

CS 66: Machine Learning

Prof. Sara Mathieson

Spring 2019



Outline for April 3

- Continue backpropagation
 - Neural network architectures
 - Choice of non-linearity (activation function)
 - Choice of loss function
 - Choice of weight initialization
- Lab 6 due Friday (check in today during lab)
 - Lab 7 released Friday (fill out partner forms!)
 - Alternative presentation time: May 13, 1-3pm

Turing award goes to neural network research

- LeCun, Hinton, Bengio win Turing award for their work on neural networks

<https://www.nytimes.com/2019/03/27/technology/turing-award-ai.html>



From left, Facebook, via Associated Press; Aaron Vincent Elkaim for The New York Times; Chad Buchanan/Getty Images

Big picture for today

- Neural networks can approximate any function!

Big picture for today

- Neural networks can approximate any function!
- For our purposes in ML, we want to use them to approximate a function from our inputs to our outputs

Big picture for today

- Neural networks can approximate any function!
- For our purposes in ML, we want to use them to approximate a function from our inputs to our outputs
- We will train our network by asking it to minimize the loss between it's output and the true output

Big picture for today

- Neural networks can approximate any function!
- For our purposes in ML, we want to use them to approximate a function from our inputs to our outputs
- We will train our network by asking it to minimize the loss between it's output and the true output
- We will use backpropagation to minimize loss

Outline for April 3

- Continue backpropagation
- Neural network architectures
- Choice of non-linearity (activation function)
- Choice of loss function
- Choice of weight initialization

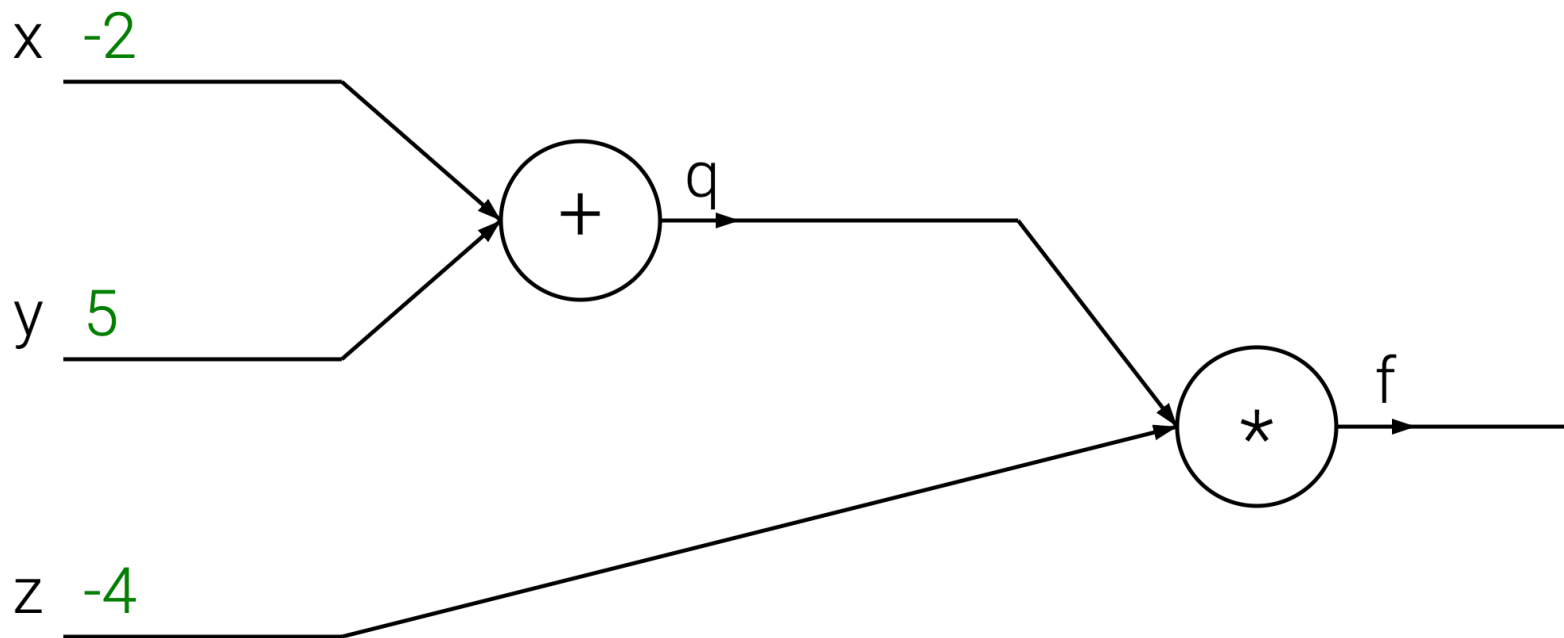
Following Stanford CNN course reading:

<http://cs231n.github.io/optimization-2/>

<http://cs231n.github.io/neural-networks-1/>

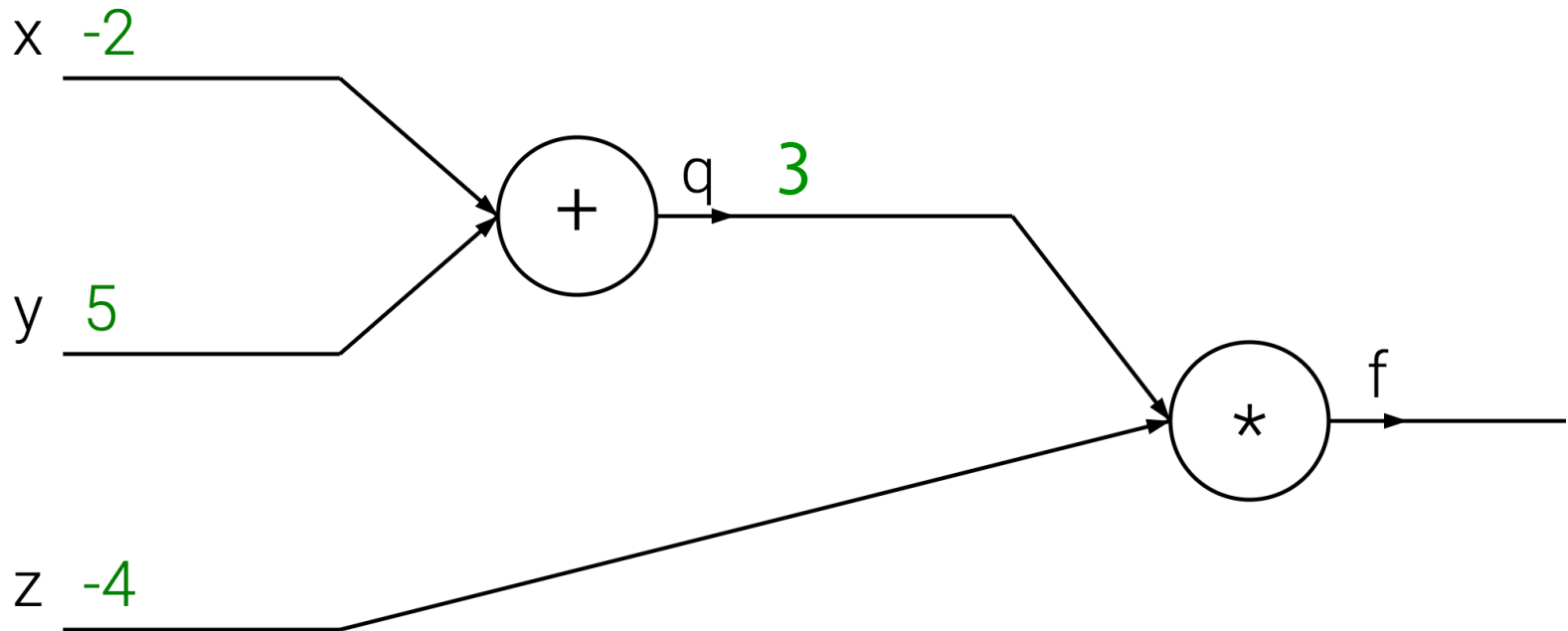
Backpropagation: Example 1

Forward pass: compute values



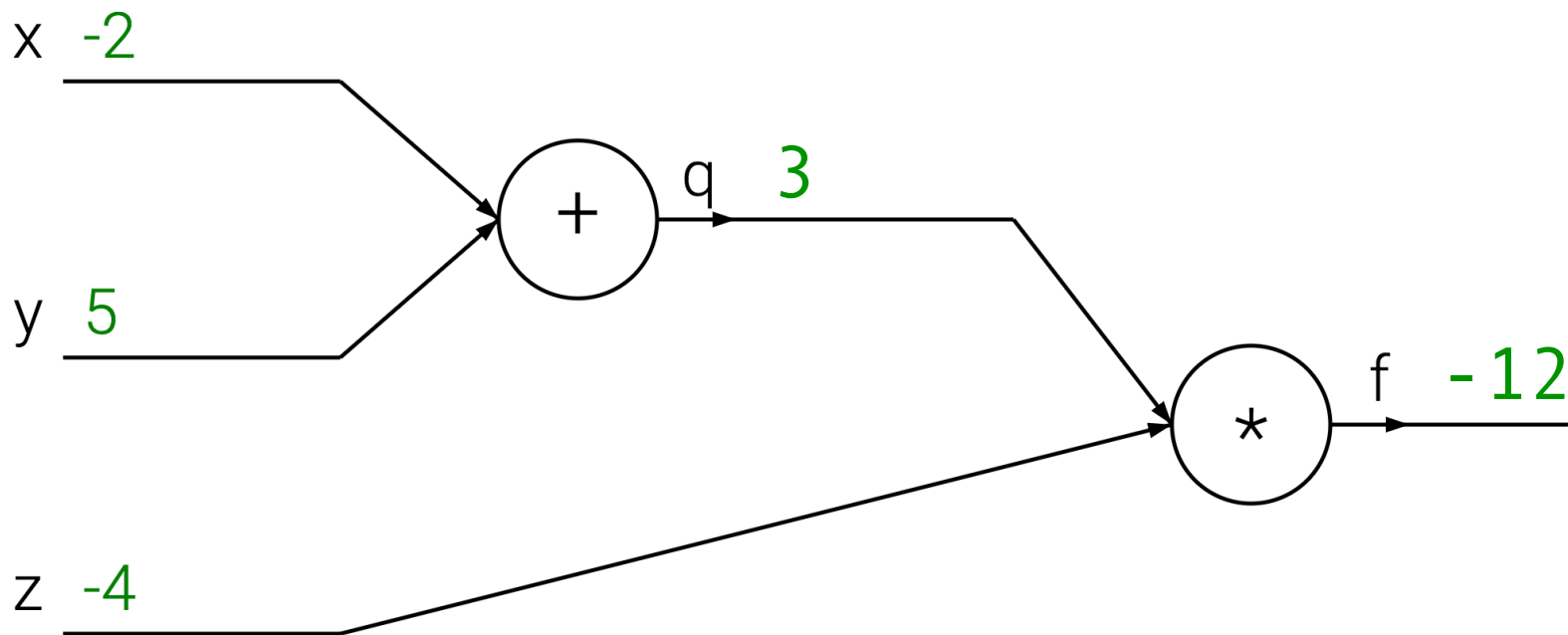
