Department of
Linguistics and Translation
香港城市大學
City University of Hong Kong

Computational Linguistics
LT3233

Jixing Li

Lecture 9: Backpropagation and Computational Graph

Slides adapted from Dan Jurafsky

Lecture plan

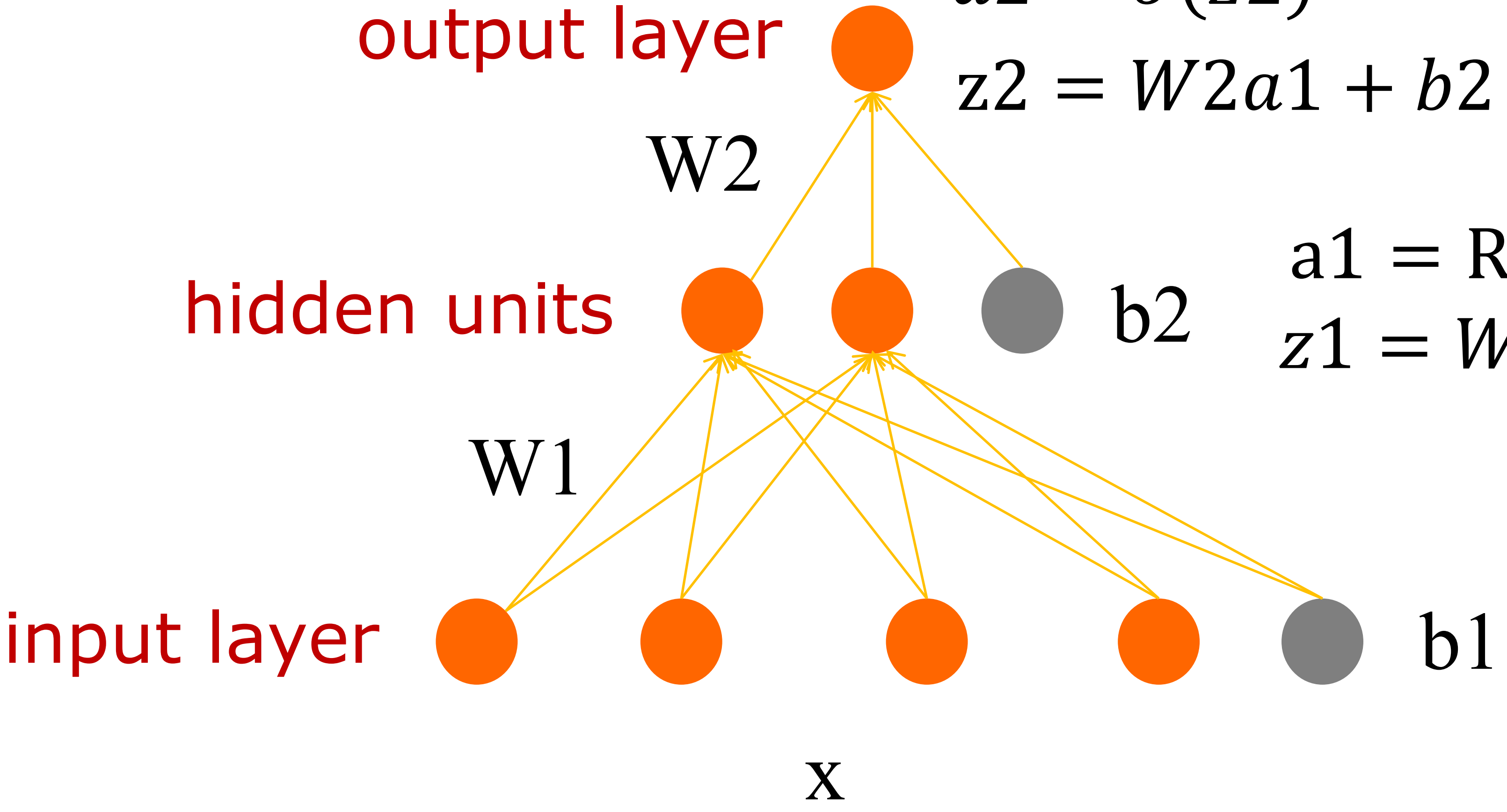
- Overview of feedforward neural networks
- Backpropagation and computational graphs
- Implementing feedforward neural networks from scratch
- Short break (15 mins)
- Hands-on exercises

Ask for help if you need it:

- office hour: 3-5 pm Tuesdays at LI-5459
- Zoom meetings: by schedule

Feedforward neural networks

Two-layer network with scalar output



$$a2 = \sigma(z2)$$
$$z2 = W2a1 + b2$$
$$a1 = \mathrm{ReLu}(z1)$$
$$z1 = W1x + b1$$

Example

chinese_name	major	gender	n1_male	n2_male	n1_uniqueness	n2_uniqueness
林加敏	LLA	F	0.442	-0.562	2.795	2.087

x = [0.442,-0.562,2.795,2.087]

