# CSC311 Introduction to Machine Learning

# The Backpropagation Algorithm

Alice Gao and Marina Tawfik

## Learning Outcomes

By the end of this lecture, students should be able to

- Explain why the backpropagation algorithm can efficiently compute many gradients for a neural network.
- Derive non-vectorized expressions for the gradients with respect to weights, biases, pre-activations, and activations in a small neural network.
- Explain the role of the forward pass in preparing quantities required for backpropagation.

# Outline

# Challenge of Training a Neural Network

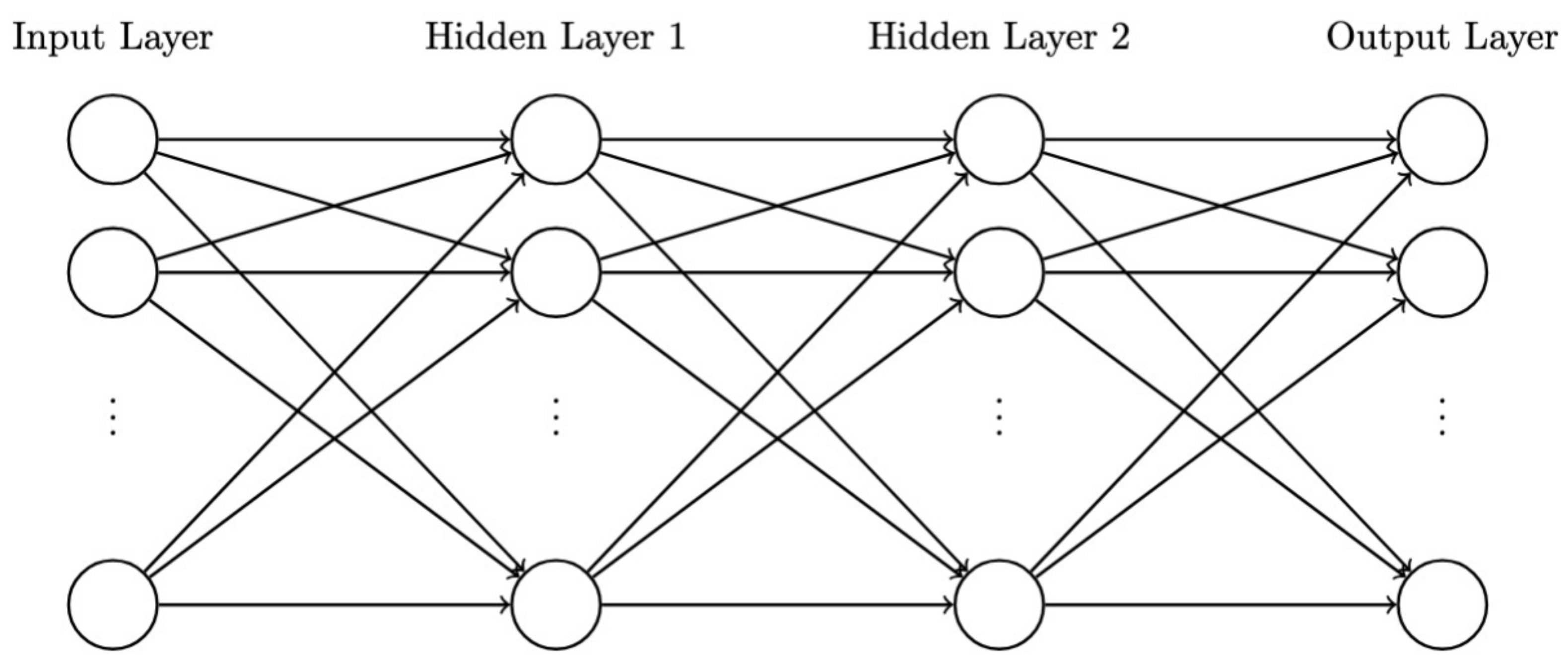
UNIVERSITY OF TORONTO

DEFY GRAVITY

# Optimizing a Neural Network

How do we find good weights for a neural network?

Use gradient descent to adjust the weights to reduce the loss.



# Gradient Descent for a Neural Network

1. Initialize the weights $\mathbf{w}$.

This is what we called $\alpha$ in previous lectures.

2. Perform the updates.

$\alpha$

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{E}(\mathbf{w})$$

where

- $\eta$ is the learning rate
- $\nabla_{\mathbf{w}} \mathcal{E}(\mathbf{w})$ is the gradient of cost function with respect to the weights $\mathbf{w}$

# The Challenge of Computing All the Gradients

To train a neural network with gradient descent

Need the partial derivative of loss function with respect to every weight.

But a neural network can have millions of weights.

How can we compute the gradient of the cost function efficiently?

The Backpropagation Algorithm

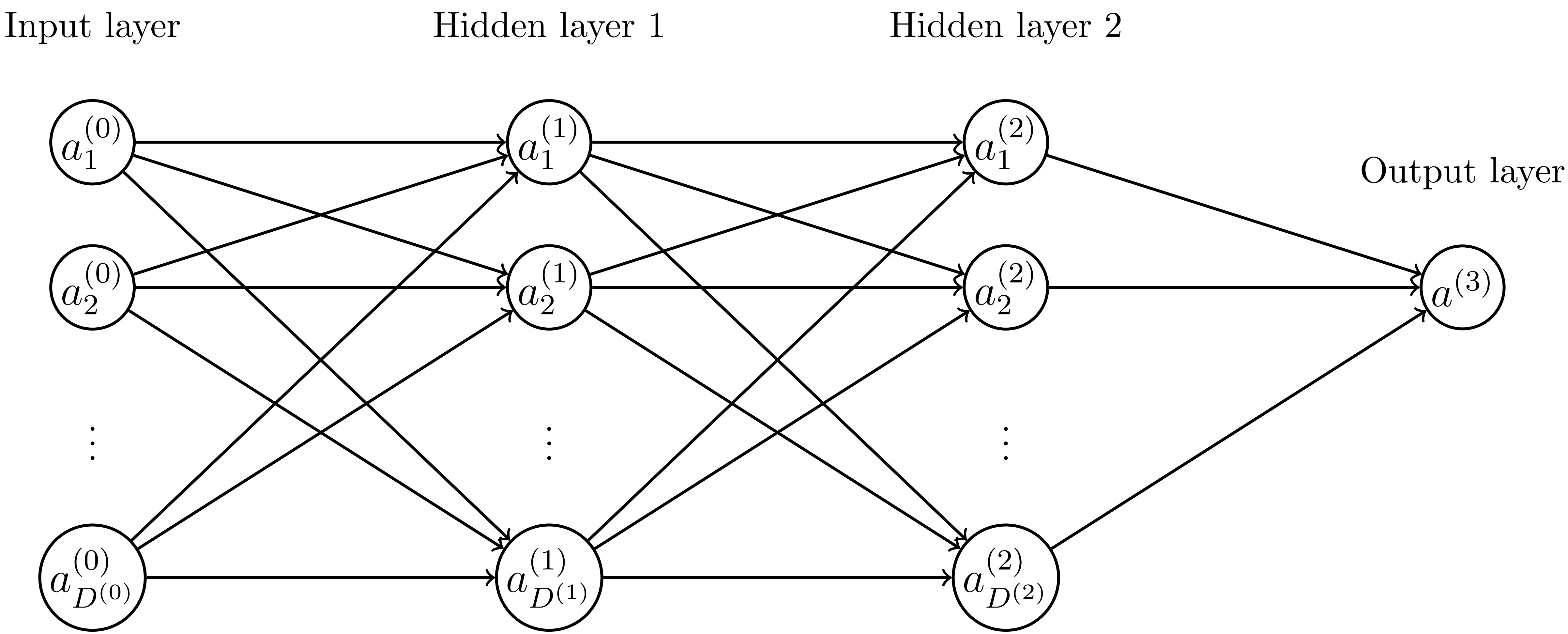
# Gradients for 3-Layer Neural Network

UNIVERSITY OF TORONTO

DEFY GRAVITY

# 3-Layer Neural Network

Here, we have a single output i.e. this would work for regression & binary classification.