Backpropagation: Example 1

Forward pass: compute values



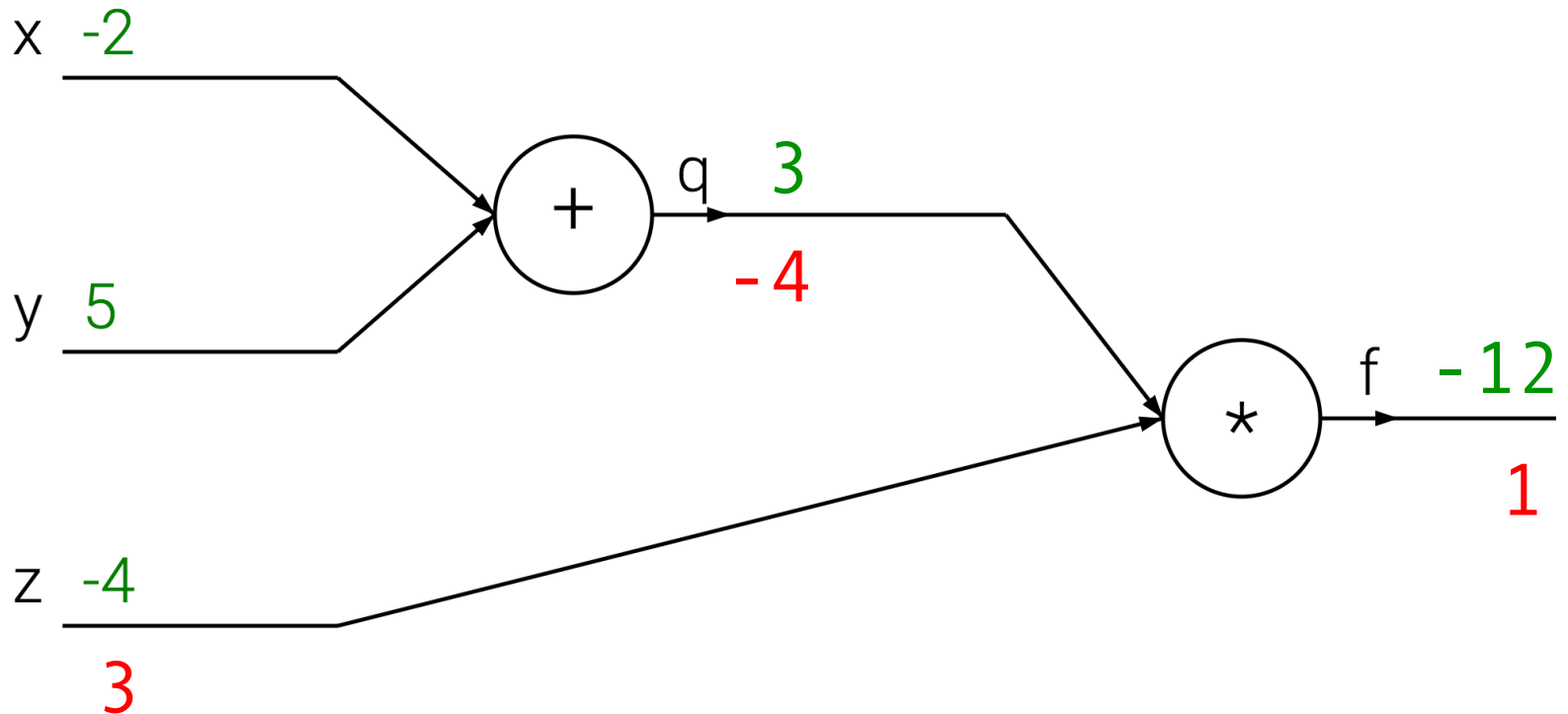
Backpropagation: Example 1

Forward pass: compute values



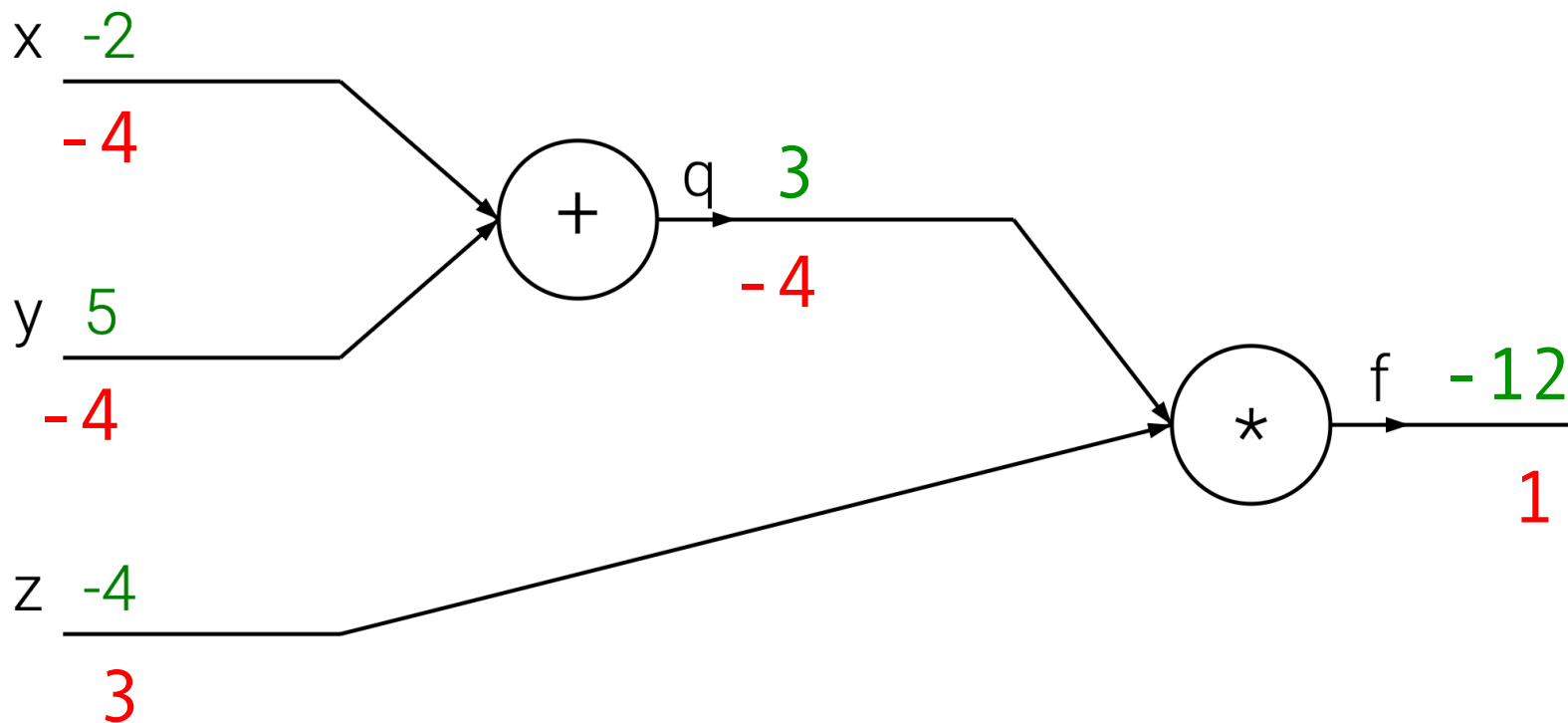
Backpropagation: Example 1

