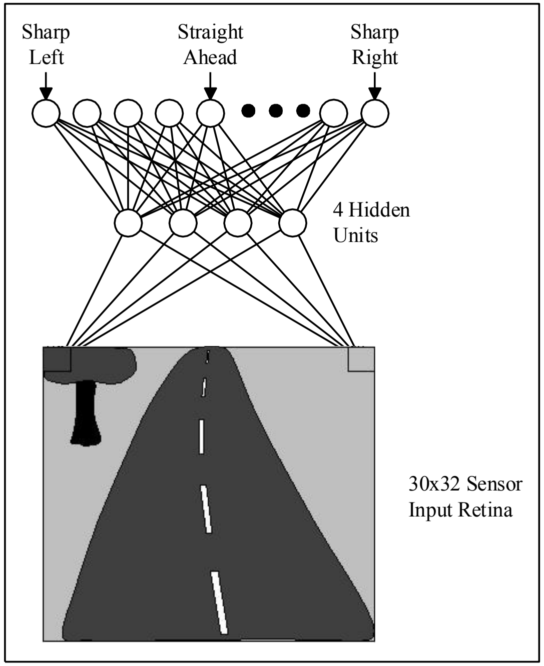
SGD and Deep Learning for Classification

UCSB CS293S, 2022, T. Yang

Motivation and Table of Content

- What we have learned so far for ranking and classification
  - Decision trees: entropy-based, or regression
  - Ensembles, boosting, and bagging. Random forests
- Focus of this slide set
  - Stochastic gradient descent (SGD) for general optimization
  - Derive weights for minimizing a loss function in a large network-based classification
  - Example of neural nets and optimization
- Why?
  - Successful in neural classification tasks for image and audio processing with machine learning
  - Effective for text oriented document classification and ranking



1

Partial Derivatives and Gradient

Single-variable functions

Notation for the Derivative

$$f'(x), \quad y', \quad \frac{dy}{dx} \quad \Bigg\} \quad \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Multi-variable functions

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Gradient

Scalar-valued multivariable function

$$\nabla f(x_0, y_0, \dots) = \begin{bmatrix} \frac{\partial f}{\partial x}(x_0, y_0, \dots) \\ \frac{\partial f}{\partial y}(x_0, y_0, \dots) \\ \vdots \end{bmatrix}$$

∇f takes the same type of inputs as f

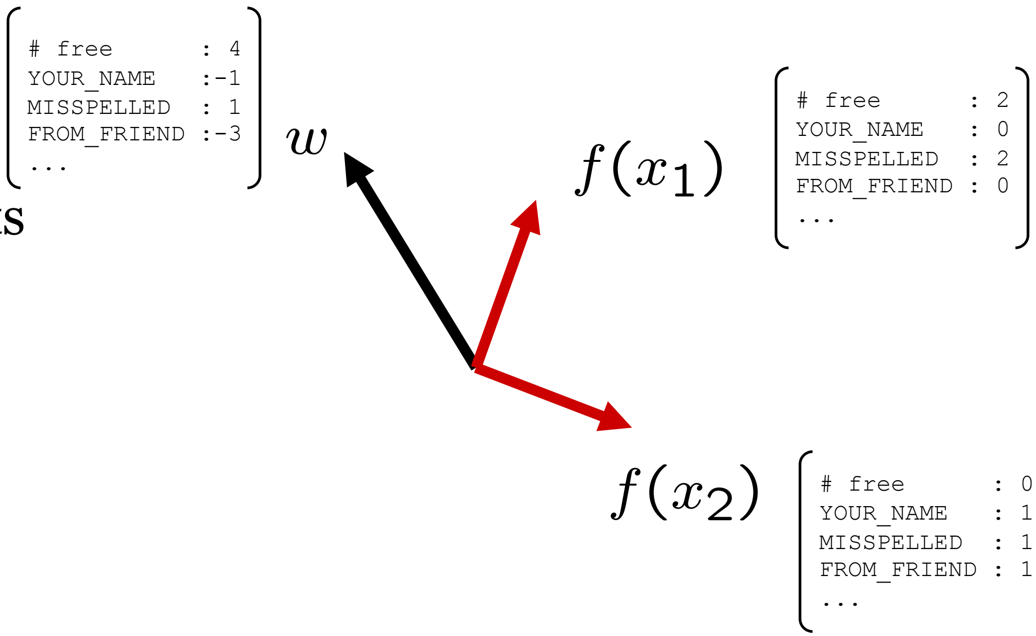
Notation for gradient, called "nabla".

∇f outputs a vector with all possible partial derivatives of f.

SGD training for Binary Classifier

Figure out the weight vector from training instances

- Start with weights = 0
- For each training instance:
  - Classify with current weights
  - f (x) is feature vector of x



$$y = \begin{cases} +1 & \text{if } \; w \cdot f(x) \geq 0 \\ -1 & \text{if } \; w \cdot f(x) < 0 \end{cases}$$

  - If correct (i.e., predicted y=target y*), no change!
  - If wrong: adjust the weight vector by adding or subtracting the feature vector. Subtract if y* is -1.

$$w = w + y^* \cdot f$$

Why?

Optimization Problem for Classification

Given training set $\{(x_1, y_1), \ldots (x_n, y_n)\}$

Given a loss function $\ell(h, y)$ (hinge loss, logistic,...)

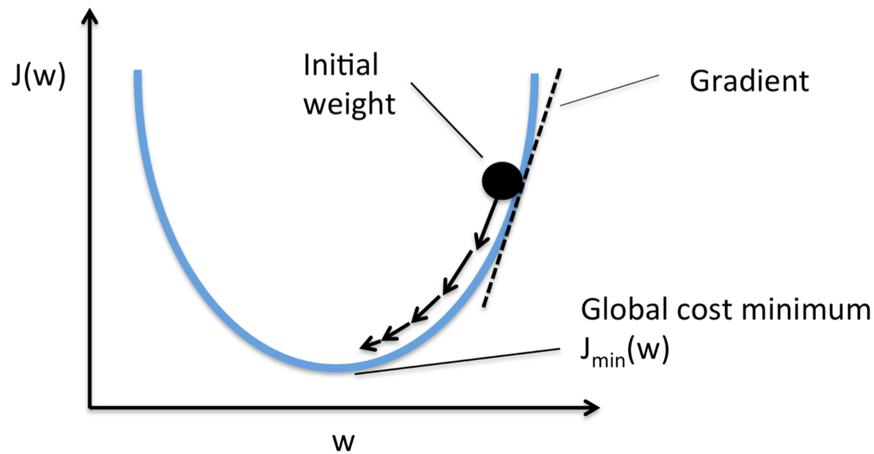
Find a prediction function $h(x; w)$ (linear, DNN,...)

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i; w), y_i)$$

- "y_i" is the classification label for a training instance
- "w" is the set of parameters to be found through training
- What does prediction function h() look like?
- How to find parameters involved in h() that minimize an objective function?

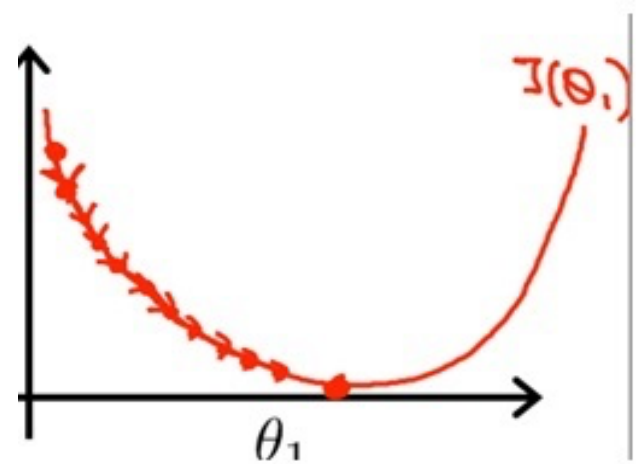
How to find parameters that minimize the loss function?

- How to find parameters that minimize a loss function J with parameter vector w?



Learning rate

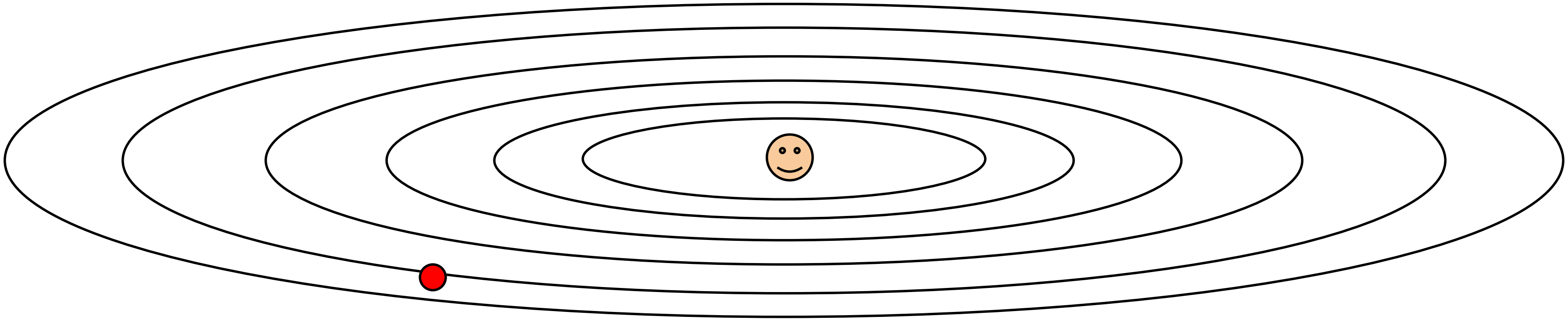
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$



- Gradient Descent Method (SGD) for Optimization
  - Start somewhere
  - Repeat: Take a step in the steepest descent direction

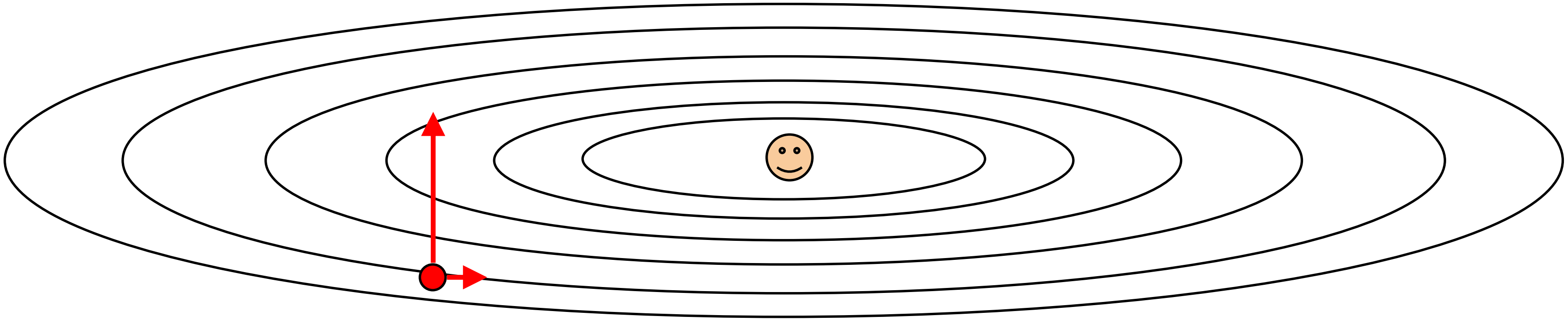
w is θ_1

Illustration of gradient descent to refine multiple parameters



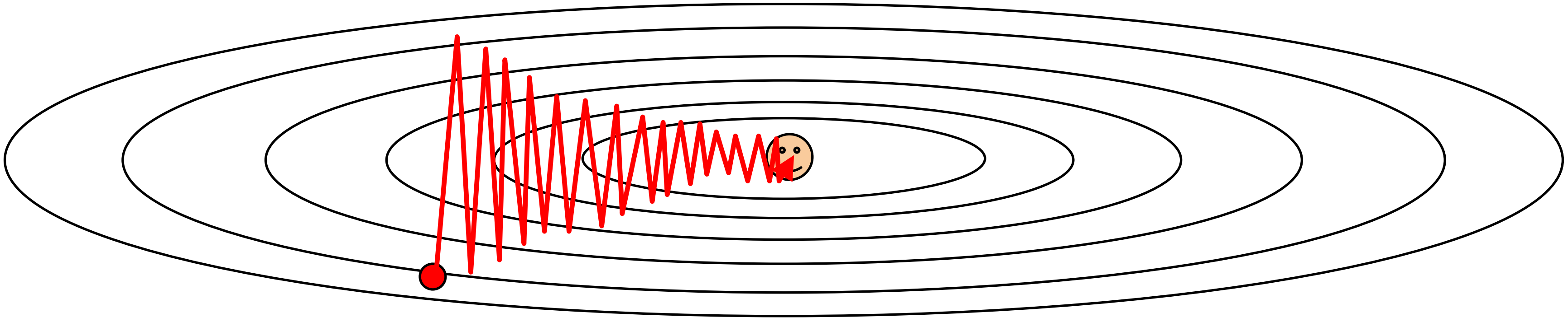
Q: What is the trajectory along which we converge towards the minimum with SGD?

Suppose loss function is steep vertically but shallow horizontally:



Q: What is the trajectory along which we converge towards the minimum with SGD?

Suppose loss function is steep vertically but shallow horizontally:



Q: What is the trajectory along which we converge towards the minimum with Gradient Descent? very slow progress along flat direction, jitter along steep one

8

Generally, Steepest Direction with n parameters

- Given loss function g and learning rate α
- Steepest Direction = direction of the gradient
- Parameter vector $w=(w_1, w_2,\ldots, w_n)$
- Gradient Descent: Update weight vector w by using a sequence of training instance i

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial w_1} \\ \frac{\partial g}{\partial w_2} \\ \cdots \\ \frac{\partial g}{\partial w_n} \end{bmatrix}$$

- Init:
- For i = 1, 2, …

$$w \leftarrow w - \alpha * \nabla g(w)$$

1. Stop after a fixed number of iterations.
2. Or when loss is close to a lower bound or has not improved much in a long tme.
3. Or when the validation error has not improved in a long time.

