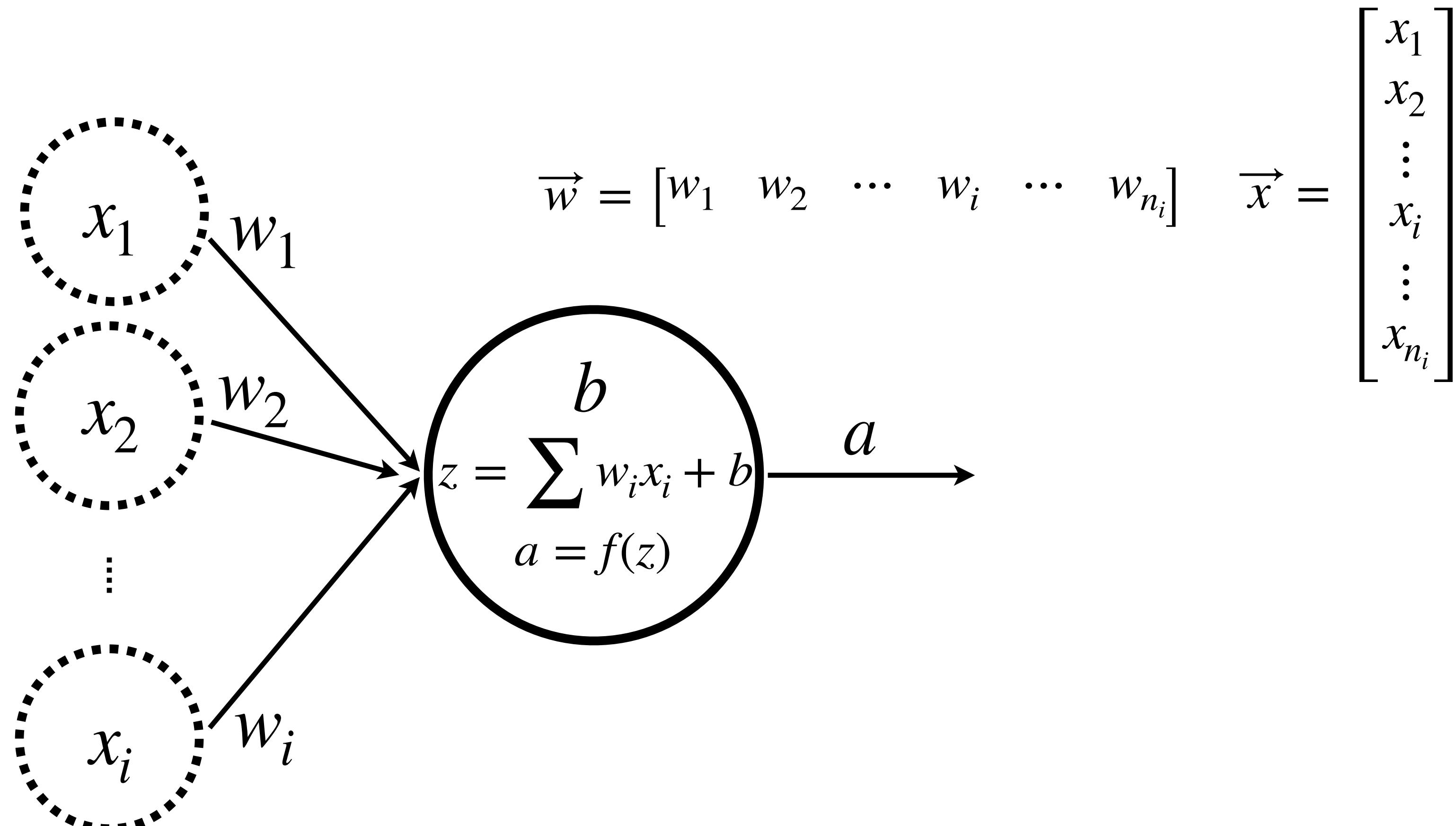
# Single example, single input - single layer, single neuron



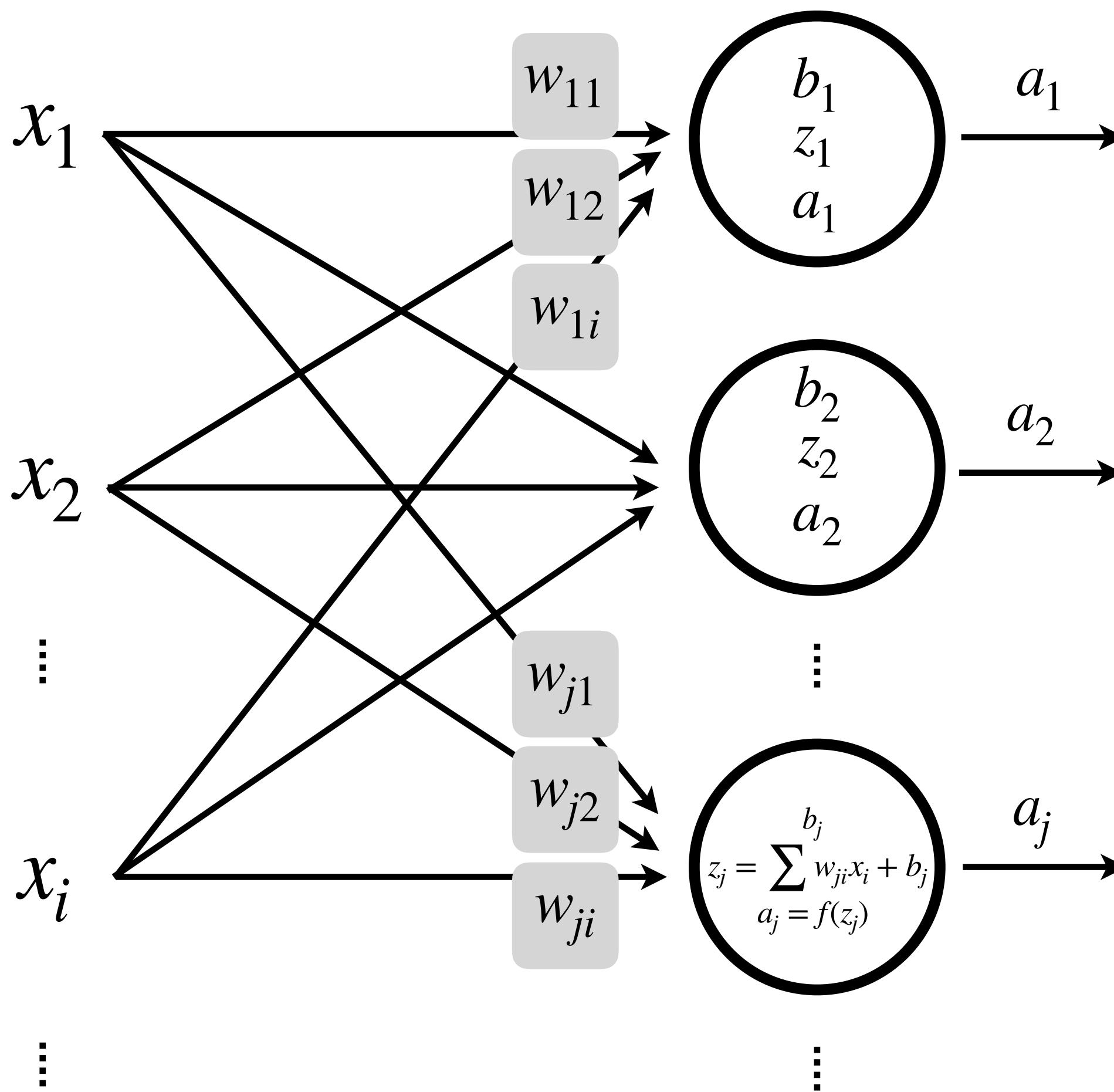
# Single example, single input - single layer, single neuron



# Single example, **multiple** inputs - single layer, single neuron



# Single example, **multiple** inputs - single layer, **multiple** neurons



$$\begin{aligned} z_1 &= \sum_i w_{1i}x_i + b_1 \\ &= w_{11}x_1 + w_{12}x_2 + \dots + w_{1i}x_i + \dots \end{aligned}$$

$$a_1 = f(z_1)$$

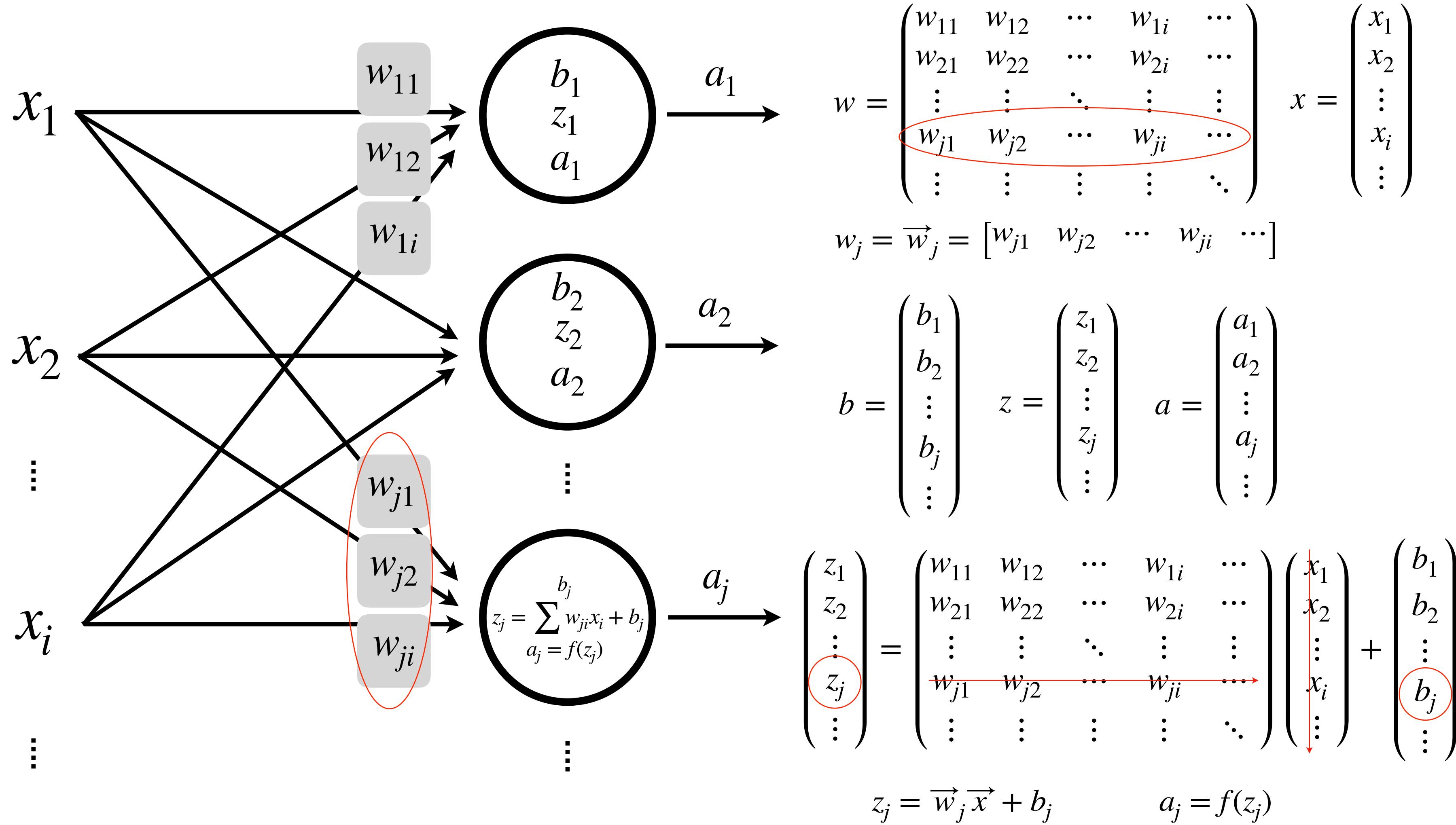
$$\begin{aligned} z_2 &= \sum_i w_{2i}x_i + b_2 \\ &= w_{21}x_1 + w_{22}x_2 + \dots + w_{2i}x_i + \dots \end{aligned}$$