W1 = [[1,3,2,4],
[2,1,4,3]]

W2 = [-1,2]

b1 = [1,-1], b2 = 2

z1 = W1x+b1

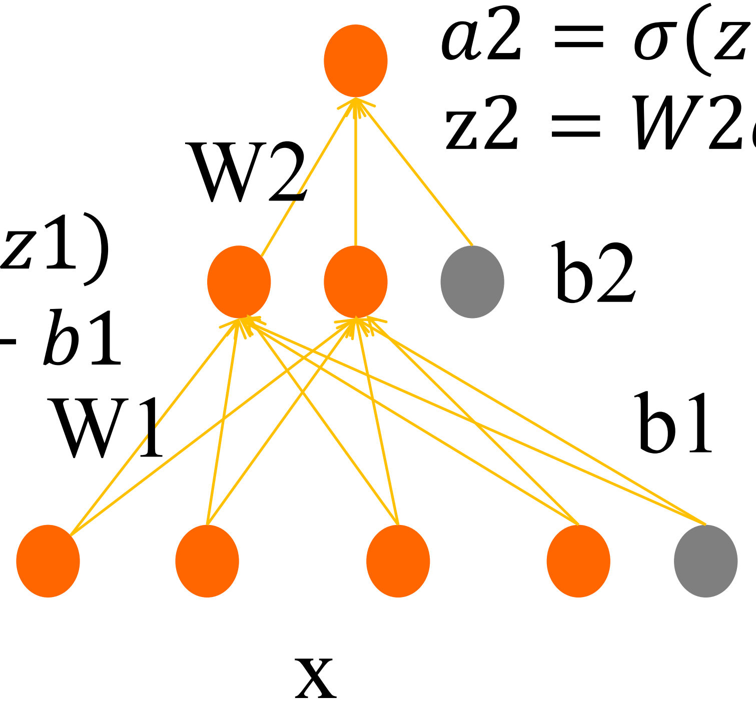
a1 = ReLU(z1) = max(z1,0)

z2 = W2a1+b2

a2 = σ(z2)=1/(1+e^{-z2})

$a1 = \mathrm{ReLu}(z1)$

$z1 = W1x + b1$



$a2 = \sigma(z2)$

$z2 = W2a1 + b2$

Example

x = [0.442,-0.562,2.795,2.087]
W1 = [[1,3,2,4],
[2,1,4,3]]
W2 = [-1,2]
b1 = [1,-1], b2 = 2
z1 = W1x+b1
a1 = ReLU(z1) = max(z1,0)
z2 = W2a1+b2
$a2 = \sigma(z2) = 1/(1+e^{-z2})$

$$\underset{W1}{\begin{bmatrix} 1,3,2,4 \\ 2,1,4,3 \end{bmatrix}} \text{ x } \underset{x}{\begin{bmatrix} 0.442 \\ -0.562 \\ 2.795 \\ 2.087 \end{bmatrix}} + \underset{b1}{\begin{bmatrix} 1 \\ -1 \end{bmatrix}}$$

$$z1 = \begin{bmatrix} 1x0.442 + 3x\text{-}0.562 + 2x2.795 + 4x2.087 \\ 2x0.442 + 1x\text{-}0.562 + 4x2.795 + 3x2.087 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$z1 = \begin{bmatrix} 12.694 \\ 17.763 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$z1 = \begin{bmatrix} 13.694 \\ 16.763 \end{bmatrix}$$

a1 = ReLU(z1) = max(z1,0) = z1

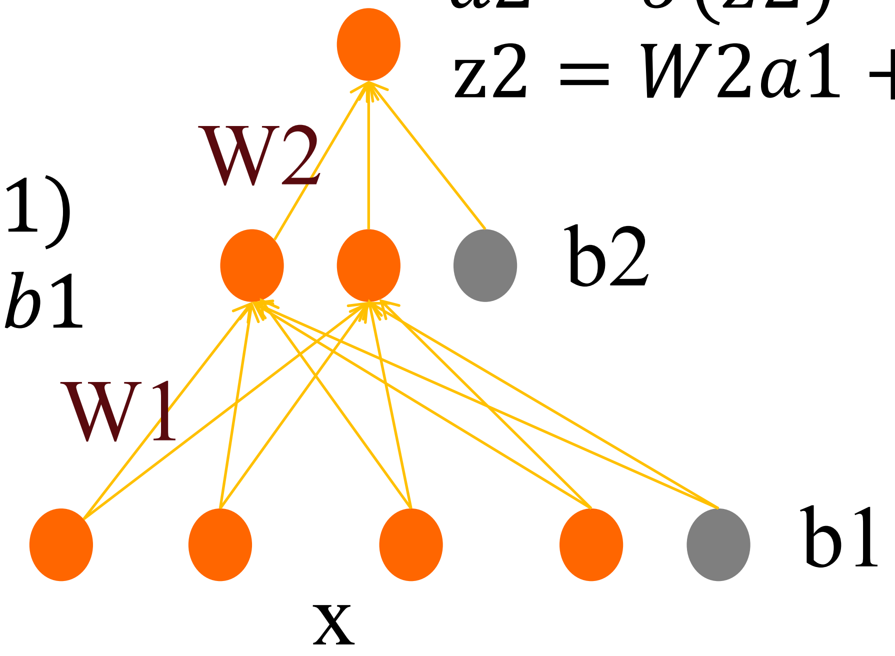
$$z2 = W2a1+b2 = [-1,2] \text{ x } \begin{bmatrix} 13.694 \\ 16.763 \end{bmatrix} + 2 = -1x13.694 + 2x16.763 + 2 = 21.832$$