Backward pass: compute local gradients



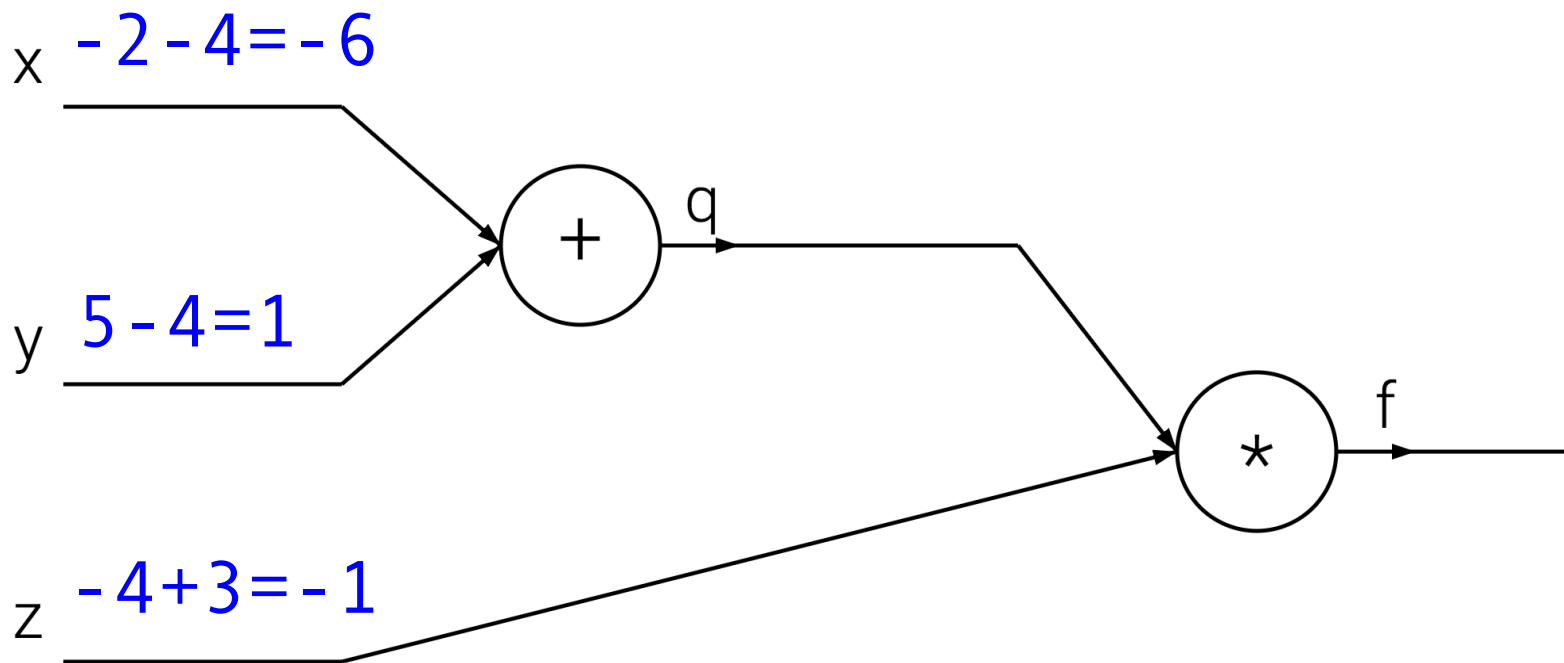
Backpropagation: Example 1

Backward pass: compute local gradients



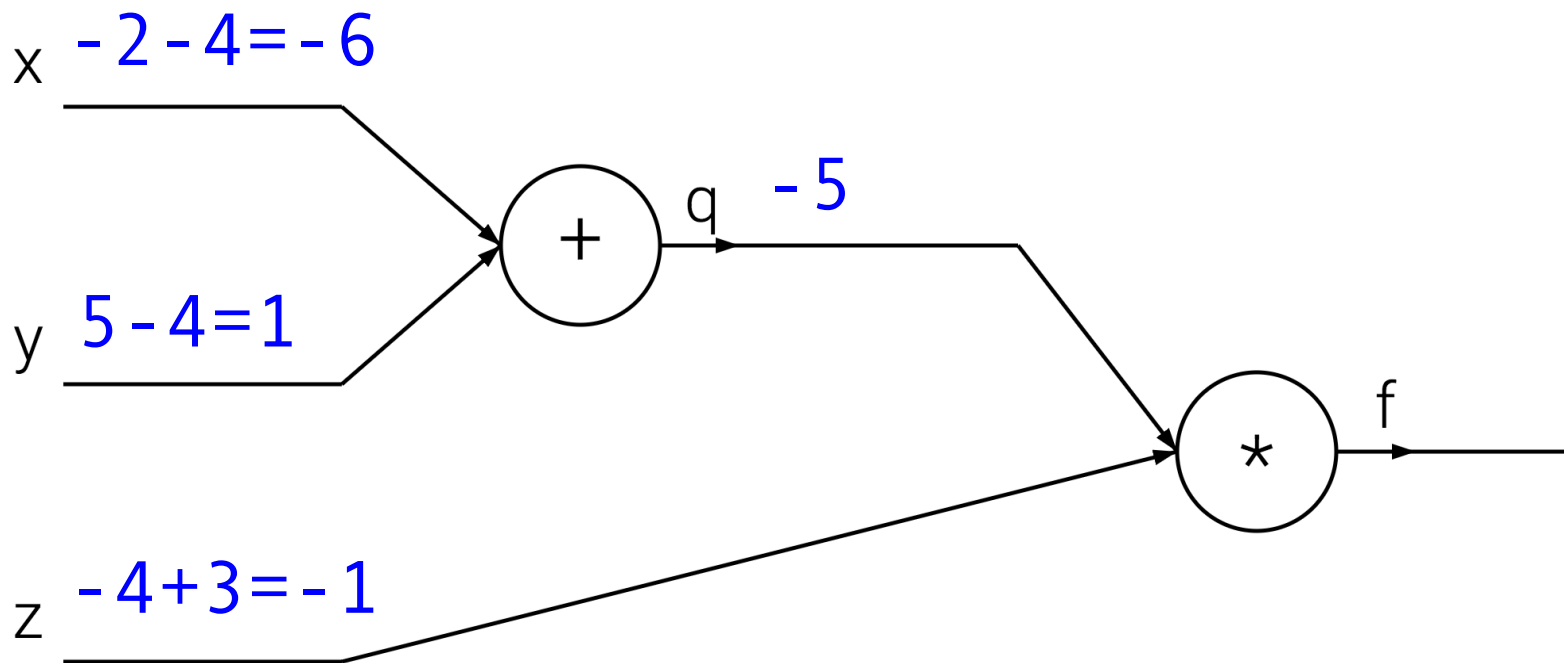
Backpropagation: Example 1