# Notation and Definitions

| | | | |
|---|---|---|---|
| $L$ | | $m$ | |
| | | $D^{(m)}$ | |
| $\mathbf{a}^{(0)}$ | | $\mathbf{W}^{(m)}$ | |
| | | $\mathbf{b}^{(m)}$ | |
| $t$ | | $\sigma^{(m)}$ | |
| $C(\mathbf{a}^{(L)}, y)$ | | $\mathbf{z}^{(m)}$ | |
| | | $\mathbf{a}^{(m)}$ | |

# Notation and Definitions

W $\vec{x}$

output x input    input x 1

| | | | |
|---|---|---|---|
| $L$ | Number of layers in the model | $m$ | Index of layer, $m \in [1, L]$ |
| | | $D^{(m)}$ | Dimensionality of layer $m$ |
| $\mathbf{a}^{(0)}$ | Input to the model ($\mathbf{x}$) | $\mathbf{W}^{(m)} \in \mathbb{R}^{D^{(m)} \times D^{(m-1)}}$ | Weight matrix for layer $m$ |
| | | $\mathbf{b}^{(m)} \in \mathbb{R}^{D^{(m)}}$ | Bias vector for layer $m$ |
| $t$ | Target output | $\sigma^{(m)}$ | Non-linearity for layer $m$ |
| $C(\mathbf{a}^{(L)}, y)$ | Cost (or loss) function | $\mathbf{z}^{(m)} \in \mathbb{R}^{D^{(m)}}$ | Pre-activations for layer $m$ |
| | | $\mathbf{a}^{(m)} \in \mathbb{R}^{D^{(m)}}$ | Activations for layer $m$ |

# of units in the m^th layer

output dim    input dim    going to layer m

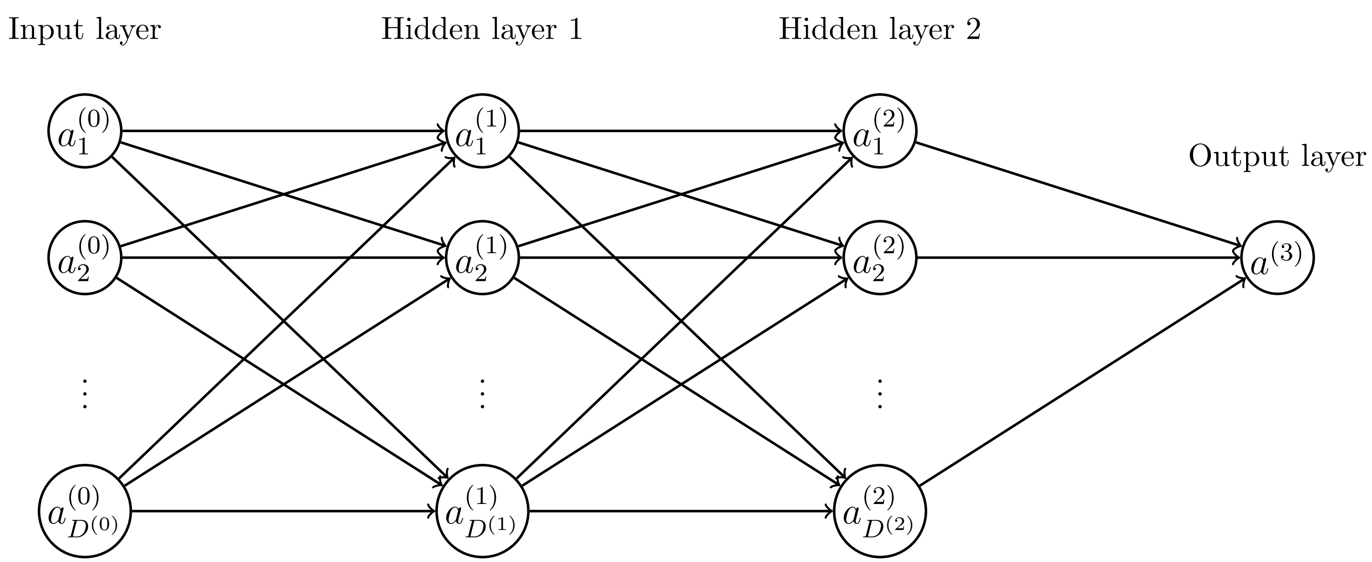
# Computations in the 3-Layer Neural Network

$$z_i^{(2)} = \sum_{j=1}^{D^{(1)}} W_{ij}^{(2)} a_j^{(1)} + b_i^{(2)}, i = 1, \ldots, D^{(2)}$$