$$a_2 = f(z_2)$$

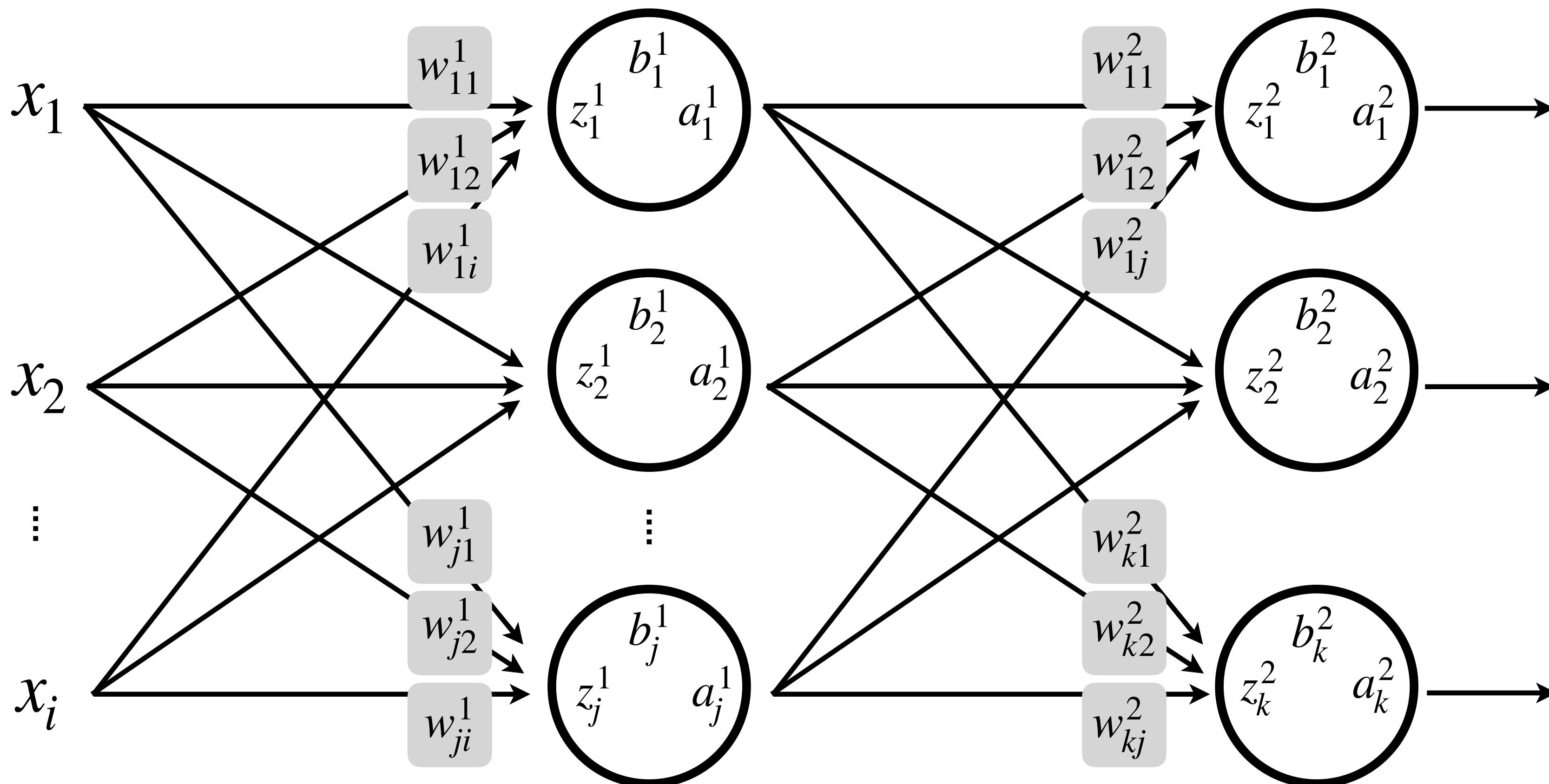
$$\begin{aligned} z_j &= \sum_i w_{ji}x_i + b_j \\ &= w_{j1}x_1 + w_{j2}x_2 + \dots + w_{ji}x_i + \dots \end{aligned}$$

$$a_j = f(z_j)$$

# Single example, **multiple** inputs - single layer, **multiple** neurons



# Single example, **multiple** inputs - **multiple** layers, **multiple** neurons



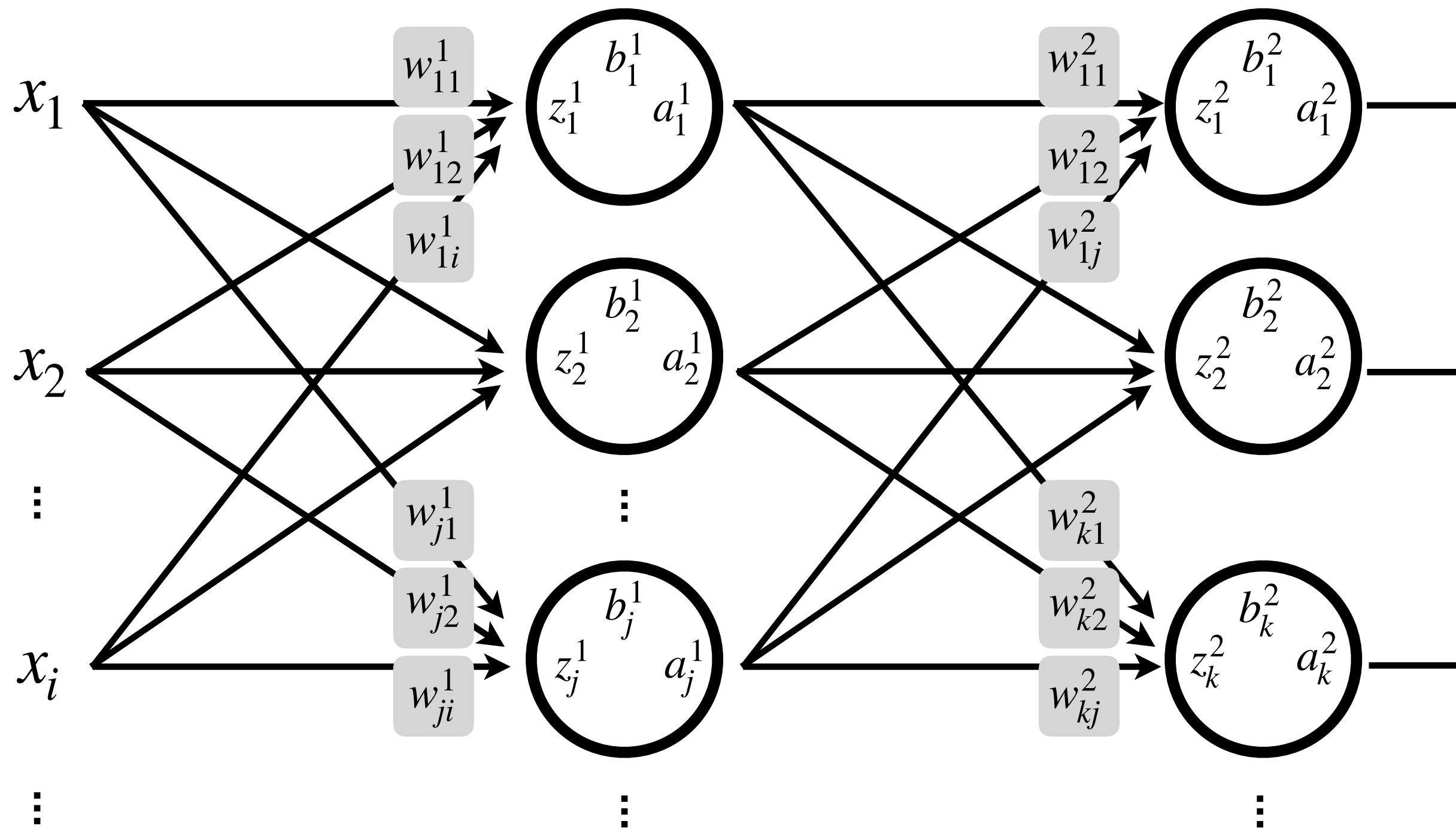
$w_{ji}^l$  is the **weight** to neuron  $j$  in layer  $l$  from neuron  $i$  in layer  $l-1$

