# Neural Networks and Back Propagation


**Dr. Santosh Kumar Vipparthi**

Dept. of C.S.E

Website: https://visionintelligence.github.io/

**Malaviya National Institute of Technology (MNIT), Jaipur**

## What is a Feature?

A feature is a significant piece of information extracted from an image which provides more detailed understanding of the image.

## A Picture Is Worth More Than A Thousand Words

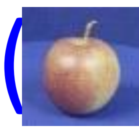
Image analysis (understanding),

image processing & computer vision plays an important role in society today because

- A picture gives a much clearer impression of a situation or an object.
- Having an accurate visual perspective of things has a high social, technical and economic value.

# The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f(  ) = “apple”

f(  ) = “tomato”

f(  ) = “cow”

# The machine learning framework

$$y = f(\mathbf{x})$$

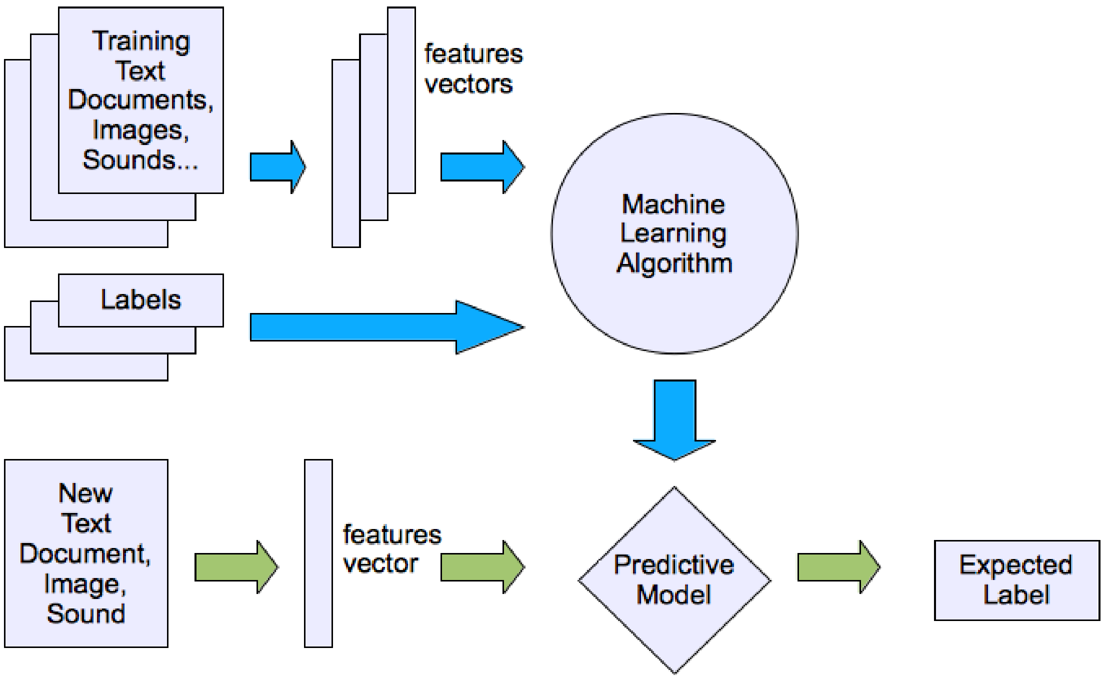
output
prediction function
Image feature

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function $f$ by minimizing the prediction error on the training set

- **Testing:** apply $f$ to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

# Machine learning structure




System Framework

Slide credit: L. Lazebnik

# Steps

## Training

Training Images

Image Features → Training → Learned model

Training Labels → Training

## Testing

Test Image

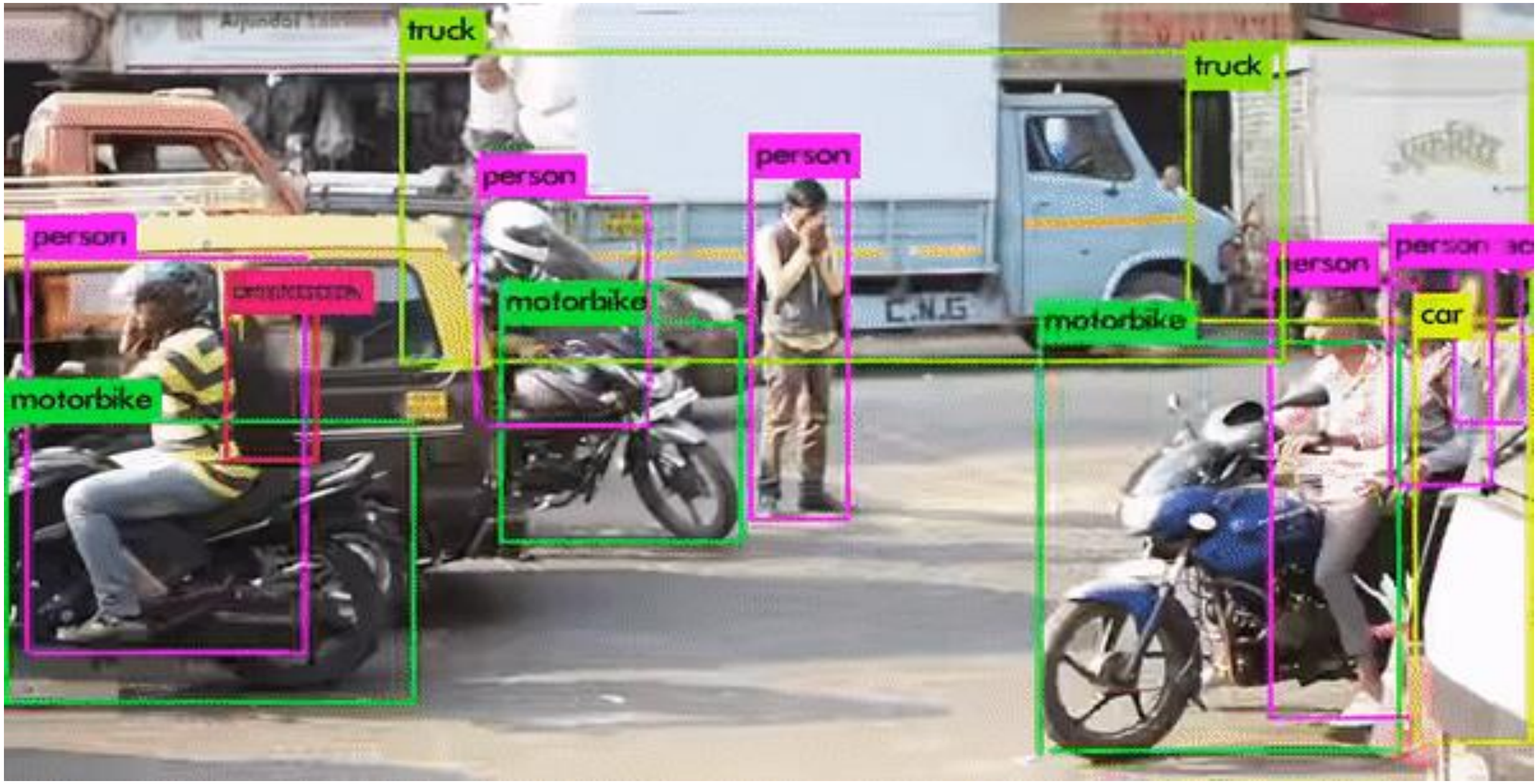
Image Features → Learned model → Prediction

Slide credit: D. Hoiem and L. Lazebnik

# Image Retrieval


Slide credit: L. Lazebnik

# Object Detection & Classification




Slide credit: L. Lazebnik

# Macro v/s Micro Expressions

Anger

Disgust

Fear

Happy

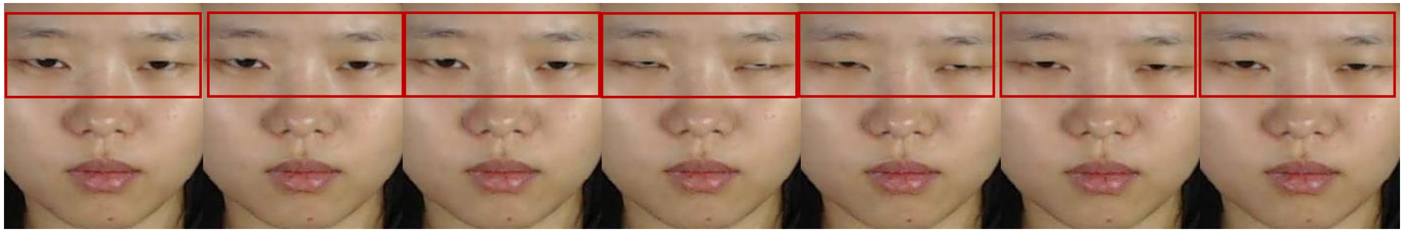
Sad

Surprise

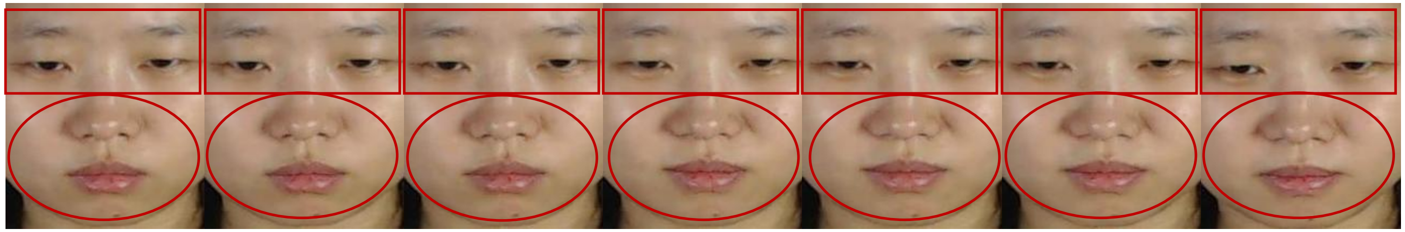
MMI Dataset

Macro Expression

# Sample Micro Expression



Disgust Expression

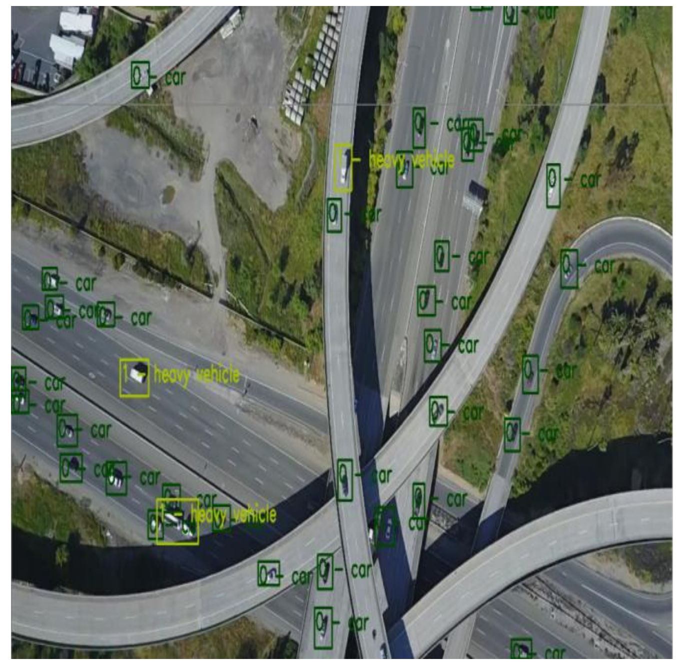


Happy expression

Slide credit: S K Vipparthi

# Regular Vs Aerial View

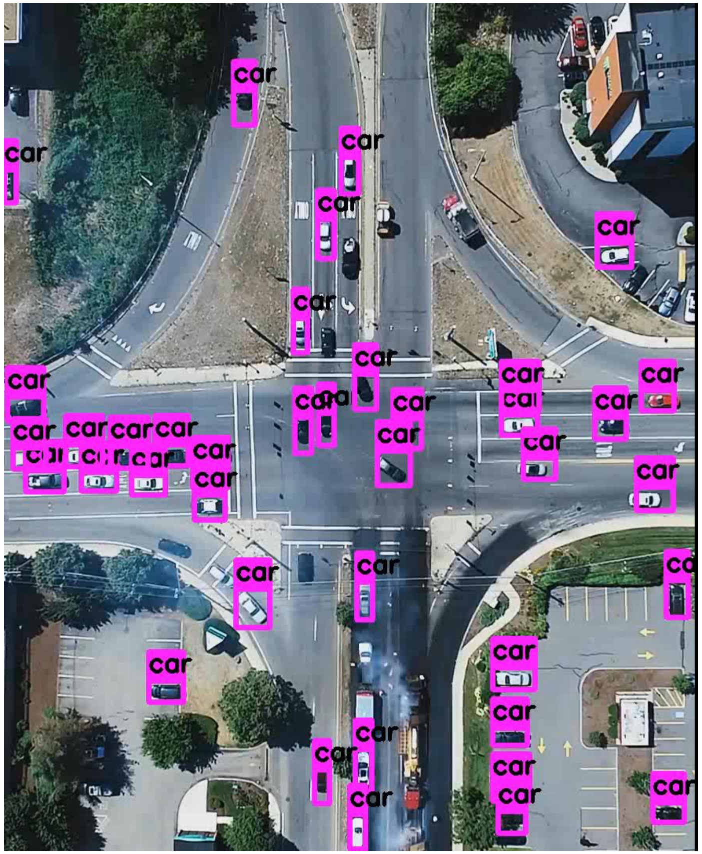

Regular View



Aerial View

Difference between regular and aerial view

# Sample Results




# Image Captioning

Automatically describing the content of an image and generate a reasonable description in plain English. NIC(Neural Image Caption) is model which take image in input and generate description.

Credit: https://www.shutterstock.com


Slide credit: S K Vipparthi

# Generalization

Training set (labels known)

Test set (labels unknown)

How well does a learned model generalize from the data it was trained on to a new test set?

Slide credit: L. Lazebnik

# Local Binary & Ternary Patterns (LBP & LTP)

**Pattern**

| | | | | |
|---|---|---|---|---|
| 9 | 7 | 5 | 2 | 1 |
| 3 | 17 | 9 | 1 | 4 |
| 6 | 8 | 9 | 14 | 5 |
| 2 | 2 | 10 | 7 | 6 |
| 1 | 9 | 8 | 9 | 10 |

**LBP**

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | | 1 |
| 0 | 1 | 0 |

| | | |
|---|---|---|
| 8 | 4 | 2 |
| 16 | | 1 |
| 32 | 64 | 128 |

| | | |
|---|---|---|
| | | |
| | 93 | |
| | | |

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 0 | | 1 |
| -1 | 0 | 0 |

$$\tau = 2[7,11]$$

**LTP**

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | | 1 |
| 0 | 0 | 0 |

| | | |
|---|---|---|
| 8 | 4 | 2 |
| 16 | | 1 |
| 32 | 64 | 128 |

| | | |
|---|---|---|
| | | |
| | 9 | |
| | | |

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 0 | | 0 |
| 1 | 0 | 0 |

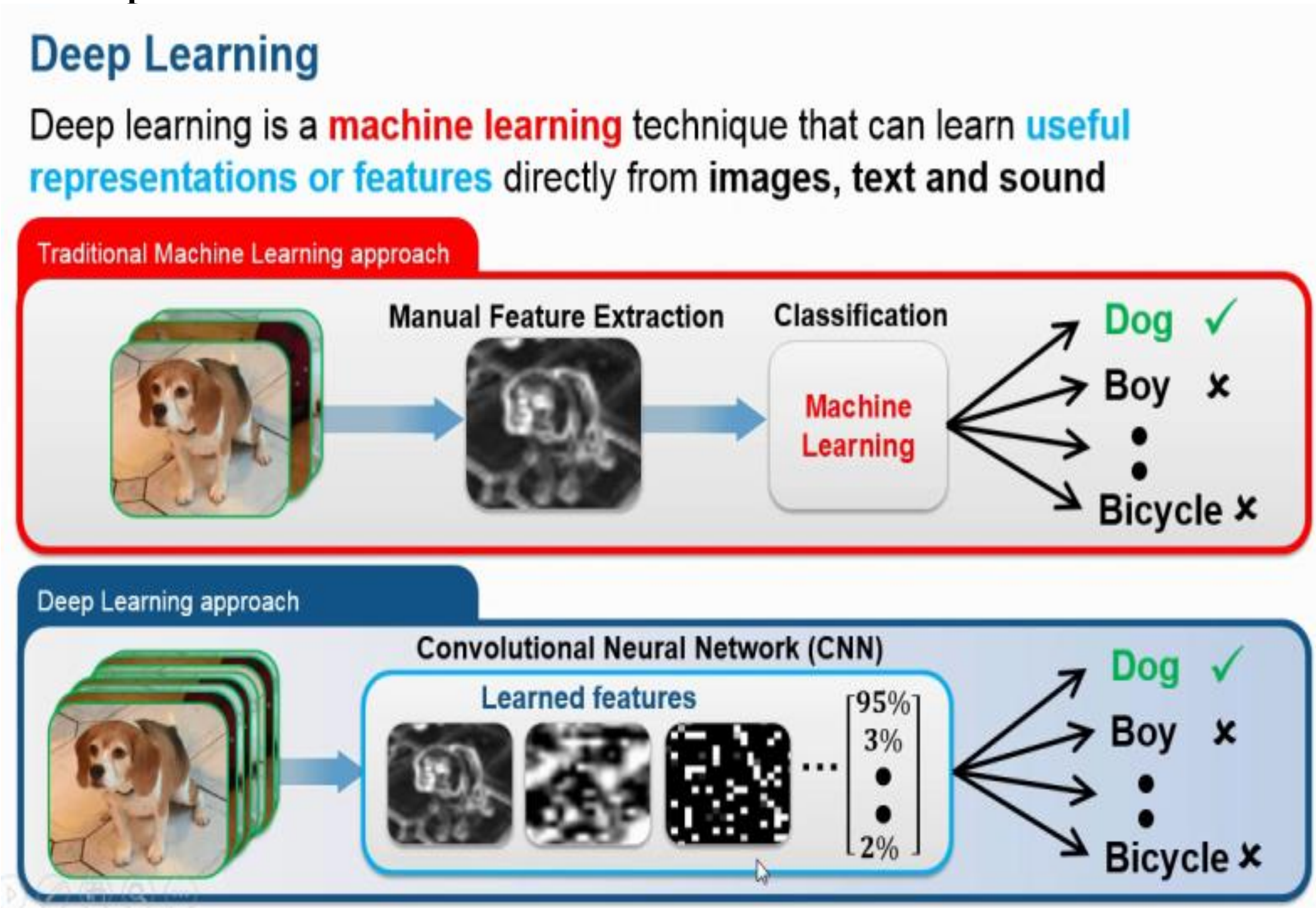
| | | |
|---|---|---|
| 8 | 4 | 2 |
| 16 | | 1 |
| 32 | 64 | 128 |

| | | |
|---|---|---|
| | | |
| | 34 | |
| | | |

Fig: Example of obtaining LBP and LTP for the $3 \times 3$ pattern

Slide credit: S K Vipparthi

# Example




Slide credit: L. Lazebnik

# Quantitative Analysis

TABLE II
recognition accuracy comparison on MMI dataset

| Methods | 6-Class Exp. | 7-Class Exp. |
|---|---|---|
| **LBP [9]** | 76.5 | 81.7 |
| **Two-Phase [10]** | 75.4 | 82.0 |
| **LDP [11]** | 80.5 | 84.0 |
| **LDN [12]** | 80.5 | 83.0 |
| **LDTexP [13]** | 83.4 | 86.0 |
| **LDTerP [14]** | 80.6 | 80.0 |
| **Spatio-Temopral* [25]** | 81.2 | - |
| **QUEST** | **83.05** | **84.0** |