$$a_i^{(2)} = \sigma^{(2)}\left(z_i^{(2)}\right), i = 1, \ldots, D^{(2)}$$

$$z^{(3)} = \sum_{j=1}^{D^{(2)}} w_j^{(3)} a_j^{(2)} + b^{(3)}$$

$$a^{(3)} = \sigma^{(3)}\left(z^{(3)}\right)$$




Cost: $C = C(a^{(3)}, t)$

# Two Exercises of Computing Derivatives

Compute

$$\frac{\partial C}{\partial W_{12}^{(2)}}$$

into ← 12 → from

Going from layer 1 unit 2 to layer 2 unit 1

Compute

$$\frac{\partial C}{\partial W_{23}^{(1)}}$$

Going from layer 0 unit 3 to layer 1 unit 2

layer 0 → input $a^{(0)}$

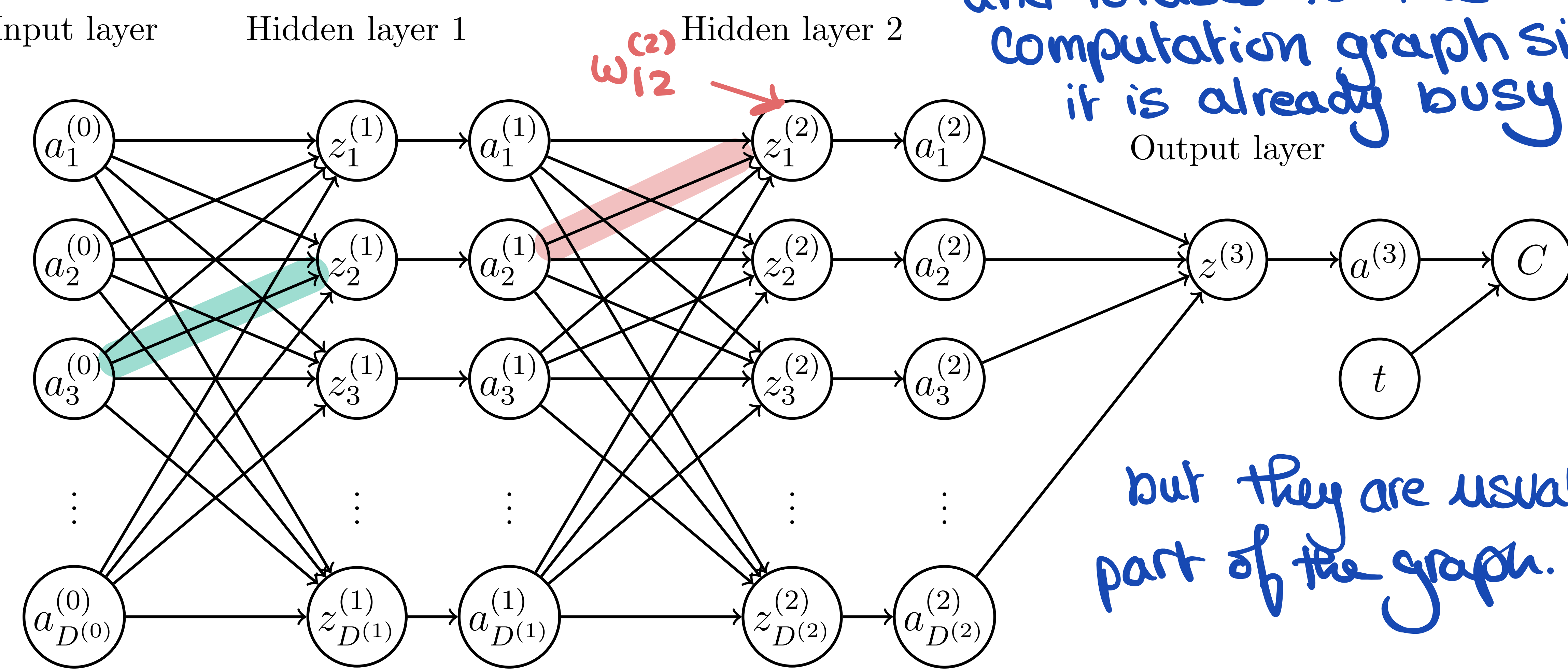


UNIVERSITY OF TORONTO

# Computation Graph

$w^{(2)}_{12}$ $w^{(1)}_{23}$

we didn't add the weights and biases to the computation graph since it is already busy.




but they are usually part of the graph.

# Review: Univariate Chain Rule

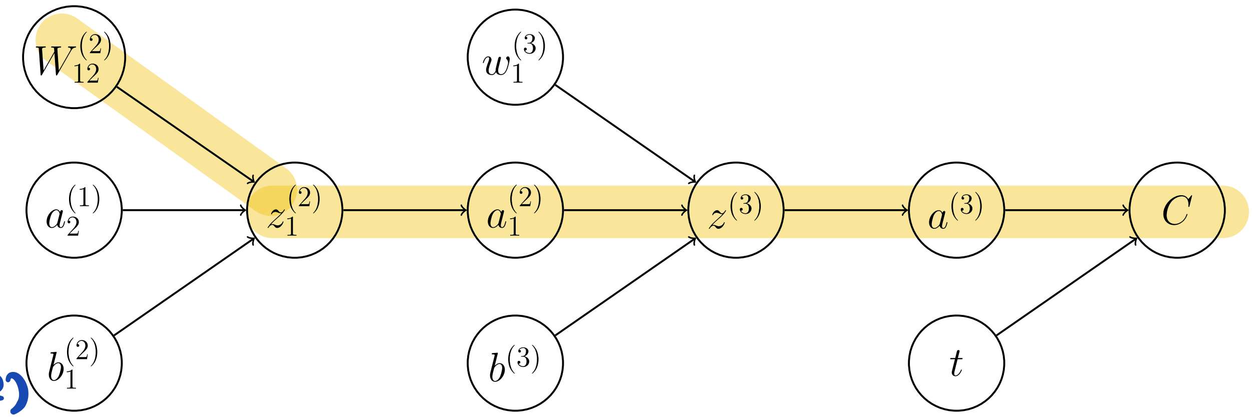
$$y = f(x) \qquad\qquad z = g(y) = g(f(x))$$

$$\frac{dz}{dx} = \frac{dz}{dy} * \frac{dy}{dx}$$

$$x \longrightarrow y \longrightarrow z$$

# Loss Derivative with respect to $W_{12}^{(2)}$

$$\frac{\partial C}{\partial W_{12}^{(2)}}$$




$$= \frac{\partial C}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial a_1^{(2)}} \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} \frac{\partial z_1^{(2)}}{\partial w_{12}^{(2)}}$$

$$=$$

$$z_i^{(2)} = \sum_{j=1}^{D^{(1)}} W_{ij}^{(2)} a_j^{(1)} + b_i^{(2)}, i = 1, \ldots, D^{(2)}$$

$$a_i^{(2)} = \sigma^{(2)}\left(z_i^{(2)}\right), i = 1, \ldots, D^{(2)}$$

$$z^{(3)} = \sum_{j=1}^{D^{(2)}} w_j^{(3)} a_j^{(2)} + b^{(3)}$$

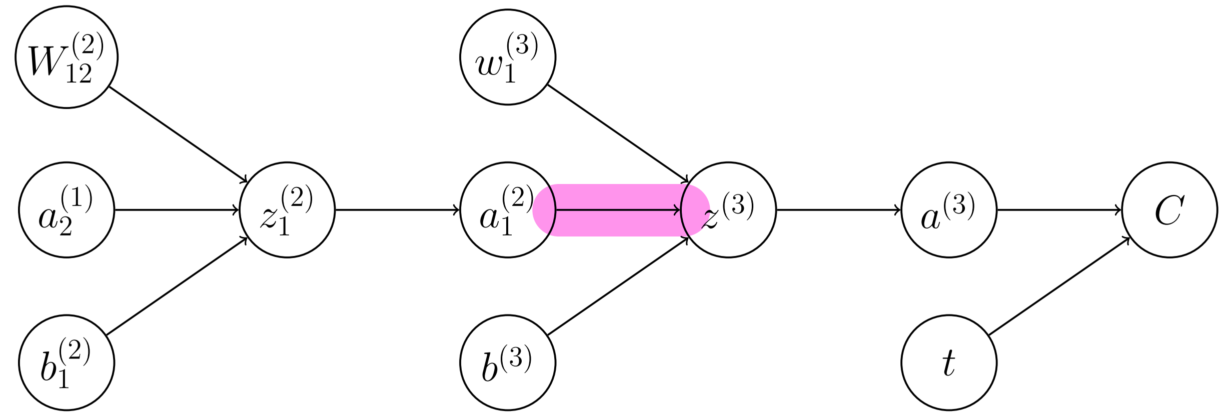
$$a^{(3)} = \sigma^{(3)}\left(z^{(3)}\right)$$

$$C = C\left(a^{(3)}, t\right)$$

# Solution: Loss Derivative with respect to $W_{12}^{(2)}$

$$\frac{\partial C}{\partial W_{12}^{(2)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} \cdot \frac{\partial z_1^{(2)}}{\partial W_{12}^{(2)}}$$

$$= C' \cdot \sigma^{(3)\prime}\left(z^{(3)}\right) \cdot \left(w_1^{(3)}\right) \cdot \sigma^{(2)\prime}\left(z_1^{(2)}\right) \cdot a_2^{(1)}$$




$$z_i^{(2)} = \sum_{j=1}^{D^{(1)}} W_{ij}^{(2)} a_j^{(1)} + b_i^{(2)}, i = 1, \dots, D^{(2)}$$