Now if we wanted to maximize f , move in direction of gradient



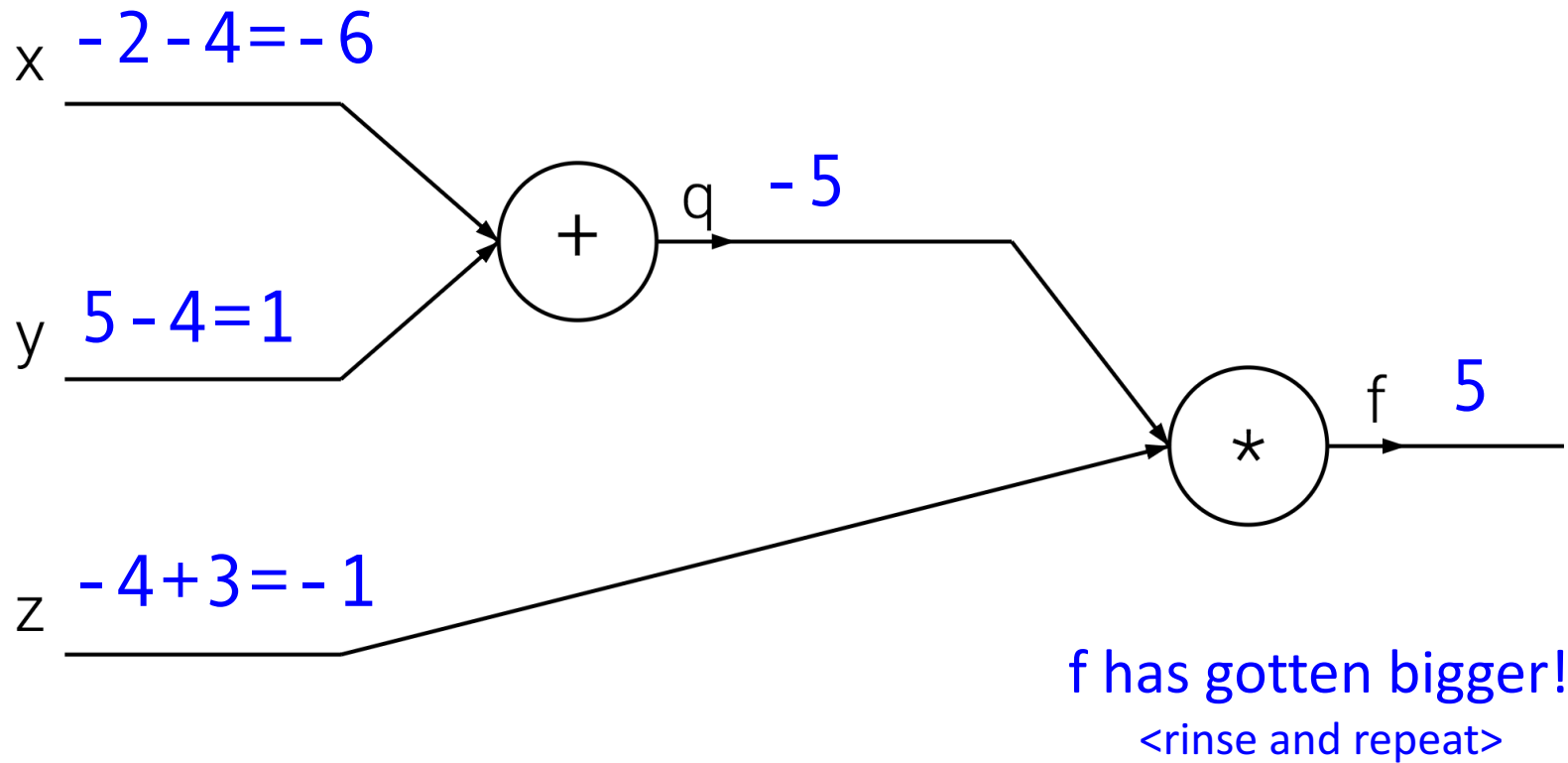
Backpropagation: Example 1

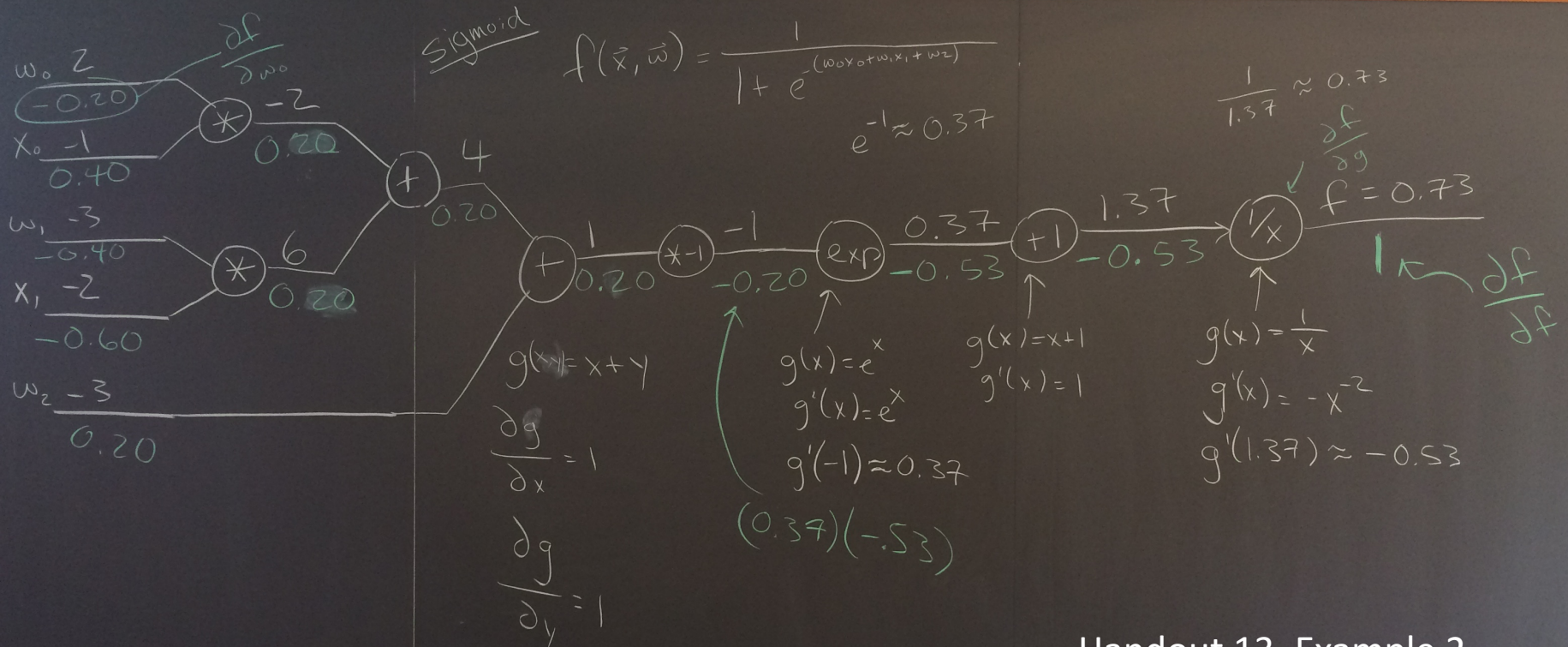
Now if we wanted to maximize f , move in direction of gradient



Backpropagation: Example 1