$$a2 = 1/(1+e^{-z2}) = 1/(1+e^{-21.832}) = 0.99$$

Compute the parameters

$$a2 = \sigma(z2)$$
$$z2 = W2a1 + b2$$

$$a1 = \text{ReLu}(z1)$$
$$z1 = W1x + b1$$



W1 x b1

$$\begin{bmatrix} 1,3,2,4 \\ 2,1,4,3 \end{bmatrix} \text{x} \begin{bmatrix} 0.442 \\ -0.562 \\ 2.795 \\ 2.087 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

How to know the weights (W1,W2) and biases (b1,b2)?

→through error **backpropagation**

which relies on **computation graphs**

Gradient descent in logistic regression

[卓, 琳, Cheuk, Lam, LLA] → $x = [0.5, 0.7, 0.5, 0.6, 0.8]$, $y=1$

1. initialize w and b, set η

$w = [0, 0, 0, 0, 0]$, $b = 0$, $\eta = 0.1$

2. compute $\hat{y}$

$\hat{y} = \sigma(wx + b) = 0.5$

3. compute the gradients for w and b

$Gw = (\hat{y} - y)x = (0.5 - 1)[0.5, 0.7, 0.5, 0.6, 0.8] = [-0.25, -0.35, -0.25, -0.3, -0.4]$

$Gb = \hat{y} - y = 0.5 - 1 = -0.5$

4. update w and b

$w_{t+1} = w_t - \eta Gw = [0, 0, 0, 0, 0] - 0.1 * [-0.25, -0.35, -0.25, -0.3, -0.4]$