$$a_i^{(2)} = \sigma^{(2)}\left(z_i^{(2)}\right), i = 1, \dots, D^{(2)}$$

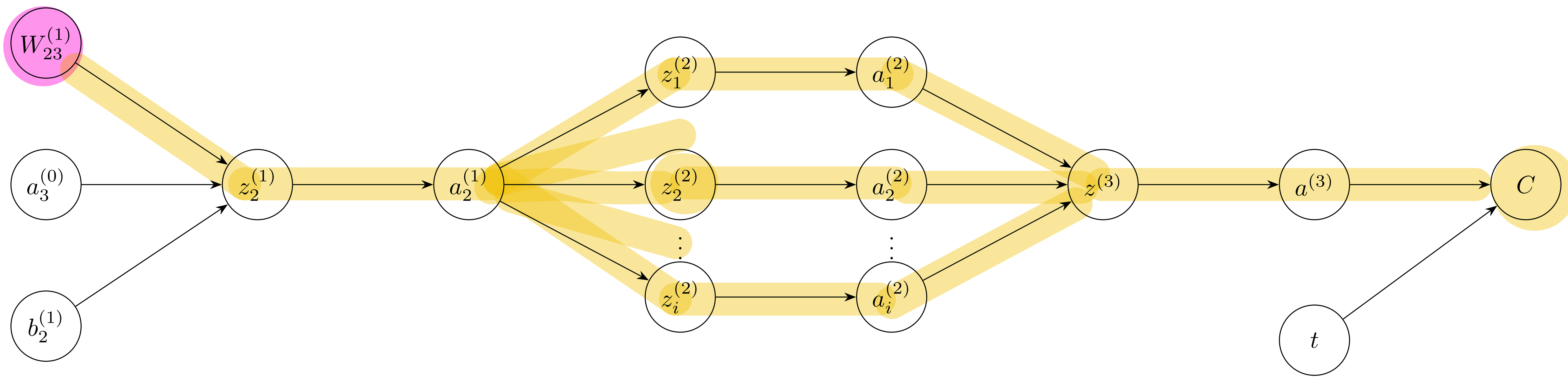
$$z^{(3)} = \sum_{j=1}^{D^{(2)}} w_j^{(3)} a_j^{(2)} + b^{(3)}$$

$$a^{(3)} = \sigma^{(3)}\left(z^{(3)}\right)$$

$$C = C\left(a^{(3)}, t\right)$$

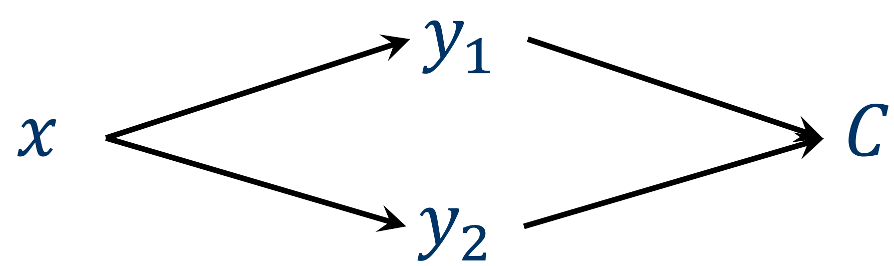
# Computing Loss Derivative with respect to $W_{23}^{(1)}$

$\frac{\partial C}{\partial \omega_{23}^{(1)}}$



# Review: Multi-Variate Chain Rule

$$y_1 = f_1(x), \qquad y_2 = f_2(x), \qquad C = g(y_1, y_2)$$




$$\frac{dC}{dx} = \frac{\partial C}{\partial y_1} * \frac{dy_1}{dx} + \frac{\partial C}{\partial y_2} * \frac{dy_2}{dx}$$

## Computing the Loss

layer 1

$$z_i^{(1)} = \sum_{j=1}^{D^{(0)}} W_{ij}^{(1)} a_j^{(0)} + b_i^{(1)},$$
$$i = 1, \dots, D^{(1)}$$

$$a_i^{(1)} = \sigma^{(1)}\left(z_i^{(1)}\right),$$
$$i = 1, \dots, D^{(1)}$$

layer 2

$$z_i^{(2)} = \sum_{j=1}^{D^{(1)}} W_{ij}^{(2)} a_j^{(1)} + b_i^{(2)},$$
$$i = 1, \dots, D^{(2)}$$

$$a_i^{(2)} = \sigma^{(2)}\left(z_i^{(2)}\right),$$
$$i = 1, \dots, D^{(2)}$$

layer 3

$$z^{(3)} = \sum_{j=1}^{D^{(2)}} W_j^{(3)} a_j^{(2)} + b^{(3)}$$

$$a^{(3)} = \sigma^{(3)}\left(z^{(3)}\right)$$

$$C = C(a^{(3)}, t)$$

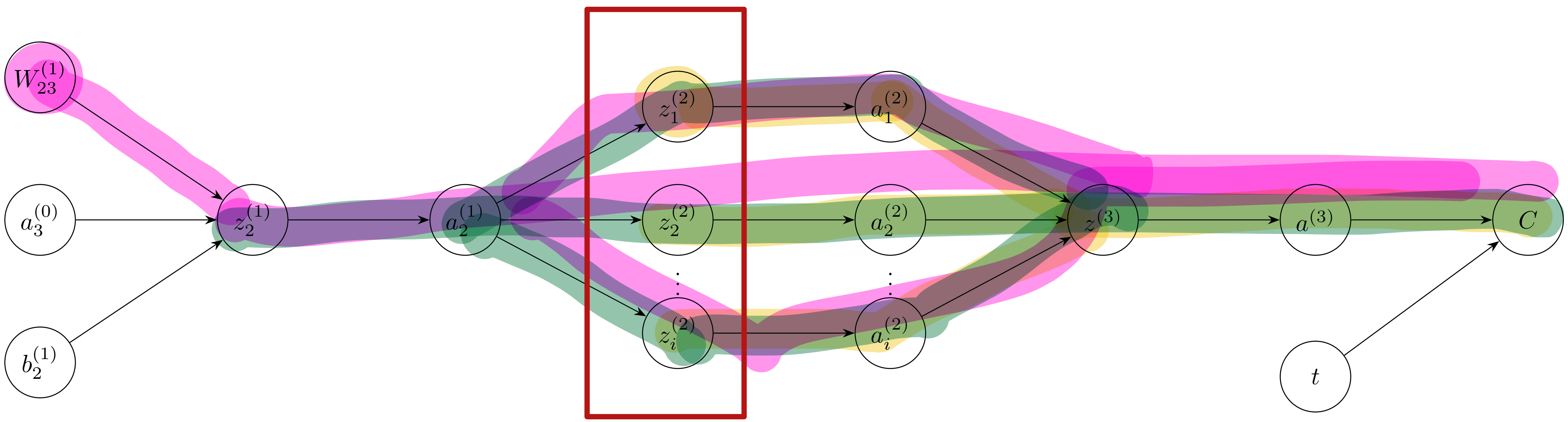
# Loss Derivative with respect to $W_{23}^{(1)}$

$$\frac{\partial C}{\partial z_2^{(1)}} = \left( \sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \frac{\partial z_i^{(2)}}{\partial a_2^{(1)}} \right) \frac{\partial a_2^{(1)}}{\partial z_2^{(1)}}$$

$$\frac{\partial C}{\partial W_{23}^{(1)}} = \frac{\partial C}{\partial z_2^{(1)}} \frac{\partial z_2^{(1)}}{\partial w_{23}^{(1)}}$$

$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}}$$

$$\frac{\partial C}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_i^{(2)}} \cdot \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}}, i = 1, \dots D^{(2)}$$