TABLE III
recognition accuracy comparison on GEMEP-FERA dataset

| Methods | 5-Class Exp. | 6-Class Exp. |
|---|---|---|
| **LBP [9]** | 92.2 | 87.8 |
| **Two-Phase [10]** | 88.6 | 85.0 |
| **LDP [11]** | 94.0 | 90.0 |
| **LDN [12]** | 93.4 | 91.0 |
| **LDTexP [13]** | 94.0 | 91.8 |
| **QUEST** | **94.3** | **91.33** |

Slide credit: S K Vipparthi

# Feed Forward & Backpropagation in Neural Networks

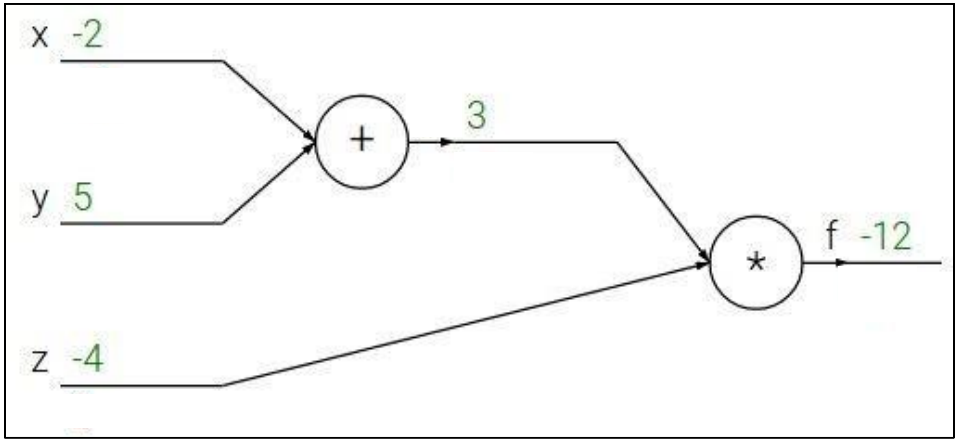
Credits to:

1. http://cs231n.stanford.edu/
2. http://cs231n.github.io/optimization-2/
3. http://neuralnetworksanddeeplearning.com/chap2.ht3
4. https://mattmazur.com/2015/03/17/

Backpropagation: a simple example

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

e.g. x = -2, y = 5, z = -4



## Backpropagation: a simple example

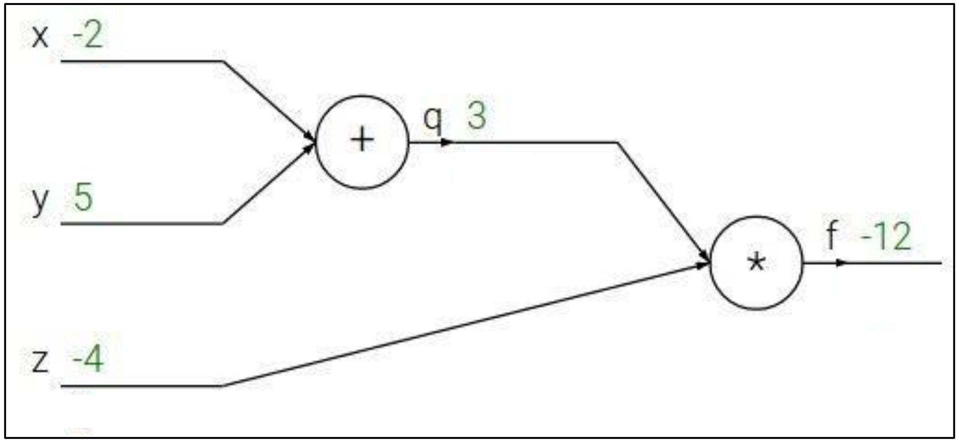
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

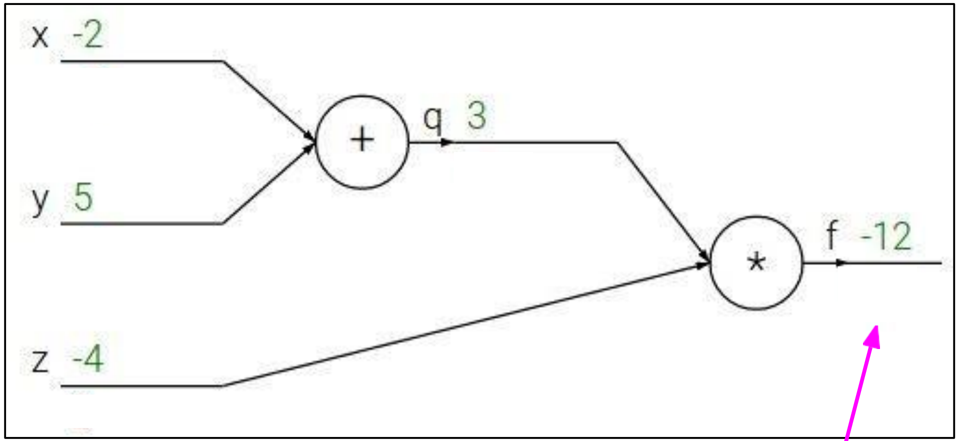
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



## Backpropagation: a simple example

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

e.g. x = -2, y = 5, z = -4




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

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

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

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

Backpropagation: a simple example

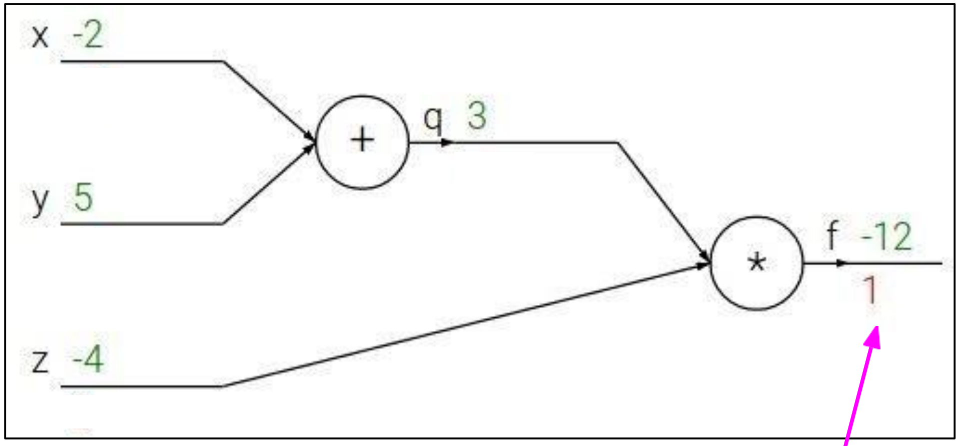
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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




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

## Backpropagation: a simple example

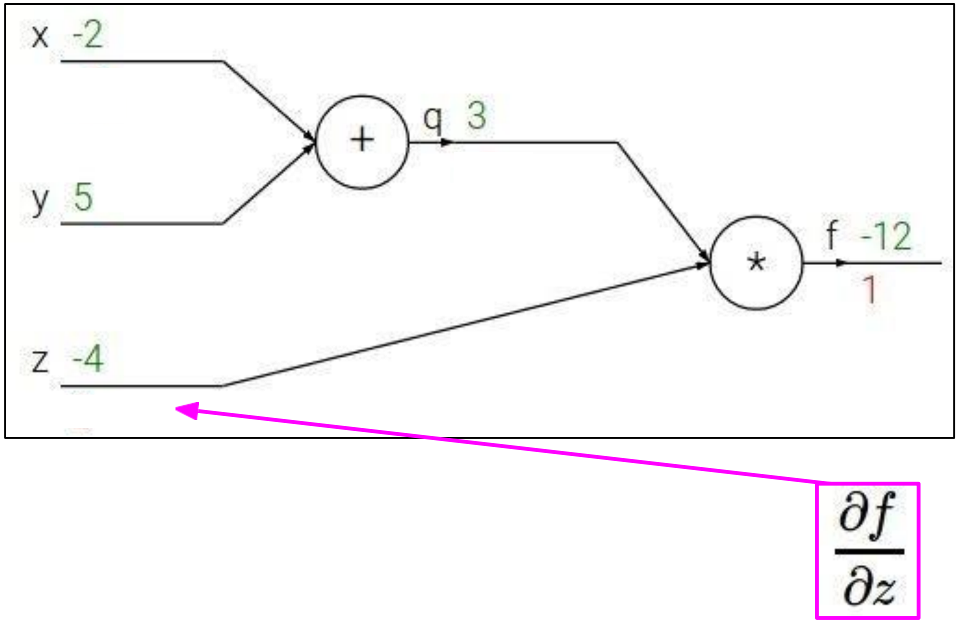
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



Backpropagation: a simple example

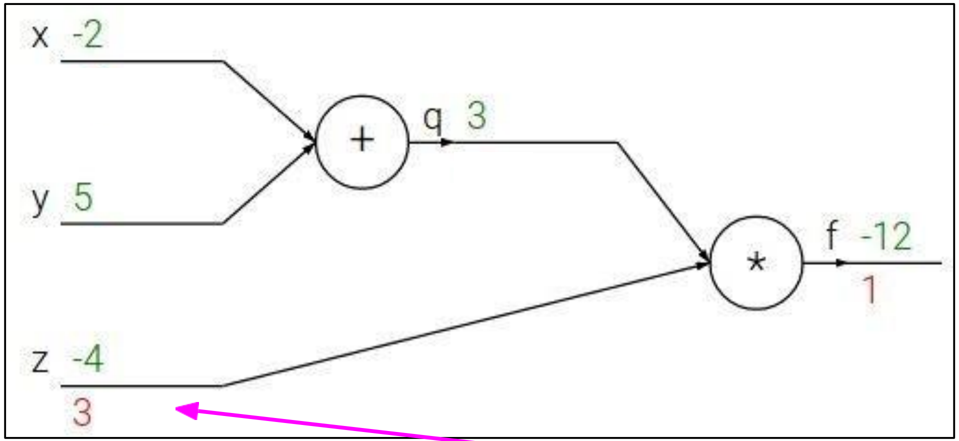
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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




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

Slide credit: Fei-Fei Li

## Backpropagation: a simple example

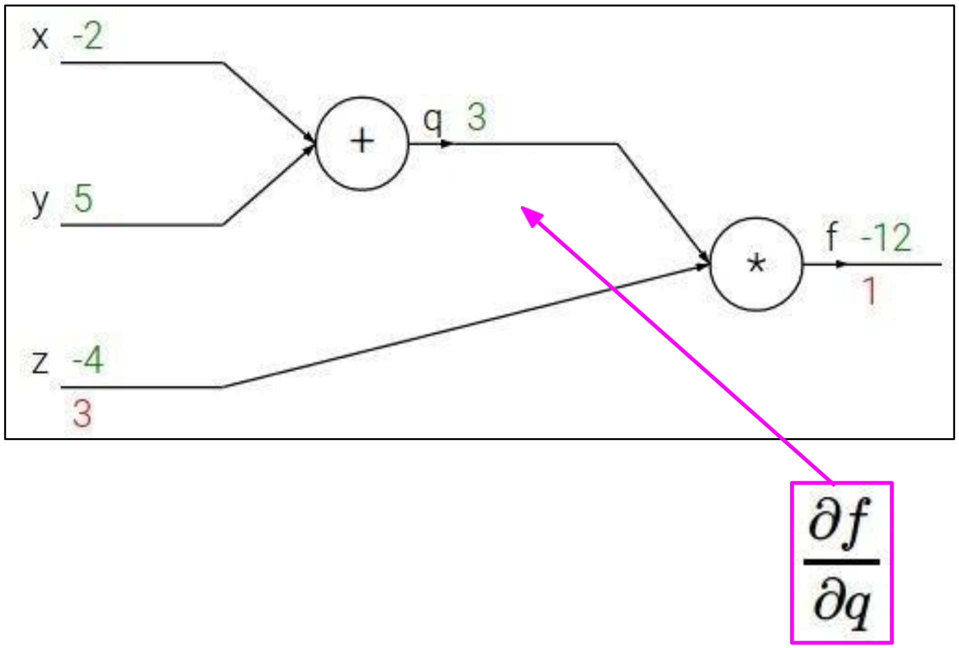
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



Backpropagation: a simple example

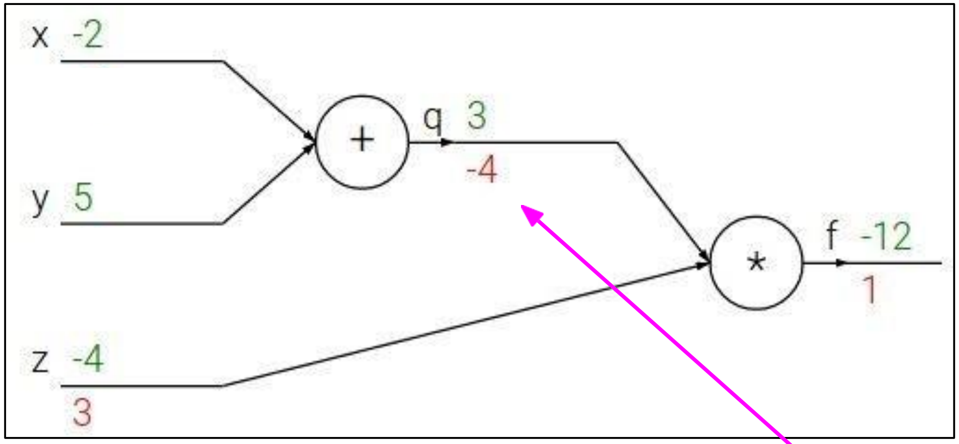
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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




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

Slide credit: Fei-Fei Li

Backpropagation: a simple example

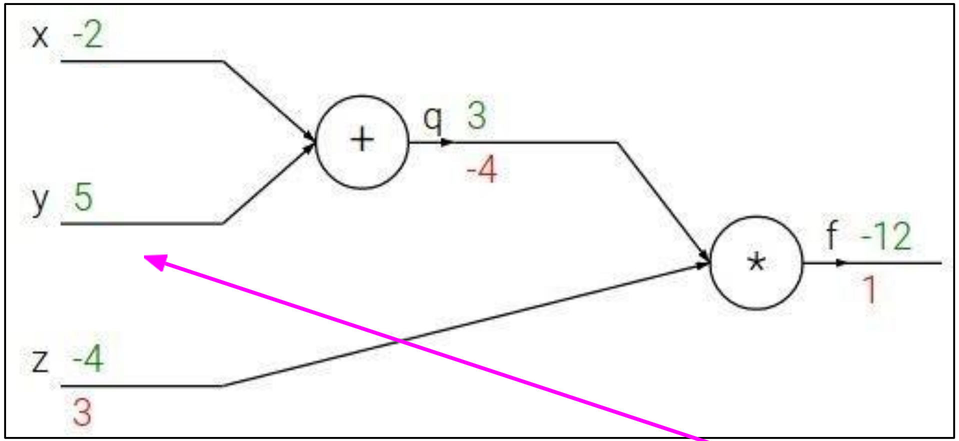
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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




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

# Backpropagation: a simple example

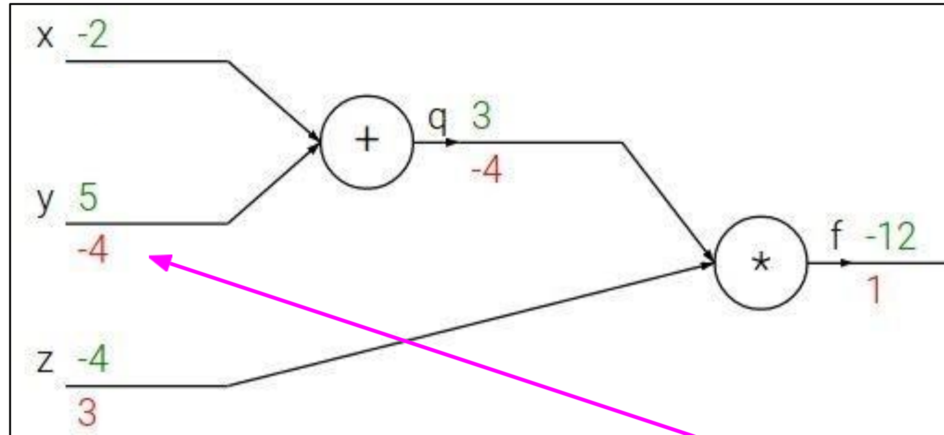
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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