$= [0.025, 0.035, 0.025, 0.03, 0.04]$

$b_{t+1} = b_t - \eta Gb = 0 - 0.1 * (-0.5) = 0.05$

Backpropagation

$x = [0.442, -0.562, 2.795, 2.087]$, $y=1$

1. initialize $W1, W2$ and $b1, b2$, set η

$W1 = [[1,1,1,1],$
$[1,1,1,1]]$

$W2 = [1,1]$

$b1 = [1,1]$

$b2=[1]$

$\eta = 0.1$

2. forward propagation

$z1 = W1x + b1$

$a1 = ReLU(z1) = max(z1,0)$

$z2 = W2a1 + b2$

$a2 = \sigma(z2) = 1/(1 + e\text{-}z2)$

3. backpropagation

GW1

GW2

Gb1

Gb2

4. update $W1, W2$ and $b1, b2$

$W1_{t+1} = W1_t - \eta GW1$

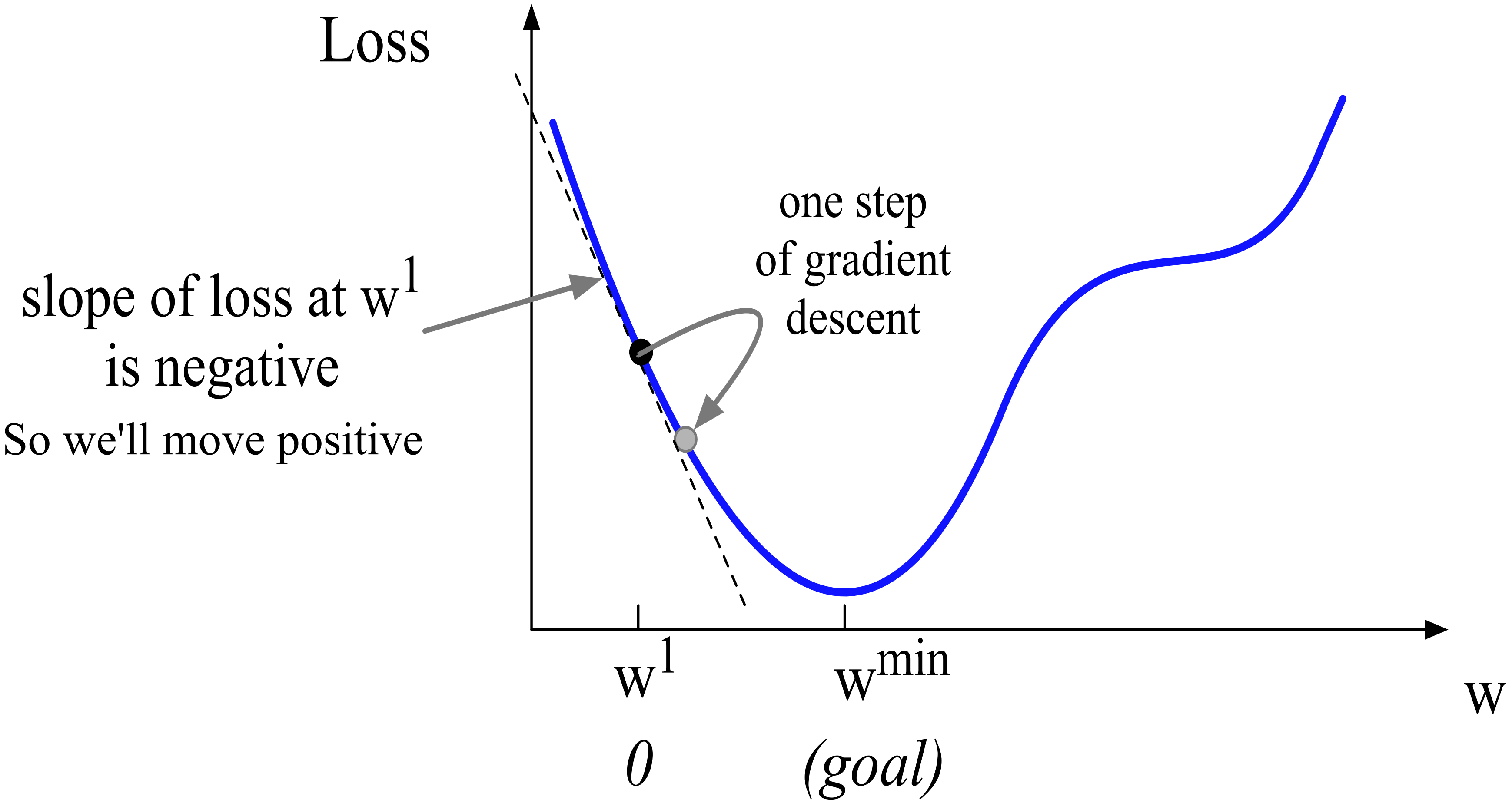
$W2_{t+1} = W2_t - \eta GW2$

$b1_{t+1} = b1_t - \eta Gb1$

$b2_{t+1} = b2_t - \eta Gb2$

Gradient descent (again)

Minimize loss: Given the current w, move w in the reverse direction from the slope of the function



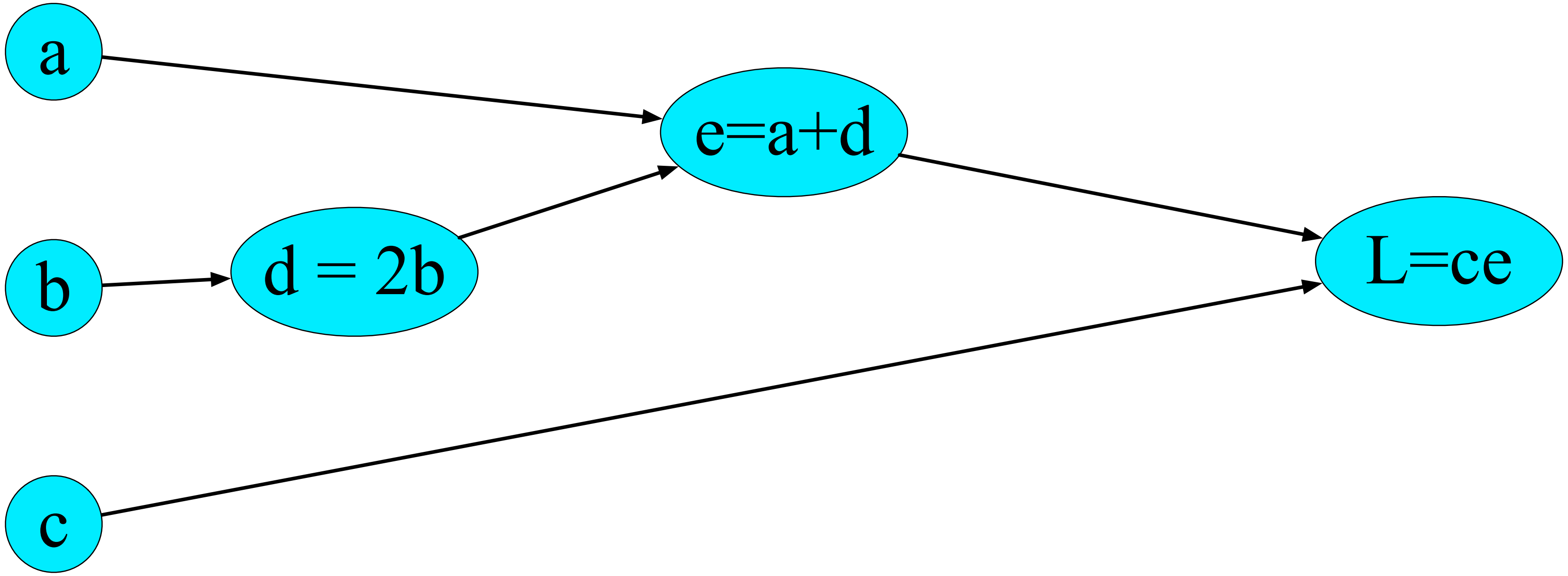
concept of derivative

Computation graph

A computation graph represents the process of computing a mathematical expression