Start with Simple Binary Text Classifier

Also called perceptron

x $f(x)$ y

Result classification:
Positive, output +1
Negative, output -1

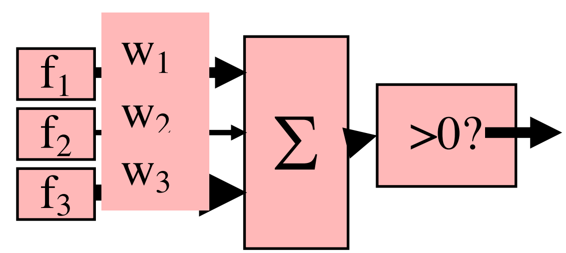
```
Hello,

Do you want free printr
cartriges?  Why pay more
when you can get them
ABSOLUTELY FREE!  Just
```

```
# free        : 2
YOUR_NAME     : 0
MISSPELLED    : 2
FROM_FRIEND   : 0
...
```

SPAM
or
+

```
PIXEL-7,12   : 1
PIXEL-7,13   : 0
...
NUM_LOOPS    : 1
...
```

"2"



$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

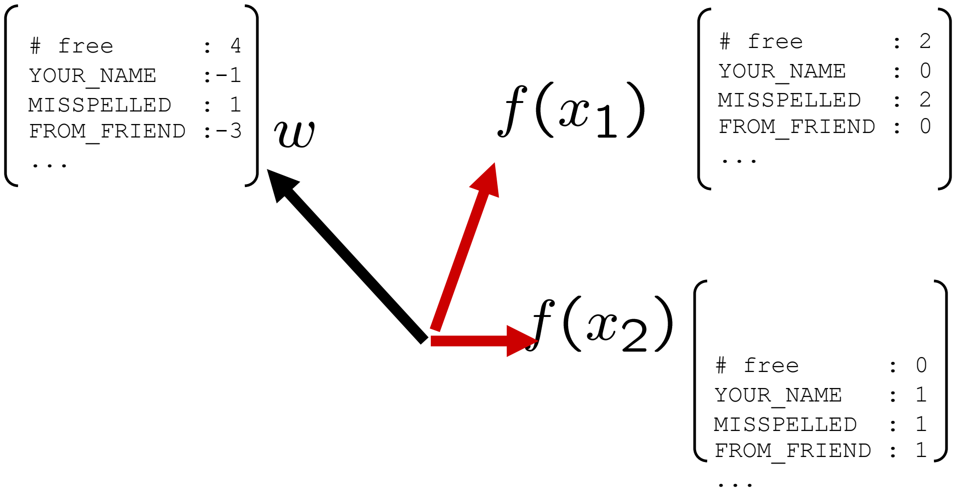
Positive dot product $w \cdot f$ *means the positive class*

SGD training for Binary Classifier

Figure out the weight vector from training instances

- Start with weights = 0
- For each training instance:
  - Classify with current weights
  - f (x) is feature vector of x

$$y = \begin{cases} +1 & \text{if } \; w \cdot f(x) \geq 0 \\ -1 & \text{if } \; w \cdot f(x) < 0 \end{cases}$$



SGD with learning rate 1:

Do until satisfied:

- For each training example *(y*, f)*

1. Compute the gradient ∇E where E is squared error

2. Update w = w - ∇E

Namely no change with correct prediction

Otherwise w= w+ y · f*

$E = 0.5(y^* - w\, f(x))^2$

$\nabla E = \partial E / \partial w = -(y^* - y) f$

$= 0 \text{ if } y^* = y$

$\text{else } -y^* \; f$

Example of SGD Learning from training data

- *Classifier model*: $f(x) =$ Size $*w_1+$color $*w_2$ $+$shape$*w_3$
  Use sign of f(x) to classify

Initially $w_1=w_2=w_3=0$

Instance	Size	Color	Shape	Category
x_1	Small 0	Red 0	Circle 0	Positive 1
x_2	Large 2	Red 0	Circle 0	Positive 1
x_3	Small 0	Red 0	Triangle 1	Negative -1
x_4	Large 2	Blue 1	Circle 0	Negative -1

With Instance 1: $\text{sign}(f(x_1))=\text{sign}(0)=1$. No weight change

With Instance 2: $\text{sign}(f(x_2))=\text{sign}(0)=1$. No weight change.

With Instance 3: $\text{sign}(f(x_3))=\text{sign}(0)=1$. Wrongly classified

$w= w+(-1) *(0,0,1) =(0,0,-1)$

With Instance 4: $\text{sign}(f(x_4))=\text{sign}(0)=1$. Wrongly classified

$w= w+(-1) *(2,1,0) =(-2,-1,-1)$

Incremental vs Batch Mode in SGD

SGD in an incremental mode:

Update weights instance by instance

Do until satisfied:

- For each training example *d* in *D*

  1. Compute the gradient $\nabla E_d[\vec{w}]$

  2. $\vec{w} \leftarrow \vec{w} - \eta \nabla E_d[\vec{w}]$

$$E_d[\vec{w}] \equiv \frac{1}{2}(t_d - o_d)^2$$

$$\nabla E = \partial E / \partial w = -(t_d - o_d)x$$

x is a feature vector

t_d is the judgement label

$o_d = w$ x

SGD in a batch or minibatch mode:

Update weights by a (mini-) batch of instances (subset D)

Do until satisfied:

  1. Compute the gradient $\nabla E_D[\vec{w}]$

  2. $\vec{w} \leftarrow \vec{w} - \eta \nabla E_D[\vec{w}]$

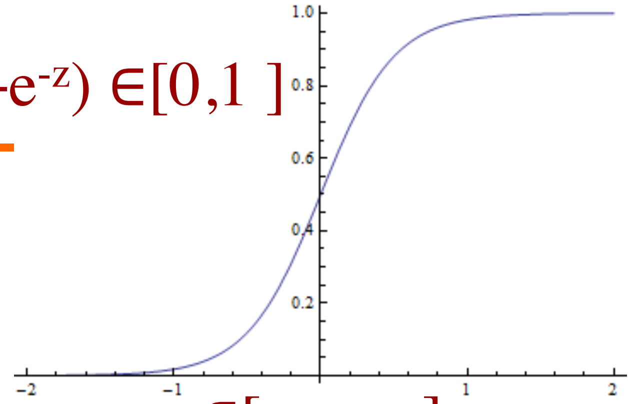
$$E_D[\vec{w}] \equiv \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2$$

Training instances are divided and utilized by batches.

Each batch can be executed fast with GPU or a parallel platform

#epoch is #passes to work through the entire training dataset

Other Classification Prediction or Loss Functions



Softmax for binary classification
Logistic regression

- Score for y=1: $w^\top f(x)$
- Score for y=-1: $-w^\top f(x)$
- Probability of label:

$$p(y=1|f(x);w) = \frac{e^{w^\top f(x^{(i)})}}{e^{w^\top f(x)} + e^{-w^\top f(x)}}$$

$$p(y=-1|f(x);w) = \frac{e^{-w^\top f(x)}}{e^{w^\top f(x)} + e^{-w^\top f(x)}}$$

- Maximize: $$l(w) = \prod_{i=1}^{m} p(y=y^{(i)}|f(x^{(i)});w)$$
  - Equivalently maximize log likelihood:

$$ll(w) = \sum_{i=1}^{m} \log p(y=y^{(i)}|f(x^{(i)});w)$$

Multi-class Softmax

- 3-class softmax – classes A, B, C
  - 3 weight vectors: w_A, w_B, w_C

- Probability of label A: (similar for B, C)

$$p(y = A|f(x); w) = \frac{e^{w_A^\top f(x)}}{e^{w_A^\top f(x)} + e^{w_B^\top f(x)} + e^{w_C^\top f(x)}}$$

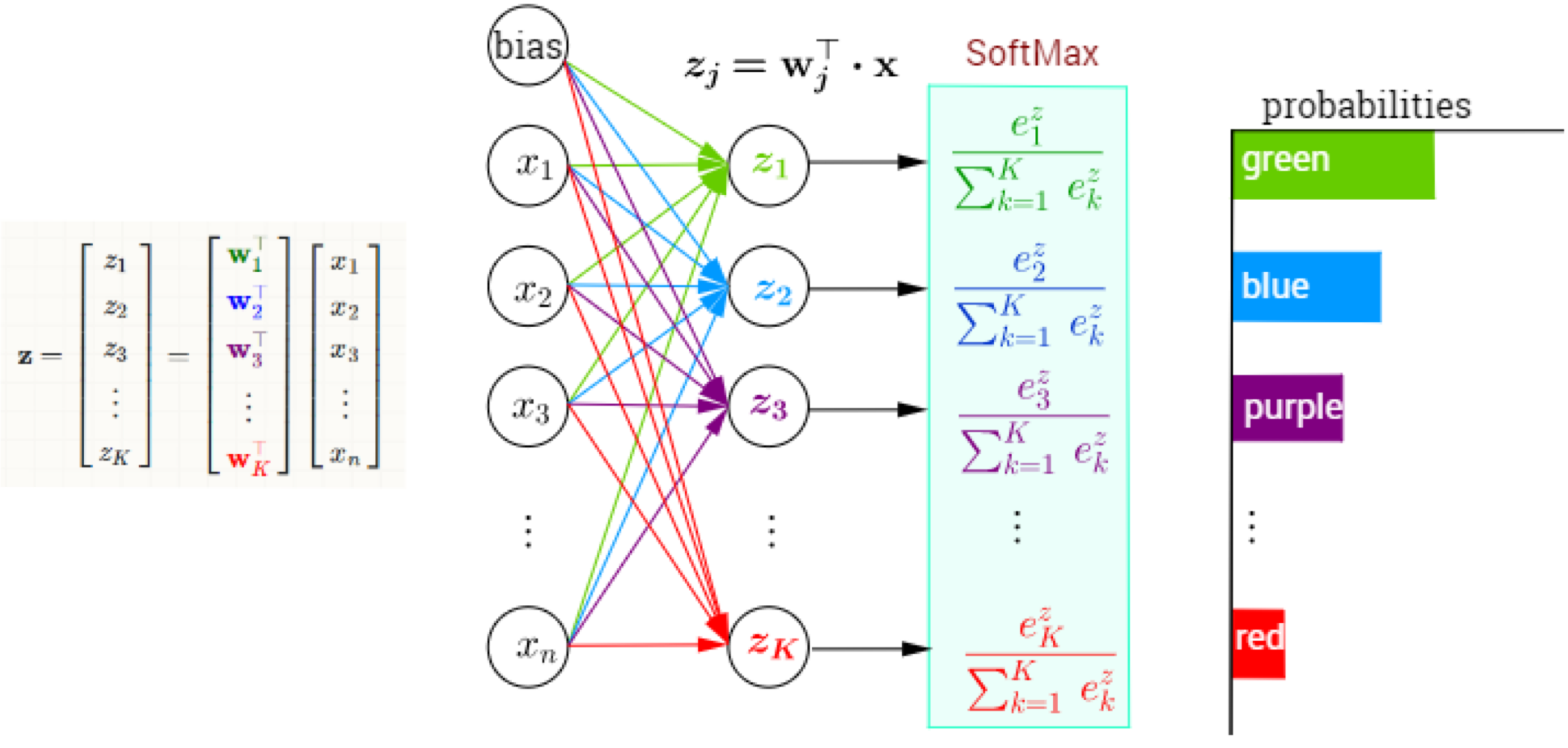
- Loss function: $l(w) = \prod_{i=1}^{m} p(y = y^{(i)}|f(x^{(i)}; w)$

- Equivalently maximize log likelihood:

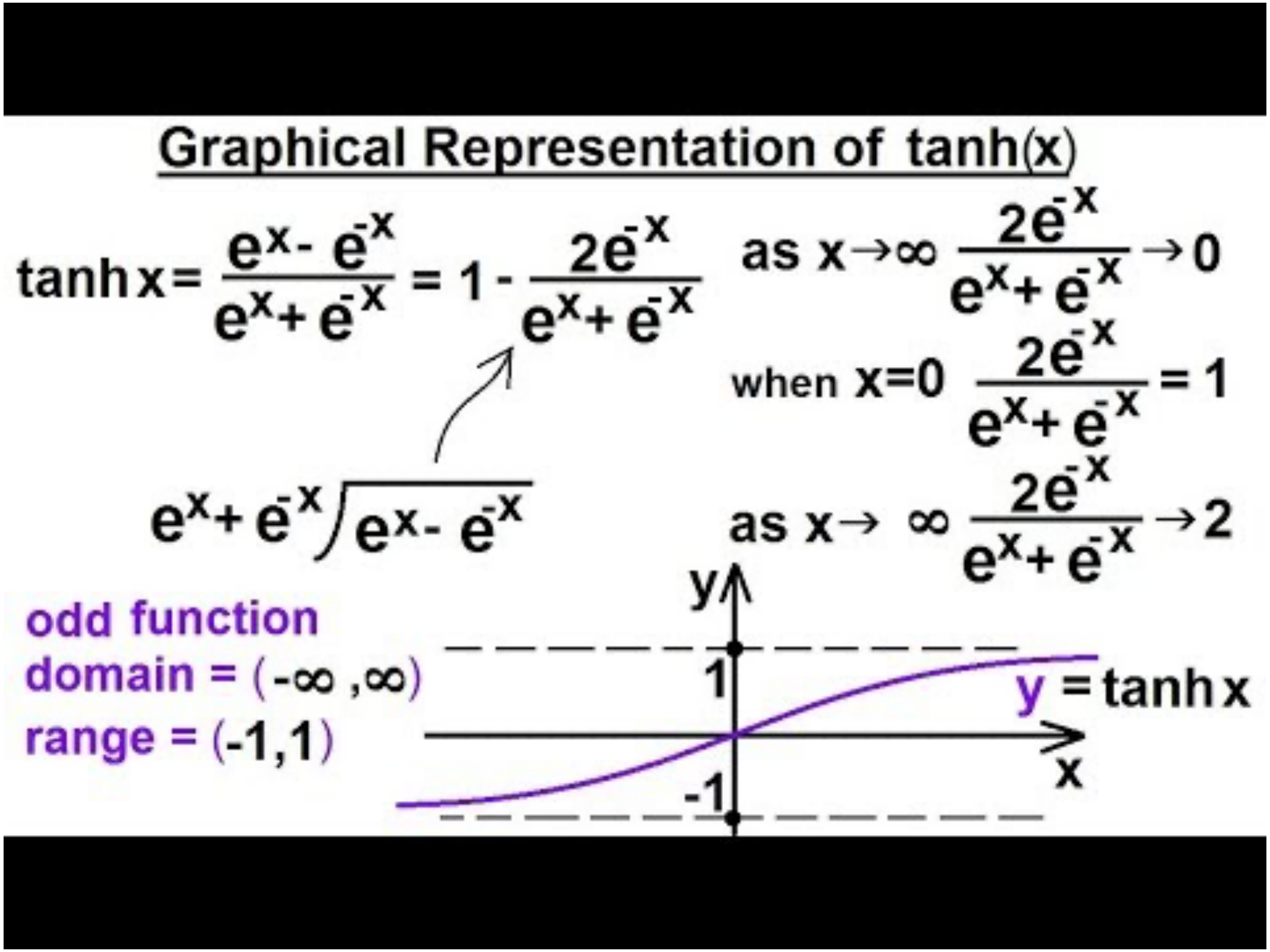
$$ll(w) = \sum_{i=1}^{m} \log p(y = y^{(i)}|f(x^{(i)}; w)$$

Multi-class Two-Layer Neural Network with SoftMax

Multi-Class Classification with NN and SoftMax Function



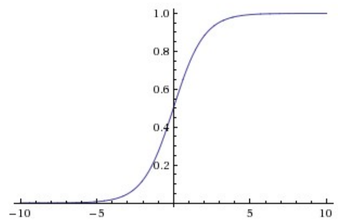
Activation Function: tanh(x)