$b_j^l$  is the **bias** of neuron  $j$  in layer  $l$

$z_j^l$  is the **weighted input** of neuron  $j$  in layer  $l$

$a_j^l$  is the **activation** of neuron  $j$  in layer  $l$

# Single example, **multiple** inputs - **multiple** layers, **multiple** neurons



$$a^0 \quad w^1 \quad b^1 \ z^1 \ a^1$$

$$w^2 \quad b^2 \ z^2 \ a^2$$

**Vectorization:**

$$x = \vec{x}$$

$$f(x) = f(\vec{x})$$

$$f(x)_j = f(x_j)$$

$$f(x_j) = x_j^2$$

$$f\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = ?$$

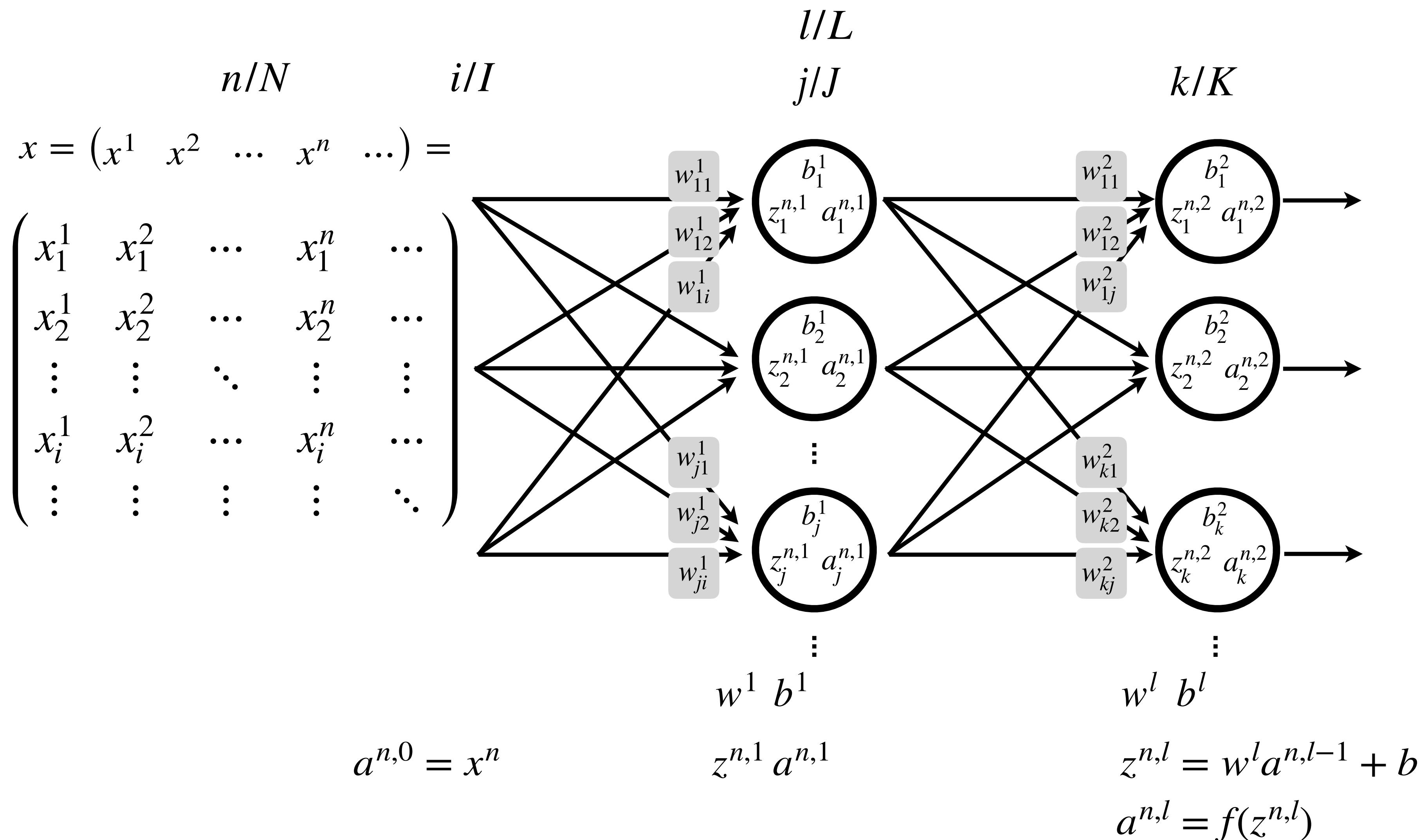
$$w^l$$

$$b^l \ z^l \ a^l$$

$$a^l = f(z^l)$$

$$z^l = w^l a^{l-1} + b^l$$

# Multiple examples, multiple inputs - multiple layers, multiple neurons



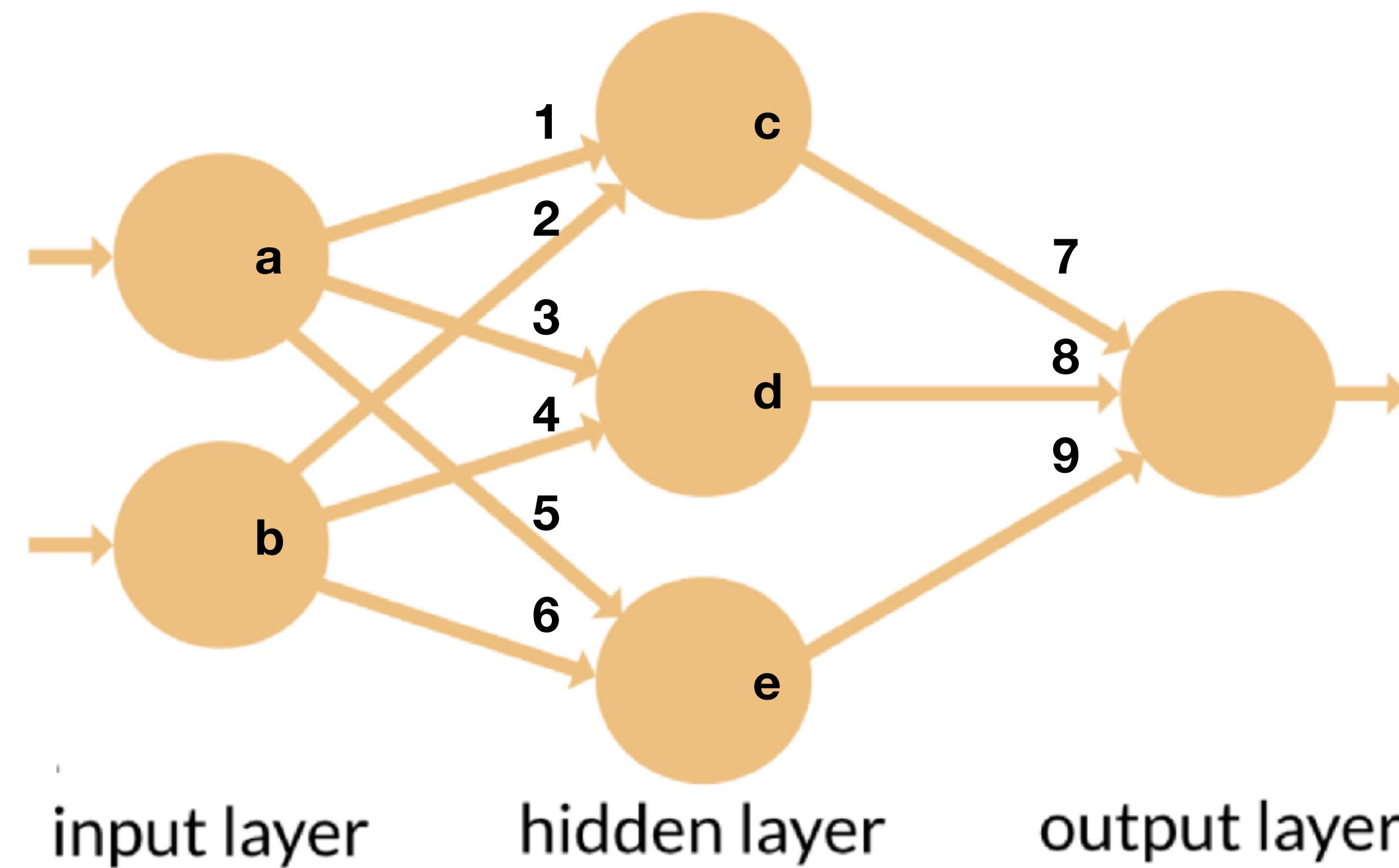
**index  $i$  = input  $i$**   
**index  $j$  = hidden neuron  $j$**   
**( $j_1, j_2, \dots$ )**  
**index  $k$  = output neuron  $k$**   
**index  $n$  = example  $n$**   
**index  $l$  = layer  $l$** 
  
  
 **$I$  = # inputs**  
 **$J$  = # neurons in hidden layer**  
 **$K$  = # neurons in output layer**  
 **$N$  = # examples (batch size)**  
 **$L$  = # layers (input = layer 0)**

# Forward pass

$$w^1 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \quad w^2 = (7 \quad 8 \quad 9)$$

**Given:** two weight matrixes. **Provide:** figure with weights

# Forward pass



**Given:** figure with weights **Provide:** two weight matrixes

**Given:**

$$z^l = w^l a^{l-1} + b^l$$

$$a^l = f(z^l)$$

$$w^1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$w^2 = [9 \quad 10 \quad 11 \quad 12]$$

$$w^3 = \begin{bmatrix} 13 \\ 14 \end{bmatrix}$$

**Find: NN Architecture**

**Activation function:**  $f(z) = \frac{1}{1 + e^{-z}}$

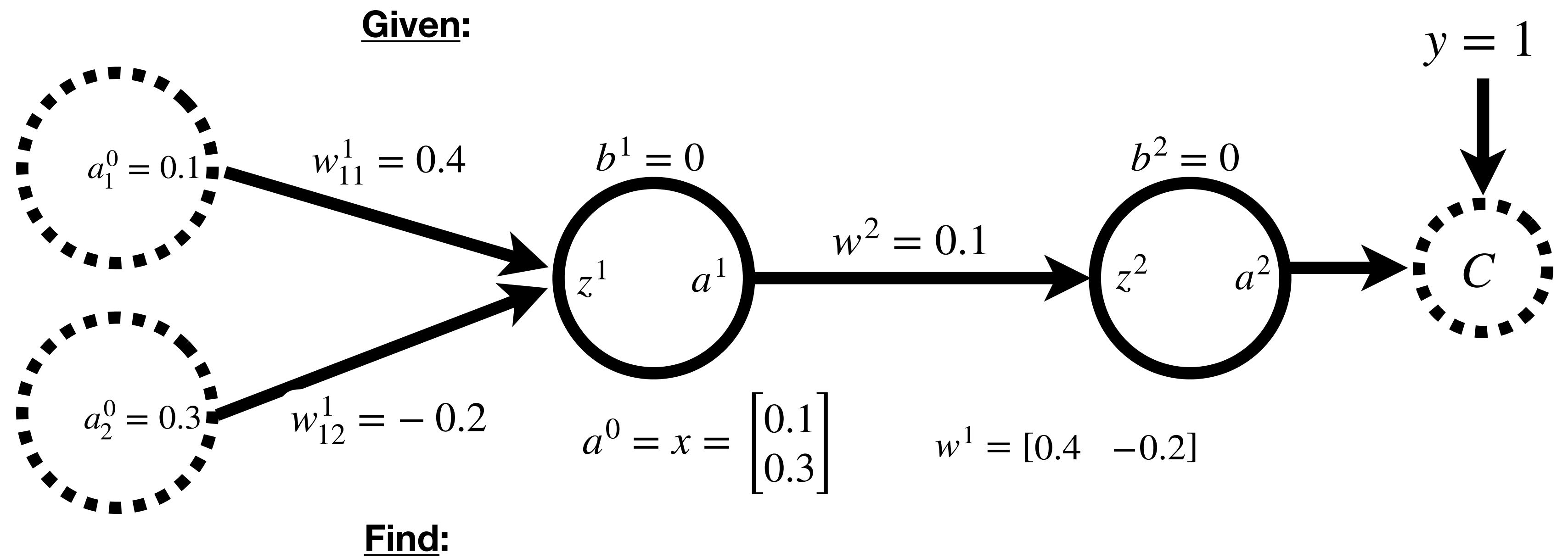
$$\frac{df(z)}{dz} = f(z)(1 - f(z))$$

**Cost function:**  $C = C(a^2) = \frac{1}{2}(y - a^2)^2$

$$\frac{\partial C}{\partial a^2} = a^2 - y$$

**Learning rate:**  $\eta = 0.01$

**Forward pass:**  $z^1 =$   
 $a^1 =$



$$z^2 =$$

$$a^2 =$$

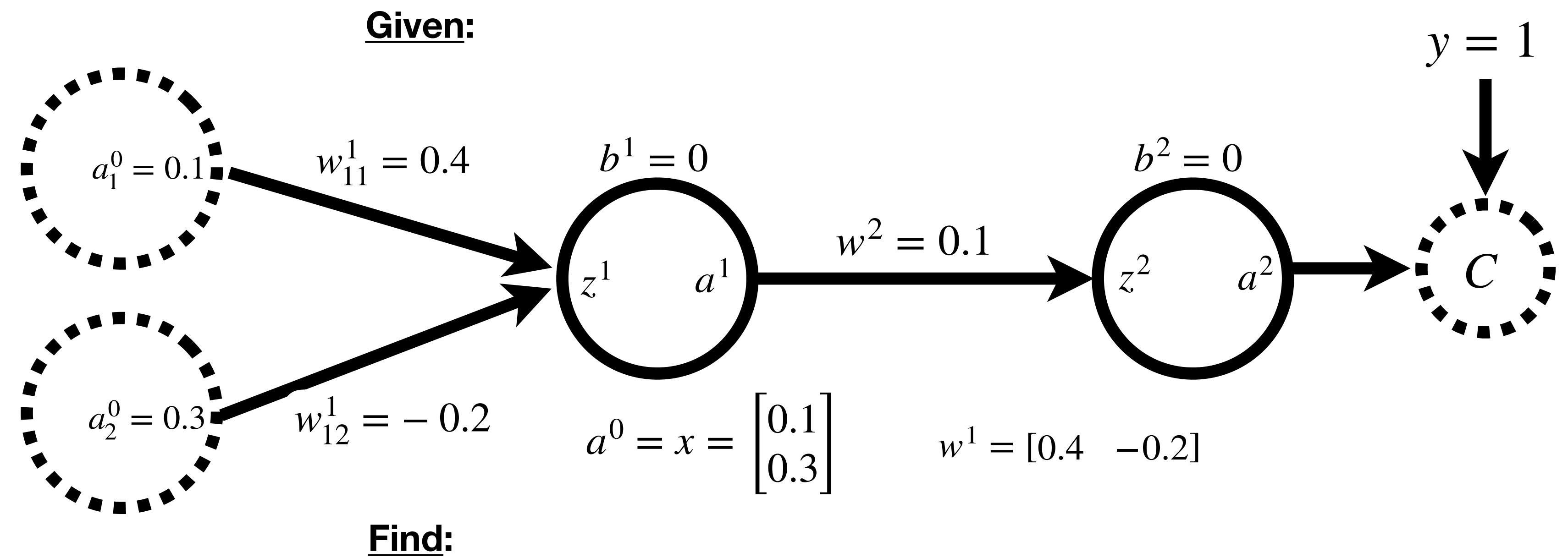
**Activation function:**  $f(z) = \frac{1}{1 + e^{-z}}$

$$\frac{df(z)}{dz} = f(z)(1 - f(z))$$

**Cost function:**  $C = C(a^2) = \frac{1}{2}(y - a^2)^2$

$$\frac{\partial C}{\partial a^2} = a^2 - y$$

**Learning rate:**  $\eta = 0.01$



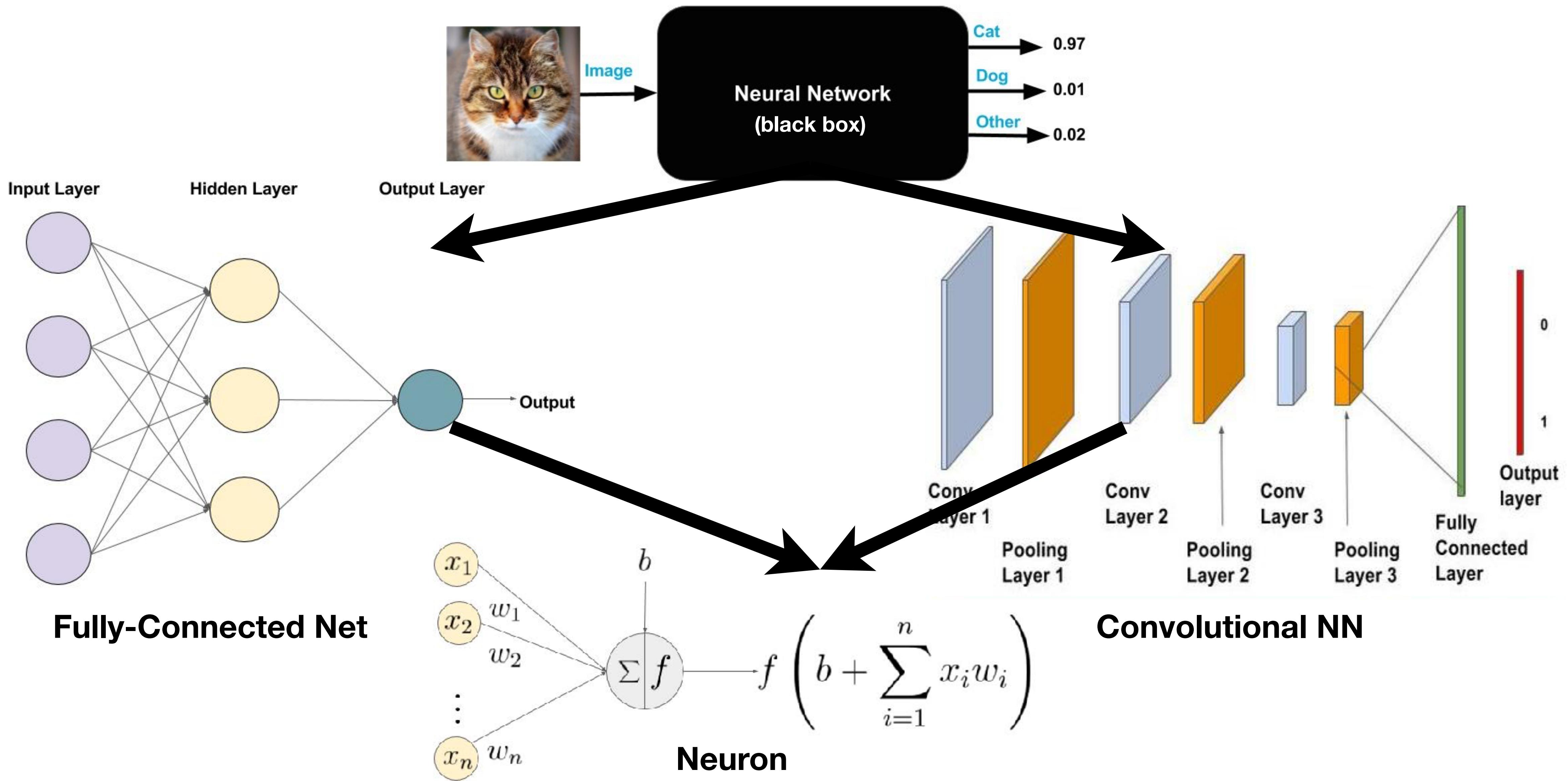
**Forward pass:**  $z^1 = w^1 a^0 + b^1 = [0.4 \quad -0.2] \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix} + 0 = 0.04 - 0.06 = -0.02$