Now if we wanted to maximize f , move in direction of gradient





Handout 13, Example 2

$$g(\omega_0, x_0) = \omega_0 x_0$$

$$\frac{\partial g}{\partial \omega_0} = x_0$$

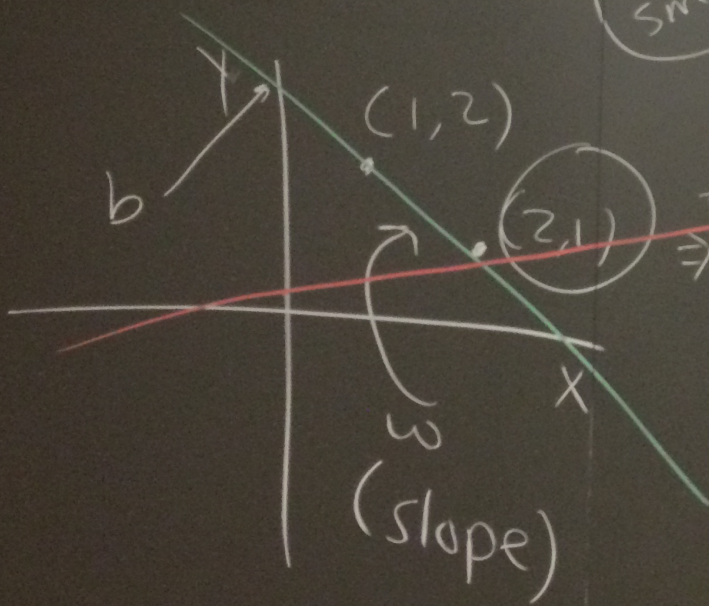
$$\frac{\partial g}{\partial x_0} = \omega_0$$

Notes

- In the previous example, we were allowing all variables to change (w_0, x_0, w_1, x_1, w_2)
- In the next (more realistic) example, x & y are fixed and we are allowing the weights (w & b) to change