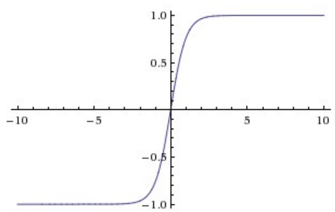
Other Activation Functions

Sigmoid

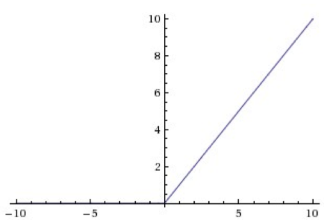
$$\sigma(x) = 1/(1+e^{-x})$$



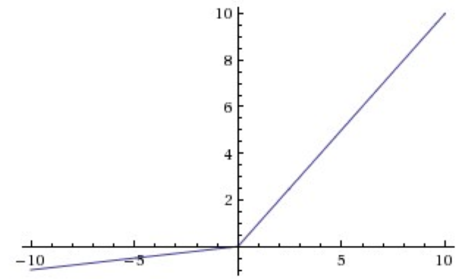
tanh tanh(x)



ReLU max(0,x)



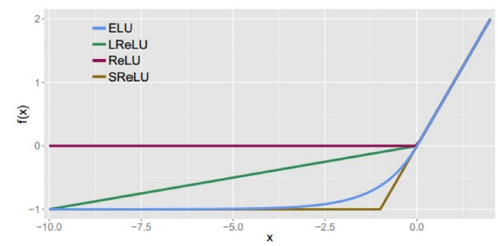
Leaky ReLU
max(0.1x, x)



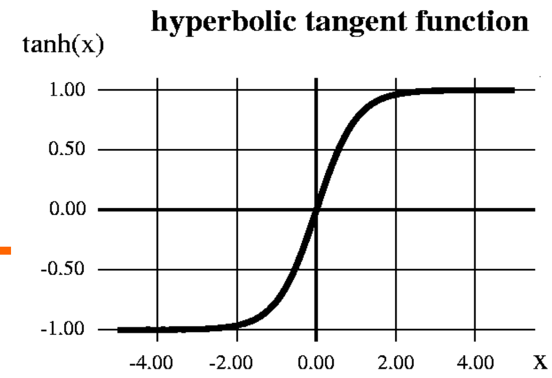
Maxout $\max(w_1^T x + b_1, w_2^T x + b_2)$

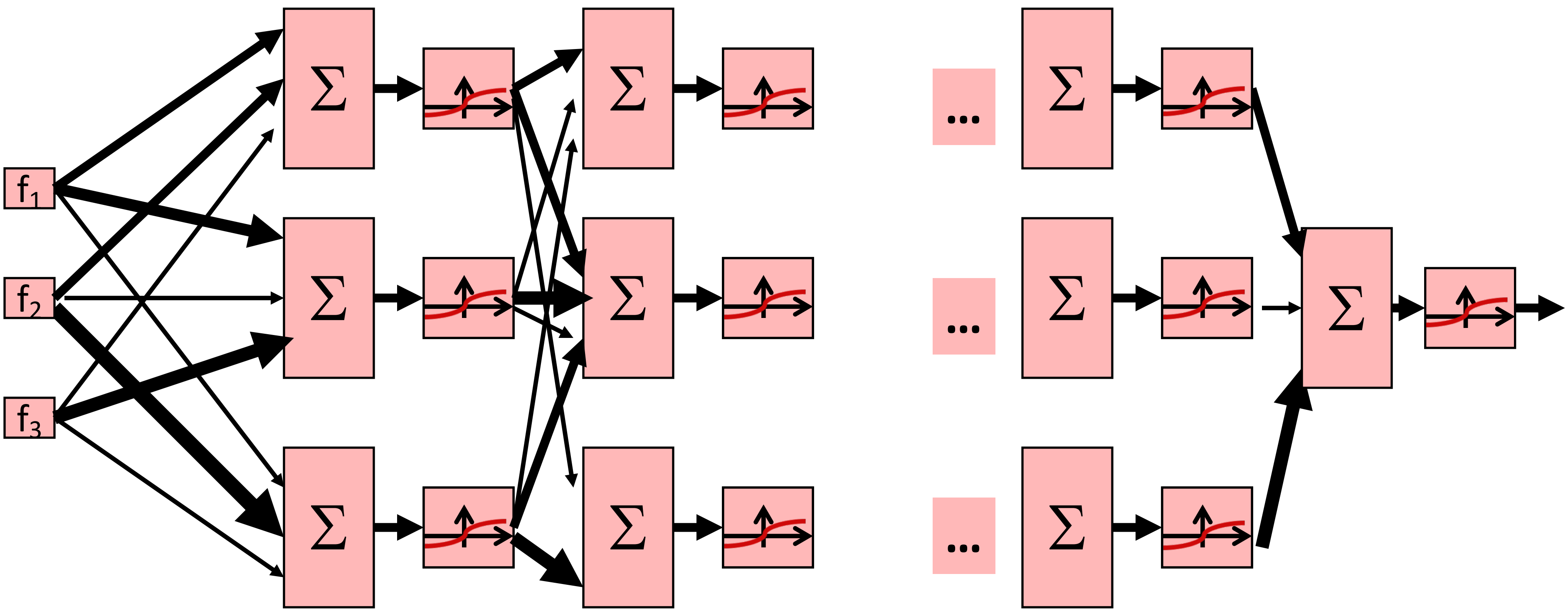
ELU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \le 0 \end{cases}$$

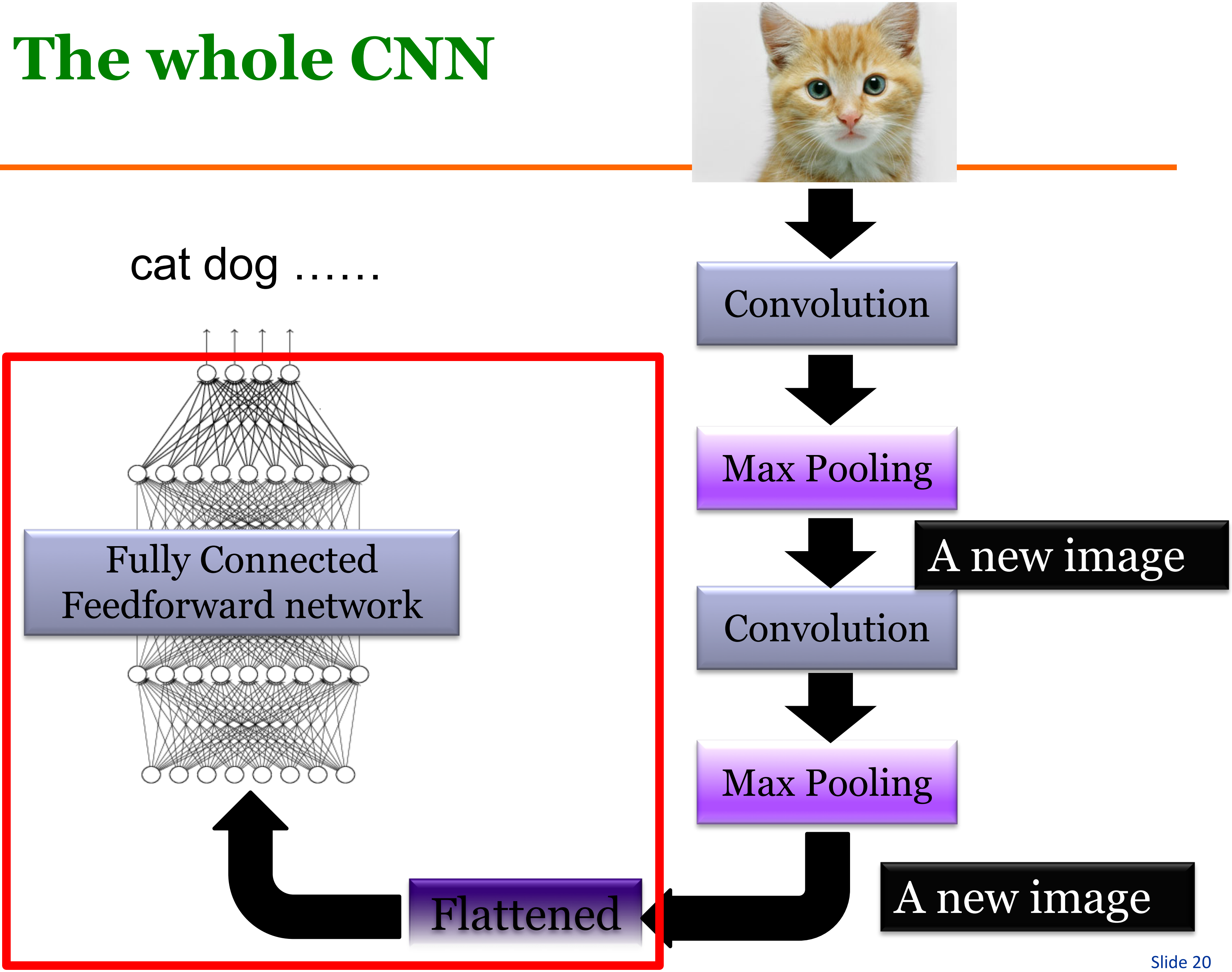


N-Layer Neural Network





The whole CNN



How to Calculate Partial Derivatives for SGD through a Computer Algorithm

- Graph representation of a loss function can be huge with thousands or even millions of parameters.
- How to compute partial derivatives of a computational graph

Example: Given a function f(x,y,z)= (x+y)z, what is the partial derivative of f with respect to x, y, z?

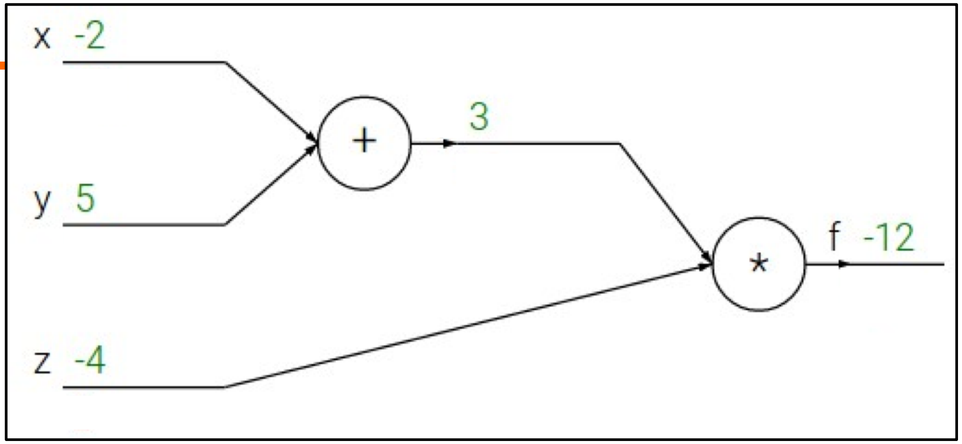
- Computer has to do it symbolically. Not easy in general
- What is the partial derivative of f with respect to x, y, z, given x = -2, y = 5, z = -4 from a training instance?

Easier to do by focusing on the given training instance

Example of Algorithmic Derivative Computation

- $f(x, y, z) = (x + y)z$

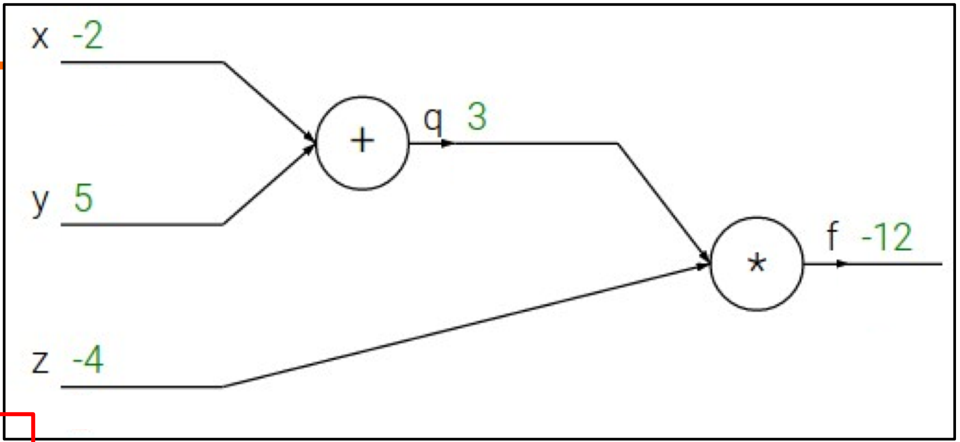
Knowing x = -2, y = 5, z = -4



Get local derivates for each node
Get the final value f via forward computation

$$f(x,y,z) = (x+y)z$$

x = -2, y = 5, z = -4, f(x,y,z)=-12



Get local derivates for each node

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

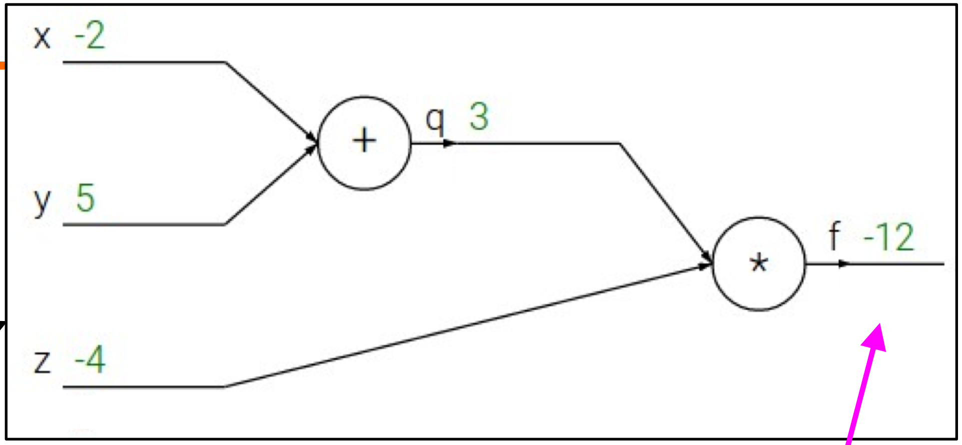
$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Now we conduct a backward propagation in this graph to compute $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Backward to get the derivative of last node $\frac{\partial f}{\partial f}$

- $f(x, y, z) = (x + y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12



$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

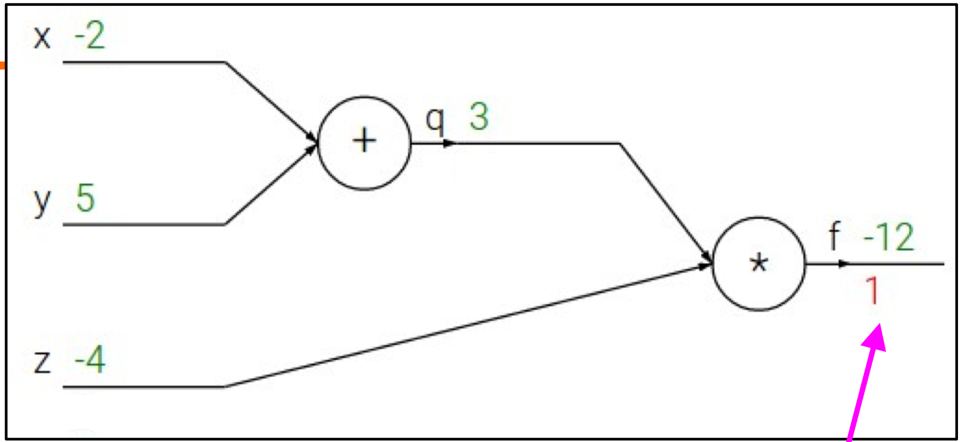
$\frac{\partial f}{\partial f}$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