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

Chain rule:

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

Backpropagation: a simple example

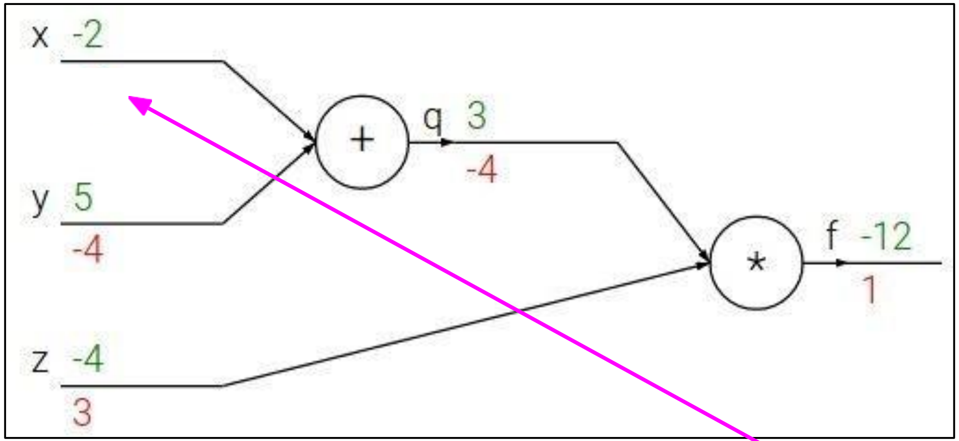
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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




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

Slide credit: Fei-Fei Li

## Backpropagation: a simple example

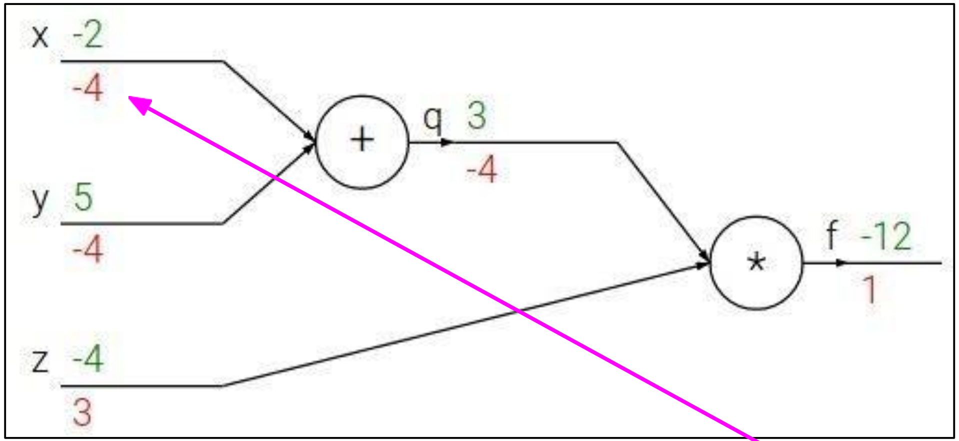
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

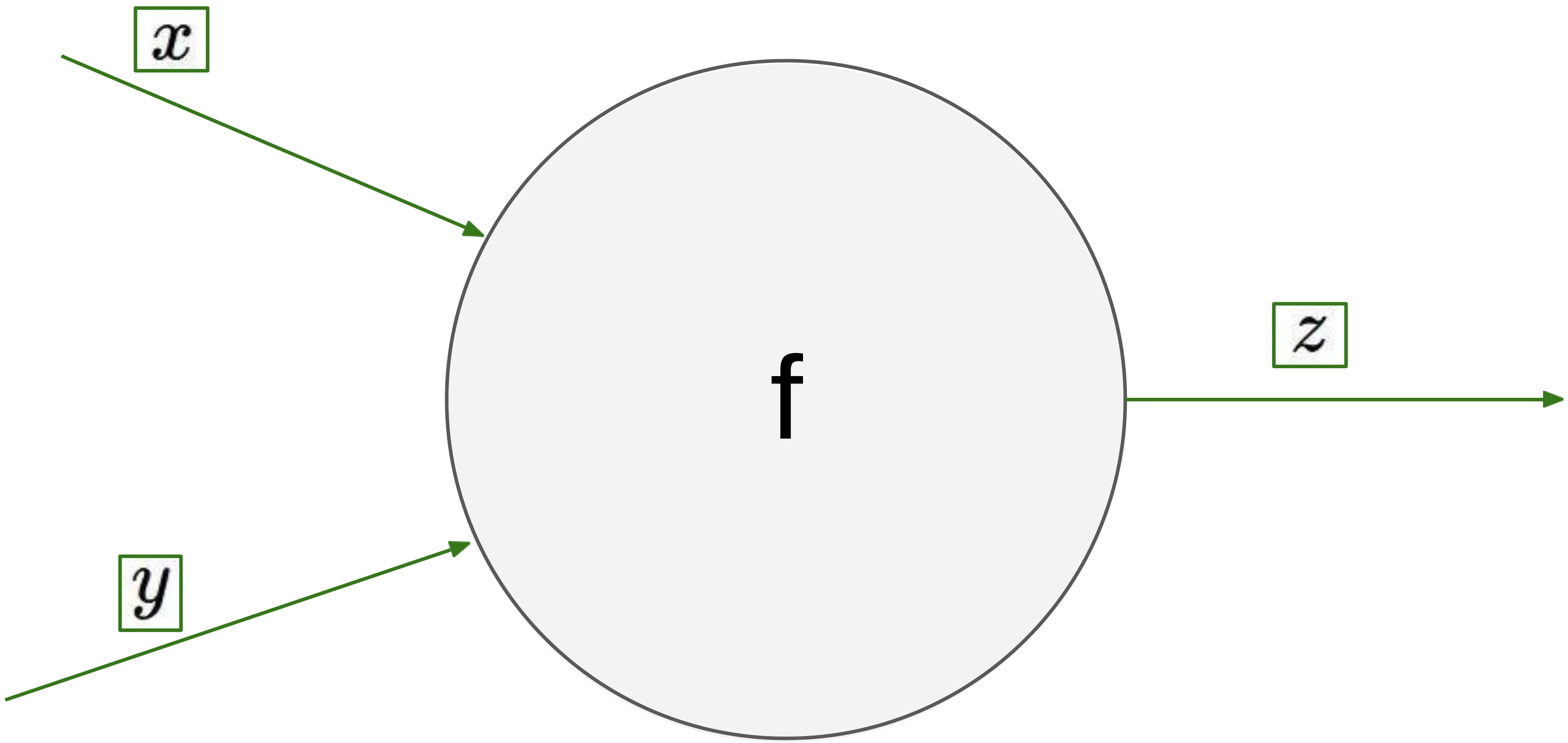
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

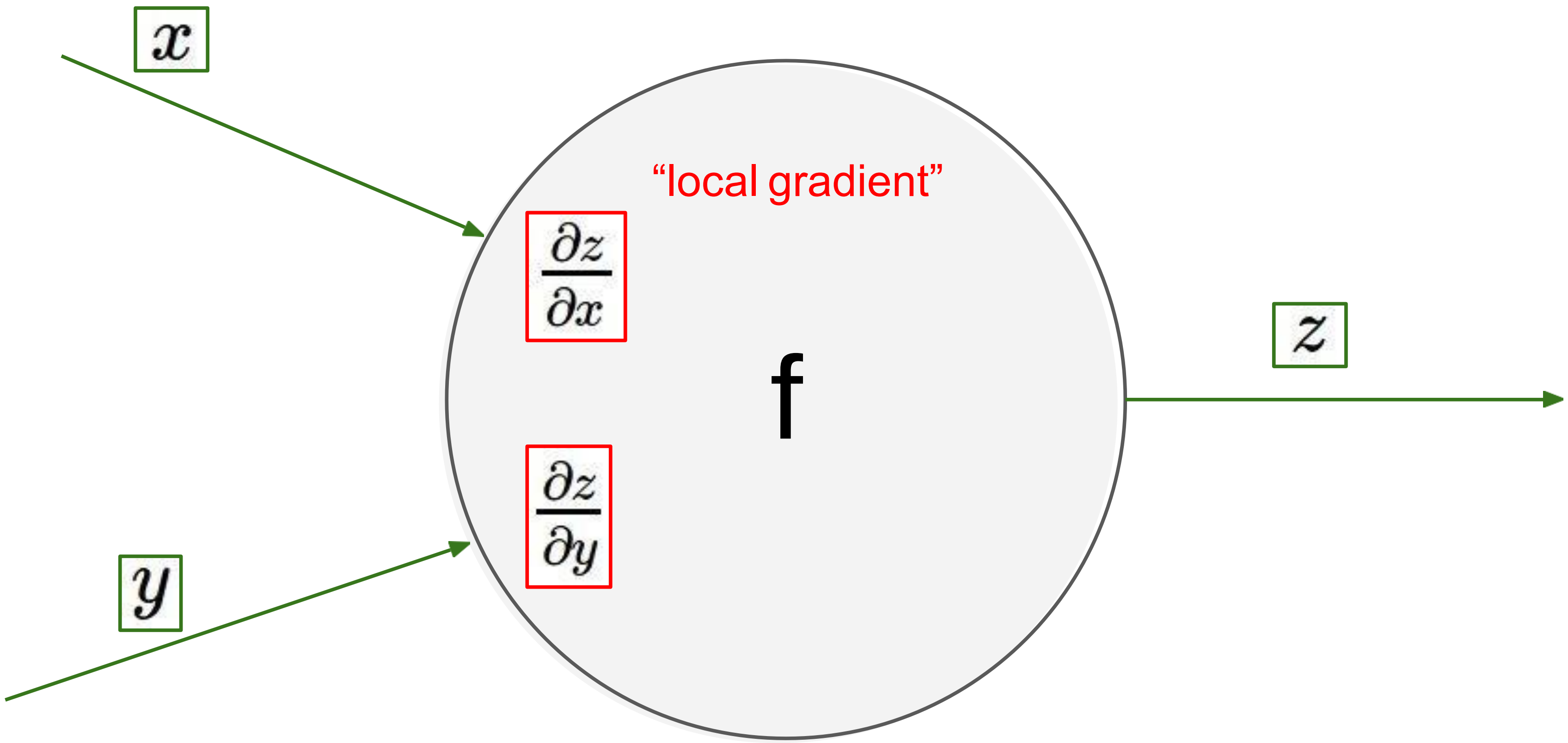


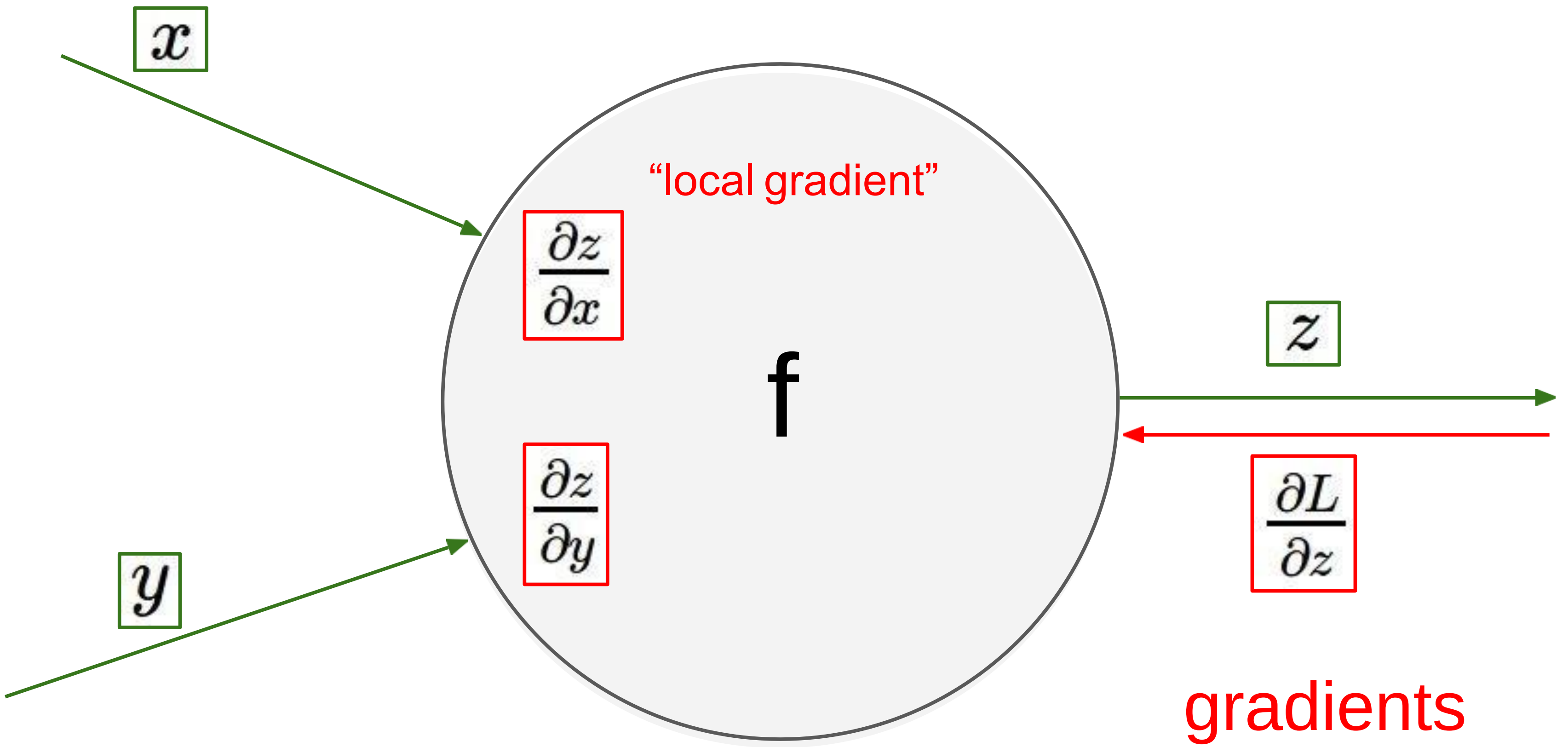

$\frac{\partial f}{\partial x}$

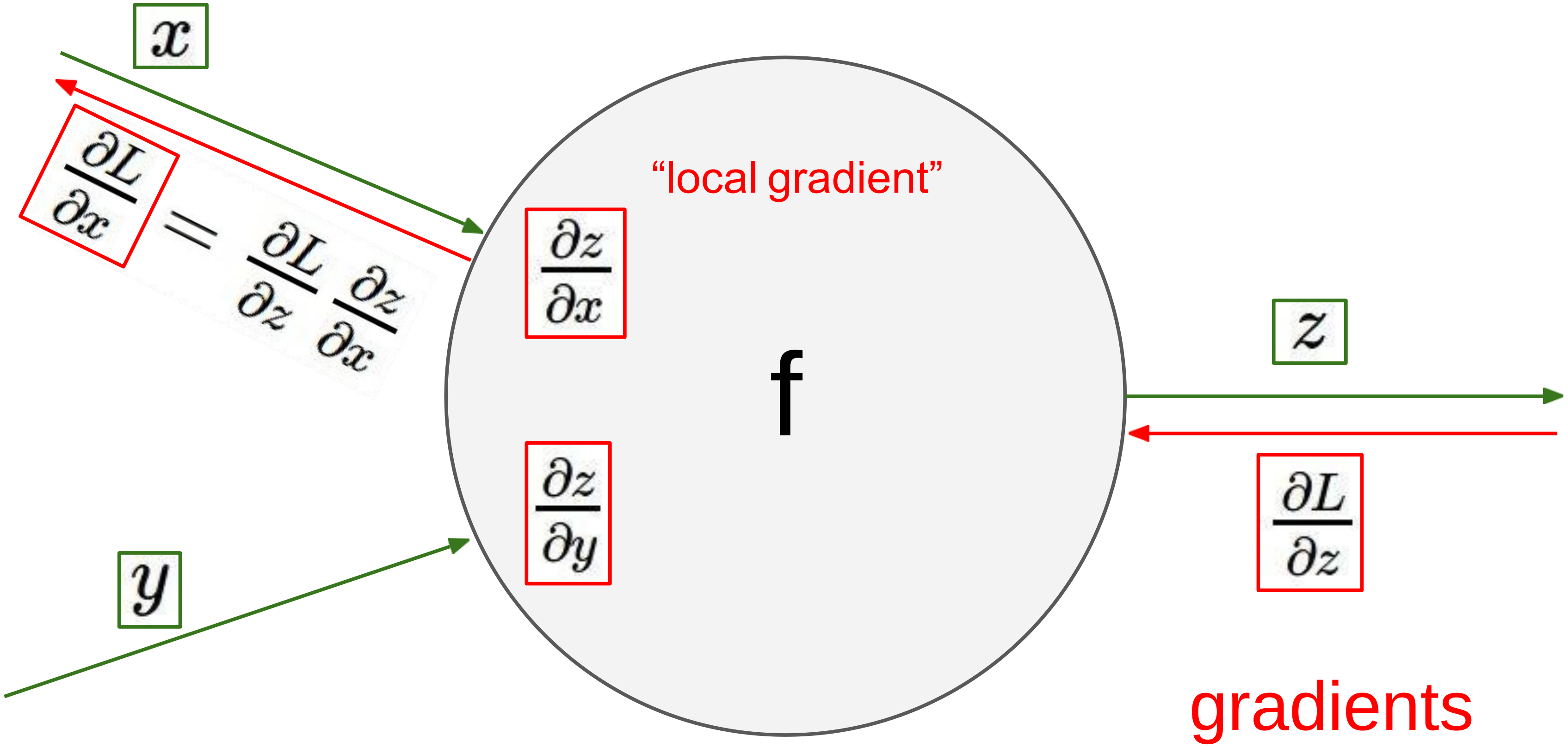
Chain rule:

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

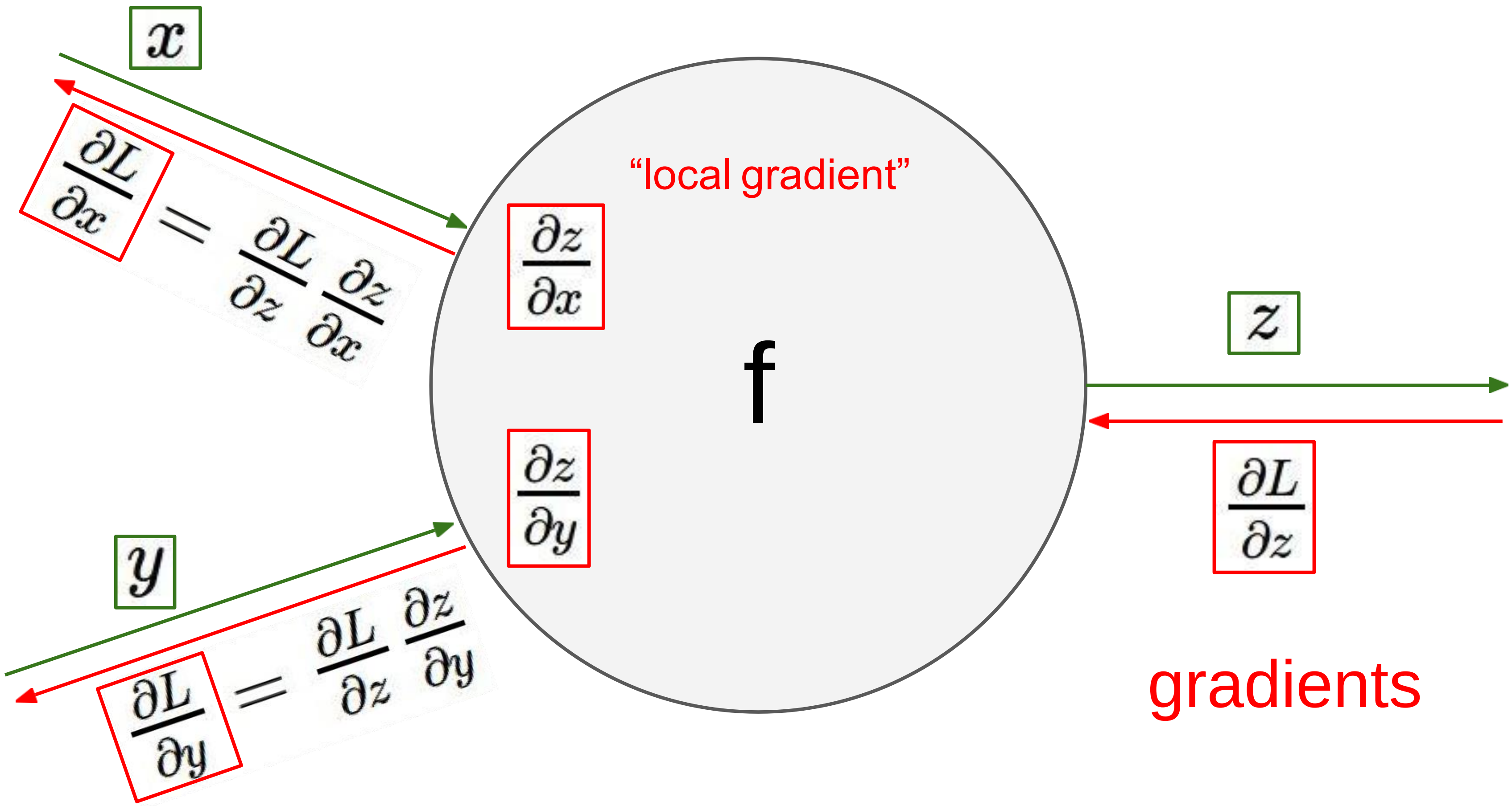

$x$
$y$
f
$z$

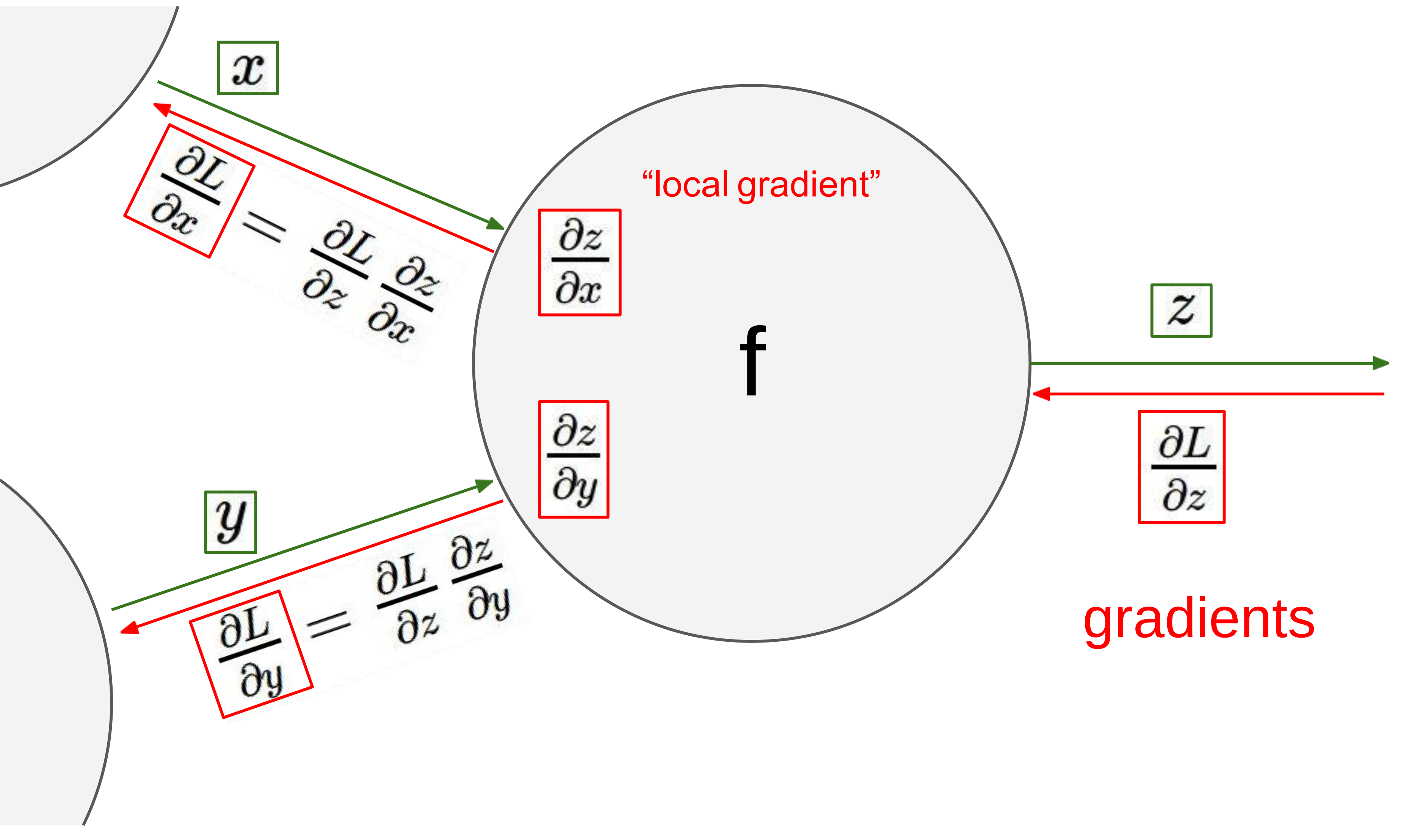

$x$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$z$
$\frac{\partial z}{\partial y}$
$y$


$x$
$y$
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
$z$
$\frac{\partial L}{\partial z}$
gradients




Slide credit: Fei-Fei Li

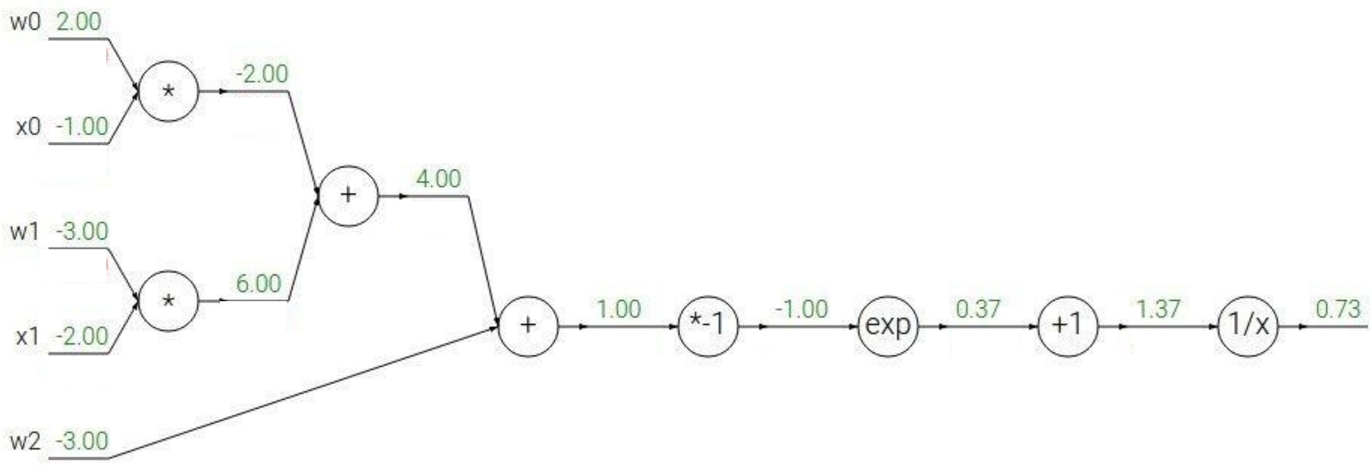

$x$
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$z$
$\frac{\partial z}{\partial y}$
$y$
$\frac{\partial L}{\partial z}$
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial y}$
gradients



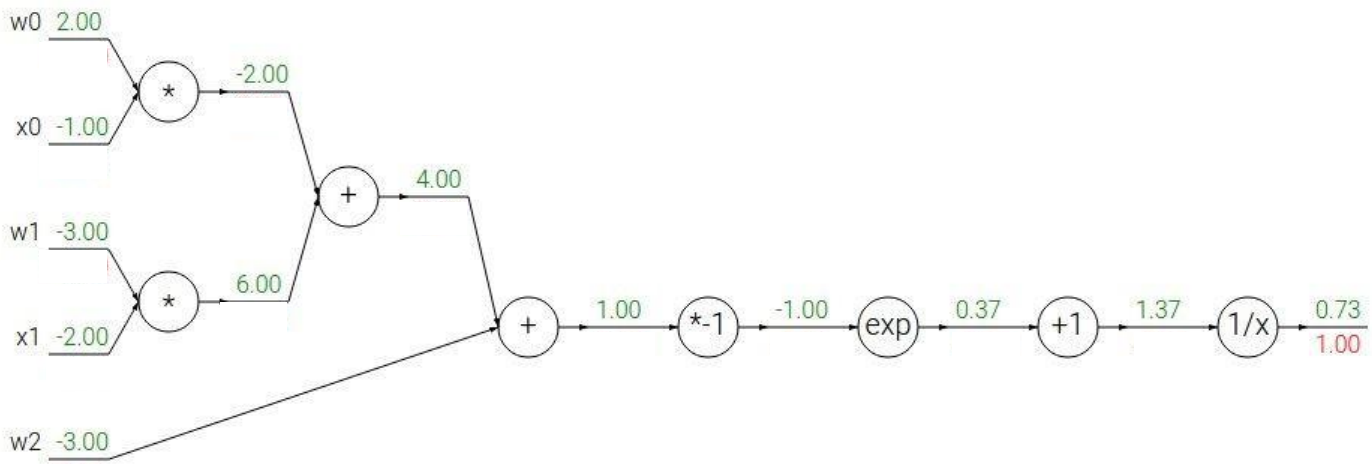

Slide credit: Fei-Fei Li

Another example:

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



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




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

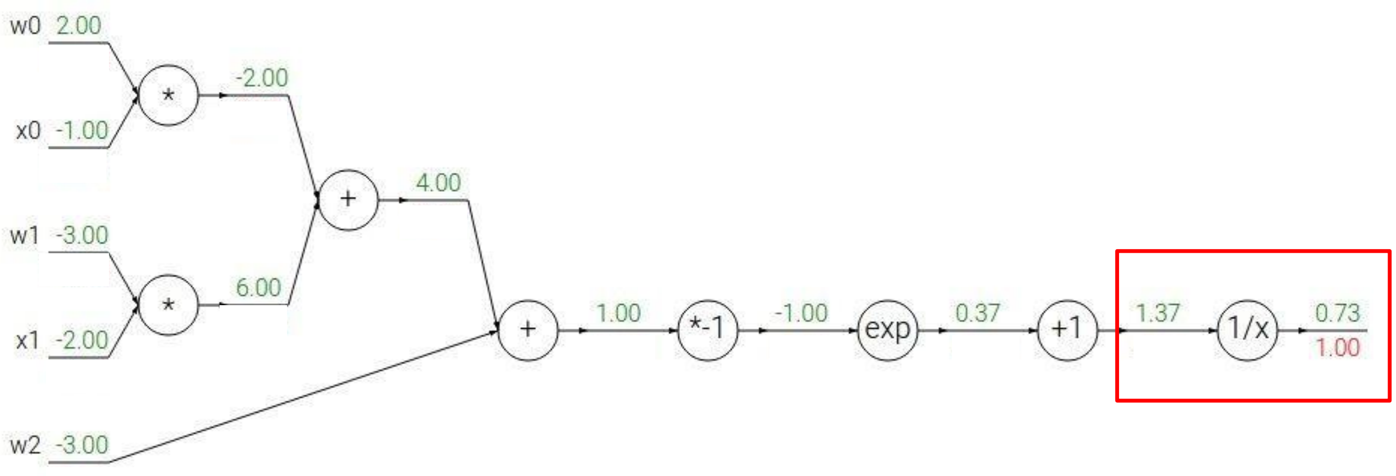
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

Slide credit: Fei-Fei Li

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




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

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

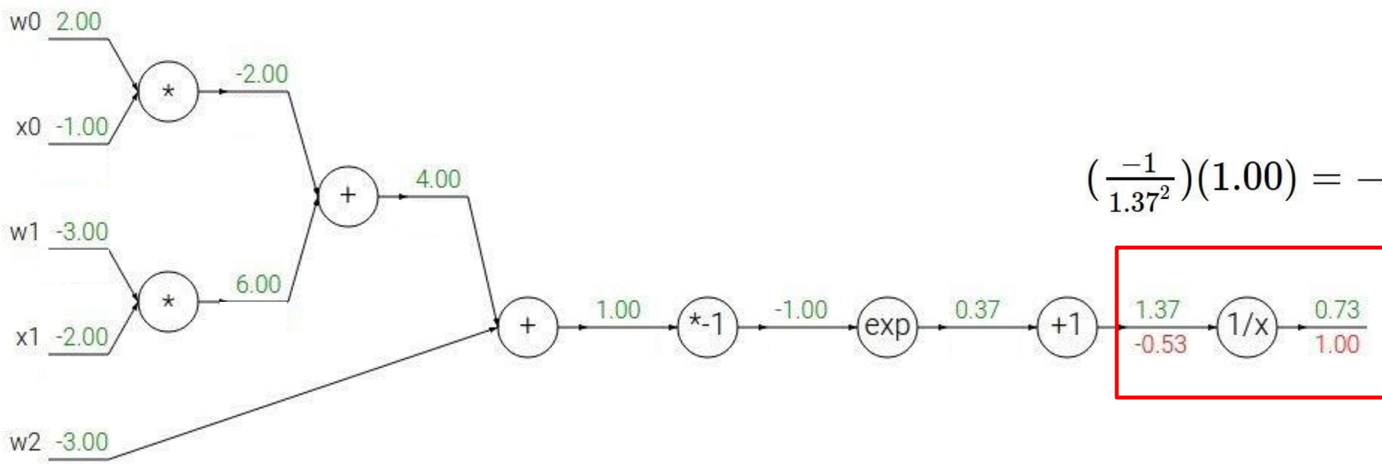
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

# Another example:

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

$$\left(\frac{-1}{1.37^2}\right)(1.00) = -0.53$$




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

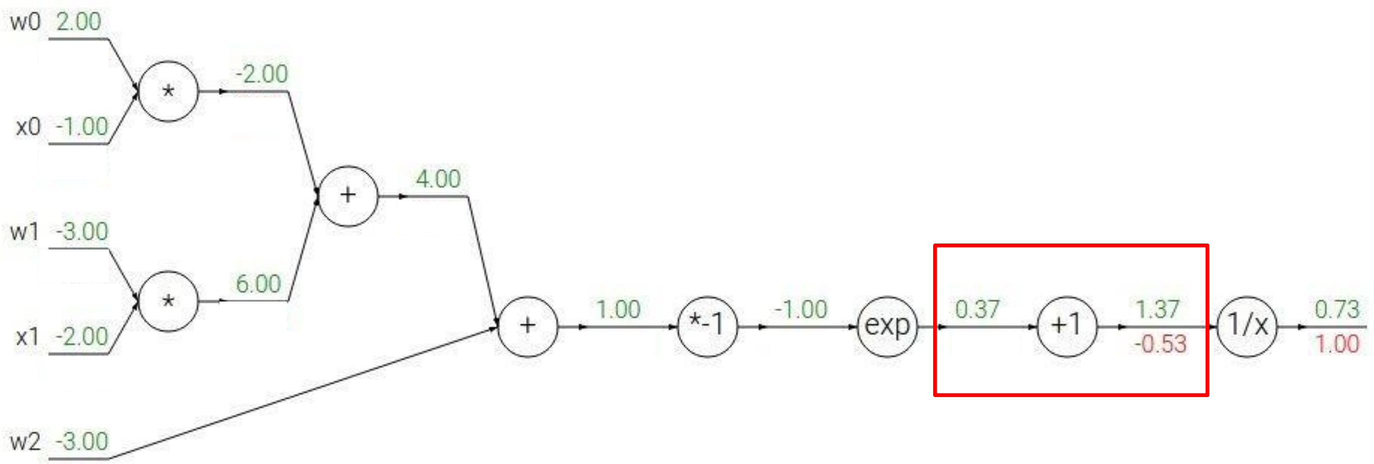
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