 $a^1 = f(z^1) = f(-0.02) = 0.495$

$$z^2 = w^2 a^1 + b^2 = 0.1 \cdot 0.495 + 0 = 0.0495$$

$$a^2 = f(z^2) = f(0.0495) = 0.5124$$

# Big picture / overview: Predictions

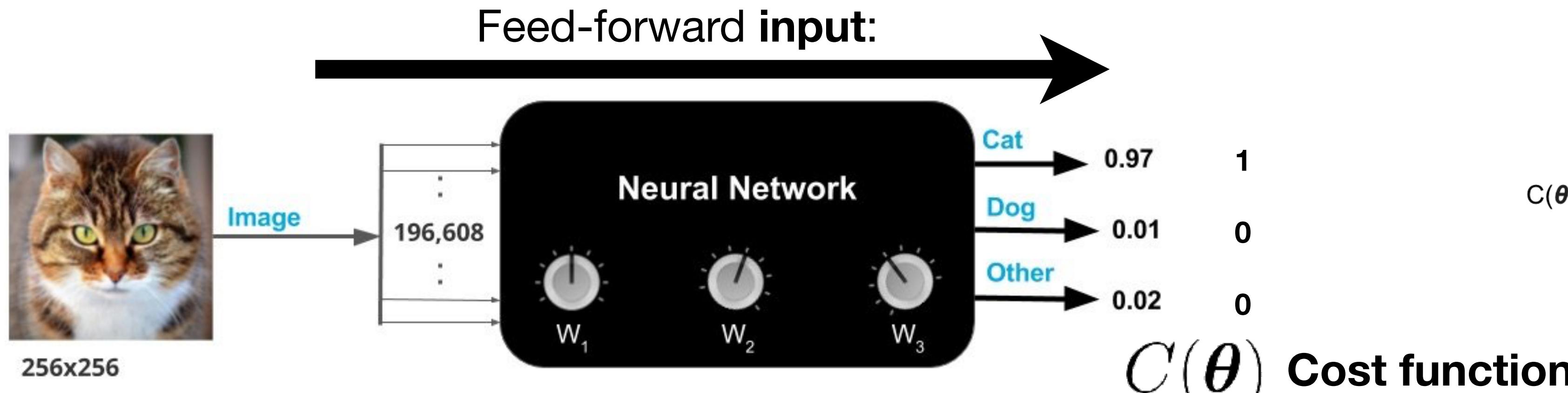


# Repetition: Learning

**Given data:**  $(x^1, y^1), (x^2, y^2), \dots (x^N, y^N)$

$$f: \mathbb{R}^I \rightarrow \mathbb{R}^J$$

$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(1)}(x)))$  **Find mapping:**  $f(x; \theta) = a^L = \hat{y} =? y$  **Desired output:**



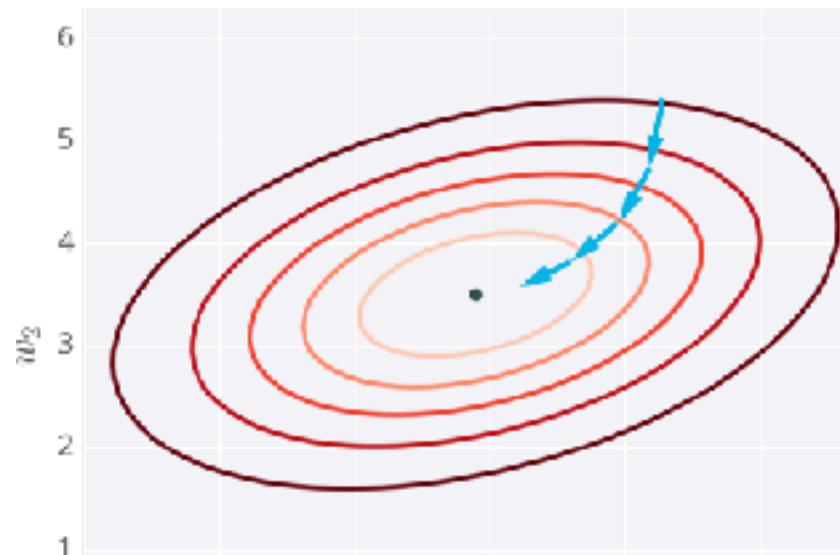
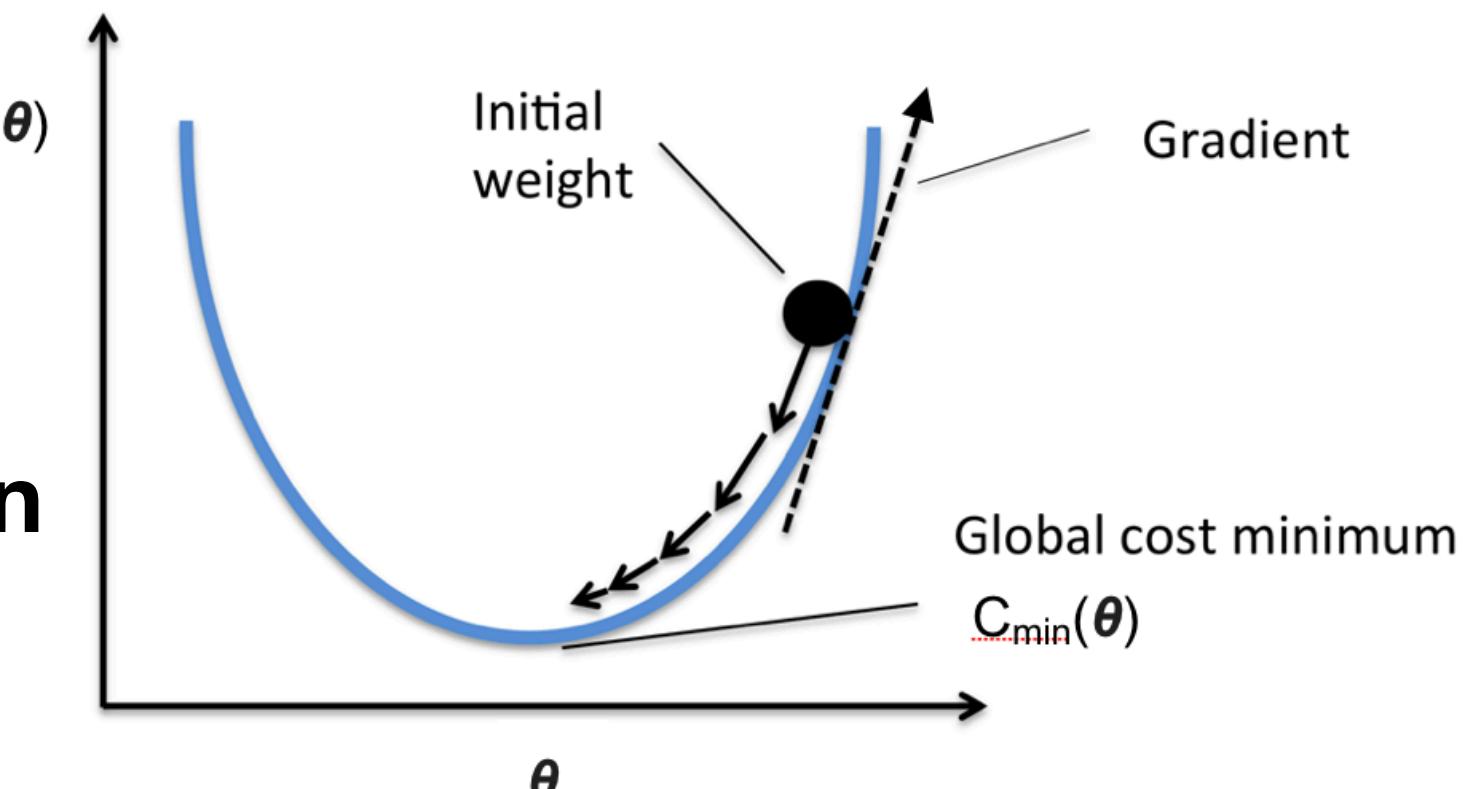
$$\frac{\partial C}{\partial w_{kj}^L} = \frac{\partial C}{\partial z^L} \frac{\partial z^L}{\partial w_{kj}^L} = \delta^L \frac{\partial z^L}{\partial w_{kj}^L}$$

$$\nabla C = \frac{\partial C}{\partial \theta_1}, \frac{\partial C}{\partial \theta_2}, \dots$$

**Gradients = partial derivatives**

$$\theta = \theta - \alpha \nabla C$$

**Update rule**



$$\delta_k^L = \frac{\partial C}{\partial z_k^L} = \frac{\partial C}{\partial a_k^L} \frac{\partial a_k^L}{\partial z_k^L} = \frac{\partial C}{\partial a_k^L} f'(z_k^L) \quad a^L = f(z^L)$$

$$\frac{\partial z^L}{\partial w_{kj}^L} = a_j^{L-1}$$

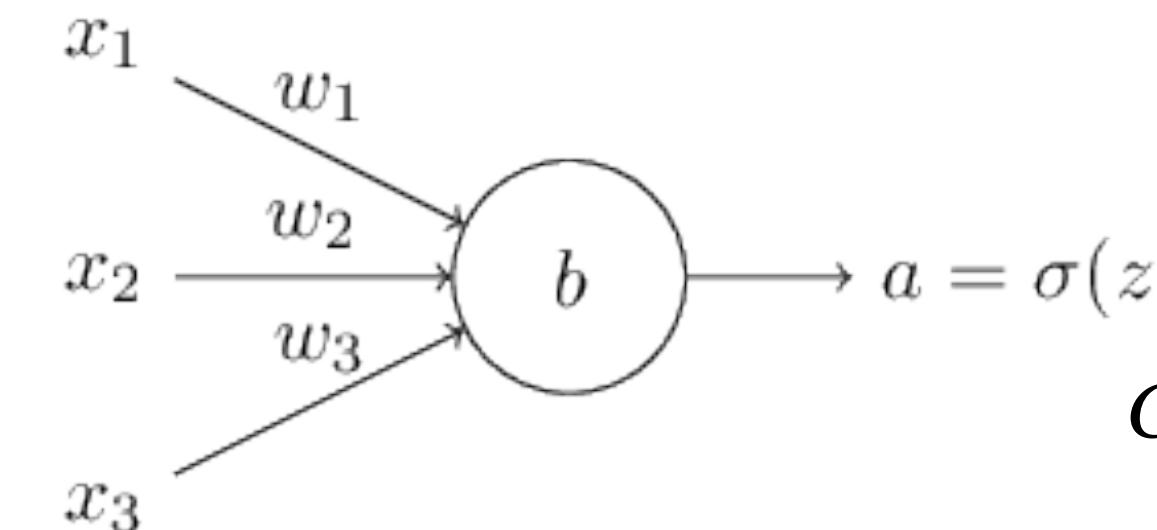
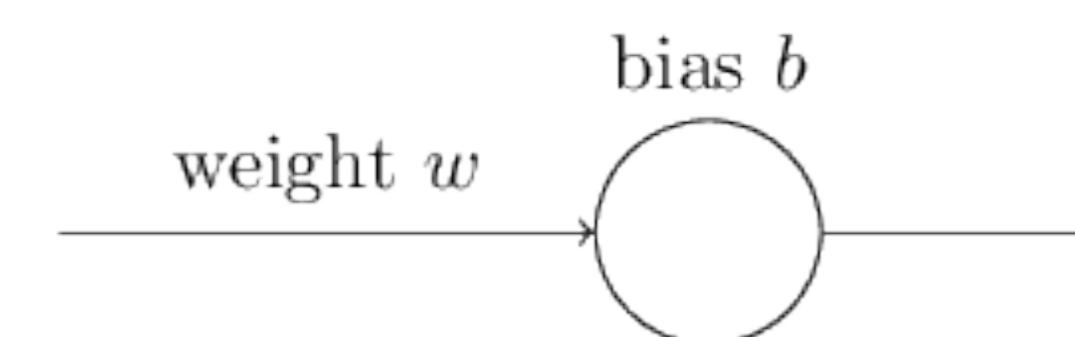
$$z^L = w^L a^{L-1} + b^L$$