$\frac{\partial f}{\partial f}$ =1 as local derivative. It is trivial

$f(x,y,z) = (x+y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12



$\frac{\partial f}{\partial f}$

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

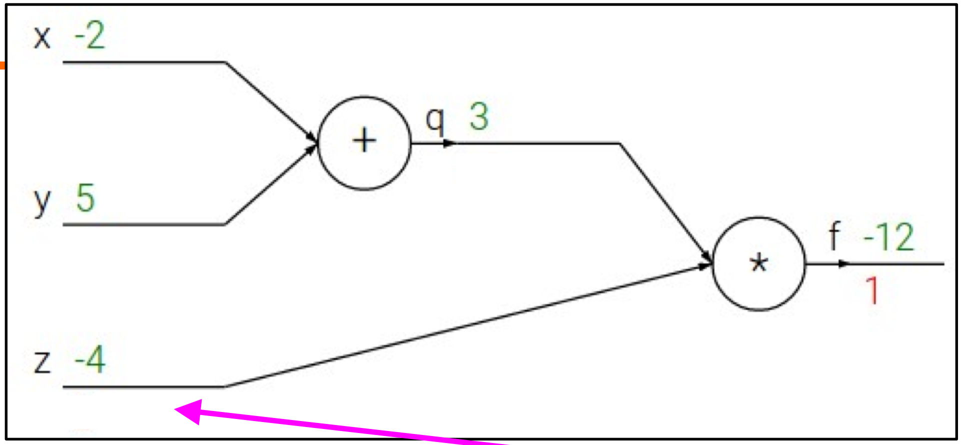
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

25

Need to get derivative $\frac{\partial f}{\partial z}$

- $f(x, y, z) = (x + y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12



$\frac{\partial f}{\partial z}$

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

26

Derive 3 as derivative $\frac{\partial f}{\partial z}$ because ∂f/ ∂z= q=3

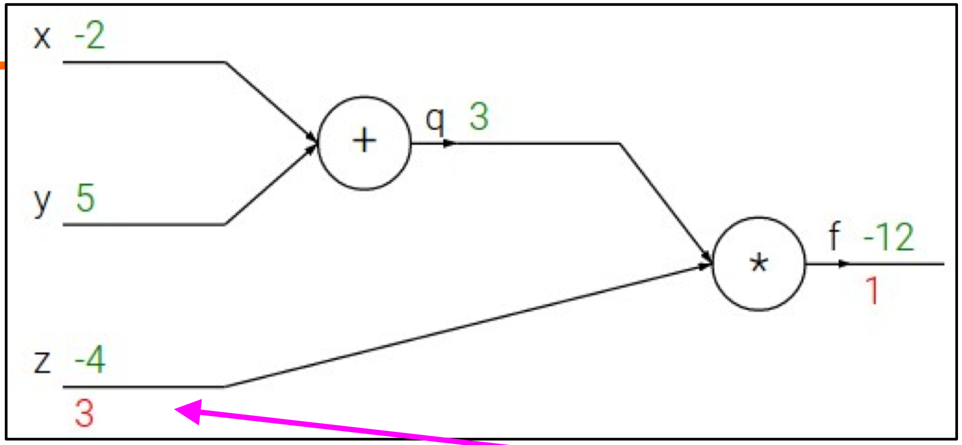
- $f(x, y, z) = (x + y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$\frac{\partial f}{\partial z}$

27

Need to get derivative $\frac{\partial f}{\partial q}$

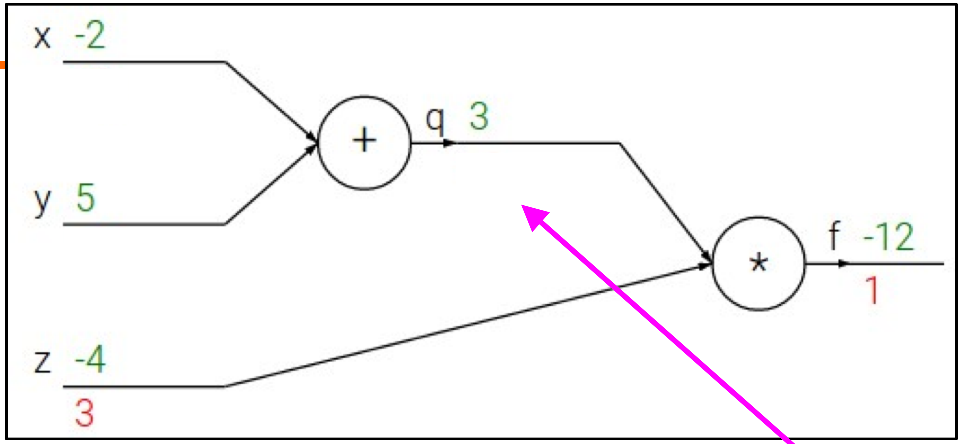
- $f(x,y,z) = (x+y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

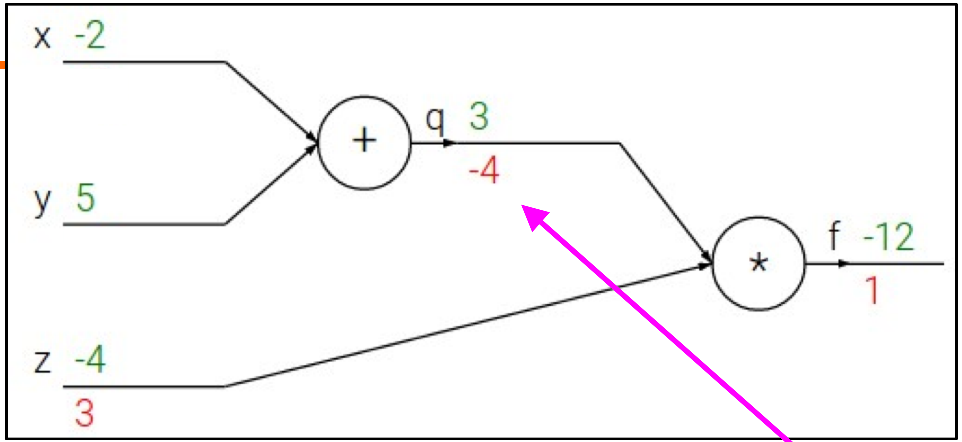


$\frac{\partial f}{\partial q}$

$\frac{\partial f}{\partial q}$ is found because ∂f/ ∂q= z=-4

$$f(x,y,z) = (x+y)z$$

x = -2, y = 5, z = -4, f(x,y,z)=-12



$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

$$\frac{\partial f}{\partial q}$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

How to compute ∂f/ ∂y ?

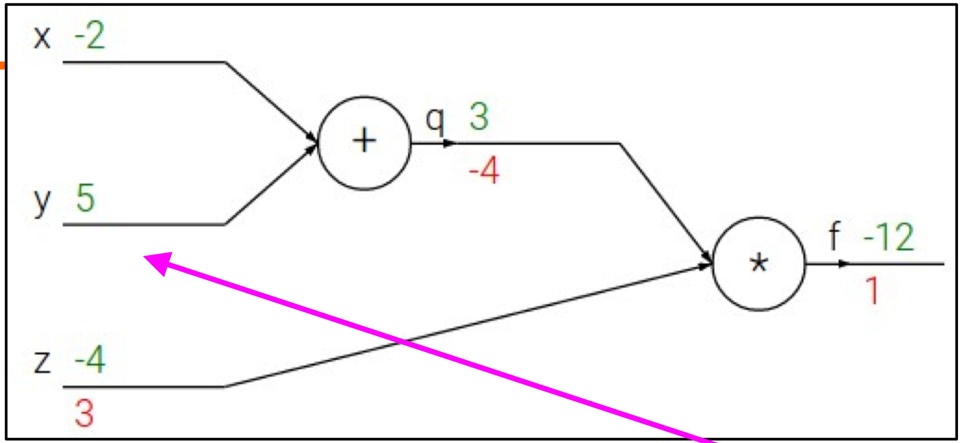
- $f(x,y,z) = (x+y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

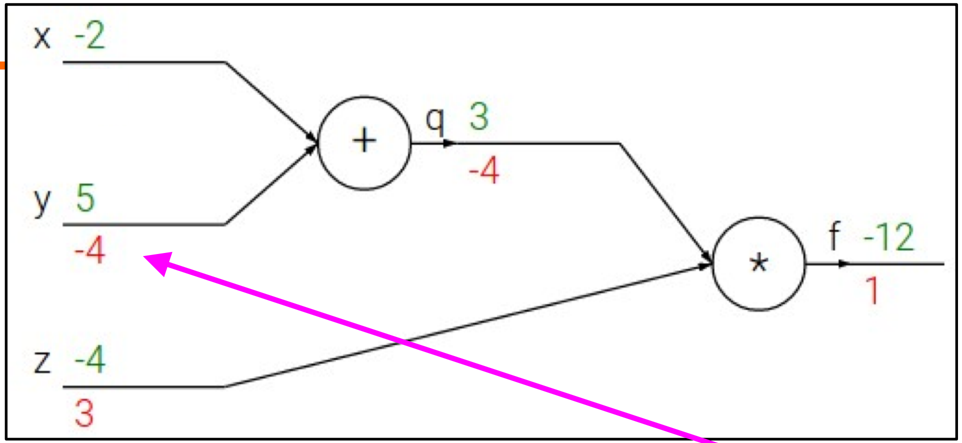


$\frac{\partial f}{\partial y}$

Use the chain rule locally to compute ∂f/ ∂y = (-4)· 1 =-4

- $f(x,y,z) = (x+y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12



$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

$$\frac{\partial f}{\partial y}$$

Chain rule:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y}$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

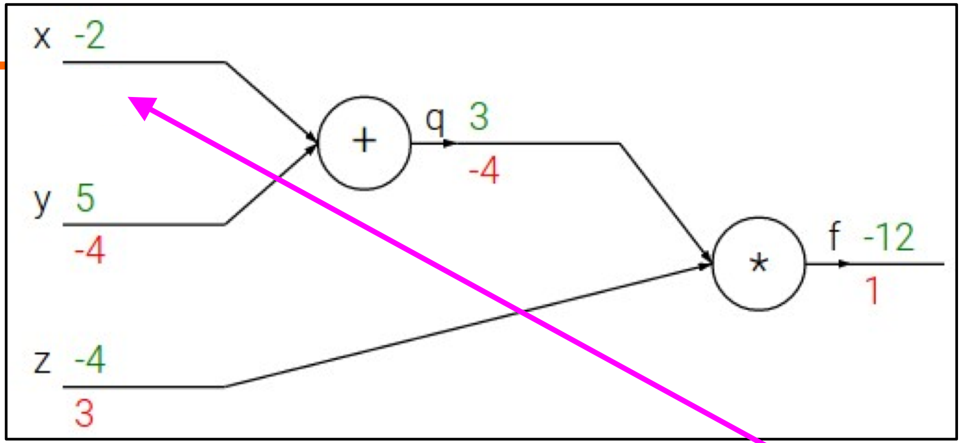
Use the chain rule locally to compute ∂f/ ∂x

$$f(x,y,z) = (x+y)z$$

x = -2, y = 5, z = -4, f(x,y,z)=-12

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$



$$\frac{\partial f}{\partial x}$$

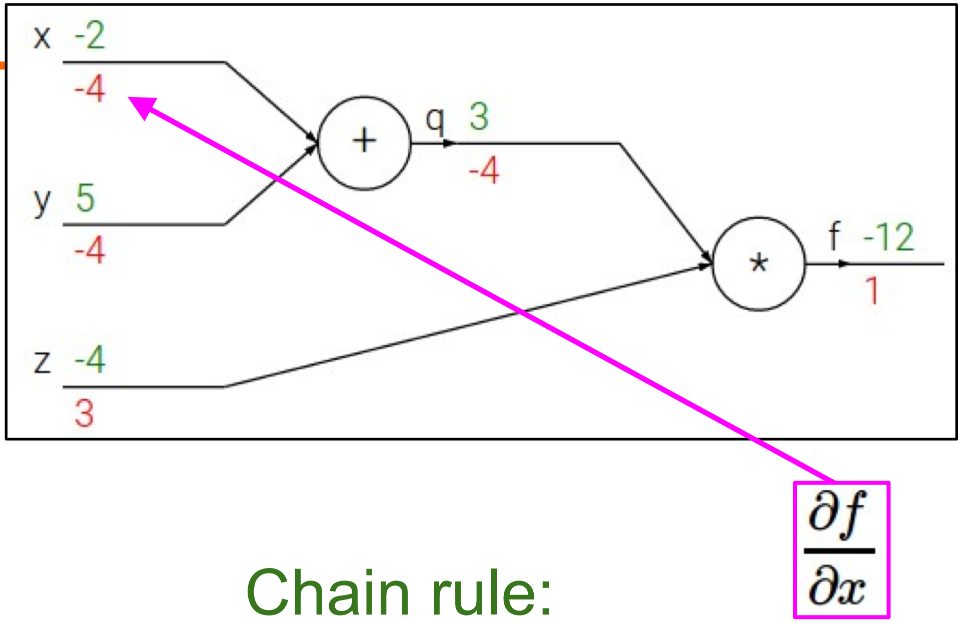
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