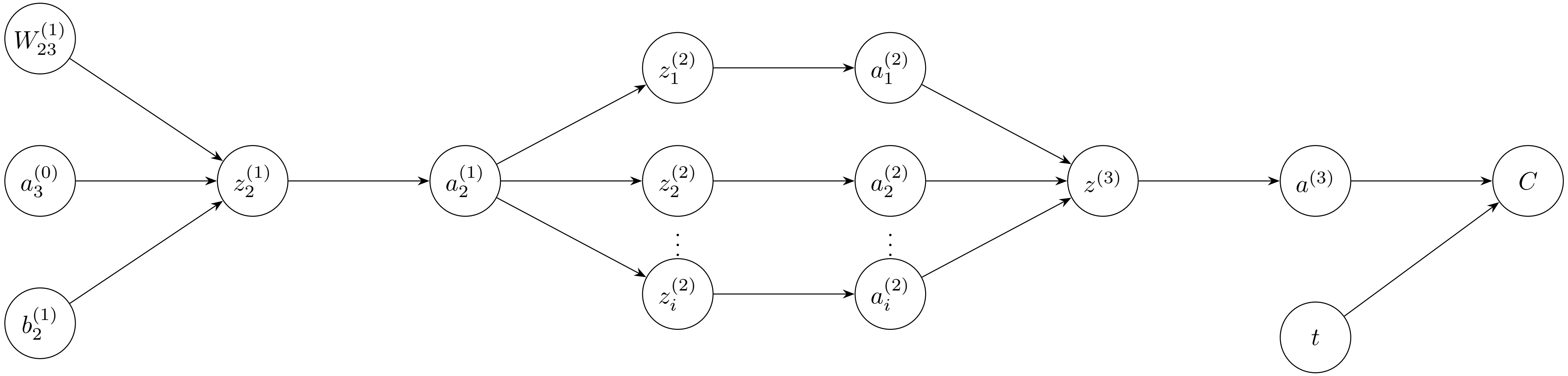
# Solution: Loss Derivative with respect to $W_{23}^{(1)}$

$$\frac{\partial C}{\partial z_2^{(1)}} = \left( \sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial a_2^{(1)}} \right) \frac{\partial a_2^{(1)}}{\partial z_2^{(1)}}$$

$$\frac{\partial C}{\partial W_{23}^{(1)}} = \frac{\partial C}{\partial z_2^{(1)}} \cdot \frac{\partial z_2^{(1)}}{\partial W_{23}^{(1)}}$$

$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}}$$

$$\frac{\partial C}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_i^{(2)}} \cdot \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}}, i = 1, \dots D^{(2)}$$



# Computing Gradients Efficiently

UNIVERSITY OF TORONTO

DEFY GRAVITY

# Notice Any Inefficiency in These Gradient Computations?

$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}}$$

$$\frac{\partial C}{\partial z_1^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}}$$

$$\frac{\partial C}{\partial W_{12}^{(2)}} = \frac{\partial C}{\partial z_1^{(2)}} \cdot \frac{\partial z_1^{(2)}}{\partial W_{12}^{(2)}}$$

$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}}$$

$$\frac{\partial C}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_i^{(2)}} \cdot \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}}, i = 1, \ldots D^{(2)}$$

$$\frac{\partial C}{\partial z_2^{(1)}} = \left( \sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial a_5^{(1)}} \right) \frac{\partial a_2^{(1)}}{\partial z_2^{(1)}}$$

$$\frac{\partial C}{\partial W_{23}^{(1)}} = \frac{\partial C}{\partial z_2^{(1)}} \cdot \frac{\partial z_2^{(1)}}{\partial W_{23}^{(1)}}$$

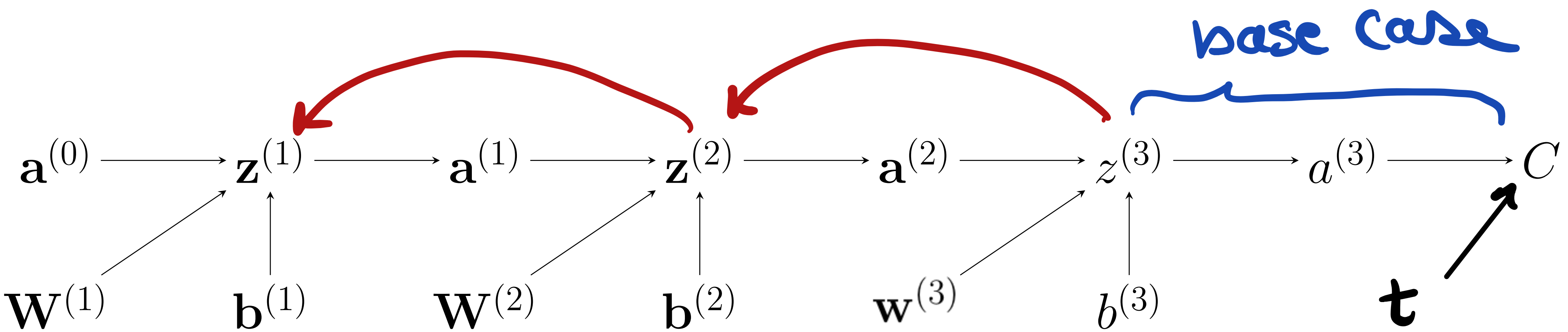
highlighted computation are repeated between the 2 derivatives.

# Reuse Computations for All Gradients

This is a vectorized computation graph.

Many repeated computations to compute each derivative

Compute all derivatives in one pass by reusing intermediate results (dynamic programming).



# Backpropagation for a 3-Layer Neural Network

UNIVERSITY OF TORONTO

DEFY GRAVITY

# Backpropagation for 3-Layer Network

Step 1: Gradients for the output layer

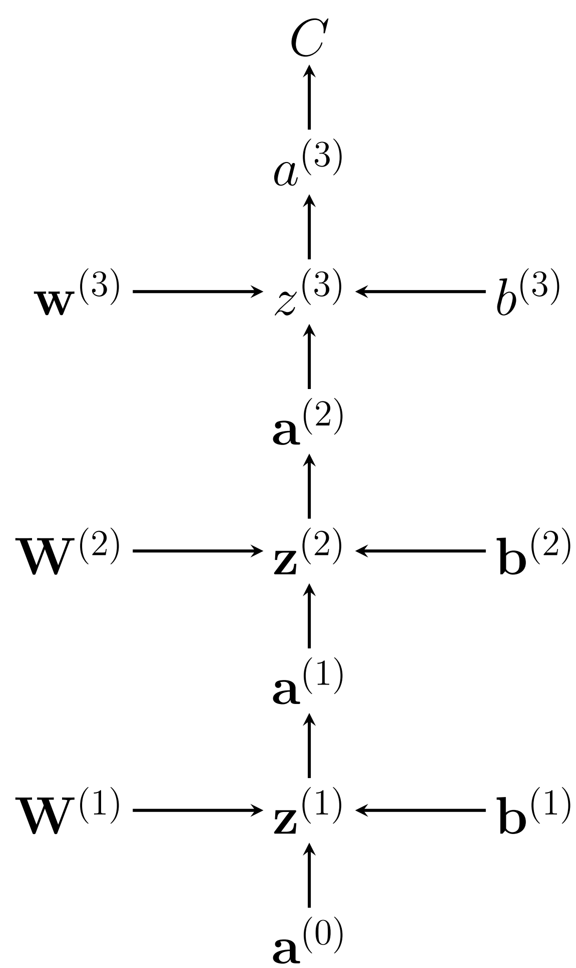
$$\frac{\partial C}{\partial z^{(3)}} = ? \qquad \frac{\partial C}{\partial w^{(3)}} = ? \qquad \frac{\partial C}{\partial b^{(3)}} = ?$$

Step 2: Gradients for hidden layer 2

$$\frac{\partial C}{\partial z_i^{(2)}} = ? \qquad \frac{\partial C}{\partial W_{ij}^{(2)}} = ? \qquad \frac{\partial C}{\partial b_i^{(2)}} = ?$$

Step 3: Gradients for hidden layer 1

$$\frac{\partial C}{\partial z_j^{(1)}} = ? \qquad \frac{\partial C}{\partial W_{ij}^{(1)}} = ? \qquad \frac{\partial C}{\partial b_i^{(1)}} = ?$$



output is a scalar

# Backprop Step 1: Gradients for Output Layer

For pre-activation $z^{(3)}$:

$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial z^{(3)}}$$

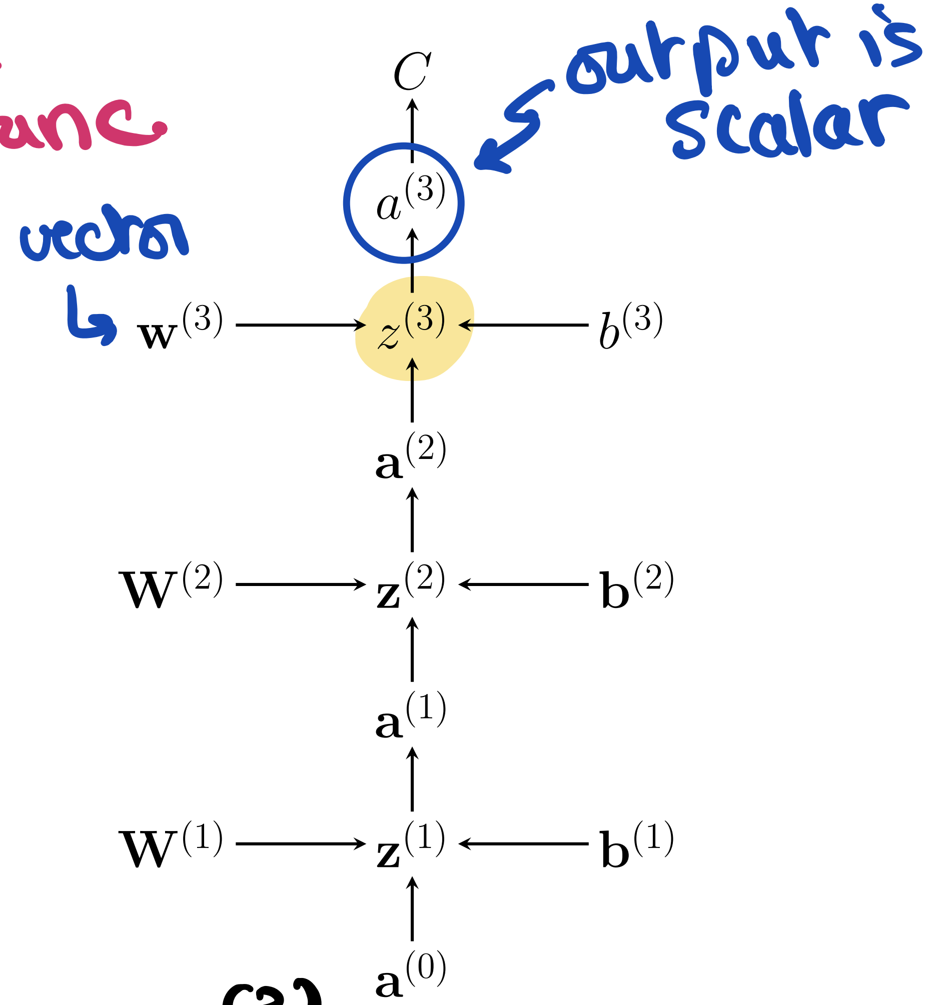
↗ depends on the activation func

For weight:

$$\frac{\partial C}{\partial w_i^{(3)}} = \frac{\partial C}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial w_i^{(3)}} = \frac{\partial C}{\partial z^{(3)}} a_i^{(2)}$$

For bias:

$$\frac{\partial C}{\partial b^{(3)}} = \frac{\partial C}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial b^{(3)}} = \frac{\partial C}{\partial z^{(3)}} \cdot 1$$




$$z^{(3)} = w_1^{(3)} a_1^{(2)} + w_2^{(3)} a_2^{(2)} + \dots + b^{(3)}$$

# Backprop Step 1: Gradients for Output Layer

For pre-activation $z^{(3)}$:

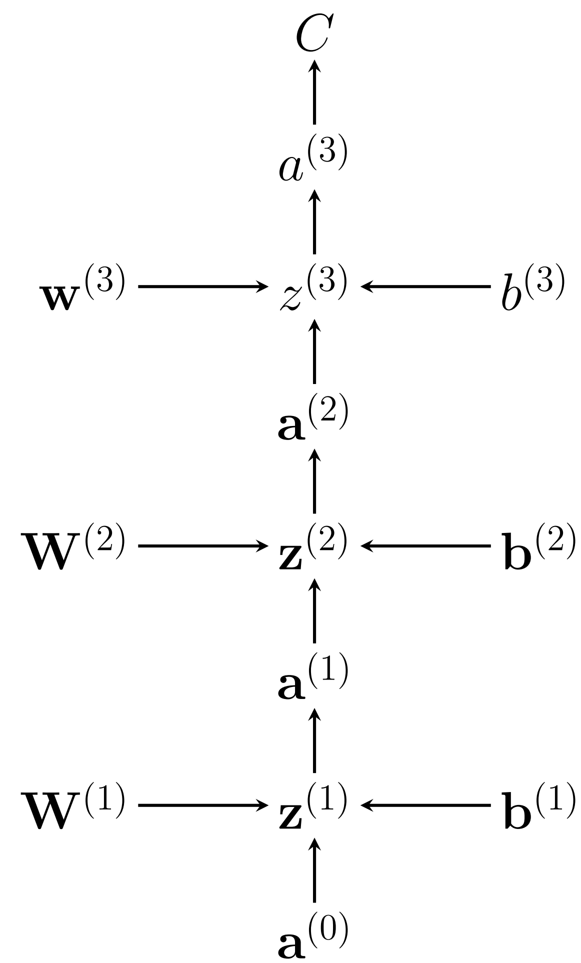
$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \sigma^{(3)\prime}\left(z^{(3)}\right)$$

For weight:

$$\frac{\partial C}{\partial w_i^{(3)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial w_i^{(3)}} = \frac{\partial C}{\partial z^{(3)}} \cdot a_i^{(2)}$$

For bias:

$$\frac{\partial C}{\partial b^{(3)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial b^{(3)}} = \frac{\partial C}{\partial z^{(3)}}$$



# Backprop Step 2: Gradients for Hidden Layer 2

For pre-activations $z_i^{(2)}$:

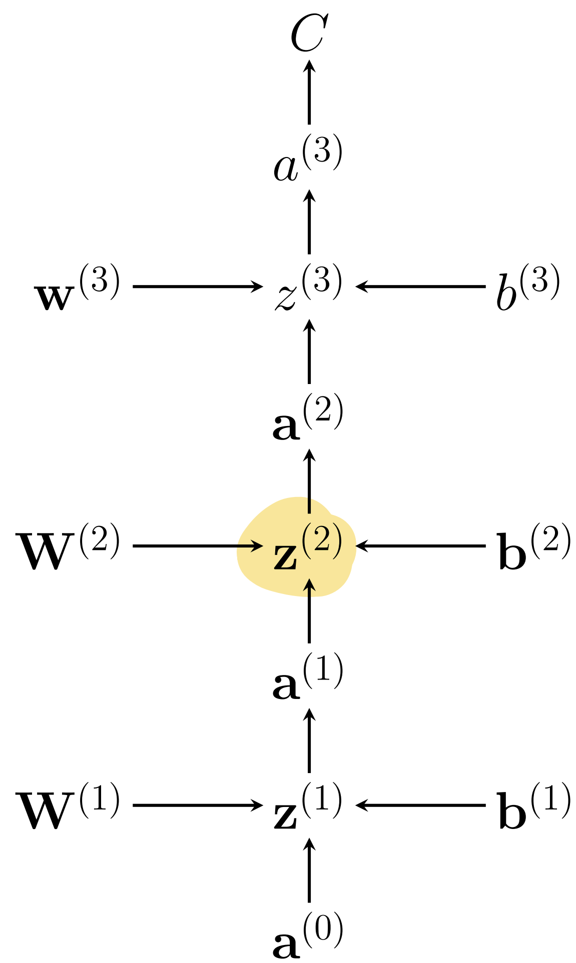
$$\frac{\partial C}{\partial z_i^{(2)}} =$$