# Cross-entropy (cost function)

Cost functions:

MSE

Cross-entropy  
Log-likelihood



$$C = \frac{(y - a)^2}{2} \quad x = 1, y = 0$$

$$\frac{\partial C}{\partial w} = (a - y)\sigma'(z)x = a\sigma'(z)$$

$$\frac{\partial C}{\partial b} = (a - y)\sigma'(z) = a\sigma'(z)$$

$$C = -\frac{1}{N} \sum [y \ln a + (1 - y) \ln(1 - a)]$$

$$\frac{\partial C}{\partial w_j} = \frac{1}{N} \sum_n x_j (\sigma(z) - y)$$

$$\frac{\partial C}{\partial b} = \frac{1}{N} \sum_n (\sigma(z) - y)$$

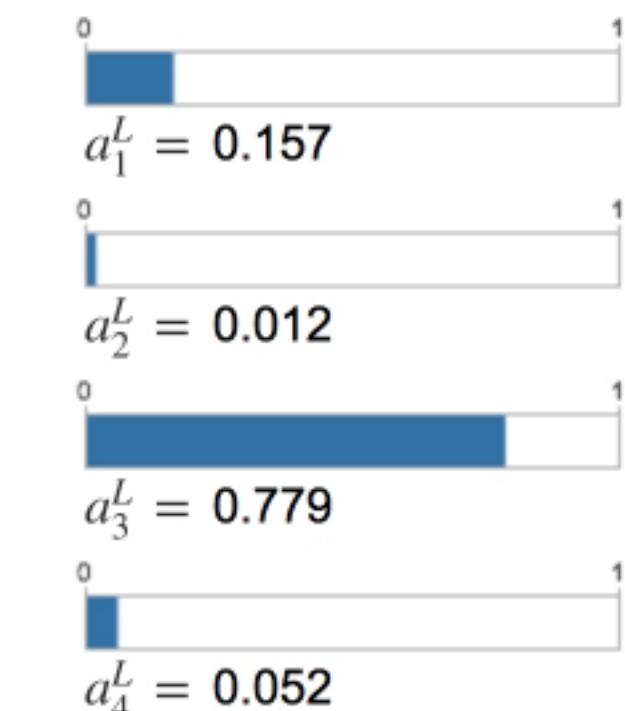
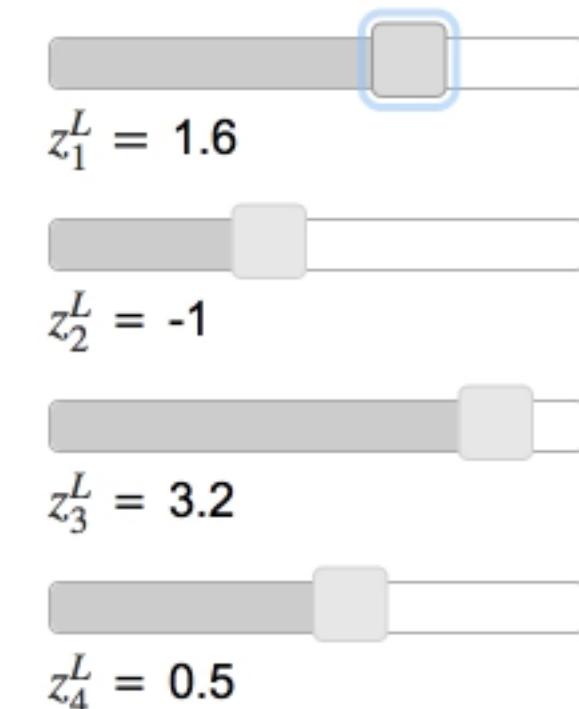
$$C = -\frac{1}{N} \sum_n \sum_j \left[ y_j \ln a_j^L + (1 - y_j) \ln(1 - a_j^L) \right]$$

# Softmax (activation function) & Log-likelihood (cost function)

- Softmax **output** layer (weighted inputs same as before)
- Example: increase one input, inc. activation but decrees the others, sum over all activations remains 1, **probability distribution**, e.g. MNIST, i.e. simple interpretation of the output activations.
- Log-likelihood cost: Model confident  $\rightarrow$  high probability (close to 1)  $\rightarrow$  cost small (other way)
- Classification problems (e.g MNIST)

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}}$$

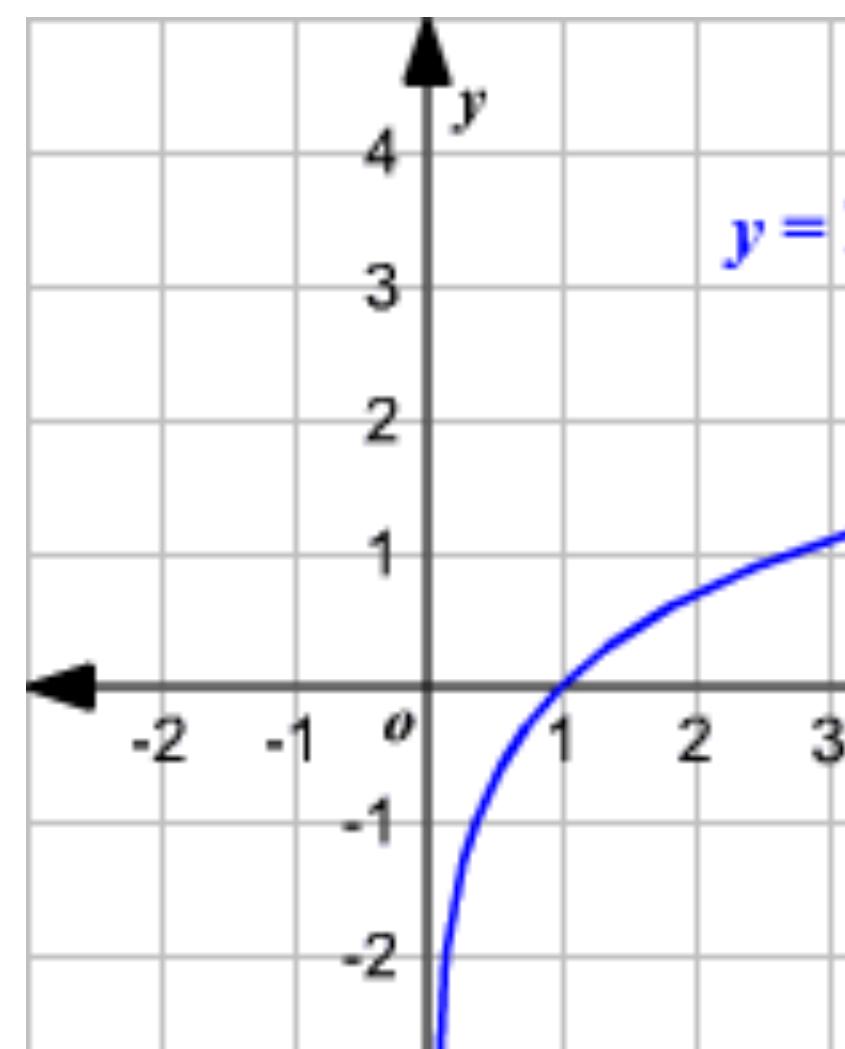
$$\sum_j a_j^L = \frac{\sum_j e^{z_j^L}}{\sum_k e^{z_k^L}} = 1$$



$$C \equiv -\ln a_y^L$$

$$\frac{\partial C}{\partial w_{kj}^L} = a_j^{L-1}(a_k^L - y_k)$$

$$\frac{\partial C}{\partial b_k^L} = a_k^L - y_k$$



# Back-propagation: Basics

## Problem:

Given a function:  $f(x)$  (loss function, weights & biases)

Compute the derivative:  $\nabla f(x)$

**Derivative:** rate of change of  $f$  wrt  $x$ , small region around a point

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

Operator:  $\frac{d}{dx}$  Applied to:  $f(x, y)$

Partial derivatives:  $\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}$  (straight line, slope)

# Back-propagation: Gradients

**Eks:**

$$f(x, y) = xy \rightarrow \frac{\partial f}{\partial x} = y \quad \frac{\partial f}{\partial y} = x$$

$$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}] = [y, x]$$

$$x = 4, y = -3 \quad f(x, y) = -12$$

$$\frac{\partial f}{\partial x} = -3 \quad \frac{\partial f}{\partial y} = 4,$$

**Sensitivity:**

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

$$(f(x + h) = f(x) + h \frac{df(x)}{dx})$$

Increase x with small h -> reduce f with 3h

Increase y with small h -> increase f with 4h

## Local process (each gate):

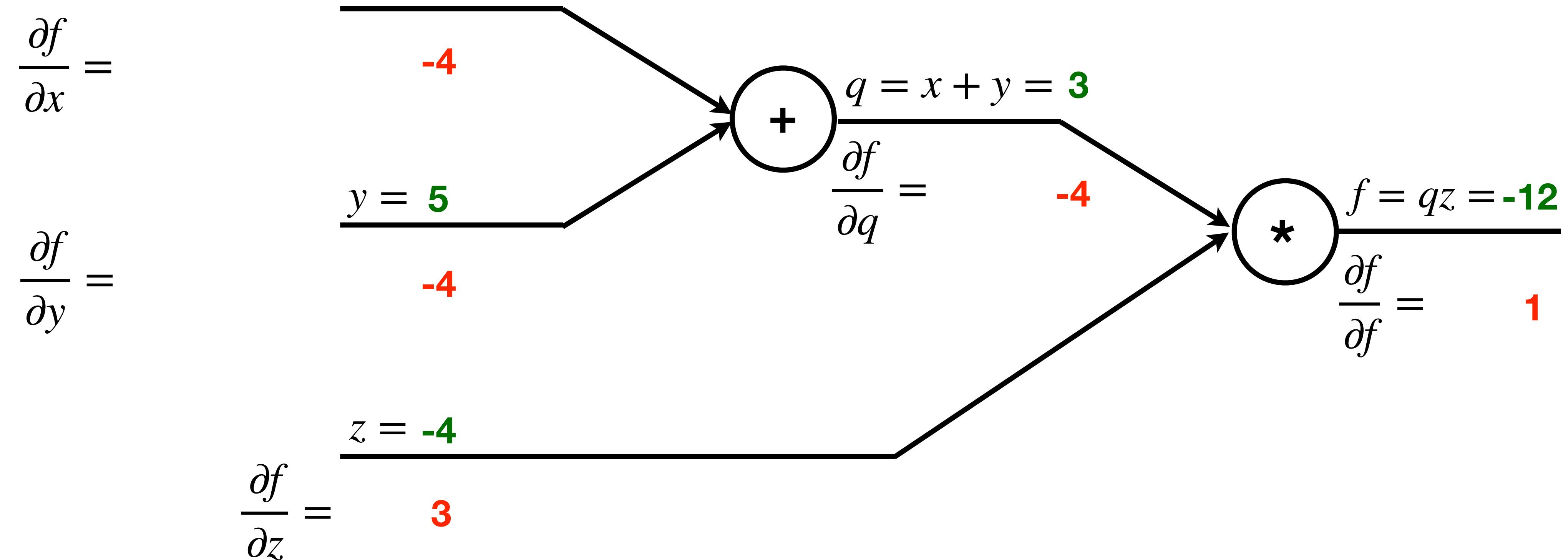
- **Forward pass:**
  - Output value
  - Local gradient
- **Backward pass:**
  - Gradient (final output)
  - Chain rule