Another example:

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




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

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

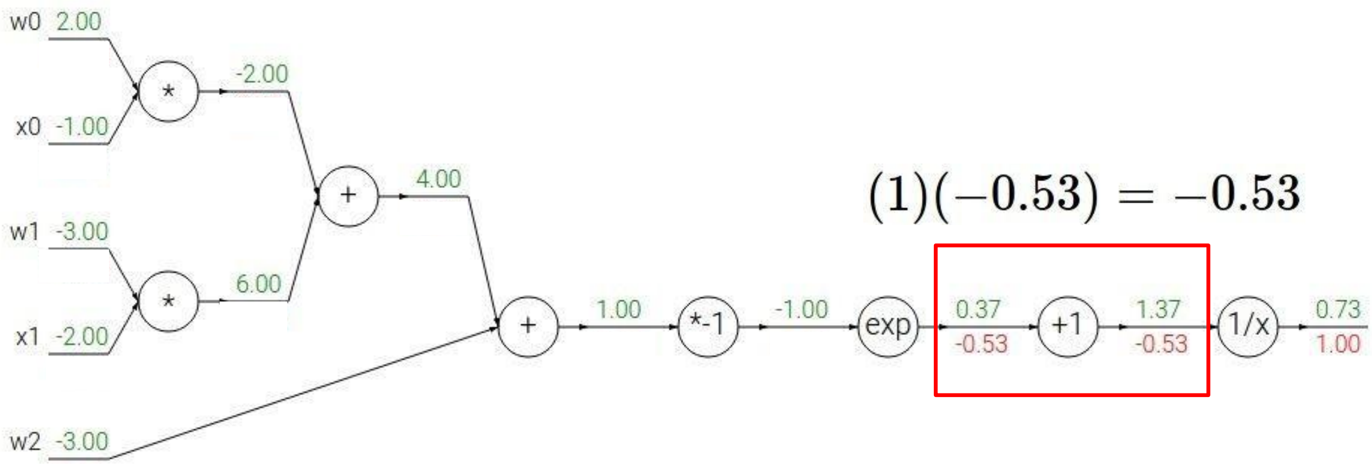
$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

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

Slide credit: Fei-Fei Li

Another example:

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




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

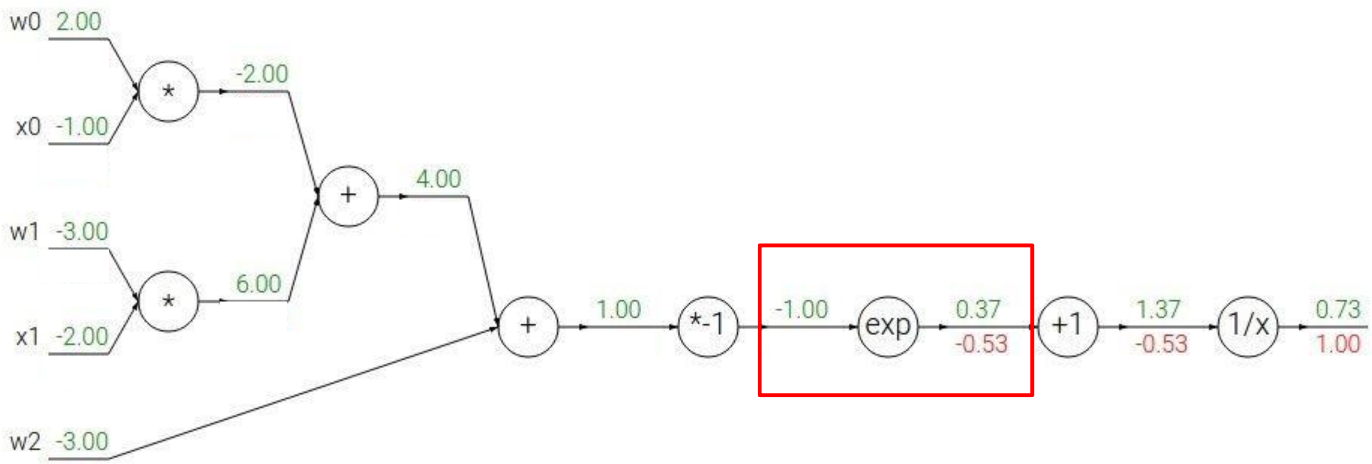
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

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

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

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

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




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

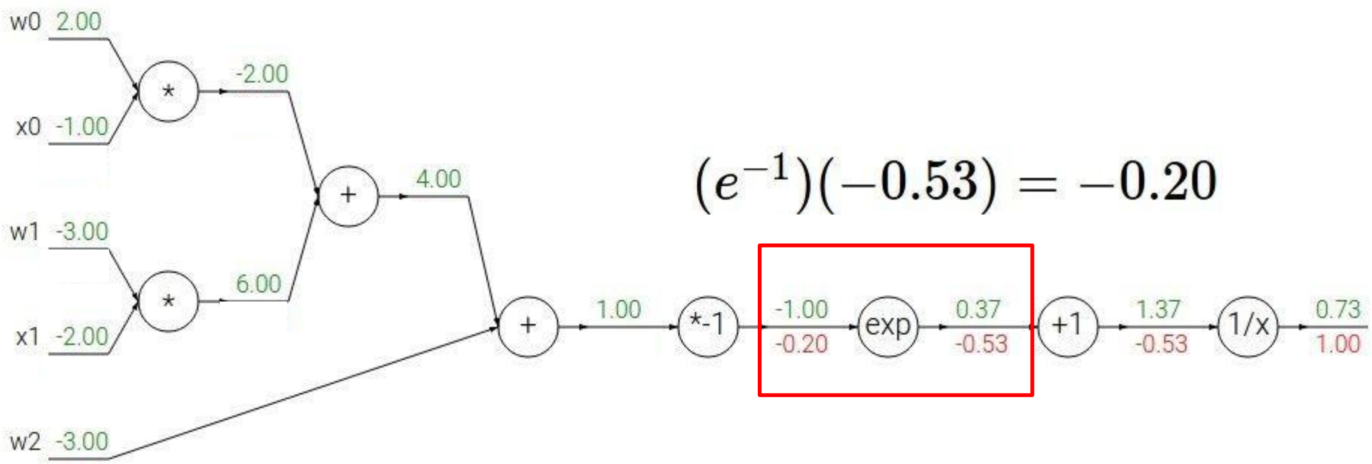
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

Another example:

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




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

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

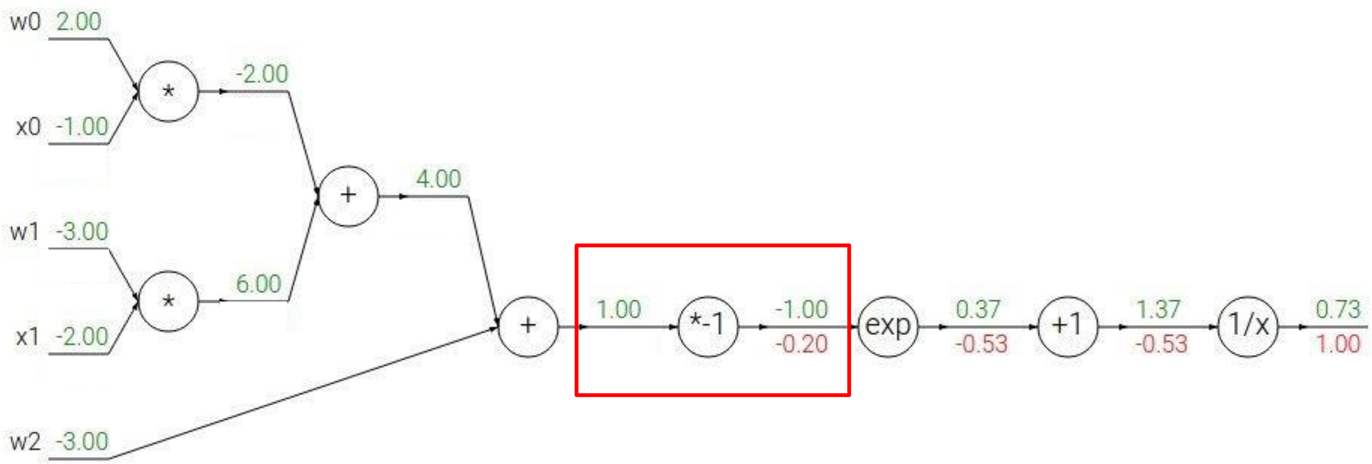
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

Slide credit: Fei-Fei Li

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




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

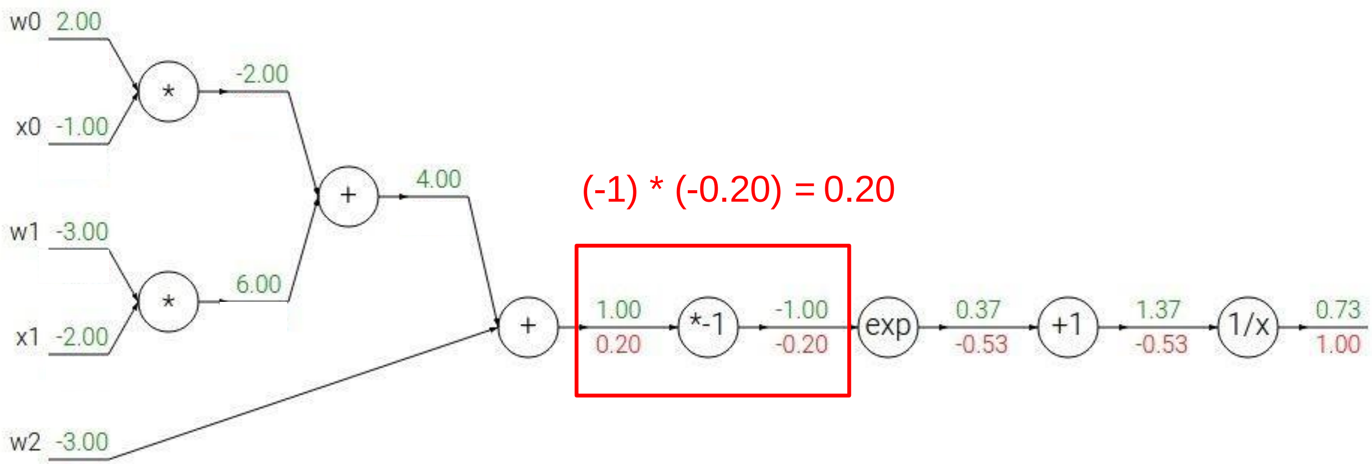
$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

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

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

Slide credit: Fei-Fei Li

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




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

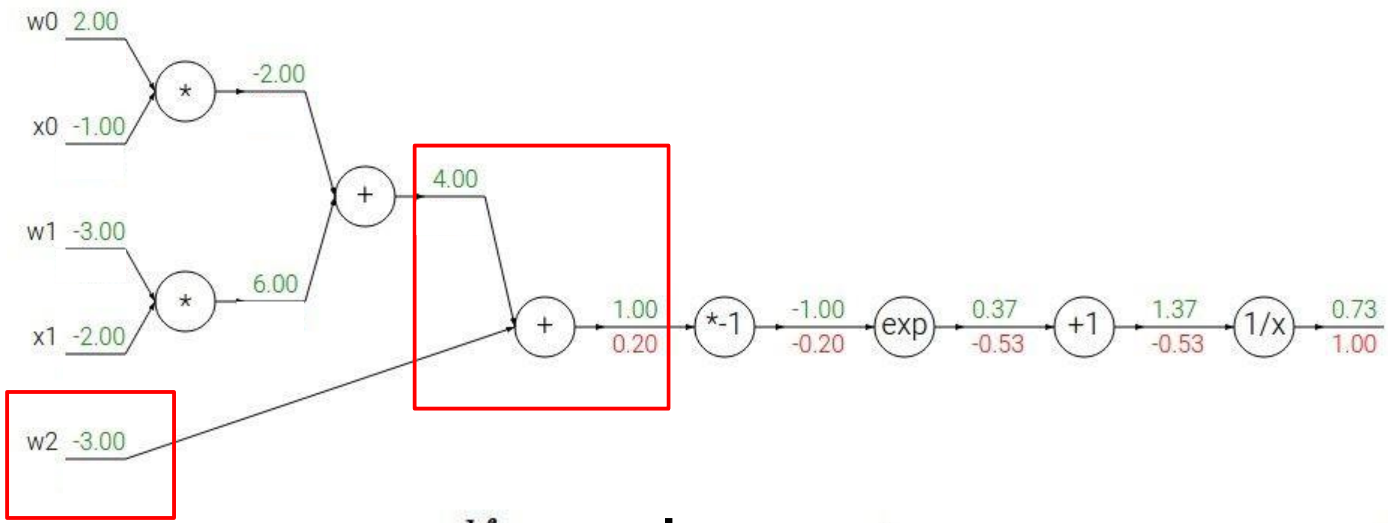
$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

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

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

# Another example:

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




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

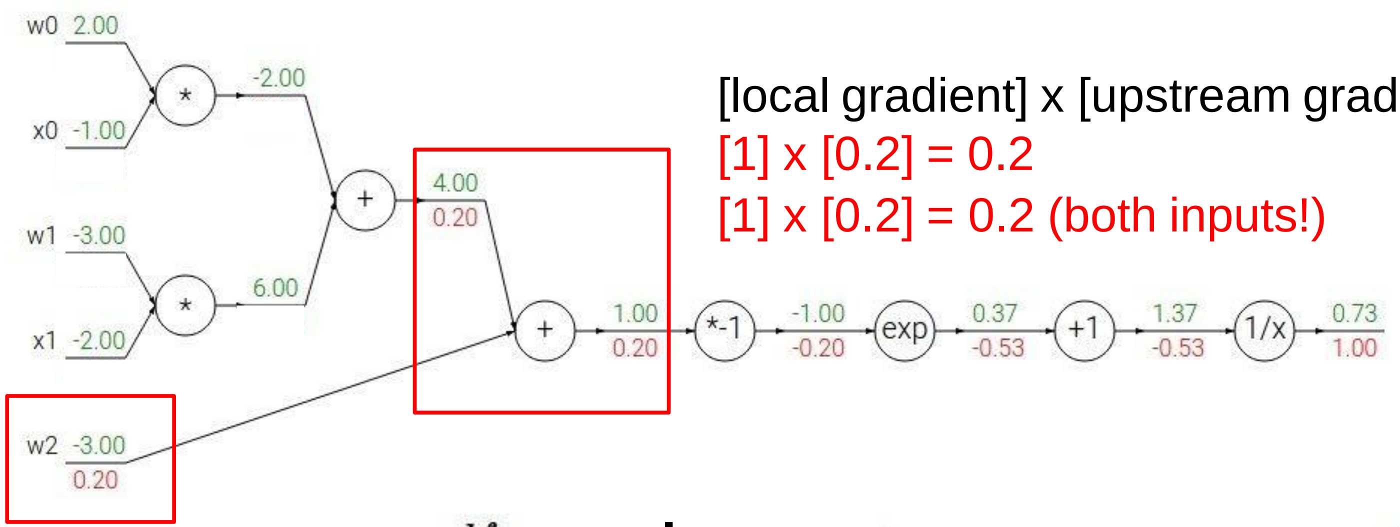
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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

Another example:

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




[local gradient] x [upstream gradient]

[1] x [0.2] = 0.2

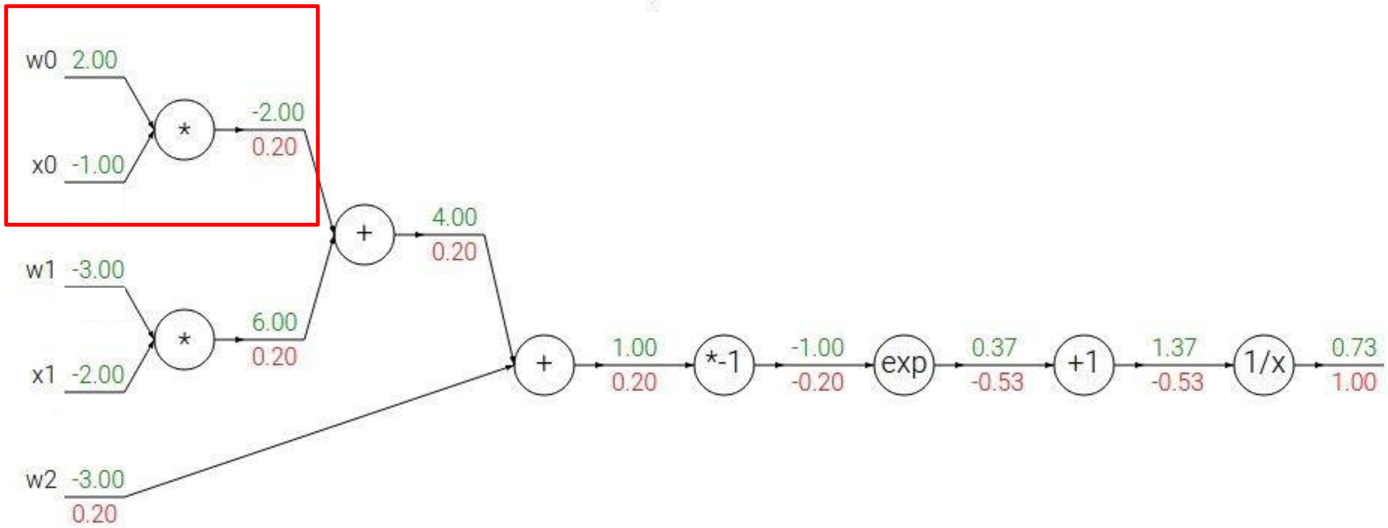
[1] x [0.2] = 0.2 (both inputs!)