For weight:

$$\frac{\partial C}{\partial W_{ij}^{(2)}} =$$

For bias:

$$\frac{\partial C}{\partial b_i^{(2)}} =$$



# Backprop Step 2: Gradients for Hidden Layer 2

$$z^{(3)} = w_1^{(3)} a_1^{(2)} + w_2^{(3)} a_2^{(2)} + \dots + b^{(3)}$$

For pre-activations $z_i^{(2)}$:

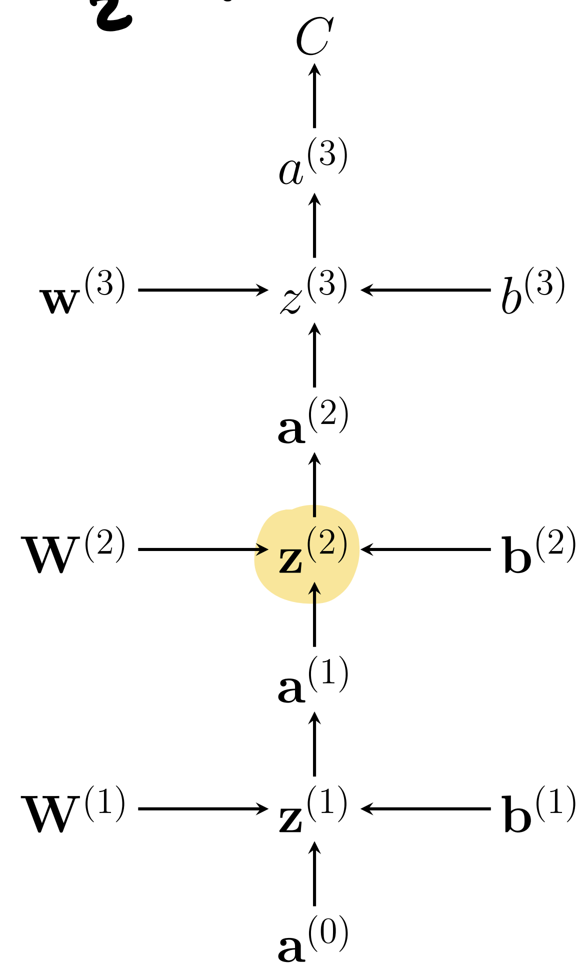
$$\frac{\partial C}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_i^{(2)}} \cdot \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot w_i^{(3)} \cdot \sigma^{(2)\prime}\left(z_i^{(2)}\right)$$

For weight:

$$\frac{\partial C}{\partial W_{ij}^{(2)}} = \frac{\partial C}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial W_{ij}^{(2)}} = \frac{\partial C}{\partial z_i^{(2)}} \cdot a_j^{(1)}$$

For bias:

$$\frac{\partial C}{\partial b_i^{(2)}} = \frac{\partial C}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial b_i^{(2)}} = \frac{\partial C}{\partial z_i^{(2)}} \quad 1$$




$$z_i^{(2)} = w_{i1}^{(2)} a_1^{(1)} + w_{i2}^{(2)} a_2^{(1)} + \dots + b_i^{(2)}$$

# Backprop Step 3: Gradients for Hidden Layer 1

For pre-activations $z_j^{(1)}$:

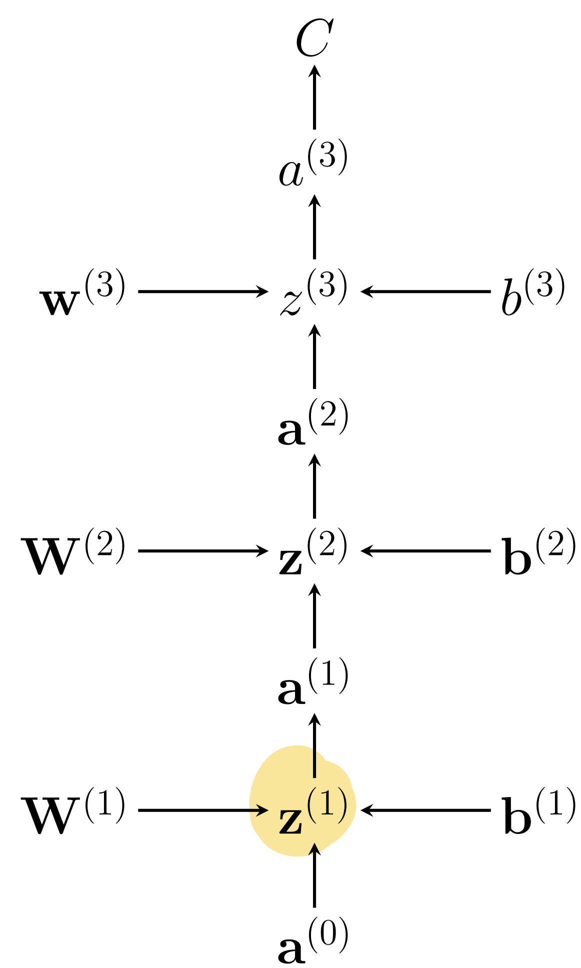
$$\frac{\partial C}{\partial z_j^{(1)}} =$$

For weight:

$$\frac{\partial C}{\partial W_{ij}^{(1)}} =$$

For bias:

$$\frac{\partial C}{\partial b_i^{(1)}} =$$



# Backprop Step 3: Gradients for Hidden Layer 1

For pre-activations $z_j^{(1)}$:

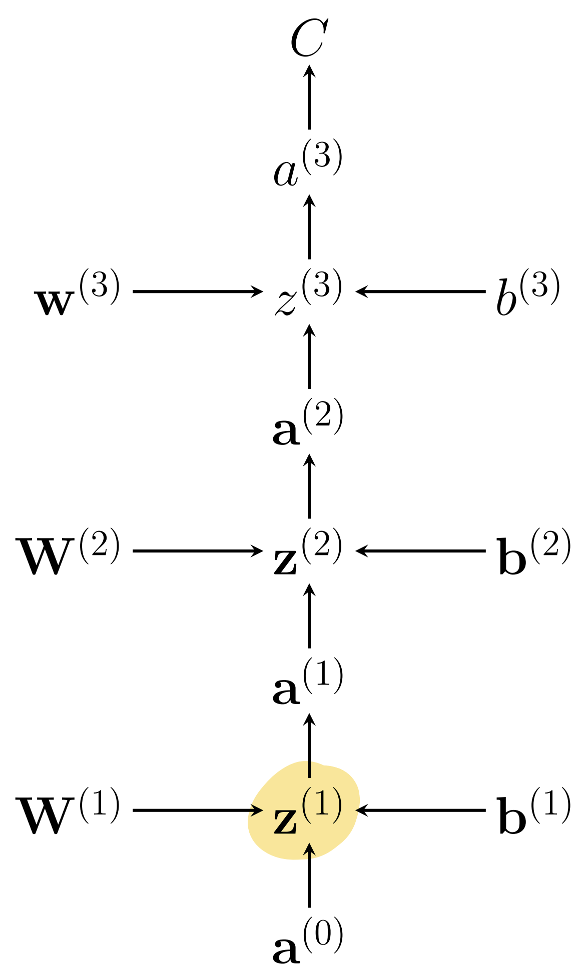
$$\frac{\partial C}{\partial z_j^{(1)}} = \left(\sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial a_j^{(1)}}\right) \cdot \frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} = \left(\sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \cdot W_{ij}^{(2)}\right) \cdot \sigma^{(1)\prime}\left(z_j^{(1)}\right)$$

For weight:

$$\frac{\partial C}{\partial W_{ij}^{(1)}} = \frac{\partial C}{\partial z_i^{(1)}} \cdot \frac{\partial z_i^{(1)}}{\partial W_{ij}^{(1)}} = \frac{\partial C}{\partial z_i^{(1)}} \cdot a_j^{(0)}$$