# Back-propagation

$$f(x, y, z) = f = (x + y)z = qz \quad \frac{\partial f}{\partial q} = z \quad \frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

$$q(x, y) = q = x + y \quad \frac{\partial f}{\partial z} = q \quad \frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

$$x = -2 \quad y = 5 \quad z = -4$$



# Back-propagation

$$\Delta q = 1$$

$$q : 3 \rightarrow 4$$

$$f = qz = 4 \cdot -4 = -16$$

$$f : -12 \rightarrow -16$$

$$\Delta f = -4$$

$$\Delta z = 1$$

$$z : -4 \rightarrow -3$$

$$f = (x+y)z = (-2+5) \cdot -3 = 3 \cdot -3 = -9$$

$$f : -12 \rightarrow -9$$

$$\Delta f = 3$$

$$\Delta y = 1$$

$$y : 5 \rightarrow 6$$

$$f = (x+y)z = (-2+6) \cdot -4 = 4 \cdot -4 = -16$$

$$f : -12 \rightarrow -16$$

$$\Delta f = -4$$

$$\Delta x = 1$$

$$x : -2 \rightarrow -1$$

$$f = (x+y)z = (-1+5) \cdot -4 = 4 \cdot -4 = -16$$

$$f : -12 \rightarrow -16$$

$$\Delta f = -4$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = z \cdot 1 = \textcolor{red}{-4}$$

$$x = \textcolor{green}{-2}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y} = z \cdot 1 = \textcolor{red}{-4}$$

$$y = \textcolor{green}{5}$$

$$\frac{\partial f}{\partial z} = q = \textcolor{red}{3}$$

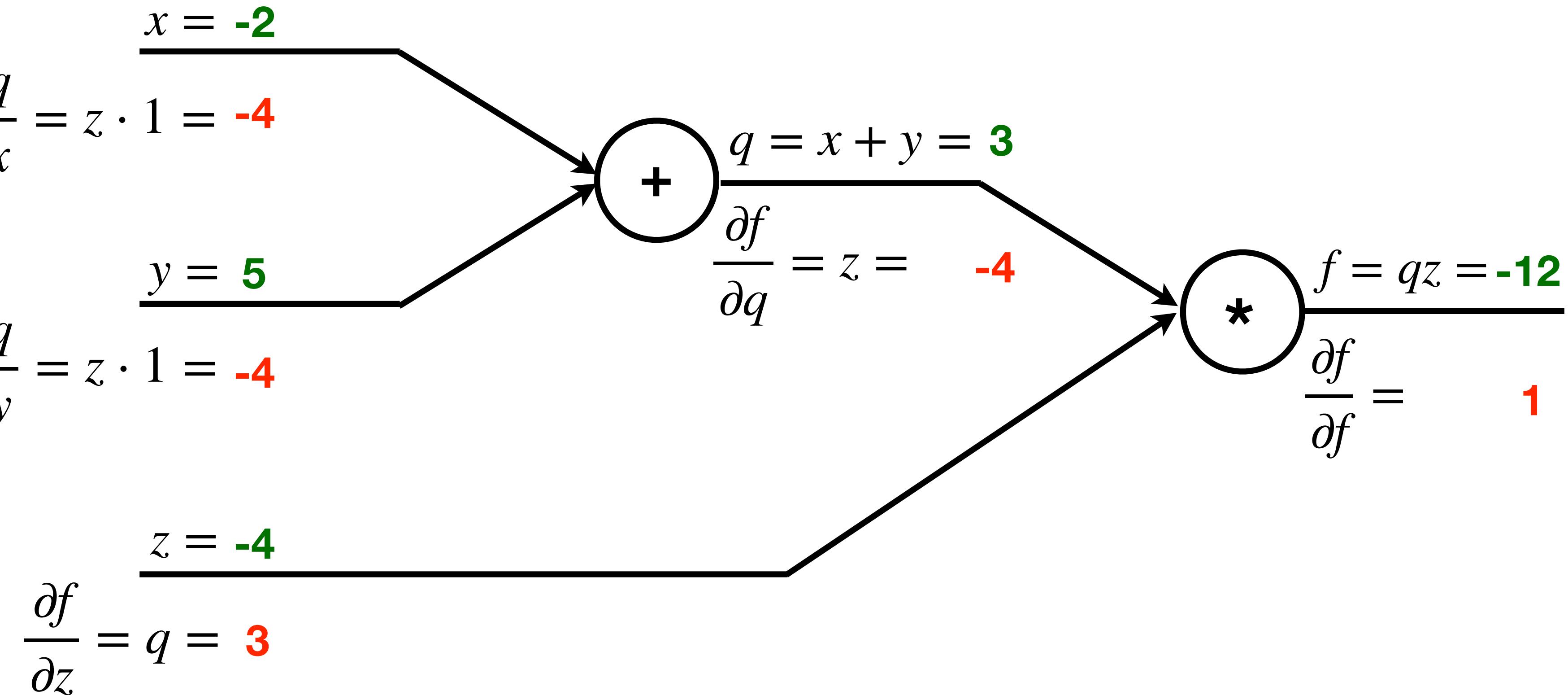
$$z = \textcolor{green}{-4}$$

$$q = x + y = \textcolor{green}{3}$$

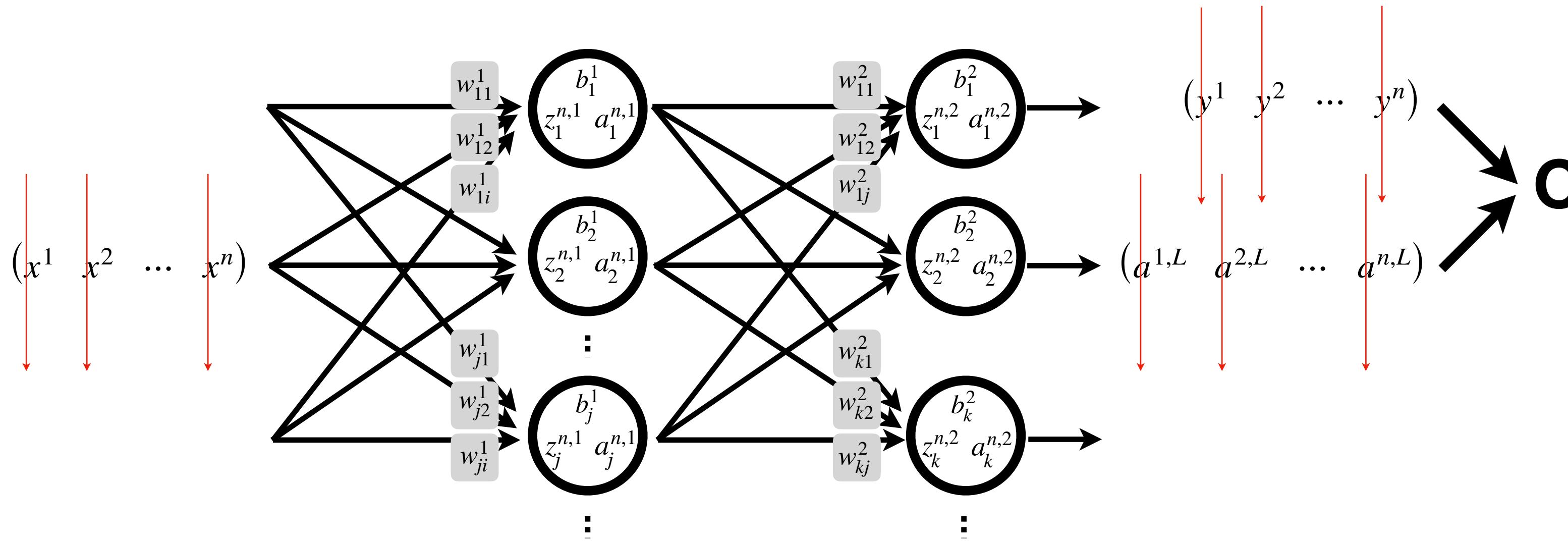
$$\frac{\partial f}{\partial q} = z = \textcolor{red}{-4}$$

$$f = qz = \textcolor{green}{-12}$$

$$\frac{\partial f}{\partial f} = \textcolor{red}{1}$$



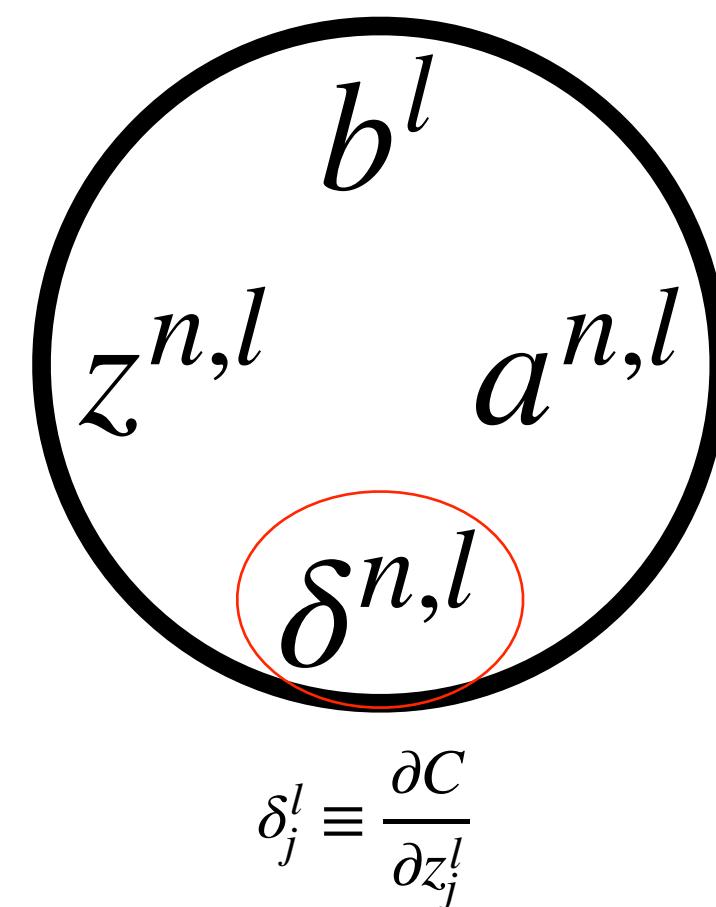
# Gradient descent & Backpropagation



**Gradient descent:**

$$\frac{\partial C}{\partial w_{ji}^l}$$

$$\frac{\partial C}{\partial b_j^l}$$



$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}$$

**Error:**

**For each training example n:**  $x^n$

**Input activation:**  $a^{n,0} = x^n$

**Feed-forward:**  $z^{n,l} = w^l a^{n,l-1} + b^l$

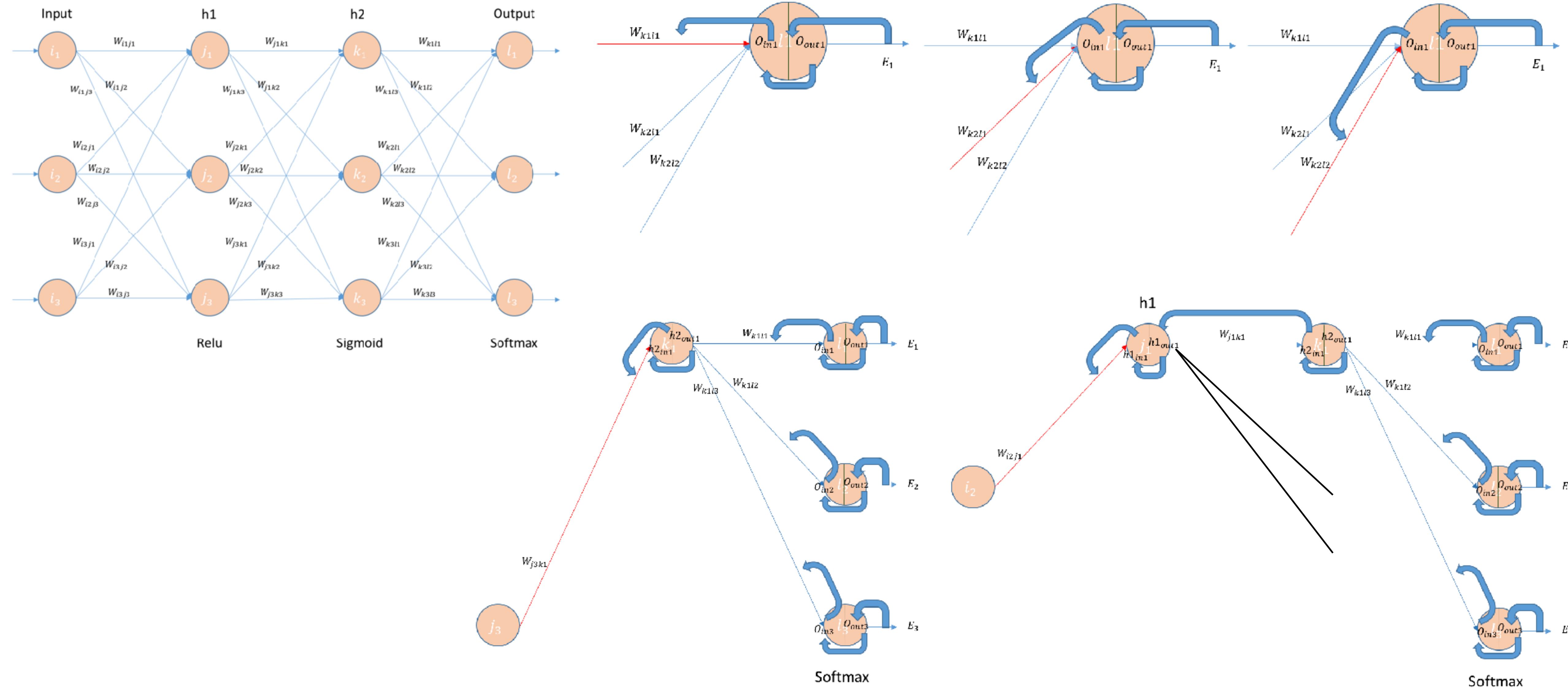
$$a^{n,l} = f(z^{n,l}) \quad l = 1, 2, \dots, L$$