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

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

Another example:

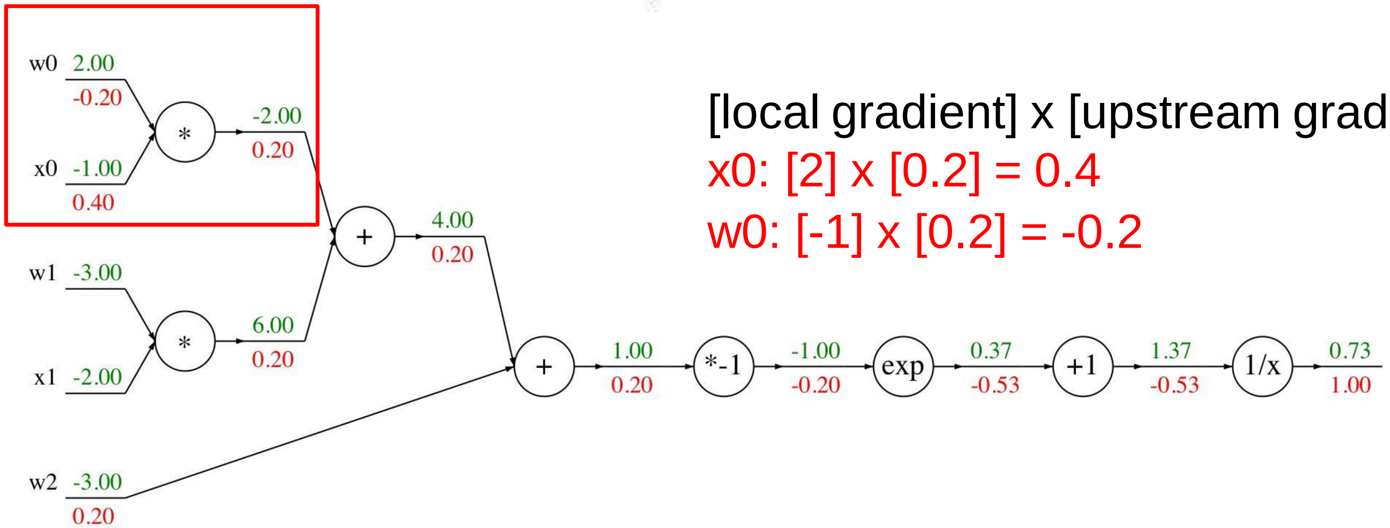
$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$




| | |
|---|---|
| $f(x) = e^x \quad \rightarrow \quad \dfrac{df}{dx} = e^x$ | $f(x) = \dfrac{1}{x} \quad \rightarrow \quad \dfrac{df}{dx} = -1/x^2$ |
| $f_a(x) = ax \quad \rightarrow \quad \dfrac{df}{dx} = a$ | $f_c(x) = c + x \quad \rightarrow \quad \dfrac{df}{dx} = 1$ |

Another example:

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




[local gradient] x [upstream gradient]

x0: [2] x [0.2] = 0.4

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

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

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

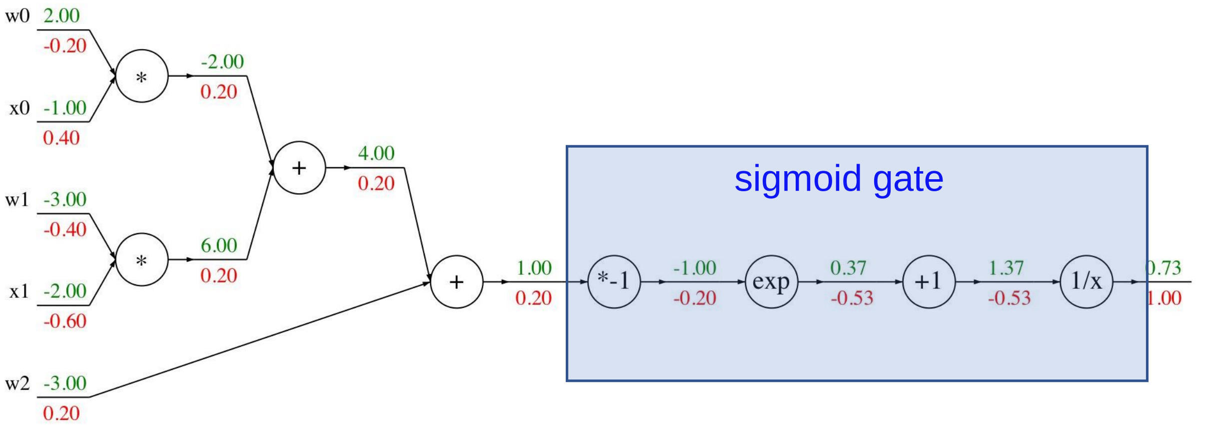
$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

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

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

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

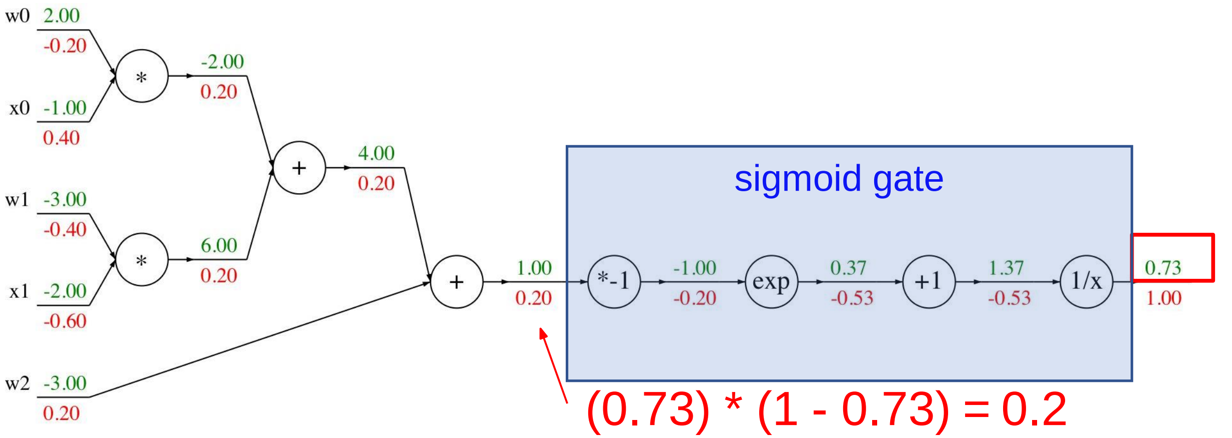
$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



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

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

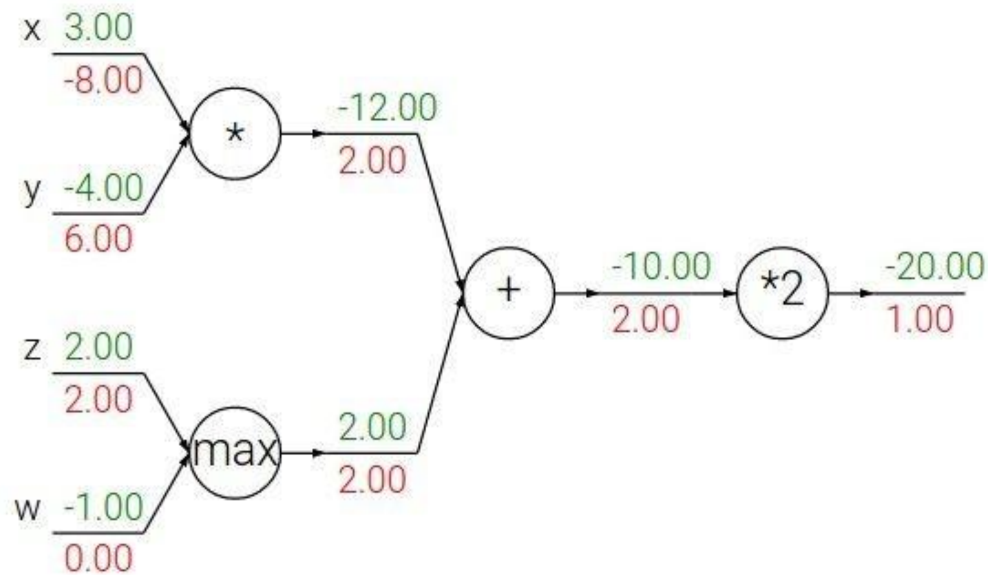
$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$




(0.73) * (1 - 0.73) = 0.2

# Patterns in backward flow

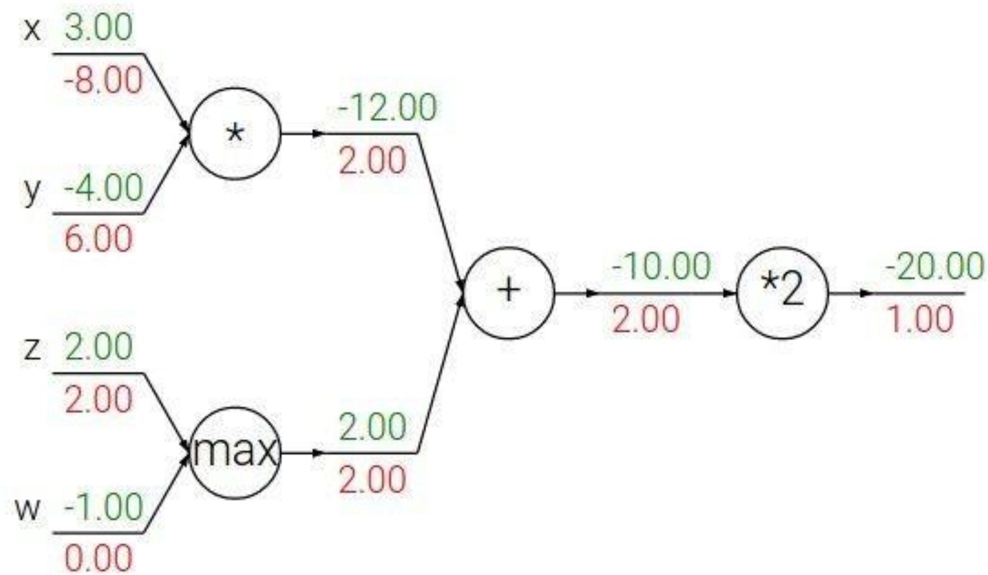
**add** gate: gradient distributor



# Patterns in backward flow

**add** gate: gradient distributor

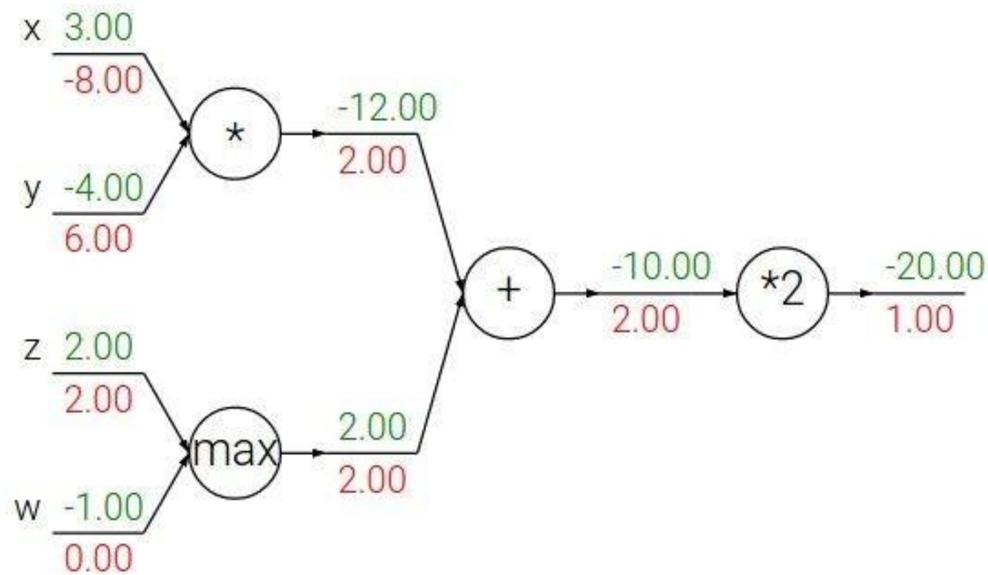
Q: What is a **max** gate?



# Patterns in backward flow

**add** gate: gradient distributor

**max** gate: gradient router

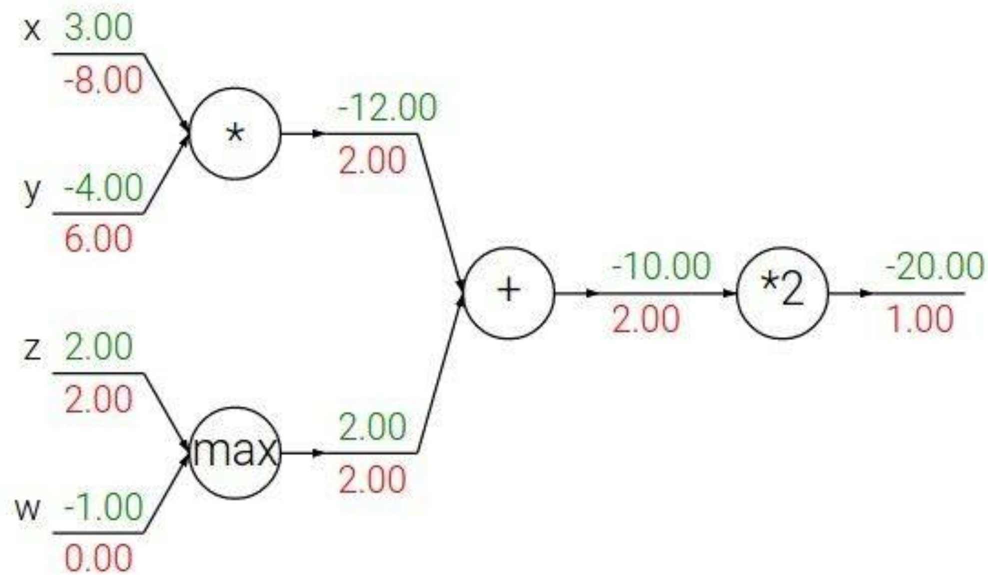


# Patterns in backward flow

**add** gate: gradient distributor

**max** gate: gradient router

Q: What is a **mul** gate?