32

Use the chain rule locally to compute ∂f/ ∂x = (-4)· 1 =-4

- $f(x,y,z) = (x+y)z$

x = -2, y = 5, z = -4, f(x,y,z)=-12



$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

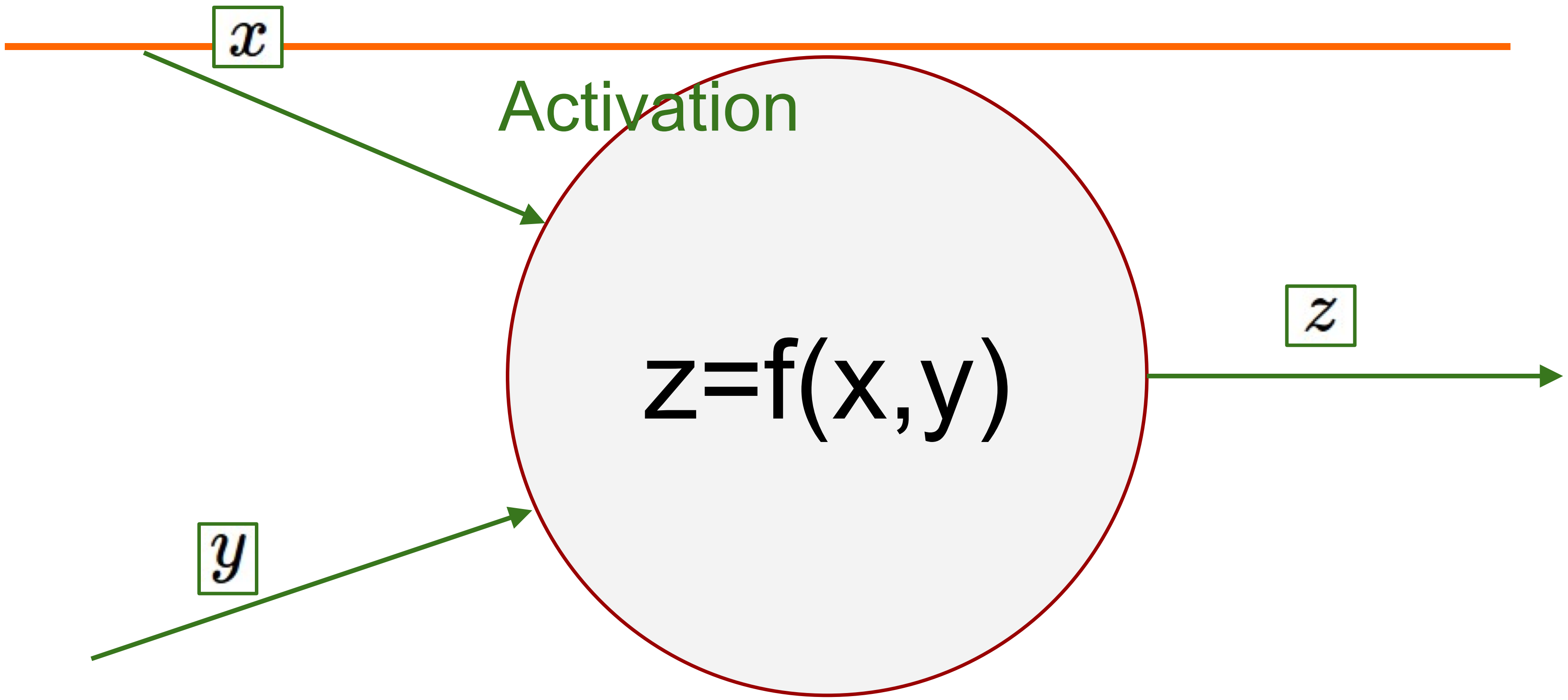
$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

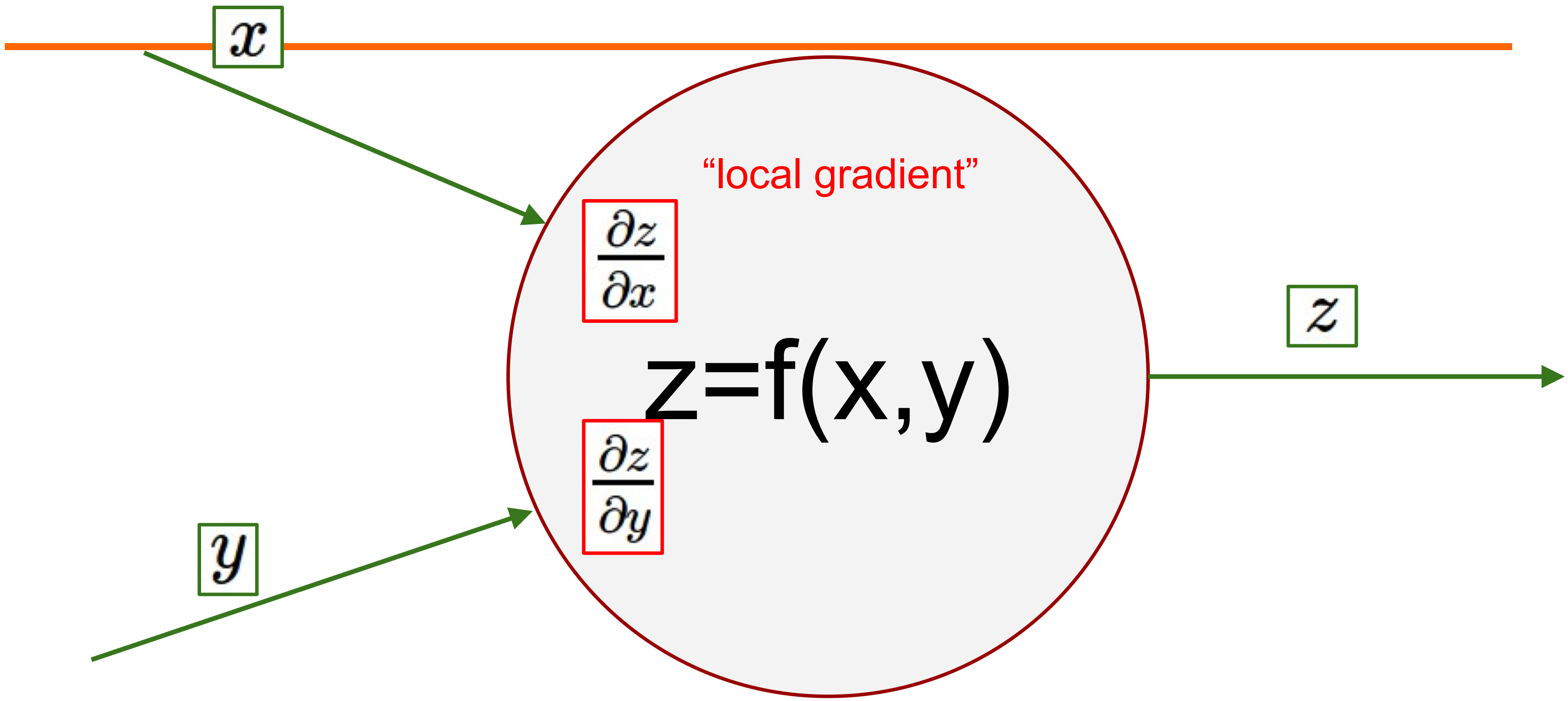
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

How to use the chain rule locally ?

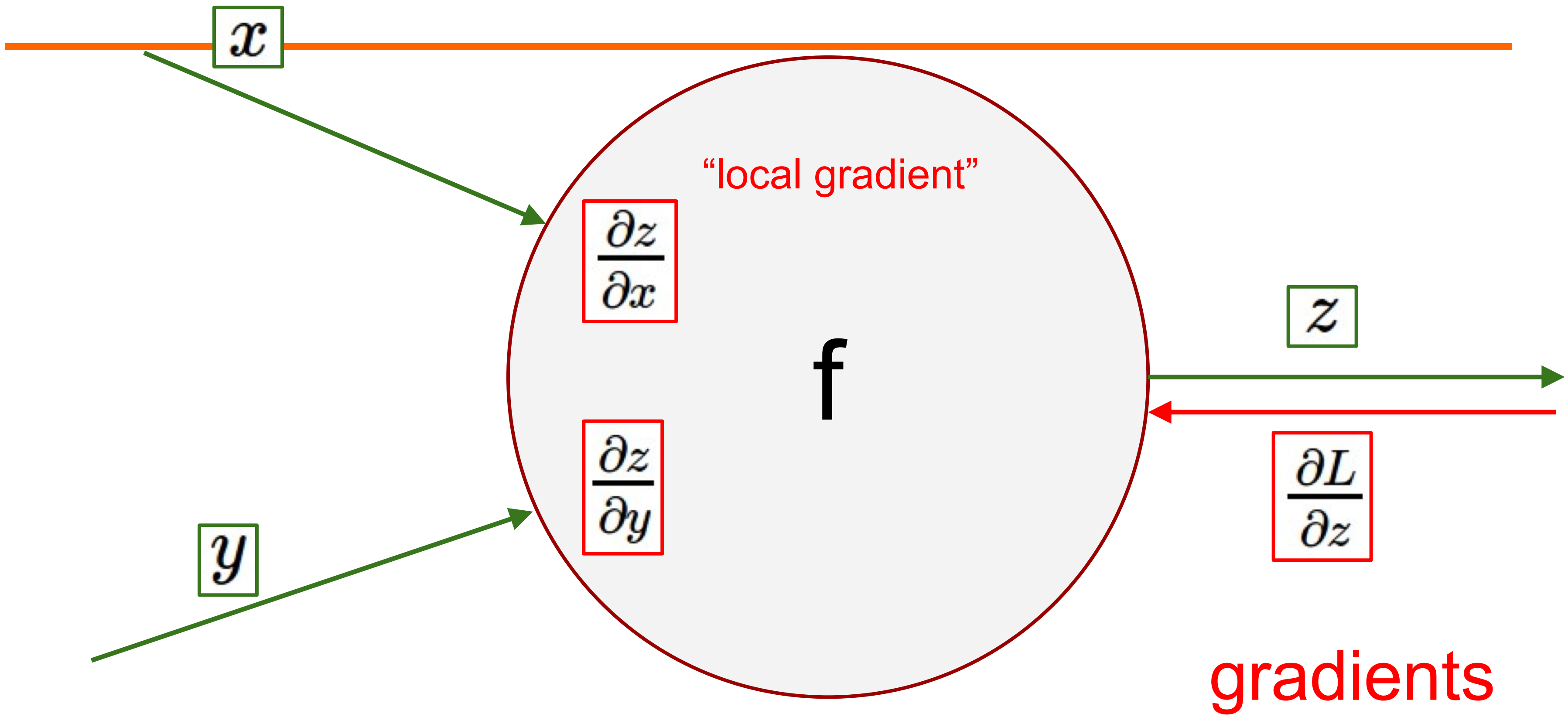


34

Compute the local gradients first

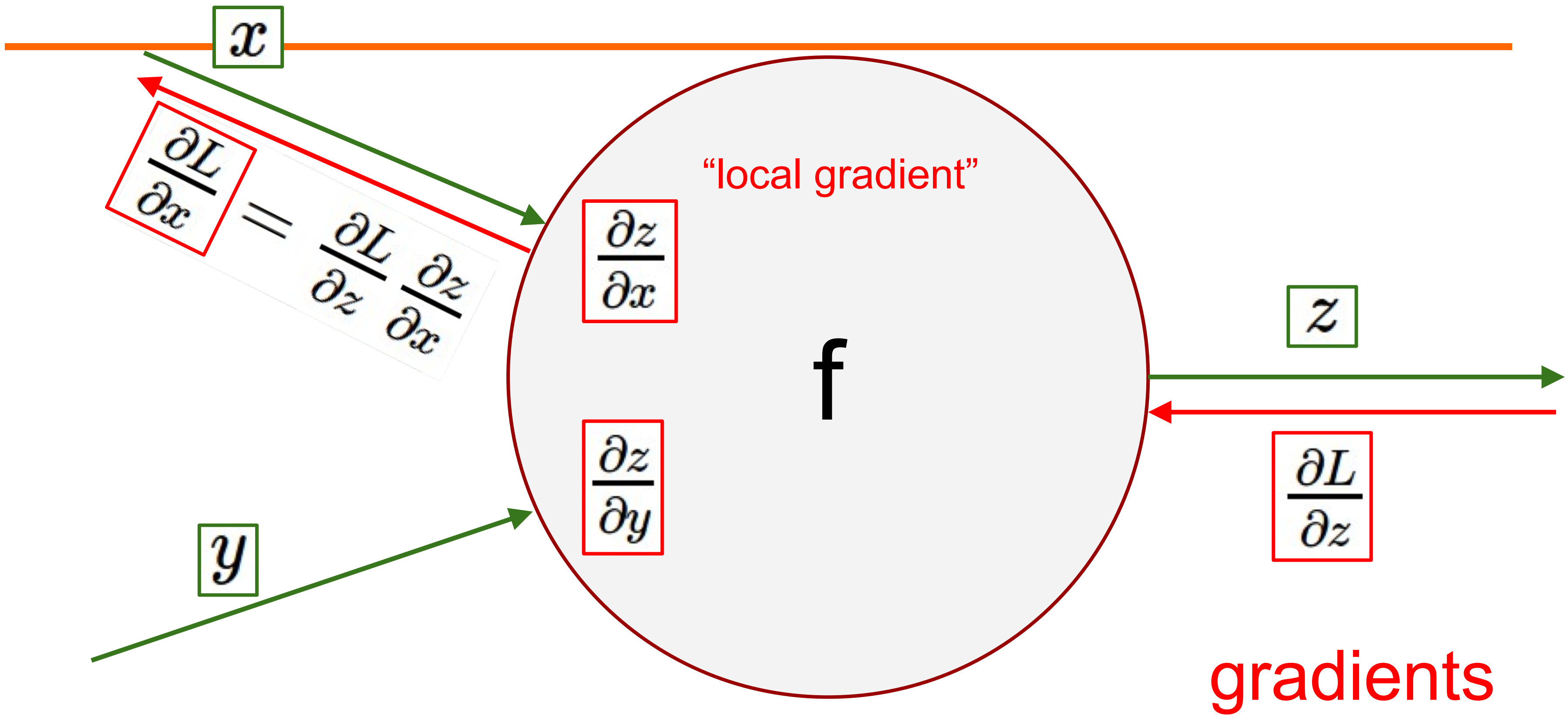


Get the incoming gradient



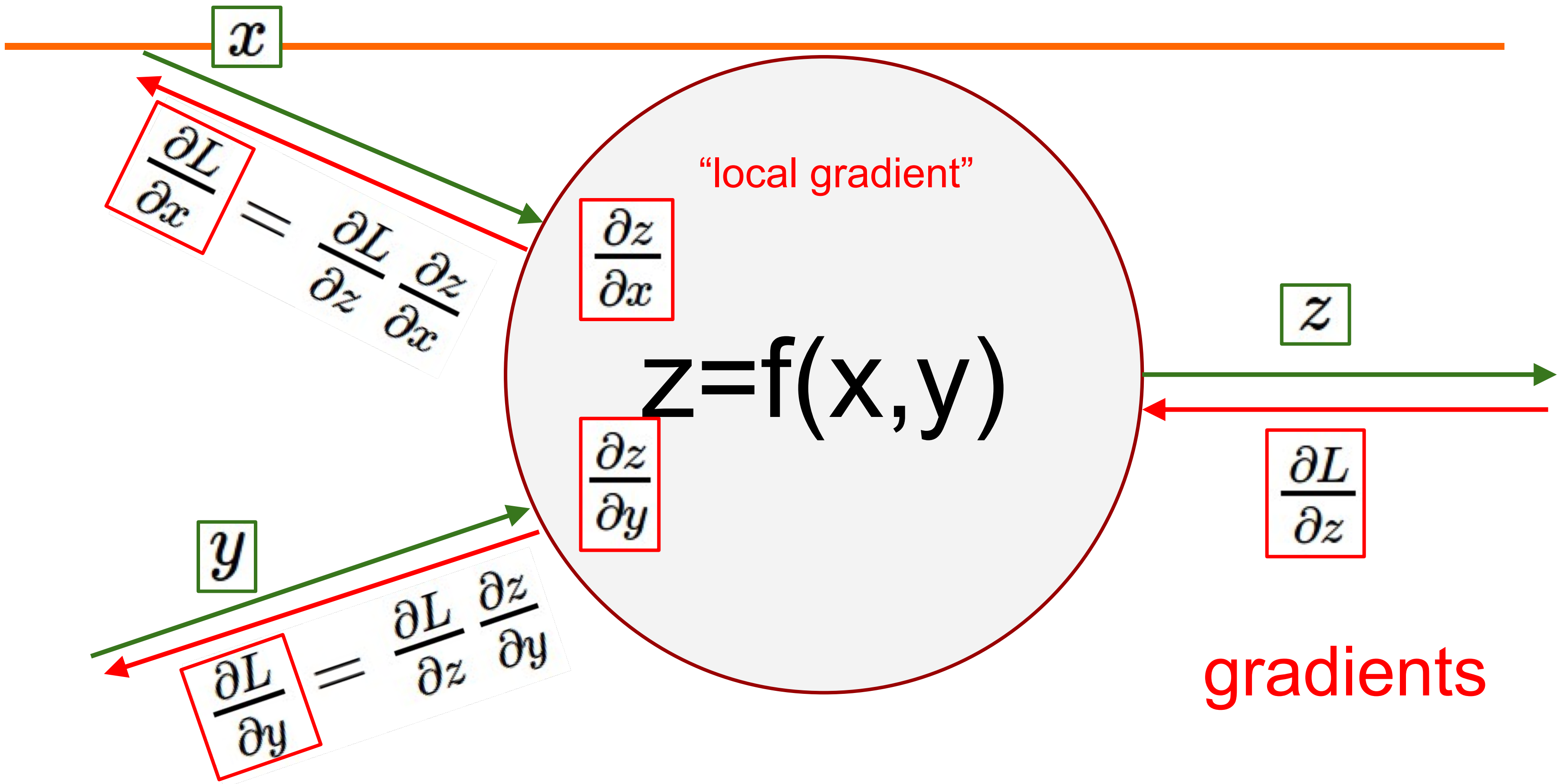
36

Apply the chain rule to compute the gradient Then propagate backward to one direction