$$L(a,b,c) = c(a+2b)$$

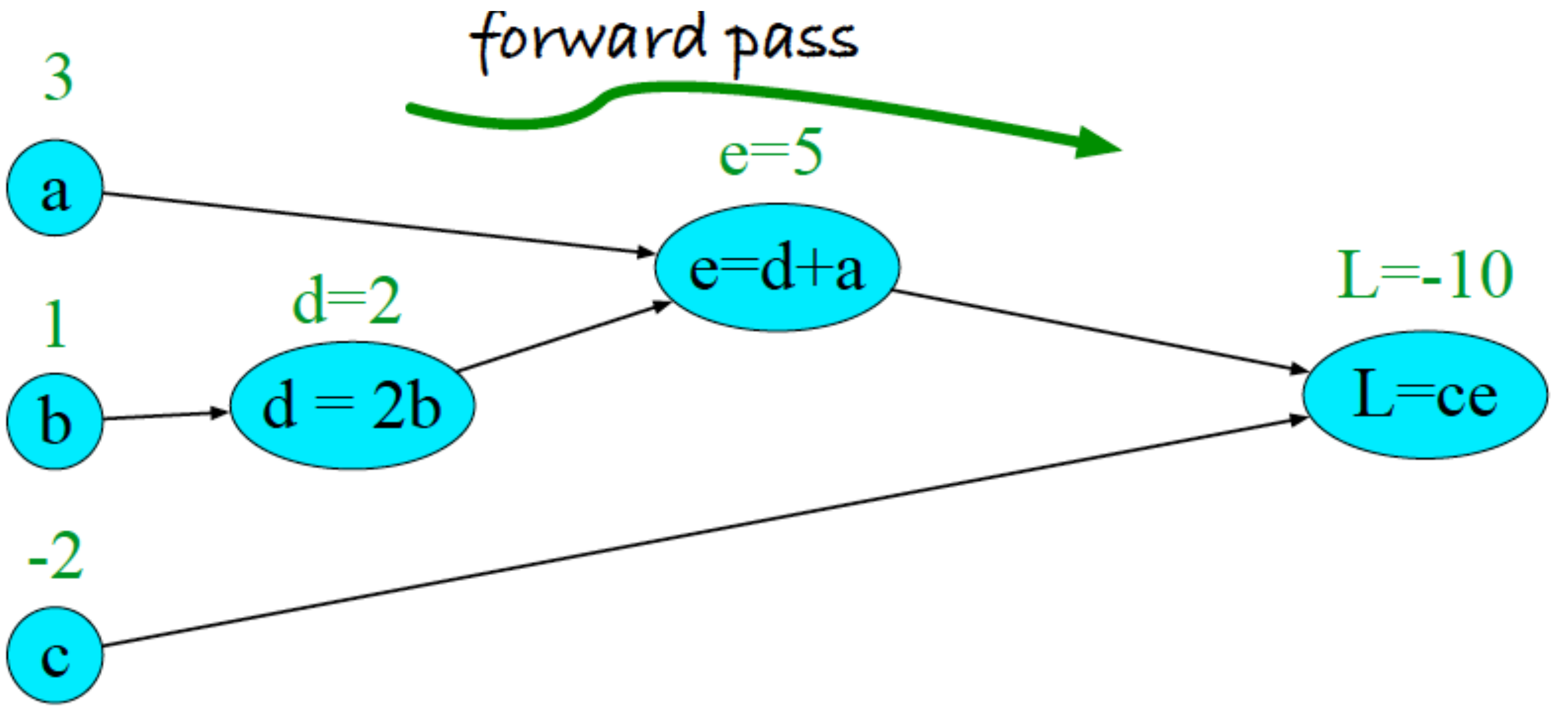
$$\begin{aligned} d &= 2*b \\ e &= a+d \\ L &= c*e \end{aligned}$$



Example

$$L(a,b,c) = c(a+2b)$$

$$\begin{aligned} d &= 2*b \\ e &= a+d \\ L &= c*e \end{aligned}$$



Example

$$L(a,b,c) = c(a+2b)$$

$$\begin{aligned} d &= 2*b \\ e &= a+d \\ L &= c*e \end{aligned}$$

We want: $\frac{\partial L}{\partial a}$, $\frac{\partial L}{\partial b}$, and $\frac{\partial L}{\partial c}$

The derivative $\frac{\partial L}{\partial a}$ tells us how much a small change in a affects L.