**Output error:**  $\delta^{n,L} = \nabla_a C_n \odot f'(z^{n,L})$

**Back-propagate:**  $\delta^{n,l} = ((w^{l+1})^T \delta^{n,l+1} \odot f'(z^{n,l}))$

$$l = L - 1, L - 2, \dots, 1$$

# Backward pass



# Repetition: Learning (2)

$$f: \mathbb{R}^I \rightarrow \mathbb{R}^J$$

$$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(1)}(x)))$$

$$(x^1, y^1), (x^2, y^2), \dots (x^N, y^N)$$

$$f(x; \theta) = a^L = \hat{y} \quad =? \quad y$$

$$\frac{\partial C}{\partial w_{kj}^L} = \underline{\frac{\partial C}{\partial z^L}} \underline{\frac{\partial z^L}{\partial w_{kj}^L}} = \underline{\delta^L} \underline{\frac{\partial z^L}{\partial w_{kj}^L}}$$

$$\frac{\partial C}{\partial b_k^L} = \underline{\frac{\partial C}{\partial z^L}} \underline{\frac{\partial z^L}{\partial b_k^L}} = \underline{\delta^L} \underline{\frac{\partial z^L}{\partial b_k^L}}$$

$$\begin{aligned} C(\theta) \\ \nabla C = \frac{\partial C}{\partial \theta_1}, \frac{\partial C}{\partial \theta_2}, \dots \\ \theta = \theta - \alpha \nabla C \end{aligned}$$

$$\underline{\delta_k^L} = \underline{\frac{\partial C}{\partial z_k^L}} = \underline{\frac{\partial C}{\partial a_k^L}} \underline{\frac{\partial a_k^L}{\partial z_k^L}} = \underline{\frac{\partial C}{\partial a_k^L}} f'(z_k^L)$$

$$\begin{aligned} a^L &= f(z^L) \\ a_k^L &= f(z_k^L) \end{aligned}$$

$$\frac{\partial z^L}{\partial w_{kj}^L} = \underline{a_j^{L-1}}$$

$$\frac{\partial z^L}{\partial b_k^L} = \underline{1}$$

$$\begin{aligned} z^L &= w^L a^{L-1} + b^L \\ z_k^L &= \sum_j w_{kj}^L a_j^{L-1} + b^L \end{aligned}$$

$$\frac{\partial C}{\partial w_{kj}^L} = \underline{\frac{\partial C}{\partial a_k^L}} f'(z_k^L) \underline{\frac{\partial z_k^L}{\partial w_{kj}^L}} = \underline{\delta_k^L} \underline{a_j^{L-1}}$$

$$\frac{\partial C}{\partial b_k^L} = \underline{\frac{\partial C}{\partial a_k^L}} f'(z_k^L) \underline{\frac{\partial z_k^L}{\partial b_k^L}} = \underline{\delta_k^L} \underline{1}$$

# SGD / BP

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} f'(z_j^L)$$

$$\delta^L = \nabla_a C \odot f'(z^L)$$

$$C = \frac{1}{2} \sum_j (y_j - a_j^L)^2$$

$$\partial C / \partial a_j^L = (a_j^L - y_j) \quad \nabla_a C = (a^L - y)$$

$$\delta^L = \nabla_a C \odot f'(z^L) = (a^L - y) \odot f'(z^L)$$

- For multiple epochs of training:
  - Generate a mini-batch of N training examples
  - For each training example n:

- Input activation:
- Forward pass / Feedforward the activation:

$$s \odot t$$

$$(s \odot t)_j = s_j t_j$$

- Output error:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 * 3 \\ 2 * 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$$

- Backward pass / Backpropagate the error:

- Gradient descent:

$$b^l \rightarrow b^l - \frac{\eta}{N} \sum_n \delta^{n,l}$$

$$l = L, L-1, \dots, 1 \quad w^l \rightarrow w^l - \frac{\eta}{N} \sum_n \delta^{n,l} (a^{n,l-1})^T$$

$$x^n$$

$$a^{n,0} = x^n$$

$$z^{n,l} = w^l a^{n,l-1} + b^l$$

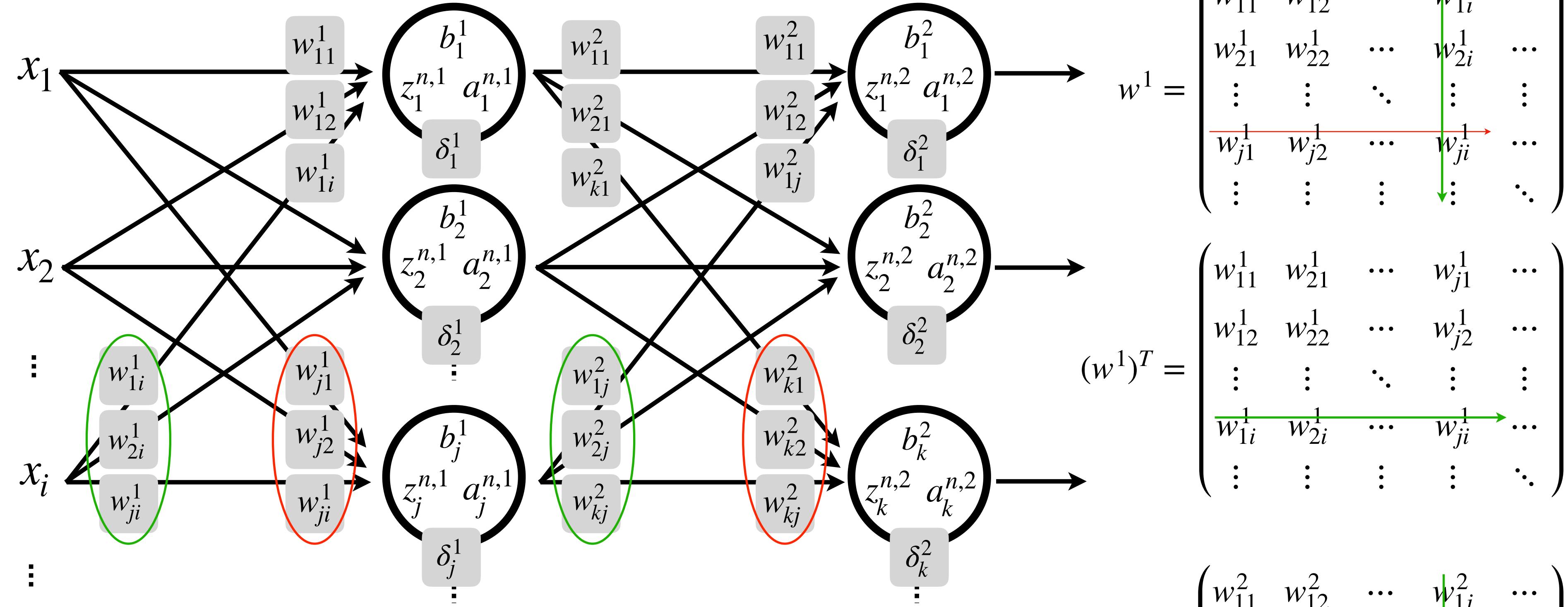
$$a^{n,l} = f(z^{n,l}) \quad l = 1, 2, \dots, L$$


$$\delta^{n,L} = \nabla_a C_n \odot f'(z^{n,L}) \quad \delta_j^l \equiv \frac{\partial C}{\partial z_j^l}$$

$$\delta^{n,l} = ((w^{l+1})^T \delta^{n,l+1} \odot f'(z^{n,l})) \quad l = L-1, L-2, \dots, 1$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

$$\frac{\partial C}{\partial w_{ji}^l} = a_i^{l-1} \delta_j^l$$



$$\begin{aligned}\delta^{n,l} &= ((w^{l+1})^T \delta^{n,l+1} \odot f'(z^{n,l})) \\ \delta^l &= ((w^{l+1})^T \delta^{l+1} \odot f'(z^l)) \\ \delta^1 &= ((w^2)^T \delta^2 \odot f'(z^1))\end{aligned}$$

$$\frac{\nabla_a C}{\partial a_k^L} \quad \delta^{n,L} = \nabla_a C_n \odot f'(z^{n,L})$$

$$\frac{\partial C}{\partial a_k^L} \quad \delta^L = \nabla_a C \odot f'(z^L)$$

$$\delta^2 = \nabla_a C \odot f'(z^2)$$

$$w^1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{ji}^1 & \cdots \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2i}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{j1}^1 & w_{j2}^1 & \cdots & w_{ji}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

$$(w^1)^T = \begin{pmatrix} w_{11}^1 & w_{21}^1 & \cdots & w_{j1}^1 & \cdots \\ w_{12}^1 & w_{22}^1 & \cdots & w_{j2}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1i}^1 & w_{2i}^1 & \cdots & w_{ji}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

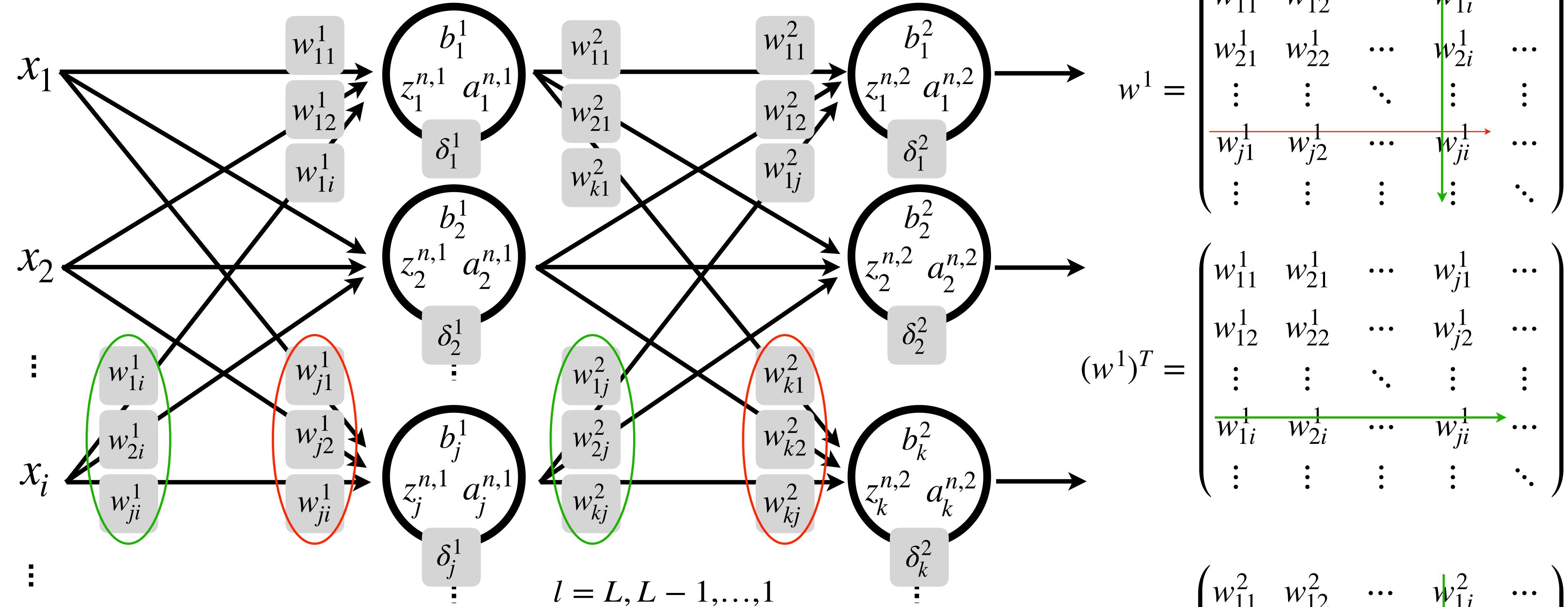
$$w^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1j}^2 & \cdots \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2j}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{k1}^2 & w_{k2}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

$$(w^2)^T = \begin{pmatrix} w_{11}^2 & w_{21}^2 & \cdots & w_{k1}^2 & \cdots \\ w_{12}^2 & w_{22}^2 & \cdots & w_{k2}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1j}^2 & w_{2j}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

?

$$\begin{pmatrix} \delta_1^1 \\ \delta_2^1 \\ \vdots \\ \delta_j^1 \\ \vdots \end{pmatrix} = \begin{pmatrix} w_{11}^2 & w_{21}^2 & \cdots & w_{k1}^2 & \cdots \\ w_{12}^2 & w_{22}^2 & \cdots & w_{k2}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1j}^2 & w_{2j}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \delta_1^2 \\ \delta_2^2 \\ \vdots \\ \delta_k^2 \\ \vdots \end{pmatrix} \odot f'(\begin{pmatrix} z_1^1 \\ z_2^1 \\ \vdots \\ z_j^1 \\ \vdots \end{pmatrix})$$

$$\begin{pmatrix} \delta_1^2 \\ \delta_2^2 \\ \vdots \\ \delta_k^2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \frac{\partial C}{\partial a_1^2} \\ \frac{\partial C}{\partial a_2^2} \\ \vdots \\ \frac{\partial C}{\partial a_k^2} \\ \vdots \end{pmatrix} \odot f'(\begin{pmatrix} z_1^2 \\ z_2^2 \\ \vdots \\ z_k^2 \\ \vdots \end{pmatrix})$$