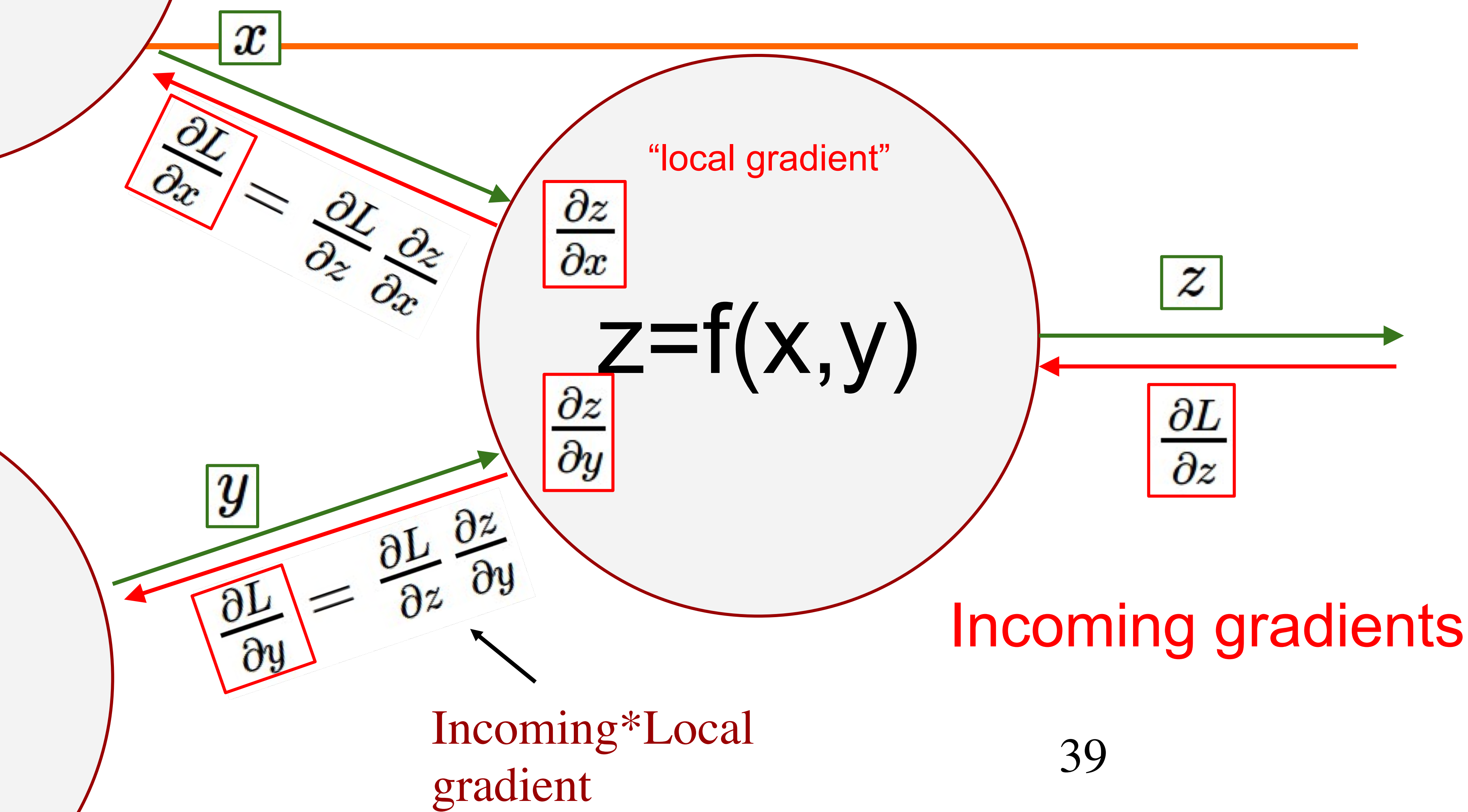
37

Apply the chain rule to compute the gradient
Propagate backward to another direction



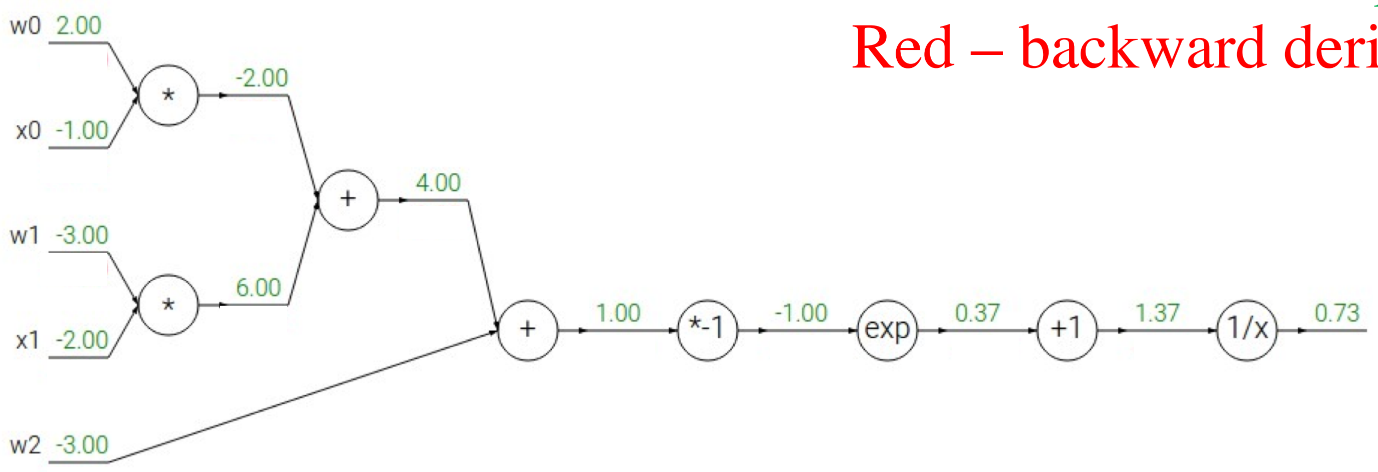
38

Summary of backward flow



Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

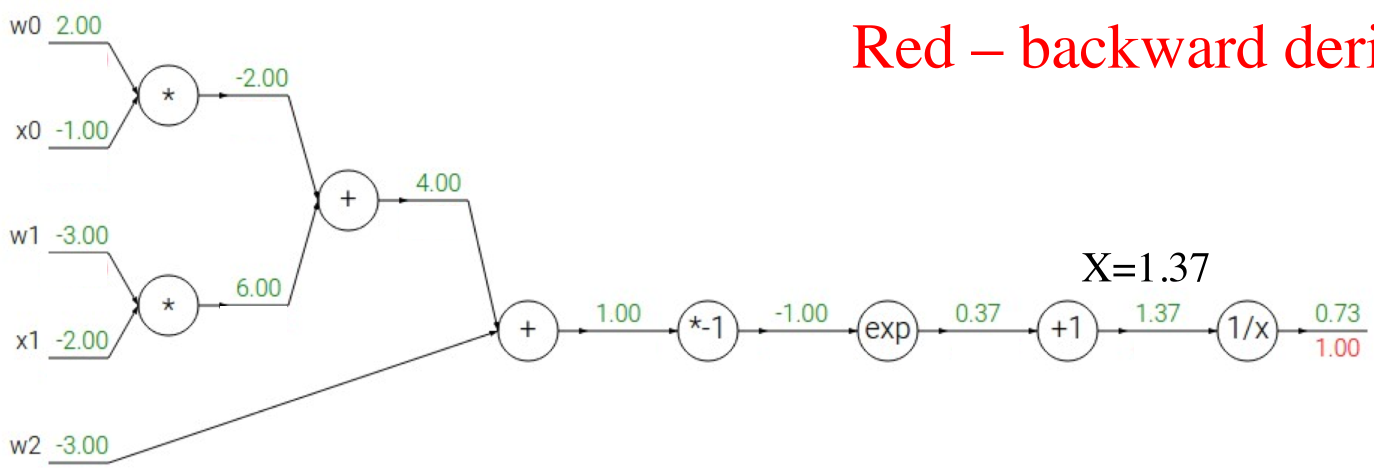
Green- forward computation
Red – backward derivatives



40

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Green- forward computation
Red – backward derivatives



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

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

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

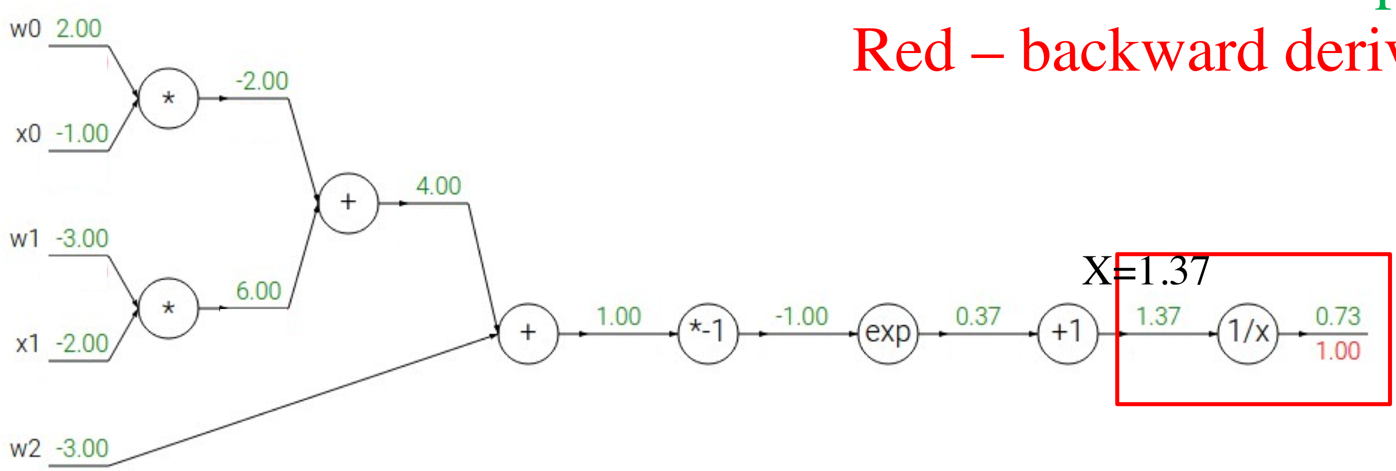
$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$

41

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

Green- forward computation

Red – backward derivatives



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

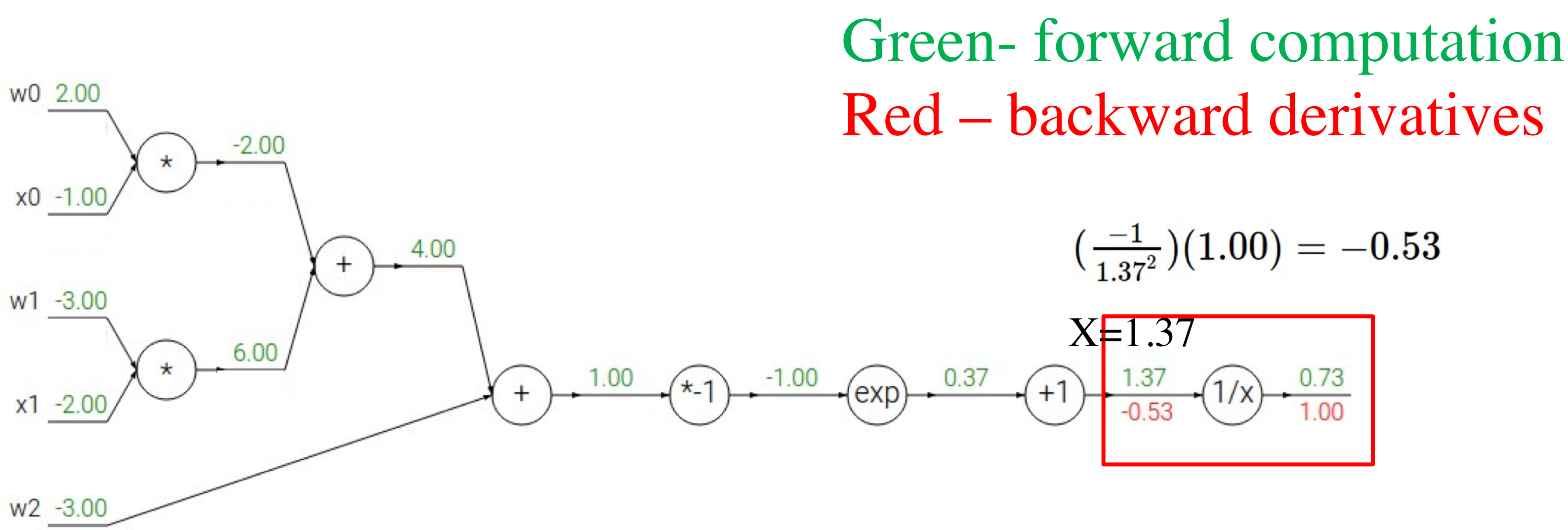
$$f_a(x) = ax \rightarrow \frac{df}{dx} = a$$