The chain rule

Computing the derivative of a composite function:

$$f(x) = u(v(x)) \qquad \frac{df}{dx} = \frac{du}{dv} \cdot \frac{dv}{dx}$$

$$f(x) = u(v(w(x))) \qquad \frac{df}{dx} = \frac{du}{dv} \cdot \frac{dv}{dw} \cdot \frac{dw}{dx}$$

Example

$$L(a,b,c) = c(a+2b)$$

$$\begin{aligned} d &= 2*b \\ e &= a+d \\ L &= c*e \end{aligned}$$

$$f(x) = \frac{1}{x} \qquad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \qquad \frac{df}{dx} = 1$$

$$f(x) = e^x \qquad \frac{df}{dx} = e^x$$

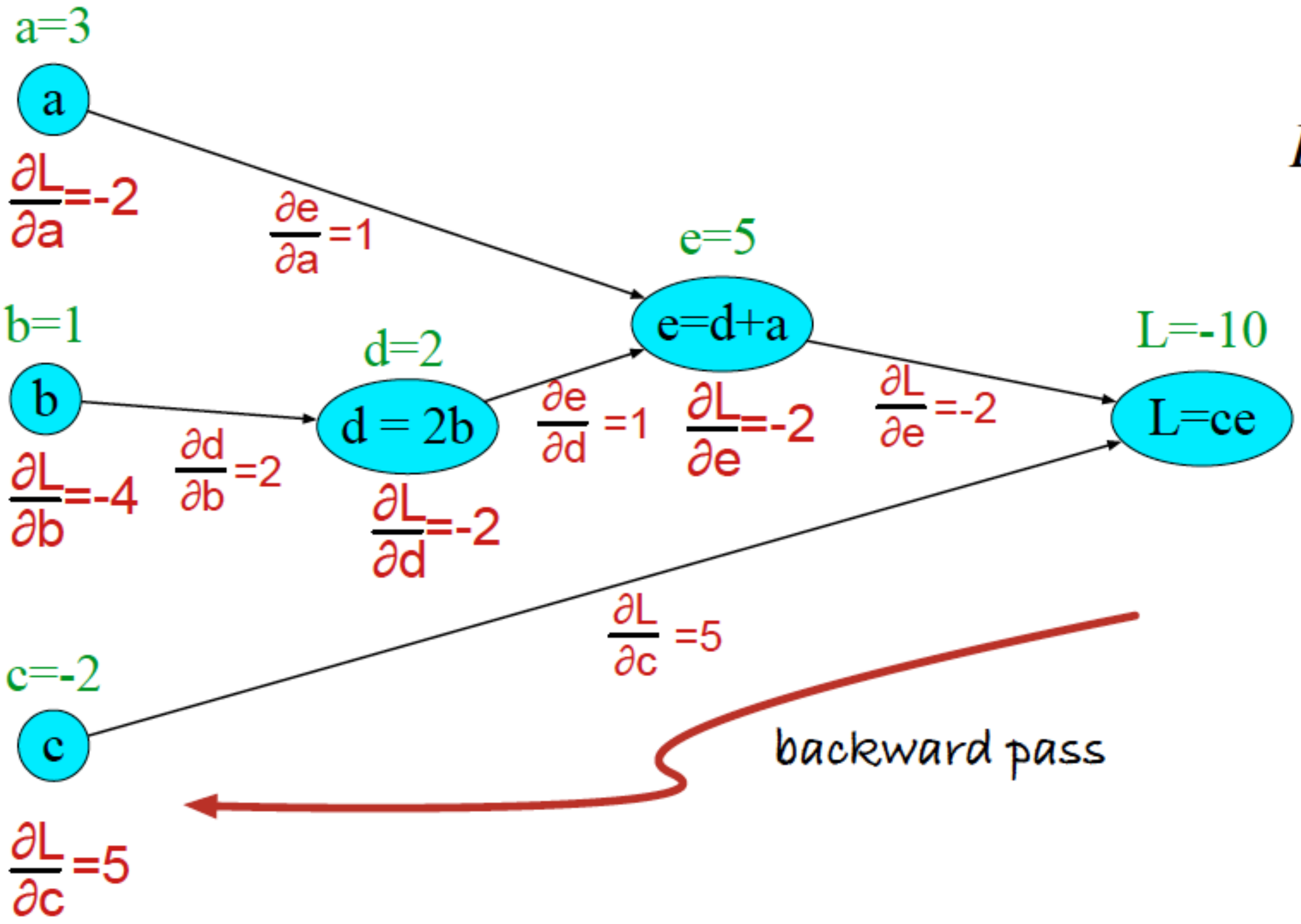
$$f_a(x) = ax \qquad \frac{df}{dx} = a$$