For bias:

$$\frac{\partial C}{\partial b_i^{(1)}} = \frac{\partial C}{\partial z_i^{(1)}} \cdot \frac{\partial z_i^{(1)}}{\partial b_i^{(1)}} = \frac{\partial C}{\partial z_i^{(1)}}$$




$$z_i^{(1)} = w_{i1}^{(1)} a_1^{(0)} + w_{i2}^{(1)} a_2^{(0)} + \cdots + b_i^{(1)}$$

# Backpropagation for 3-Layer Network

Step1: Gradients for output layer

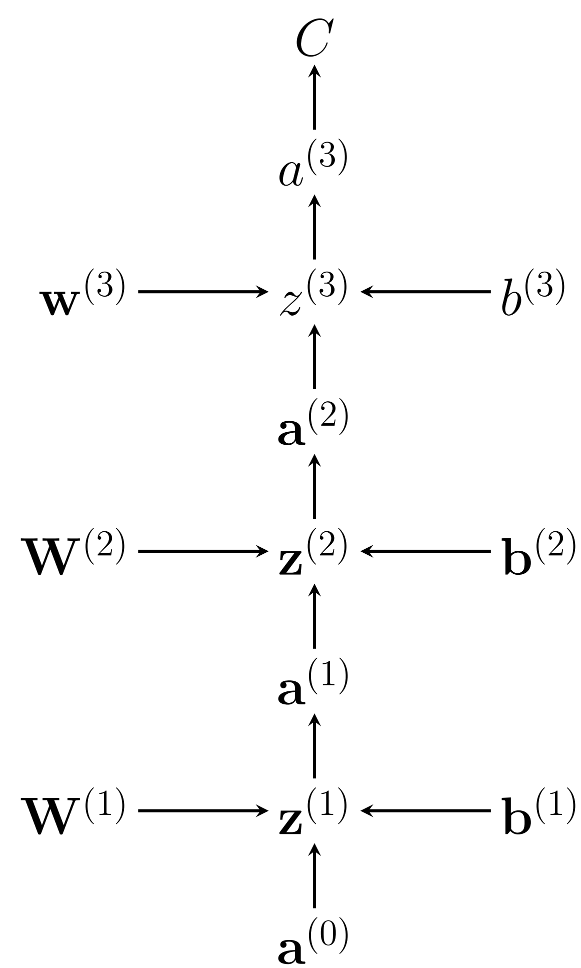
$$\frac{\partial C}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} = \frac{\partial C}{\partial a^{(3)}} \cdot \sigma^{(3)\prime}\left(z^{(3)}\right)$$

Step 2: Gradients for hidden layer 2

$$\frac{\partial C}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a_i^{(2)}} \cdot \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}} = \frac{\partial C}{\partial z^{(3)}} \cdot w_i^{(3)} \cdot \sigma^{(2)\prime}\left(z_i^{(2)}\right)$$

Step 3: Gradients for hidden layer 1

$$\frac{\partial C}{\partial z_j^{(1)}} = \left(\sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial a_j^{(1)}}\right) \cdot \frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} = \left(\sum_{i=1}^{D^{(2)}} \frac{\partial C}{\partial z_i^{(2)}} \cdot W_{ij}^{(2)}\right) \cdot \sigma^{(1)\prime}\left(z_j^{(1)}\right)$$



# The Backpropagation Algorithm (Almost Complete Version)

1. Compute gradients for output layer

$$\frac{\partial C}{\partial z^{(L)}} = \frac{\partial C}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} = \frac{\partial C}{\partial a^{(L)}} \cdot \sigma^{(L)\prime}\left(z^{(L)}\right)$$

} base case

2. Compute gradients for each hidden layer recursively

$$\frac{\partial C}{\partial z_j^{(m)}} = \left(\sum_{i=1}^{D^{(m+1)}} \frac{\partial C}{\partial z_i^{(m+1)}} \cdot \frac{\partial z_i^{(m+1)}}{\partial a_j^{(m)}}\right) \cdot \frac{\partial a_j^{(m)}}{\partial z_j^{(m)}} = \left(\sum_{i=1}^{D^{(m+1)}} \frac{\partial C}{\partial z_i^{(m+1)}} \cdot W_{ij}^{(m+1)}\right) \cdot \sigma^{(m)\prime}\left(z_j^{(m)}\right),$$

$$j = 1, \dots, D^{(m)}$$

} recursive case

3. Compute gradients for the weights

This is the backward pass

$$\frac{\partial C}{\partial W_{ij}^{(m)}} = \frac{\partial C}{\partial z_i^{(m)}} \cdot \frac{\partial z_i^{(m)}}{\partial W_{ij}^{(m)}} = \frac{\partial C}{\partial z_i^{(m)}} \cdot a_j^{(m-1)}, i = 1, \dots, D^{(m)}, j = 1, \dots, D^{(m-1)}$$

# Why The Forward Pass

UNIVERSITY OF TORONTO

DEFY GRAVITY

# Necessary Quantities for Backpropagation

What quantities do we need to carry out backpropagation?

- derivative of loss function: $\frac{\partial C}{\partial a^{(L)}}$
- derivative of non-linearity: $\sigma^{(m)'}\left(z_j^{(m)}\right)$
- weights: $W_{ij}^{(m)}$
- activations: $a_j^{(m)}$

output layer:

$$\frac{\partial C}{\partial z^{(L)}} = \frac{\partial C}{\partial a^{(L)}} \cdot \sigma^{(L)'}\left(z^{(L)}\right)$$

each hidden layer:

$$\frac{\partial C}{\partial z_j^{(m)}} = \left(\sum_{i=1}^{D^{(m+1)}} \frac{\partial C}{\partial z_i^{(m+1)}} \cdot W_{ij}^{(m+1)}\right) \cdot \sigma^{(m)'}\left(z_j^{(m)}\right)$$

weights:

$$\frac{\partial C}{\partial W_{ij}^{(m)}} = \frac{\partial C}{\partial z_i^{(m)}} \cdot a_j^{(m-1)}$$

# The Backpropagation Algorithm (Truly Complete Version)

## Forward Pass

For each $m = 1, \ldots, L$,

$$z_i^{(m)} = \sum_{j=1}^{D^{(m-1)}} W_{ij}^{(m)} a_j^{(m-1)} + b_i^{(m)}, i = 1, \ldots, D^{(m)}$$

$$a_i^{(m)} = \sigma^{(m)}\left(z_i^{(m)}\right), i = 1, \ldots, D^{(m)}$$

↳ we need these when calculating the derivatives w.r.t the weights

For output layer $L$,

$$C = C(a^L, t)$$

## Backward Pass

For output layer $L$

$$\frac{\partial C}{\partial z^{(L)}} = \frac{\partial C}{\partial a^{(L)}} \cdot \sigma^{(L)\prime}\left(z^{(L)}\right)$$

For each $m = 1, \ldots, L$,

$$\frac{\partial C}{\partial z_j^{(m)}} = \left(\sum_{i=1}^{D^{(m+1)}} \frac{\partial C}{\partial z_i^{(m+1)}} \cdot W_{ij}^{(m+1)}\right) \cdot \sigma^{(m)\prime}\left(z_j^{(m)}\right),$$

$$j = 1, \ldots, D^{(m)}$$

$$\frac{\partial C}{\partial W_{ij}^{(m)}} = \frac{\partial C}{\partial z_i^{(m)}} \cdot a_j^{(m-1)},$$

$$i = 1, \ldots, D^{(m)}, j = 1, \ldots, D^{(m-1)}$$

# Backpropagation Summary

- Efficiently compute gradients for many weights in a neural network
- The forward pass computes and stores activations.
- The backward pass
  - computes all the derivatives in one pass
  - reuses intermediate values (dynamic programming)
  - computes the derivatives w.r.t. pre-activations recursively via the chain rule.