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

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

42

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



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

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

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

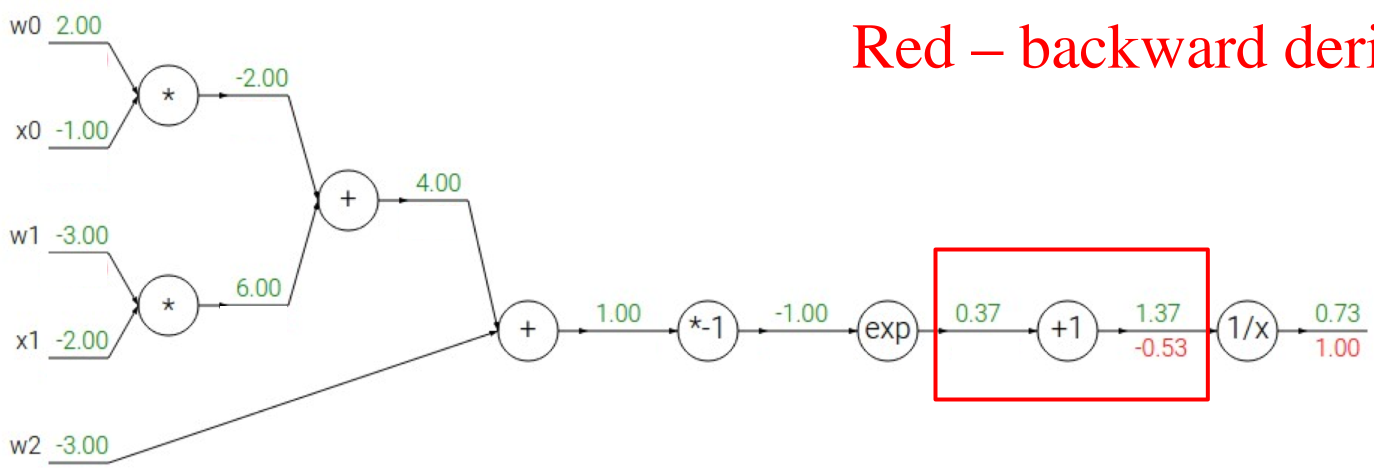
$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$

43

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

Green- forward computation

Red – backward derivatives



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

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

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

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

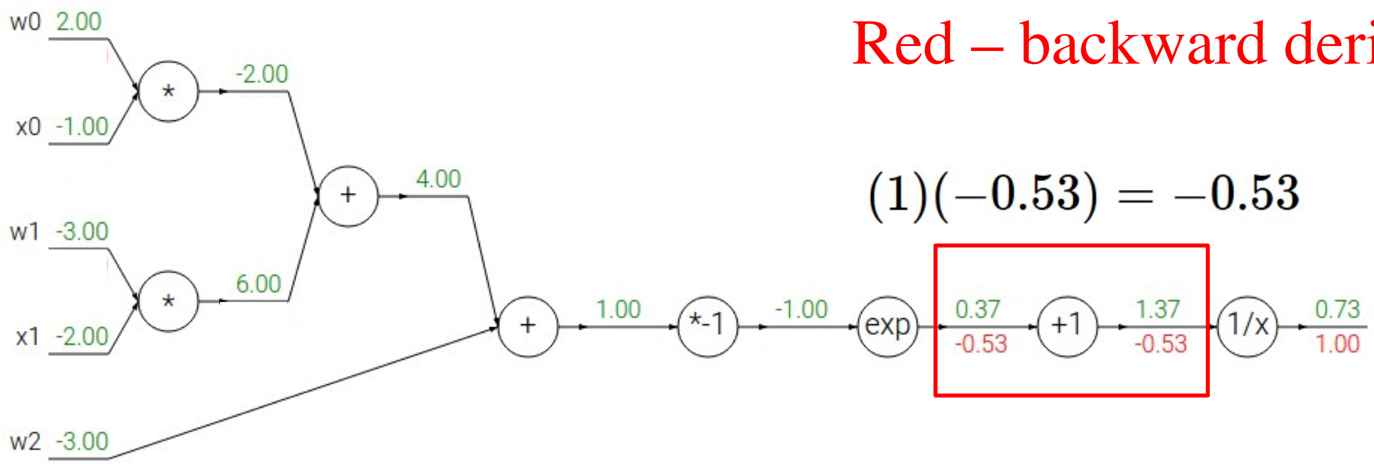
44

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

Another example: $f(w,x) = \dfrac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Green- forward computation

Red – backward derivatives



$$(1)(-0.53) = -0.53$$

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

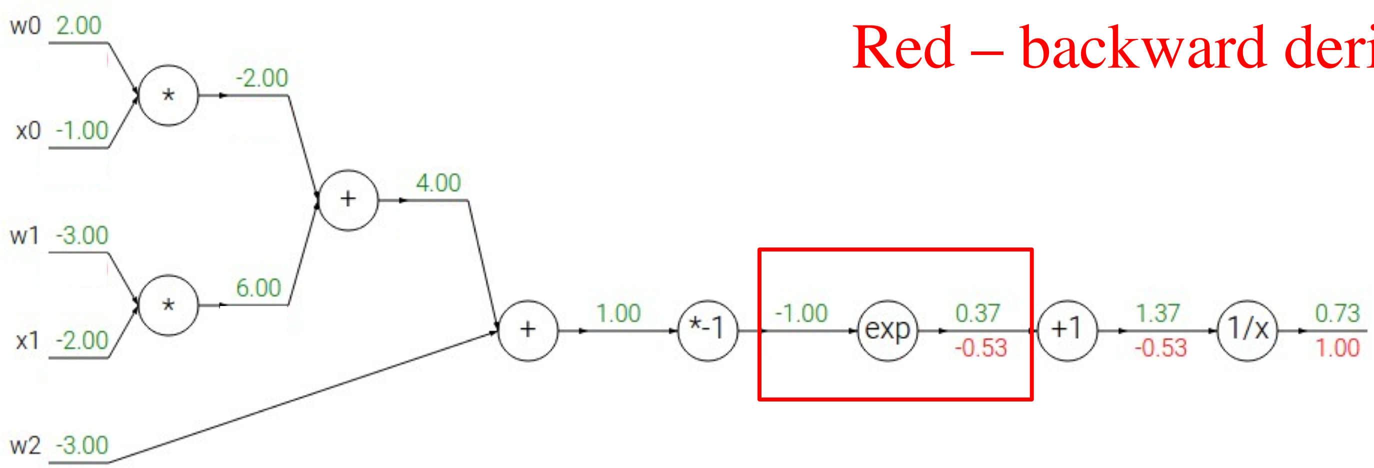
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Green- forward computation
Red – backward derivatives



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

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

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

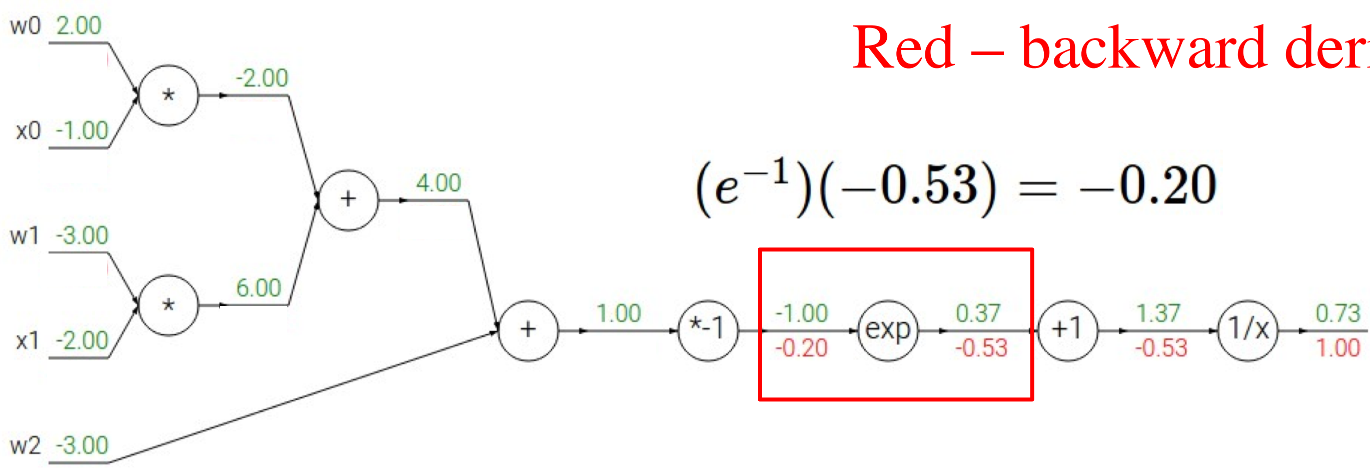
$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

Green- forward computation

Red – backward derivatives



$$(e^{-1})(-0.53) = -0.20$$

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

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

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

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

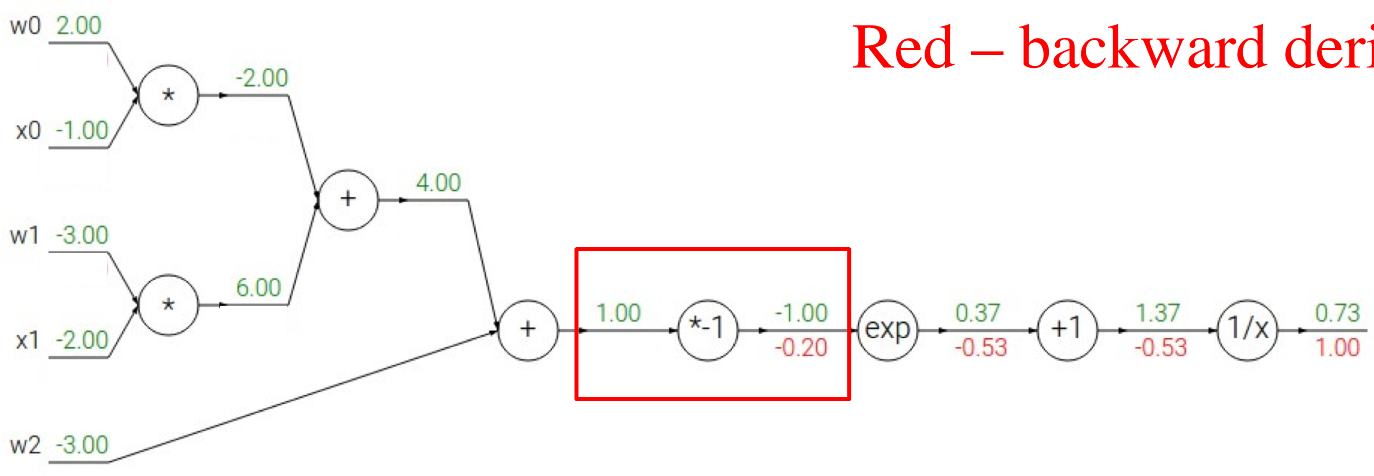
47

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Green- forward computation

Red – backward derivatives



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

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

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

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

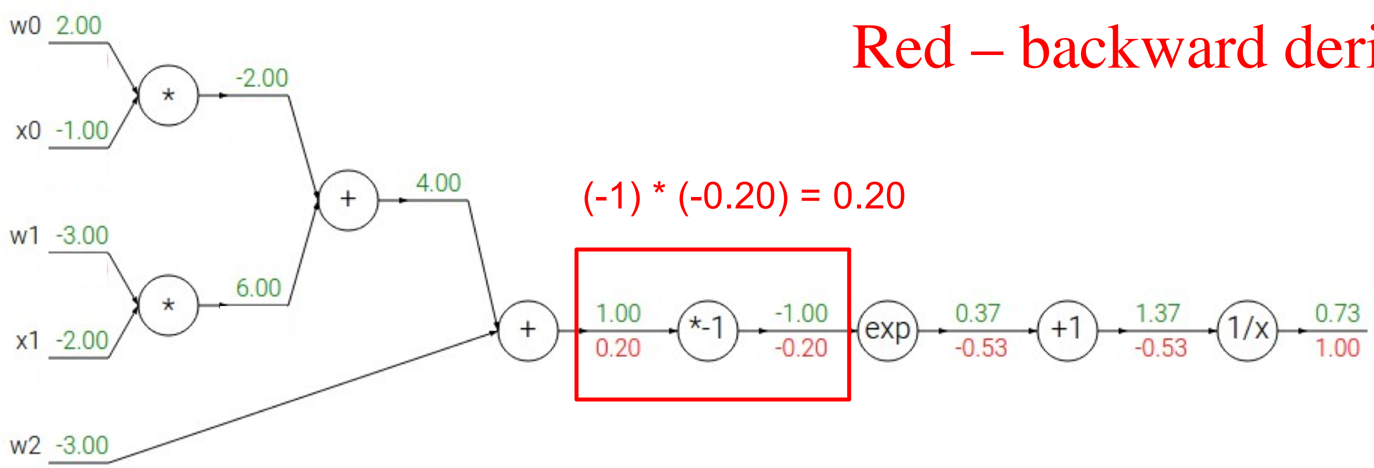
$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

Green- forward computation

Red – backward derivatives



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

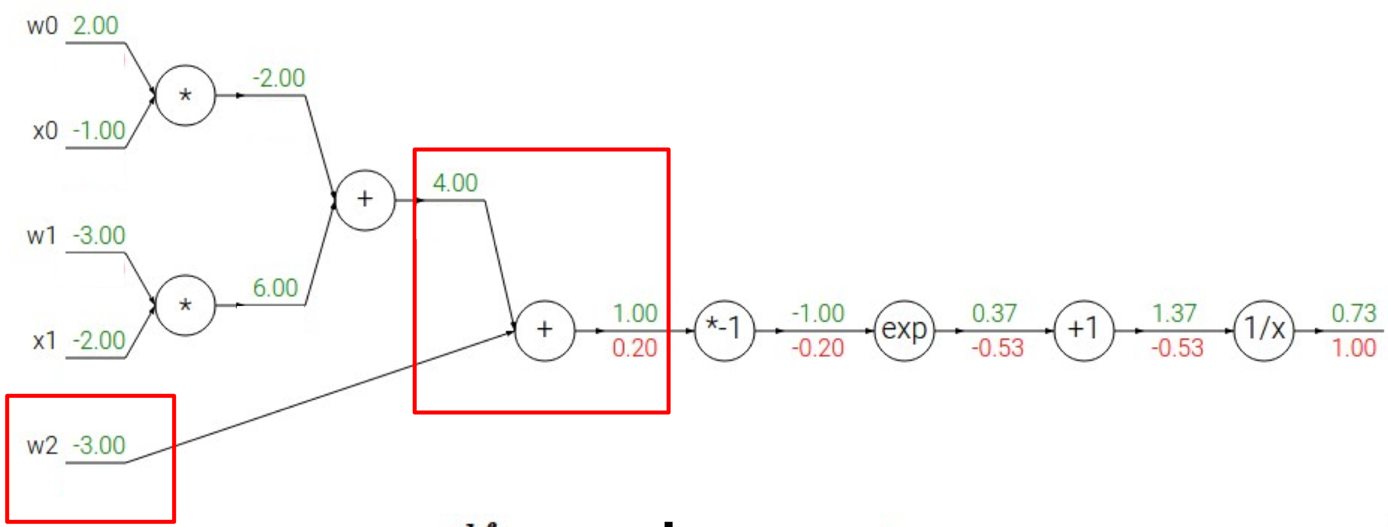
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