We want: $\frac{\partial L}{\partial a}$, $\frac{\partial L}{\partial b}$, and $\frac{\partial L}{\partial c}$

$$\begin{aligned} \frac{\partial L}{\partial c} &= e \\ \frac{\partial L}{\partial a} &= \frac{\partial L}{\partial e}\frac{\partial e}{\partial a} \\ \frac{\partial L}{\partial b} &= \frac{\partial L}{\partial e}\frac{\partial e}{\partial d}\frac{\partial d}{\partial b} \end{aligned}$$

$$\begin{aligned} L = ce &: \quad \frac{\partial L}{\partial e} = c, \frac{\partial L}{\partial c} = e \\ e = a+d &: \quad \frac{\partial e}{\partial a} = 1, \frac{\partial e}{\partial d} = 1 \\ d = 2b &: \quad \frac{\partial d}{\partial b} = 2 \end{aligned}$$

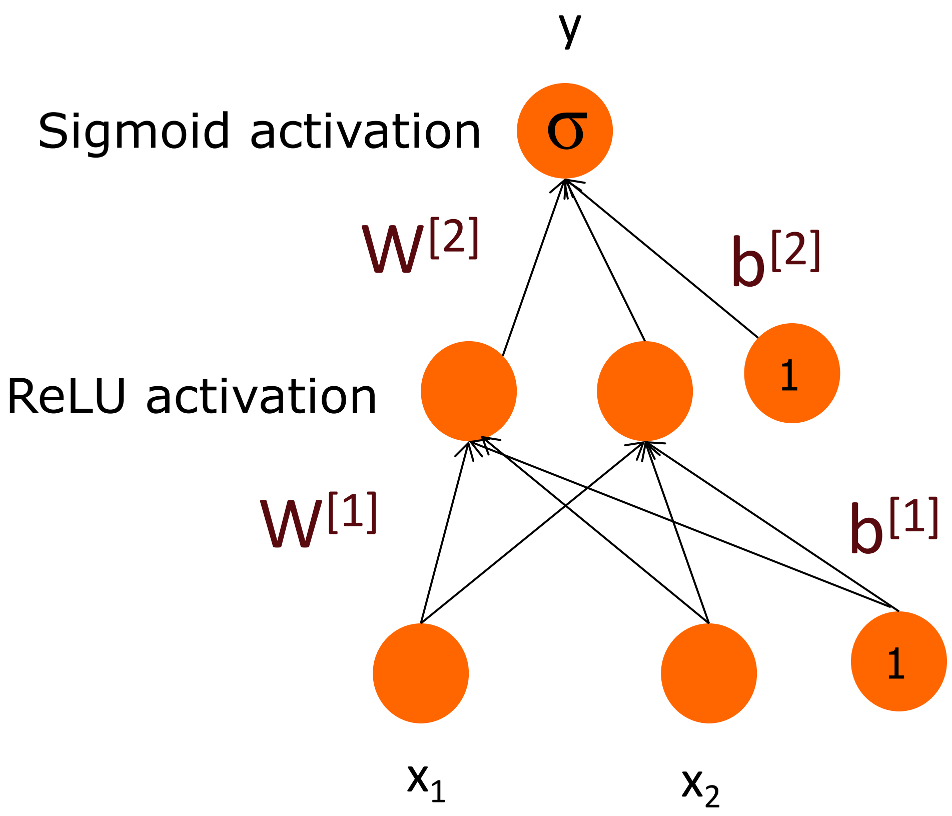
Example



$$L(a,b,c) = c(a+2b)$$

$$\begin{aligned} d &= 2*b \\ e &= a+d \\ L &= c*e \end{aligned}$$

Backprop on a two-layer network

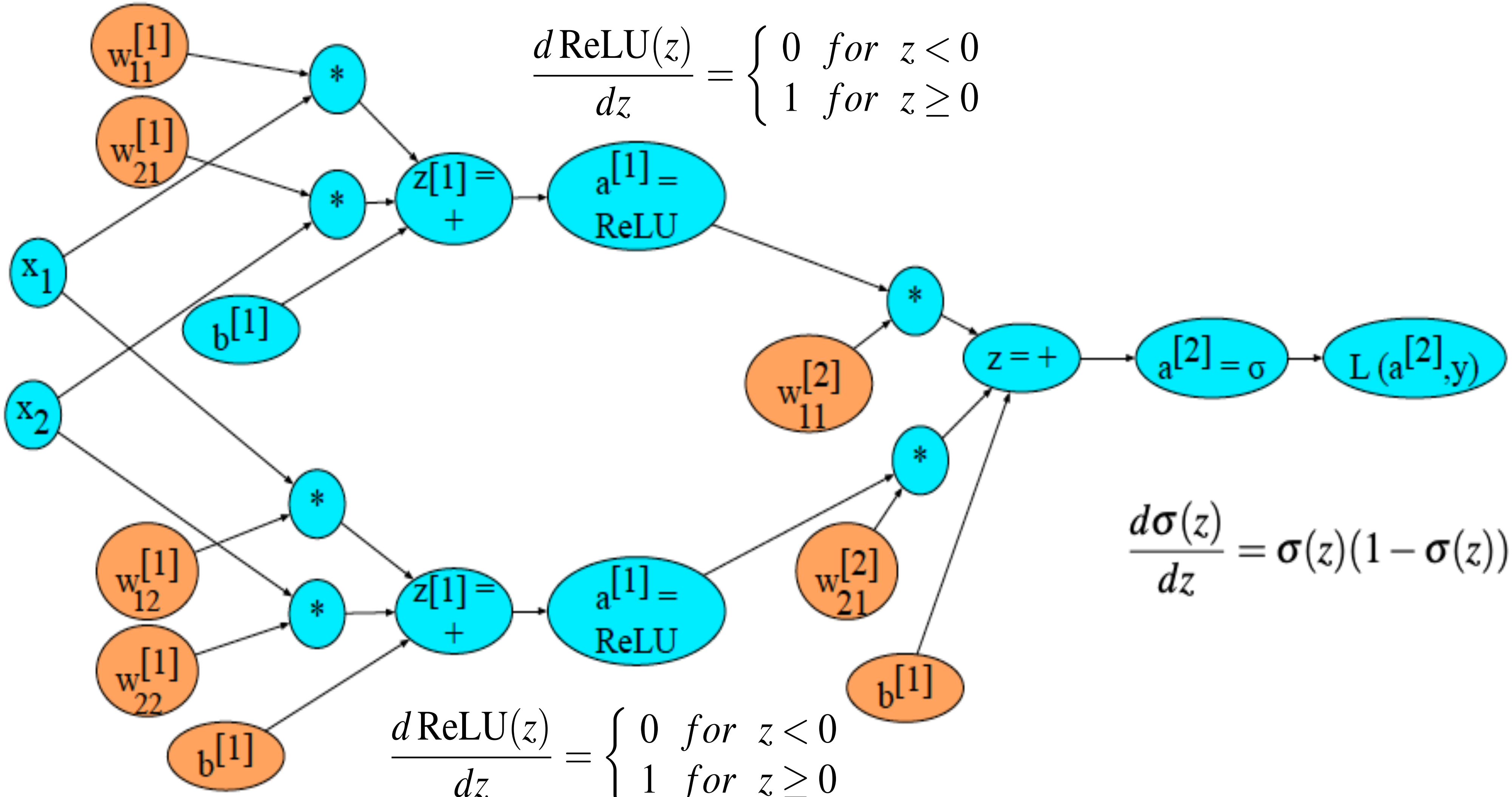


$$
\begin{aligned}
z^{[1]} &= W^{[1]}\mathbf{x} + b^{[1]} \\
a^{[1]} &= \mathrm{ReLU}(z^{[1]}) \\
z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\
a^{[2]} &= \sigma(z^{[2]}) \\
\hat{y} &= a^{[2]}
\end{aligned}
$$

$$\frac{d\,\mathrm{ReLU}(z)}{dz} = \begin{cases} 0 & for \quad z < 0 \\ 1 & for \quad z \geq 0 \end{cases}$$

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z))$$

Backprop on a two-layer network