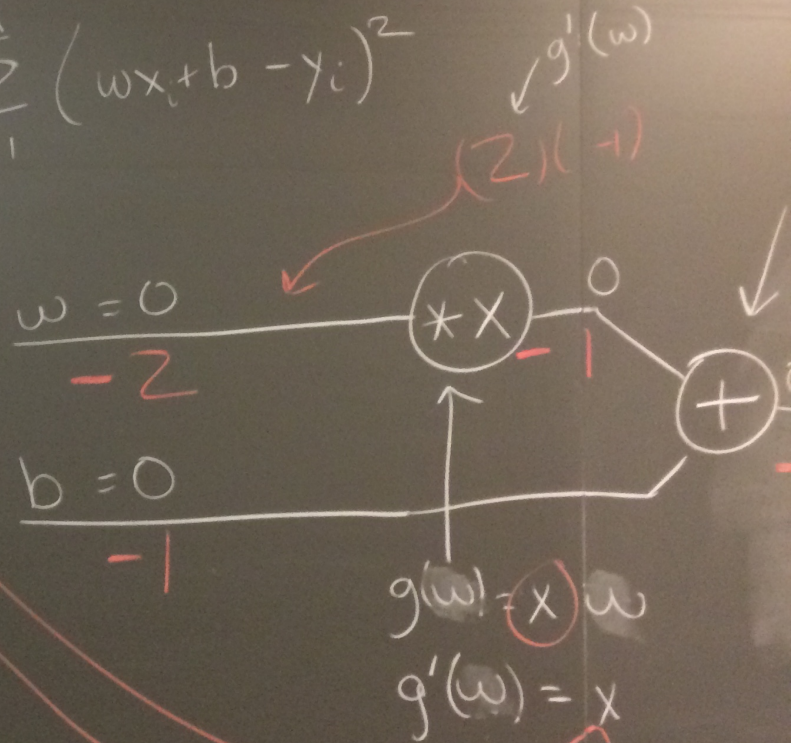
Linear Regression

Example



want small!

$$J(w, b) = \frac{1}{2} \sum_{i=1}^n (wx_i + b - y_i)^2$$



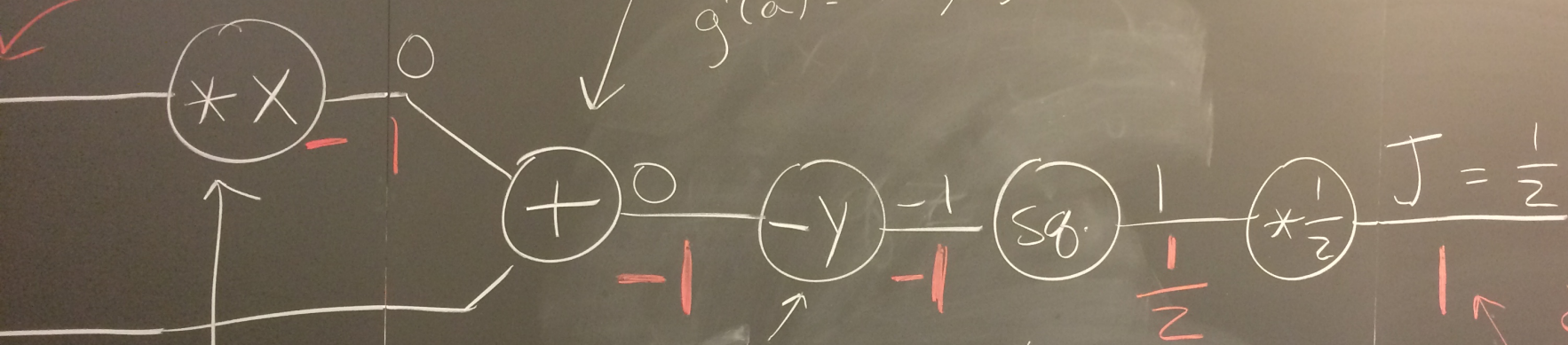
$$y_i)^2$$

$$g'(w)$$

$$(2)(-1)$$

$$g(a, b) = a + b$$

$$g'(a) = 1, g'(b) = 1$$



$$g(w) = xw$$

$$g'(w) = x$$

$$g(x) = x - y$$

$$g'(x) = 1$$

$$g(x) = x^2$$

$$g'(x) = 2x$$

$$\left(\frac{1}{2}\right)g'(-1) \rightarrow -2$$

$$\frac{\partial J}{\partial J}$$

$$w \leftarrow w - \frac{\partial J}{\partial w} \cdot x \quad \alpha = 0.1$$

$$b \leftarrow b - \frac{\partial J}{\partial b} \cdot x$$

Handwritten annotations in red: A red arrow points from '0' to the first w in the first equation. A red arrow points from '0' to the first b in the second equation. A red arrow points from '-2' to the $\frac{\partial J}{\partial w}$ term in the first equation. A red arrow points from '-1' to the $\frac{\partial J}{\partial b}$ term in the second equation. The fractions $\frac{\partial J}{\partial w}$ and $\frac{\partial J}{\partial b}$ are crossed out with red diagonal lines.

$$w = 0.2$$

$$b = 0.1$$

Update weights using
gradient descent idea

Outline for April 3

- Continue backpropagation
- Neural network architectures
- Choice of non-linearity (activation function)
- Choice of loss function
- Choice of weight initialization

NEXT TIME!

Following Stanford CNN course reading:
<http://cs231n.github.io/optimization-2/>
<http://cs231n.github.io/neural-networks-1/>