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

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

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

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

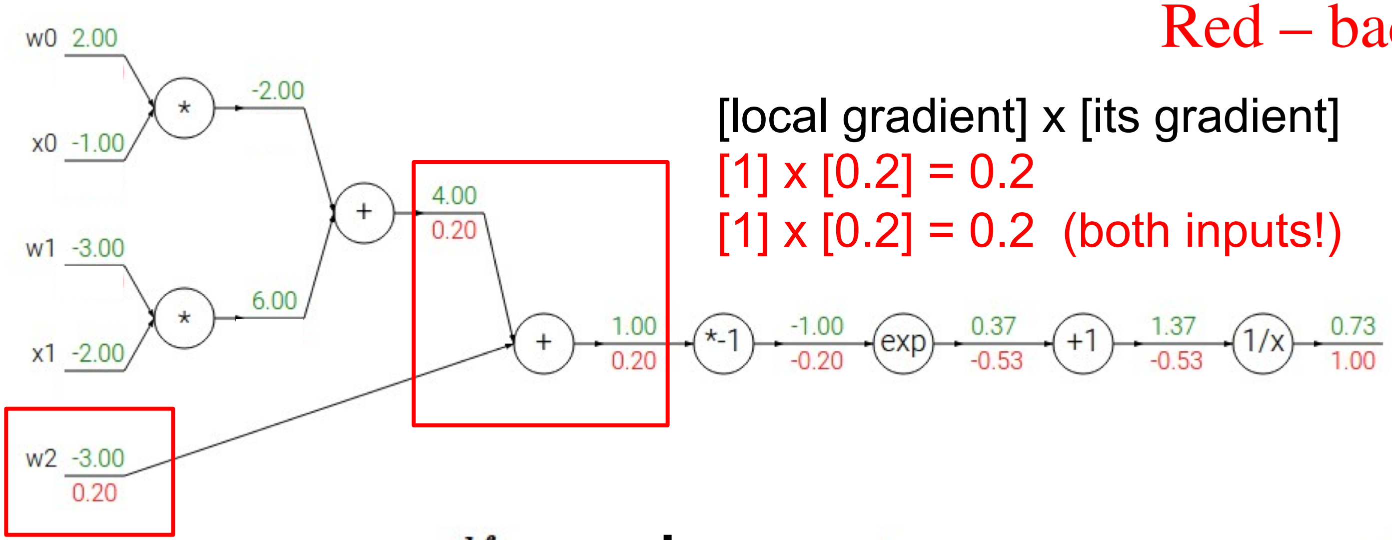
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

Green- forward

Red – backward



[local gradient] x [its gradient]

[1] x [0.2] = 0.2

[1] x [0.2] = 0.2 (both inputs!)

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

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

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

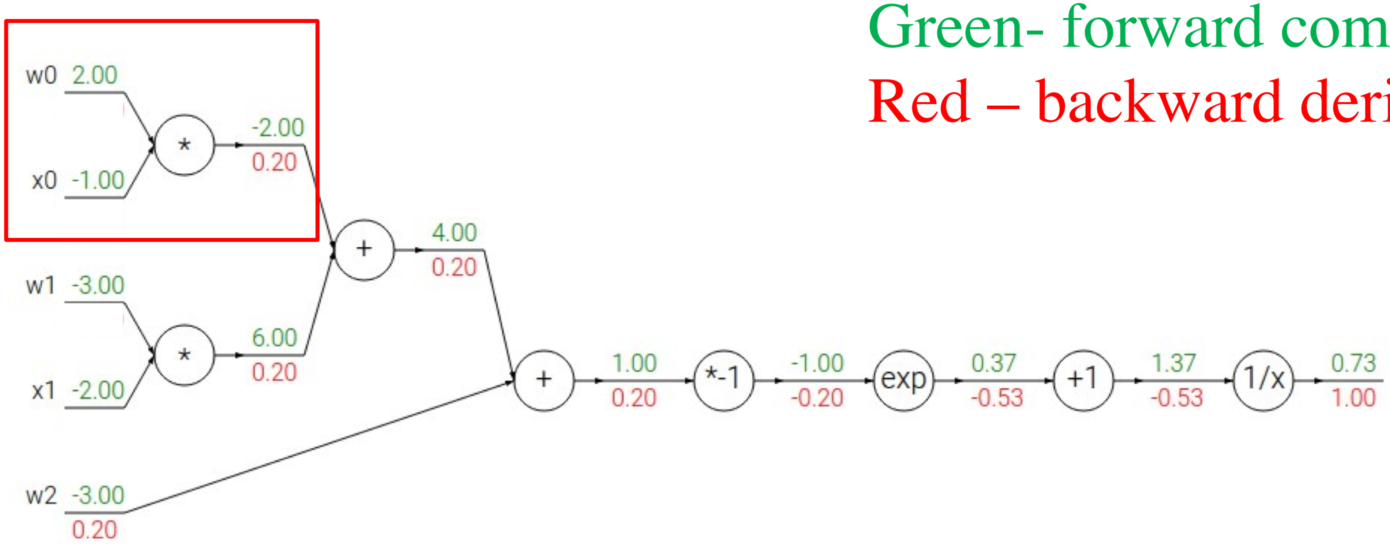
$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Green- forward computation

Red – backward derivatives



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

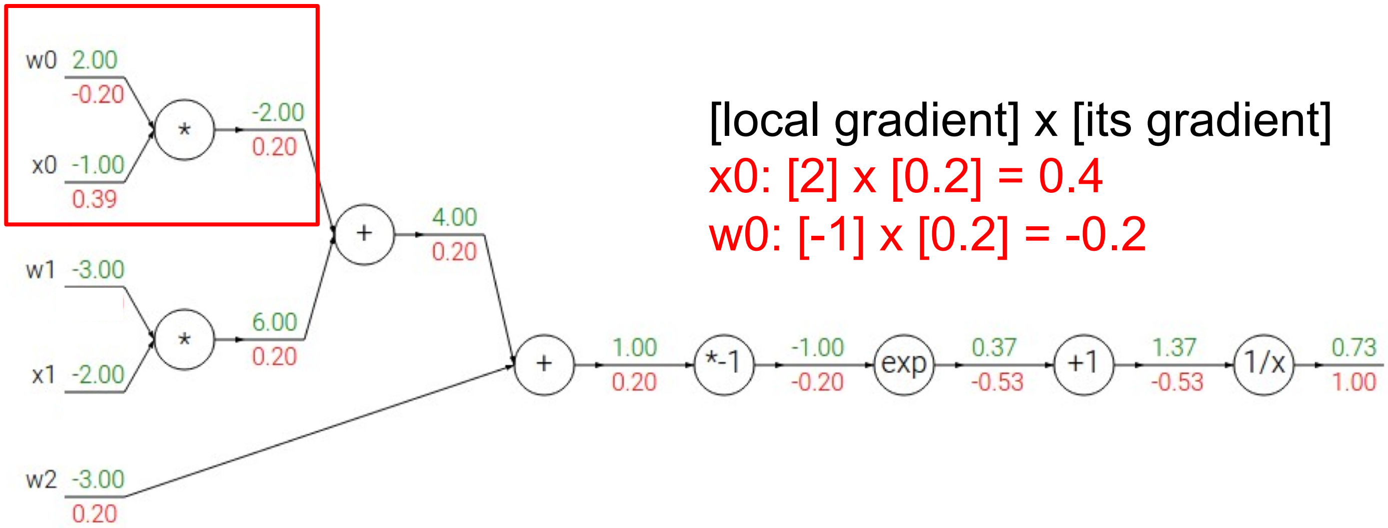
$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$

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

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

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [its gradient]

x0: [2] x [0.2] = 0.4

w0: [-1] x [0.2] = -0.2

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

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

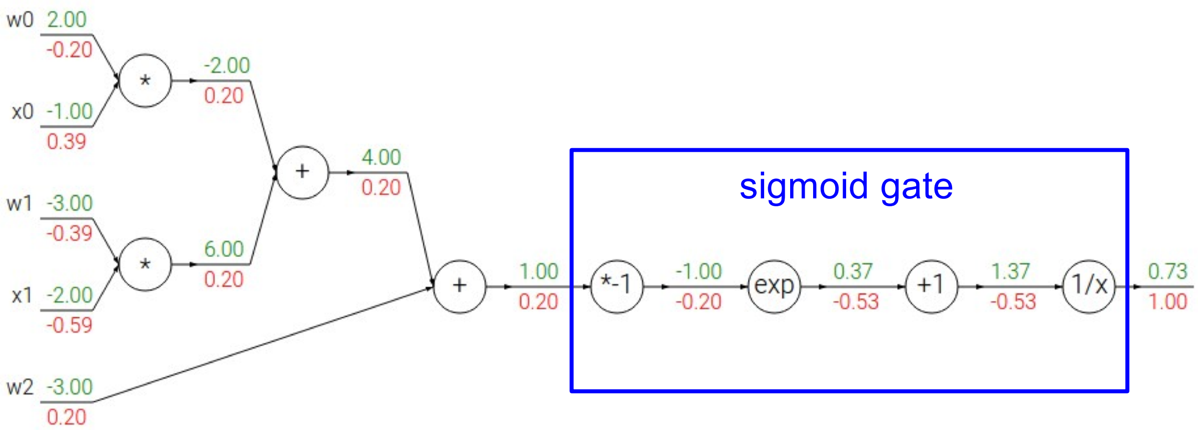
$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

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

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$

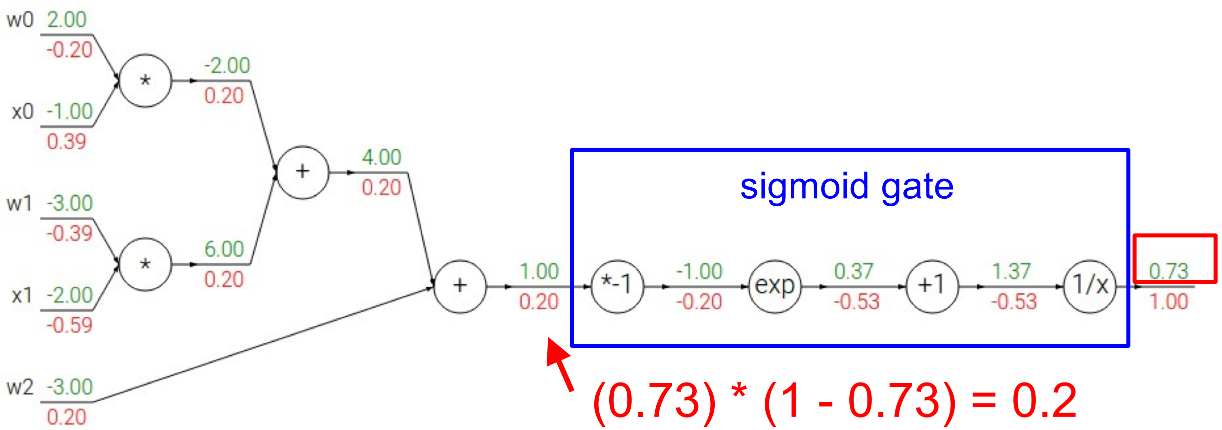


54

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

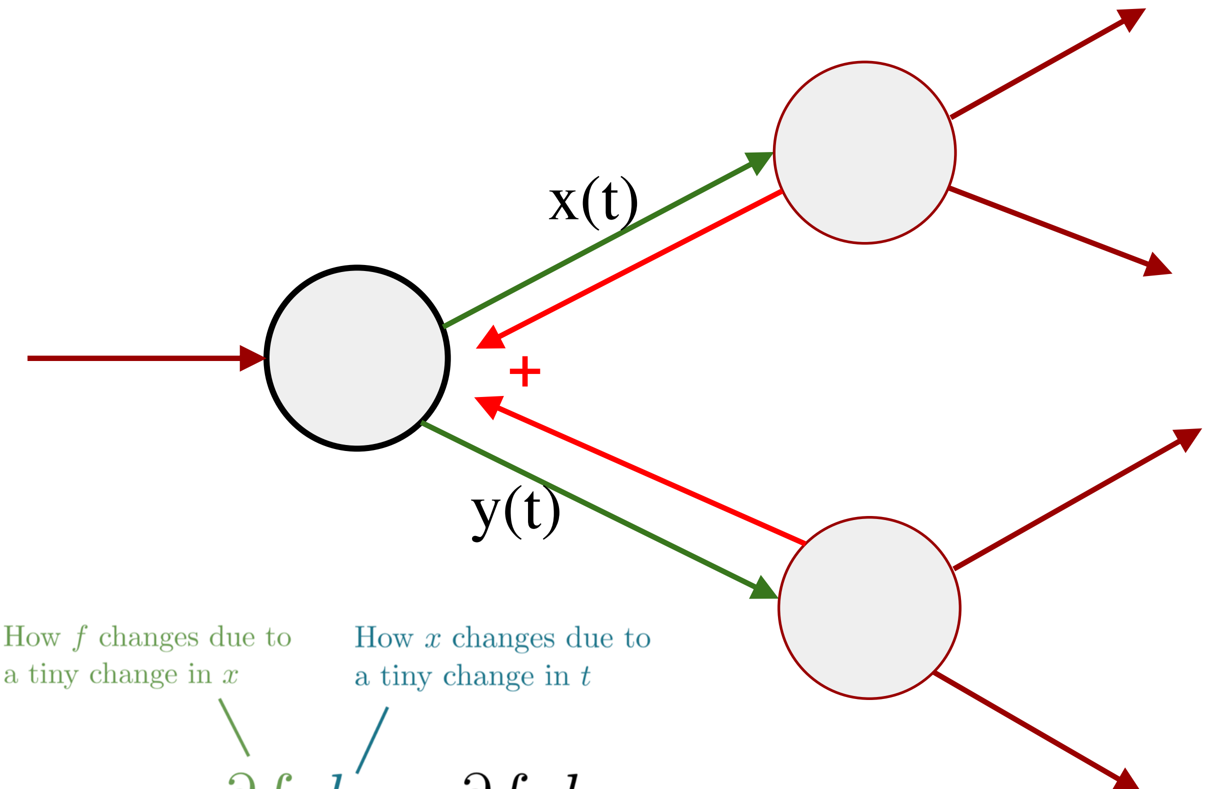
$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



Green- forward computation

Red – backward derivatives

Gradients add at branches



How f changes due to a tiny change in x

How x changes due to a tiny change in t

$$\frac{d}{dt} f(x(t), y(t)) = \underbrace{\frac{\partial f}{\partial x} \frac{dx}{dt}}_{\text{Total change in } f \text{ due to the influence } t \text{ has on } x} + \underbrace{\frac{\partial f}{\partial y} \frac{dy}{dt}}_{\text{Total change in } f \text{ due to the influence } t \text{ has on } y}$$

This is an ordinary derivative not a partial derivative $\frac{\partial}{\partial t}$, because the total composition has one input and one output.

Total change in f due to the influence t has on x

Total change in f due to the influence t has on y

Summary

- SGD
  - Simple linear classifier
  - Complex classification prediction functions
- Computing partial derivates algorithmically
  - Forward propagation to compute intermediate function values
  - Backward propagation to compute derivates
- Deep learning
  - New direction for text data processing given its success in image/audio processing
  - Frameworks and software
    - TensorFlow (Google).
    - Others: Theano, Torch, CAFFE, computation graph toolkit (CGT)