$$\frac{d\,\mathrm{ReLU}(z)}{dz} = \begin{cases} 0 & for \quad z < 0 \\ 1 & for \quad z \geq 0 \end{cases}$$

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z))$$

$$\frac{d\,\mathrm{ReLU}(z)}{dz} = \begin{cases} 0 & for \quad z < 0 \\ 1 & for \quad z \geq 0 \end{cases}$$

Starting off the backward pass

$$L(\hat{y}, y) = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y}))$$

$$L(a, y) = -(y \log a + (1-y)\log(1-a))$$

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}$$

$$\begin{aligned} z^{[1]} &= W^{[1]}\mathbf{x} + b^{[1]} \\ a^{[1]} &= \mathrm{ReLU}(z^{[1]}) \\ z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\ a^{[2]} &= \sigma(z^{[2]}) \\ \hat{y} &= a^{[2]} \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial a} &= -\left(\left(y\frac{\partial \log(a)}{\partial a}\right) + (1-y)\,\frac{\partial \log(1-a)}{\partial a}\right) \\ &= -\left(\left(y\frac{1}{a}\right) + (1-y)\,\frac{1}{1-a}(-1)\right) = -\left(\frac{y}{a} + \frac{y-1}{1-a}\right) \end{aligned}$$

$$\frac{\partial a}{\partial z} = a(1-a)$$

$$\frac{\partial L}{\partial z} = -\left(\frac{y}{a} + \frac{y-1}{1-a}\right)a(1-a) = a - y$$

To do

- Optional reading: **SLP** Ch7.6
- Tutorial on backpropagation:

https://cs231n.github.io/optimization-2/

- Gentle introduction on derivatives:

https://www.khanacademy.org/math/ap-calculus-ab/ab-differentiation-1-new