$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \quad \frac{\partial C}{\partial b_k^l} = \delta_k^l$$

$$b^1 \rightarrow b^1 - \eta \delta^1 = b^1 - \eta \frac{\partial C}{\partial b^1}$$

$$\begin{pmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_j^1 \\ \vdots \end{pmatrix} \rightarrow \begin{pmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_j^1 \\ \vdots \end{pmatrix} - \eta \begin{pmatrix} \delta_1^1 \\ \delta_2^1 \\ \vdots \\ \delta_j^1 \\ \vdots \end{pmatrix}$$

$$l = L, L-1, \dots, 1$$

$$b^l \rightarrow b^l - \frac{\eta}{N} \sum_n \delta^{n,l}$$

$$b^l \rightarrow b^l - \eta \delta^l = b^l - \eta \frac{\partial C}{\partial b^l}$$

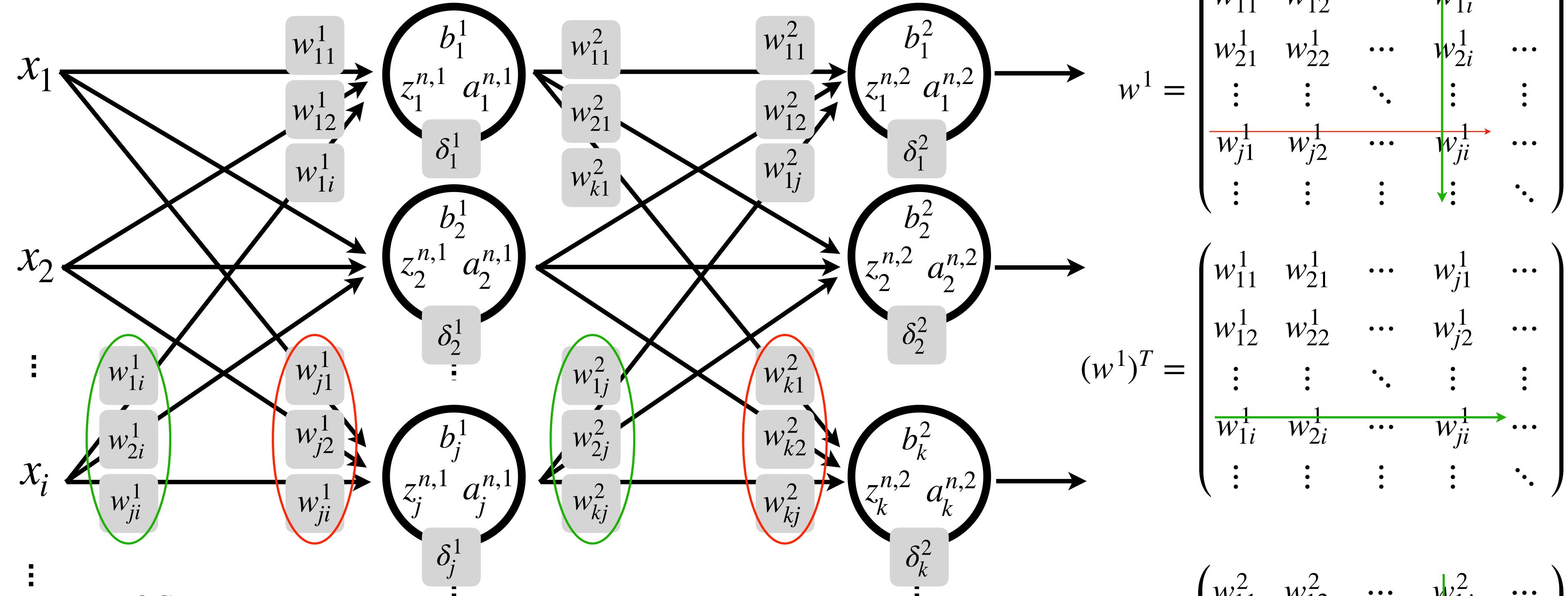
$$b^2 \rightarrow b^2 - \eta \delta^2 = b^2 - \eta \frac{\partial C}{\partial b^2}$$

$$w^1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{1i}^1 & \cdots \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2i}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{j1}^1 & w_{j2}^1 & \cdots & w_{ji}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{pmatrix}$$

$$(w^1)^T = \begin{pmatrix} w_{11}^1 & w_{21}^1 & \cdots & w_{j1}^1 & \cdots \\ w_{12}^1 & w_{22}^1 & \cdots & w_{j2}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1i}^1 & w_{2i}^1 & \cdots & w_{ji}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{pmatrix}$$

$$w^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1j}^2 & \cdots \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2j}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{k1}^2 & w_{k2}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{pmatrix}$$

$$(w^2)^T = \begin{pmatrix} w_{11}^2 & w_{21}^2 & \cdots & w_{k1}^2 & \cdots \\ w_{12}^2 & w_{22}^2 & \cdots & w_{k2}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1j}^2 & w_{2j}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{pmatrix}$$



$$\frac{\partial C}{\partial w_{ji}^l} = a_i^{l-1} \delta_j^l = \delta_j^l a_i^{l-1}$$

$$\frac{\partial C}{\partial w_{kj}^l} = a_j^{l-1} \delta_k^l = \delta_k^l a_j^{l-1}$$

$$w^1 \rightarrow w^1 - \eta \delta^1 (a^0)^T$$

$$l = L, L-1, \dots, 1$$

$$w^l \rightarrow w^l - \frac{\eta}{N} \sum \delta^{n,l} (a^{n,l-1})^T$$

$$w^l \rightarrow w^l - \eta \delta^l (a^{l-1})^T$$

$$w^2 \rightarrow w^2 - \eta \delta^2 (a^1)^T$$

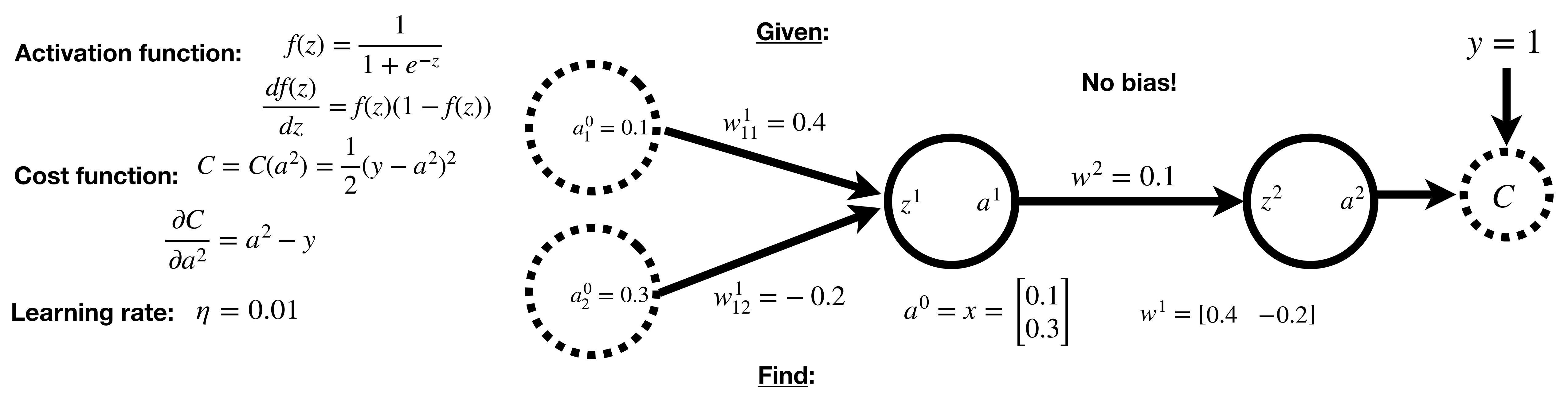
$$w^1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{1i}^1 & \cdots \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2i}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{j1}^1 & w_{j2}^1 & \cdots & w_{ji}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

$$(w^1)^T = \begin{pmatrix} w_{11}^1 & w_{21}^1 & \cdots & w_{j1}^1 & \cdots \\ w_{12}^1 & w_{22}^1 & \cdots & w_{j2}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1i}^1 & w_{2i}^1 & \cdots & w_{ji}^1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

$$w^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1j}^2 & \cdots \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2j}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{k1}^2 & w_{k2}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

$$(w^2)^T = \begin{pmatrix} w_{11}^2 & w_{21}^2 & \cdots & w_{k1}^2 & \cdots \\ w_{12}^2 & w_{22}^2 & \cdots & w_{k2}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{1j}^2 & w_{2j}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

$$\begin{pmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1j}^2 & \cdots \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2j}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{k1}^2 & w_{k2}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \rightarrow \begin{pmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1j}^2 & \cdots \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2j}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{k1}^2 & w_{k2}^2 & \cdots & w_{kj}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} - \eta \begin{pmatrix} \delta_1^2 \\ \delta_2^2 \\ \vdots \\ \delta_k^2 \end{pmatrix} \begin{pmatrix} a_1^1 & a_2^1 & \cdots & a_j^1 & \cdots \end{pmatrix} \quad (w^2)^T =$$



**Forward pass:**  $z^1 =$   
 $a^1 =$

$$z^2 =$$

$$a^2 =$$

**Backward pass:**

$$\delta^2 =$$

$$\delta^1 =$$

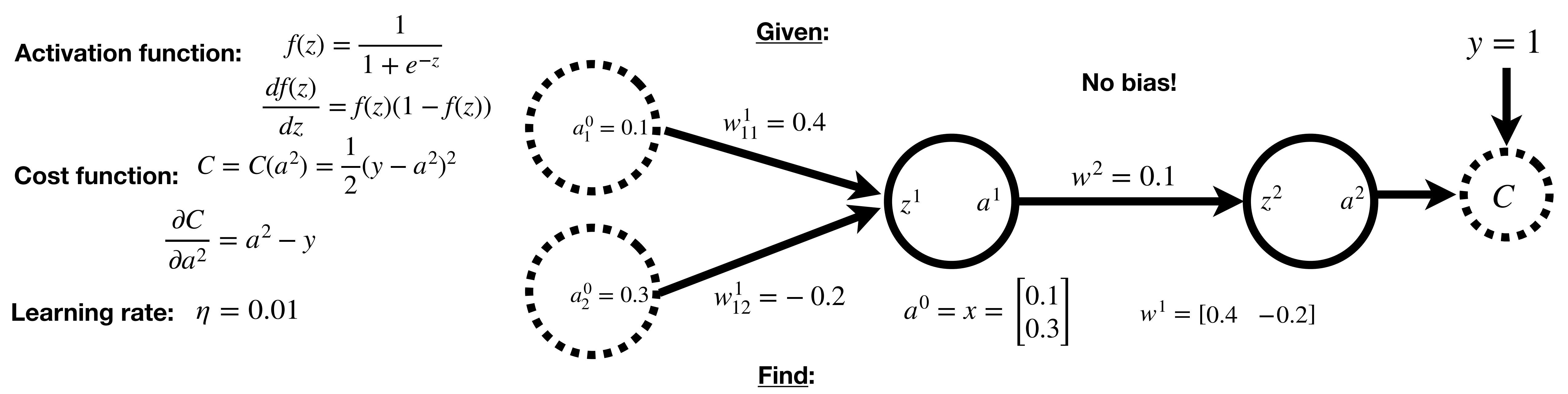
$$\frac{\partial C}{\partial w^2} =$$

$$w_{new}^2 =$$

$$w_{new}^1 =$$

**NB: outer product in matrix notation**

$$\frac{\partial C}{\partial w^1} =$$



**Forward pass:**  $z^1 = w^1 a^0 + b^1 = [0.4 \quad -0.2] \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix} + 0 = 0.04 - 0.06 = -0.02$

$$a^1 = f(z^1) = f(-0.02) = 0.495$$

$z^2 = w^2 a^1 + b^2 = 0.1 \cdot 0.495 + 0 = 0.0495$

$$a^2 = f(z^2) = f(0.0495) = 0.5124$$

### Backward pass:

$$\delta^2 = \frac{\partial C}{\partial a^2} f'(z^2) = (a^2 - y) f'(z^2) = (0.5124 - 1) \cdot (0.5124 \cdot (1 - 0.5124)) = -0.4876 \cdot 0.5124 \cdot 0.4876 = -0.1218$$

$$\delta^1 = (w^2)^T \delta^2 \cdot f'(z^1) = 0.1 \cdot (-0.1218) \cdot f'(-0.02) = 0.1 \cdot (-0.1218) \cdot 0.495 \cdot (1 - 0.495) = -0.01218 \cdot 0.25 = -0.003045$$

$$\frac{\partial C}{\partial w^2} = \delta^2 a^1 = 0.495 \cdot -0.1218 = -0.060$$

**NB: outer product in matrix notation**

$$\frac{\partial C}{\partial w^1} = \delta^1 (a^0)^T = -0.003045 \cdot [0.1 \quad 0.3] = -[0.0003045 \quad 0.0009135]$$

$$w_{new}^2 = w^2 - \eta \frac{\partial C}{\partial w^2} = 0.1 - 0.01 \cdot (-0.06) = 0.1 + 0.0006 = 0.1006$$

$$w_{new}^1 = w^1 - \eta \frac{\partial C}{\partial w^1} = [0.4 \quad -0.2] + 0.01 [0.0003045 \quad 0.0009135] = [0.400003045 \quad -0.199990865]$$

# Next DL-based topic: CNNs & ImClass