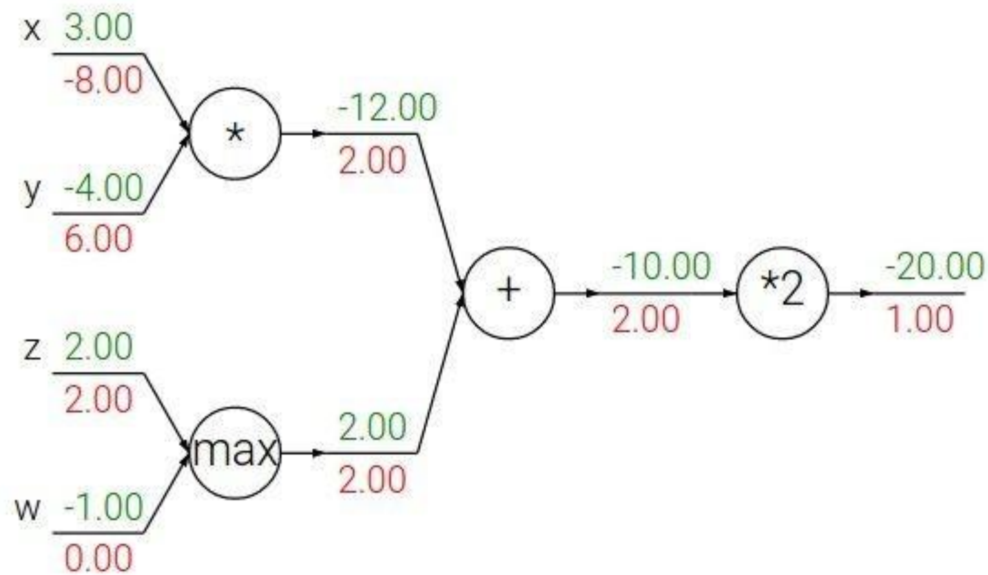


# Patterns in backward flow

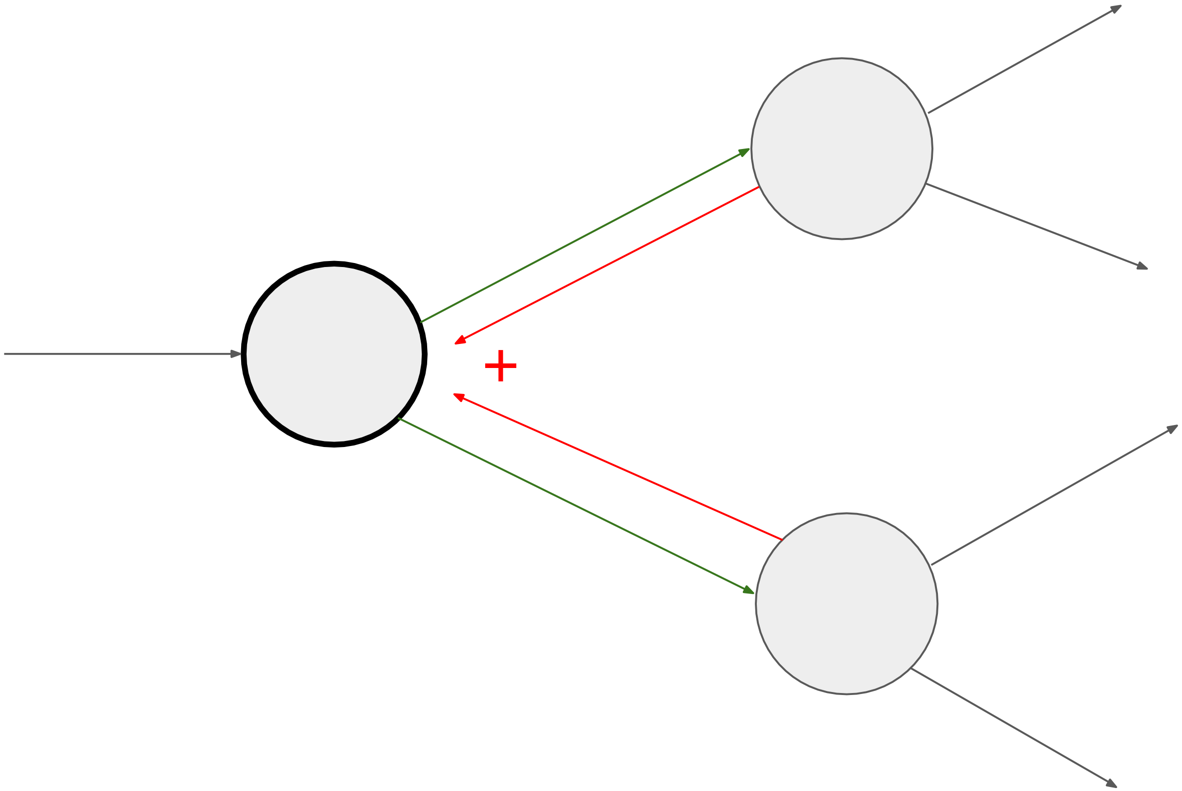
**add** gate: gradient distributor

**max** gate: gradient router

**mul** gate: gradient switcher

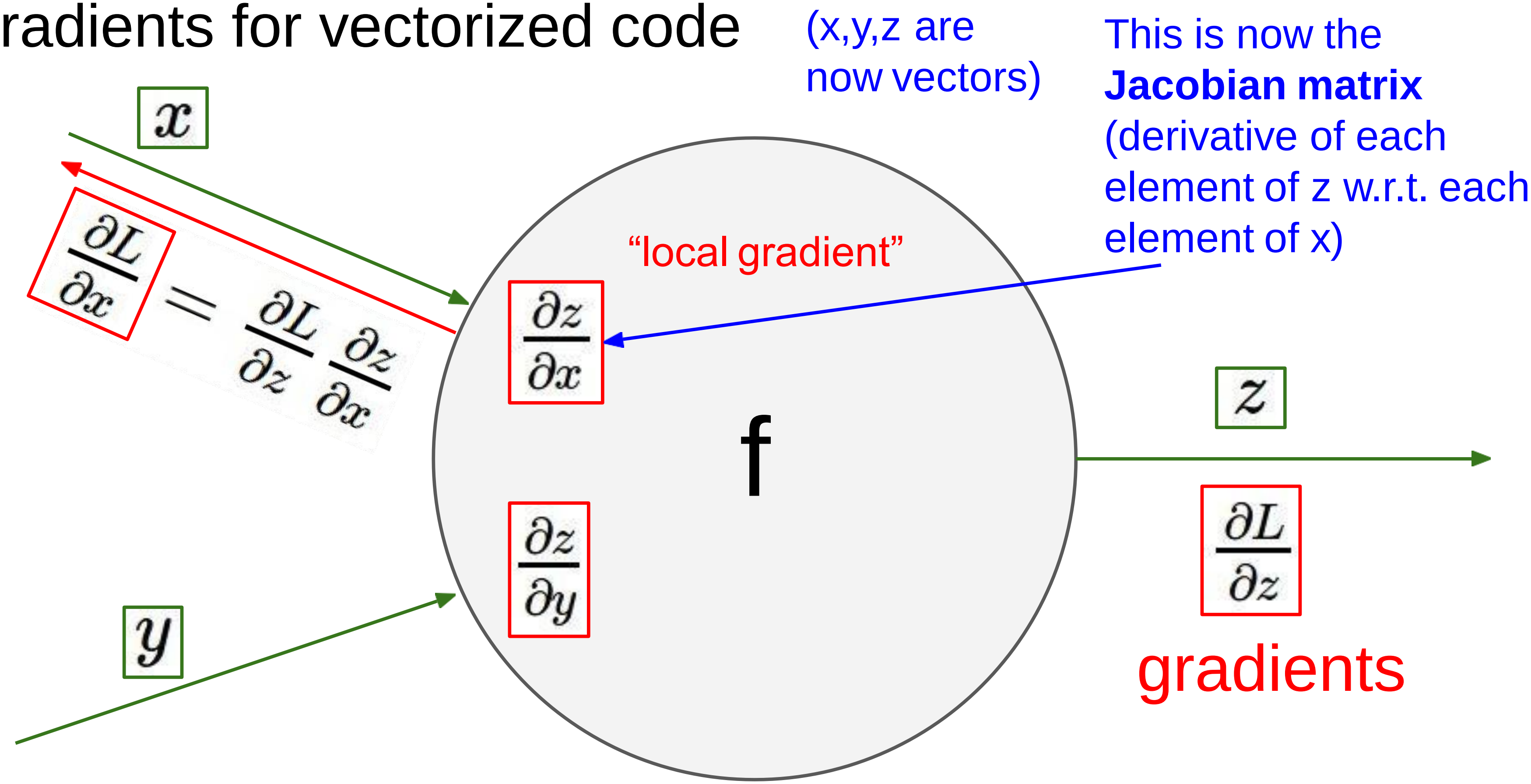


# Gradients add at branches

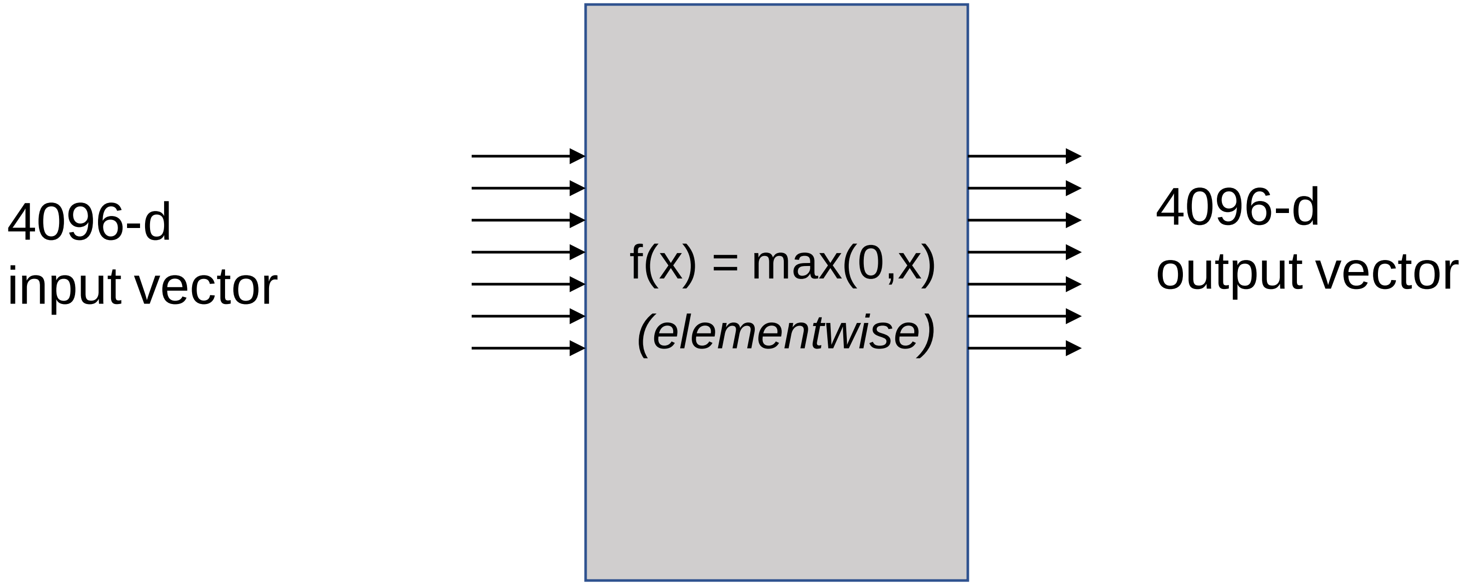



Slide credit: Fei-Fei Li

# Gradients for vectorized code



# Vectorized operations

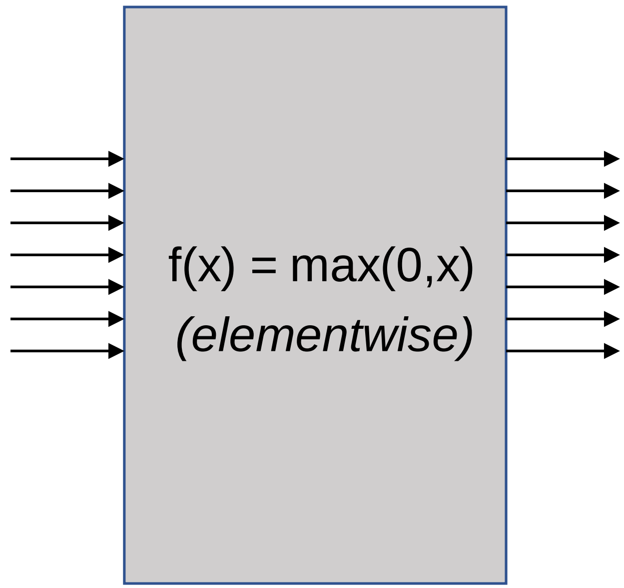


# Vectorized operations

$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

Jacobian matrix

4096-d
input vector




4096-d
output vector

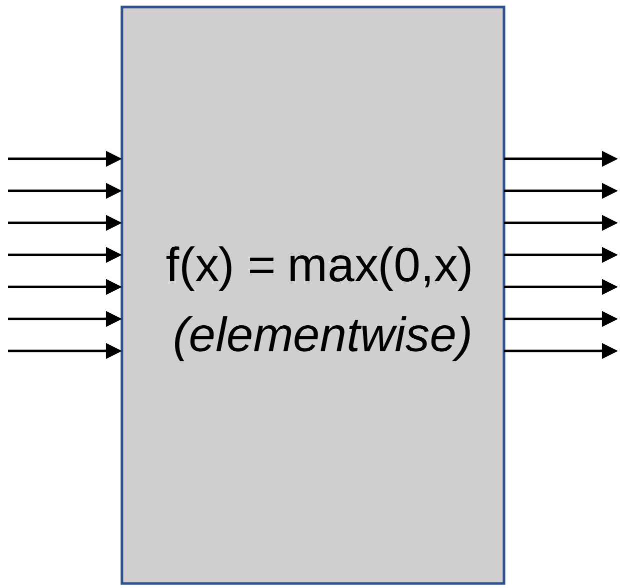
Q: what is the size of the Jacobian matrix?

# Vectorized operations

$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

Jacobian matrix

4096-d
input vector

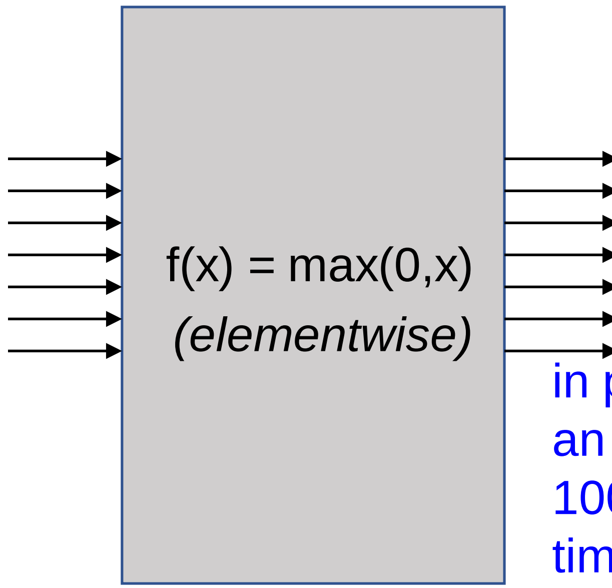



4096-d
output vector

Q: what is the size of the Jacobian matrix?
[4096 x 4096!]

# Vectorized operations

4096-d
input vector



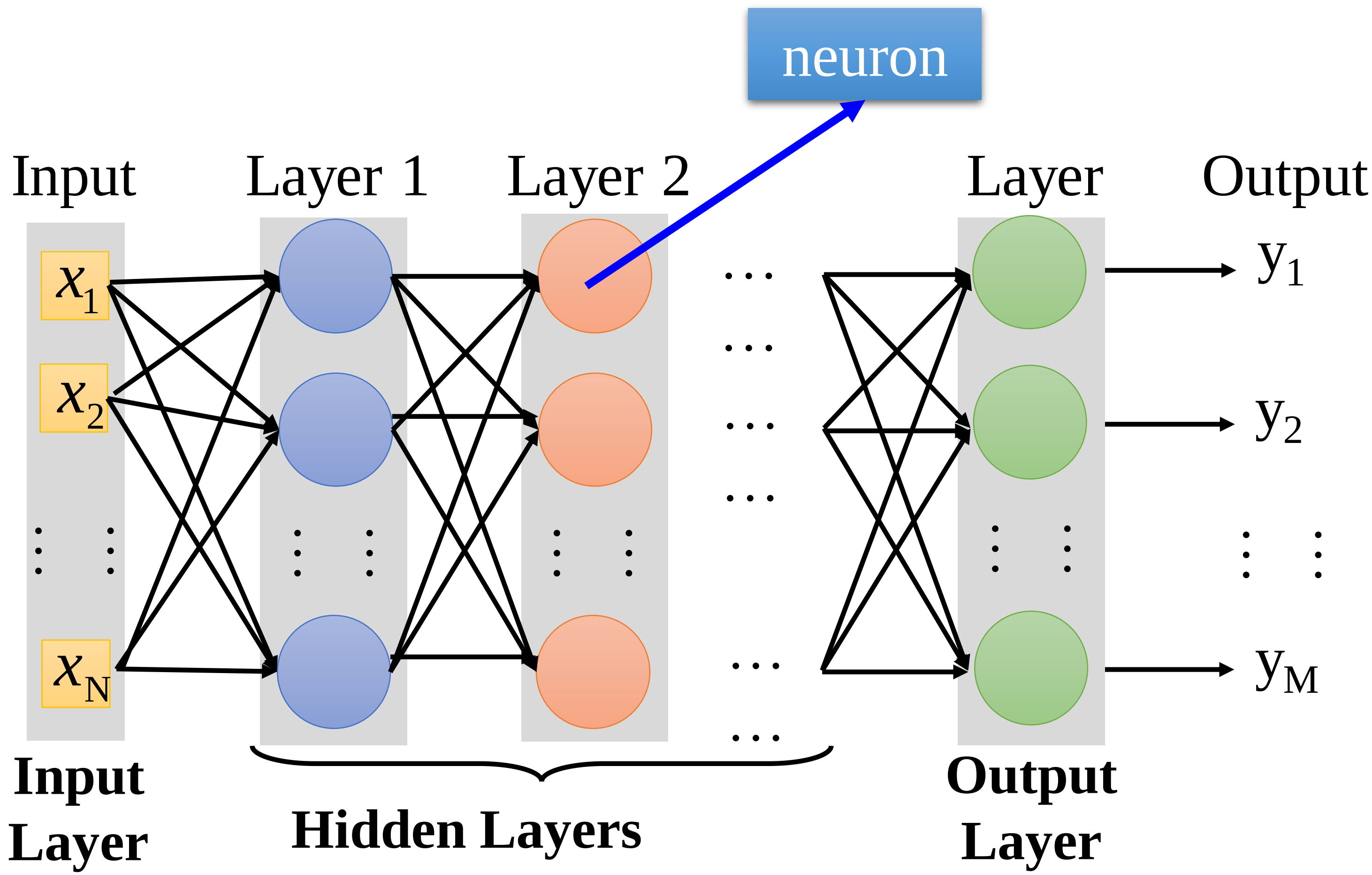

4096-d
output vector

Q: what is the size of the Jacobian matrix?
[4096 x 4096!]

in practice we process an entire minibatch (e.g. 100) of examples at one time:

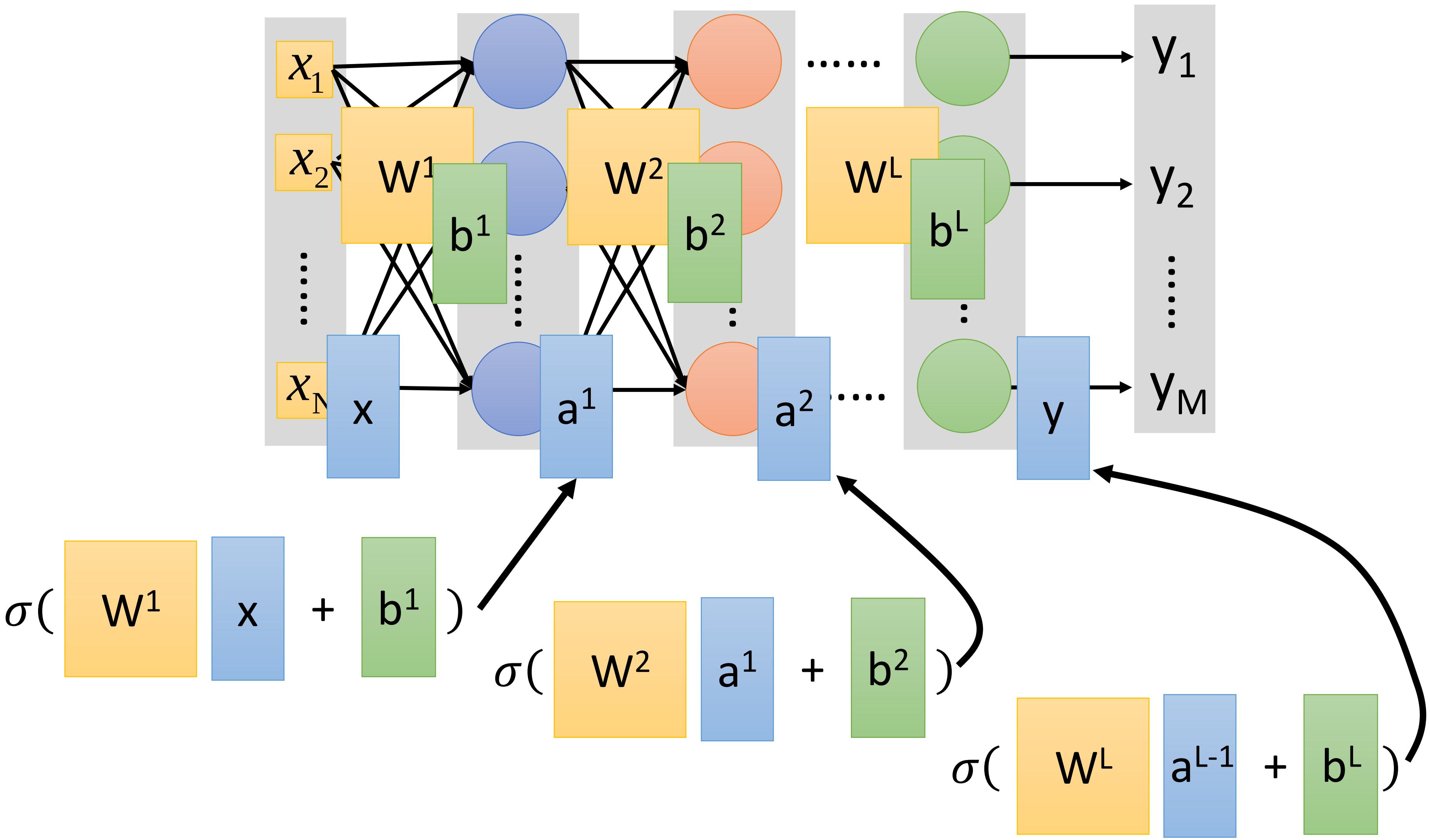
i.e. Jacobian would technically be a [409,600 x 409,600] matrix :\

# Neural Network



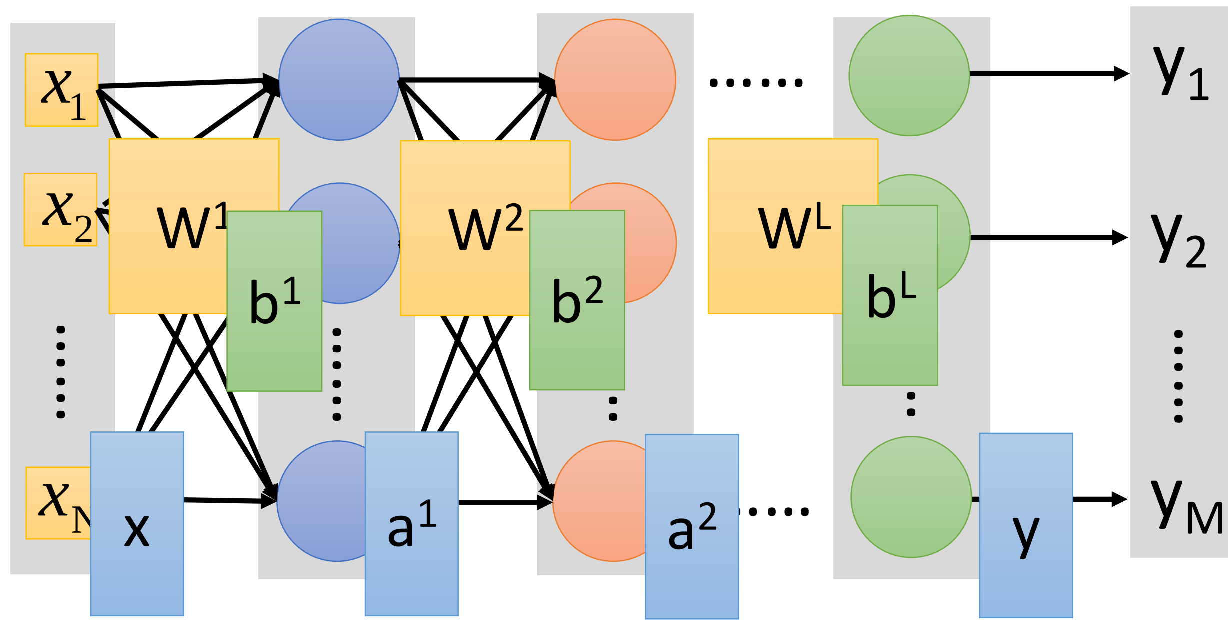

Deep means many hidden layers

# Neural Network




Slide credit: L. Lazebnik

# Neural Network




$$y = f(x)$$

Using parallel computing techniques to speed up matrix operation

$$= \sigma(W^L \cdots \sigma(W^2 \, \sigma(W^1 x + b^1) + b^2) \cdots + b^L)$$

Slide credit: L. Lazebnik

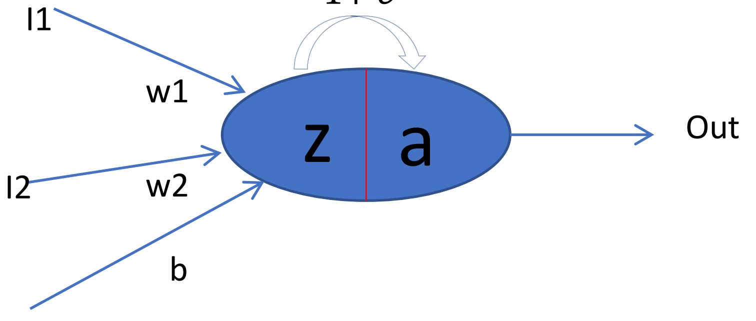
# Back Propagation In NN

- Every Hidden node and output has 2 values:
  - Net value (z)
  - Out value (a)

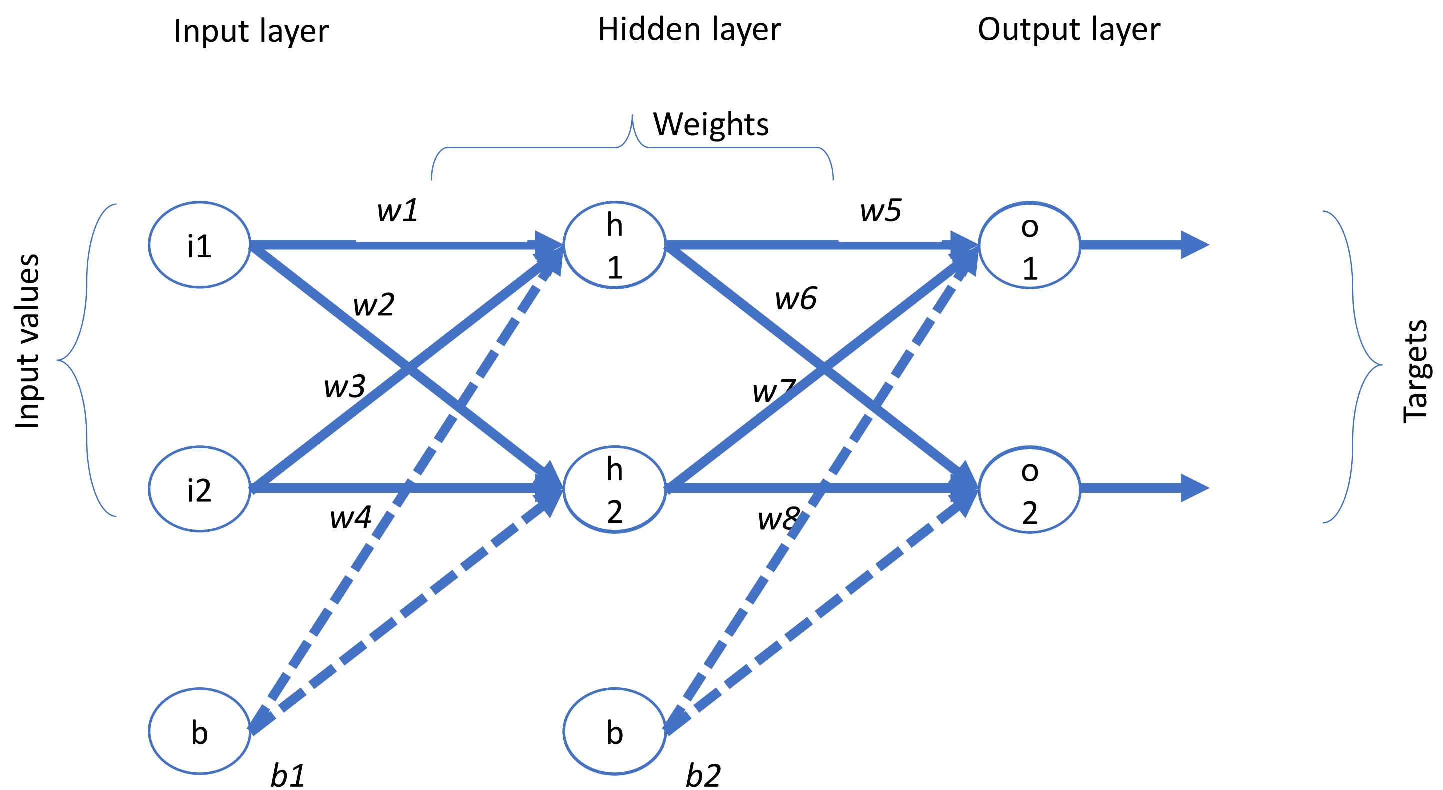
$$z = \mathrm{w1} * i1 + \mathrm{w2} * i2 + \mathrm{bias}$$

a is activation function

$$a = \frac{1}{1+ e^{-z}} \quad \text{(Sigmoid/tanh/ReLu)}$$

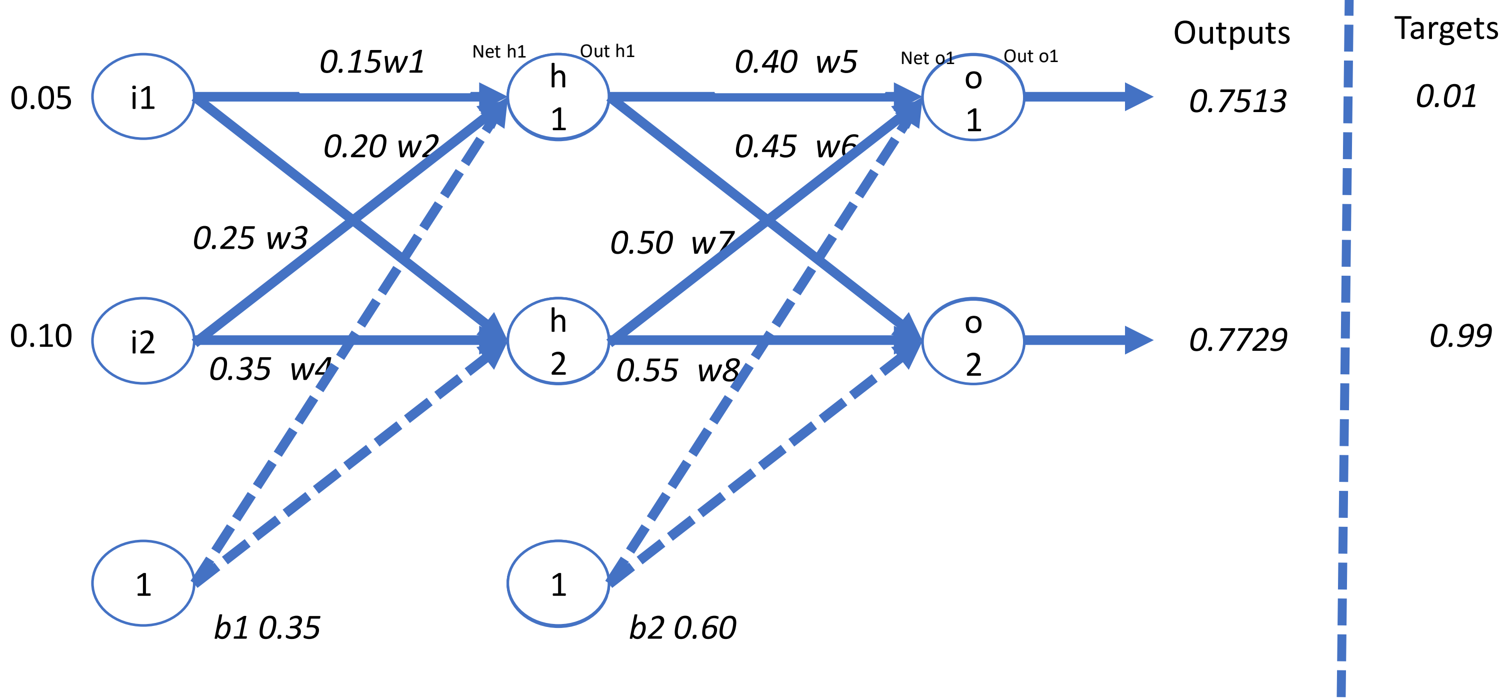


- We are going to use a neural network with:
  - two inputs,
  - two hidden neurons,
  - two output neurons.
- Additionally, the hidden and output neurons will include a bias.




**Basic Structure of NN**

Here are the **initial weights, the biases,** and training **inputs/outputs**:




**Example of NN**

# Forward Pass

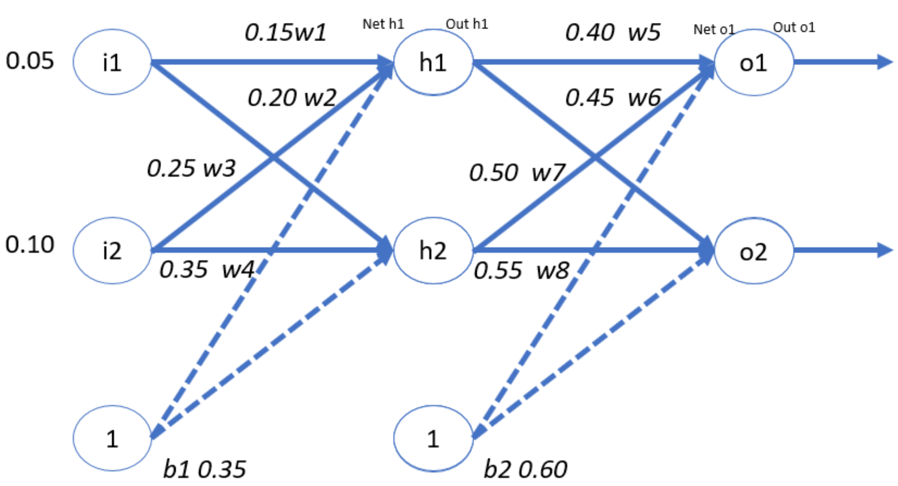
Lets see what the neural network currently predicts given the weights and biases above and inputs of 0.05 and 0.10.
=> Output for **hidden layer** with **sigmoid activation function:**

$$\boldsymbol{net_{h1} = w_1 * i_1 + w_1 * i_2 + b_1 * 1}$$

$$net_{h1} = 0.05 * 0.15 + 0.2 * 0.1 + 0.35 * 1 \quad \Rightarrow \quad \mathbf{0.3775}$$

$$\boldsymbol{out_{h1} = \frac{1}{1+ e^{-net\,h1}}} \quad (\textbf{sigmoid}\ \textit{activation function})$$

$$out\ h1 = \frac{1}{1+ e^{-0.3775}} \quad \Rightarrow \quad \mathbf{0.5932699}$$




similarly,

$$out\ h2 = \mathbf{0.5968843}$$

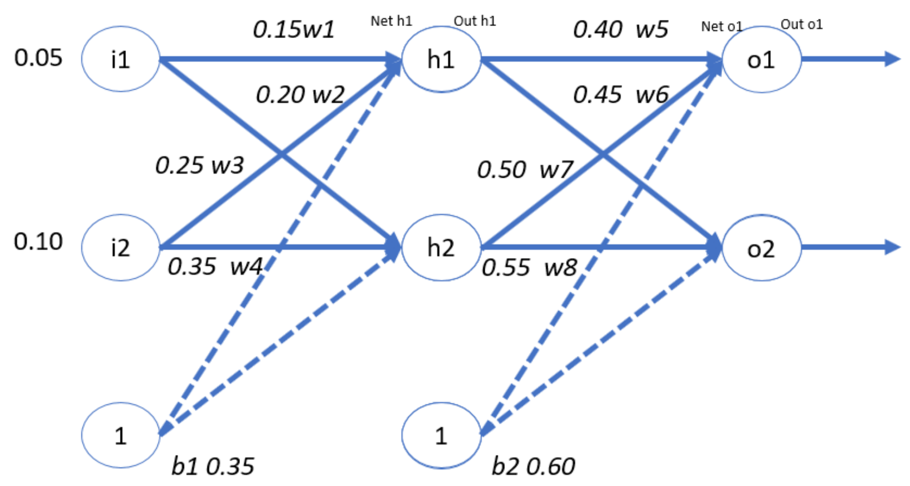
Repeat above process for the output layer neurons, using the output from the hidden layer neurons as inputs.

$$\boldsymbol{net_{o1} = w_5 * out_{h1} + w_5 * out_{h1} + b_2 * 1}$$

$$net_{o1} = 0.4 \times 0.5932699 + 0.45 \times 0.5968843 + 0.6 \times 1 \Rightarrow 1.105905967$$

$$\boldsymbol{out_{o1}} = \frac{1}{1+e^{-1.105905}} \Rightarrow 0.75136507 \quad \text{(Out1 but target is 0.01)}$$

$$\boldsymbol{out_{o2}} = 0.772928 \quad \text{(Out2 but target is 0.99)}$$



# Total Error

We can now calculate the error for each output neuron using the **squared error function** and sum them to get the total error:

$$\boldsymbol{E}_{\textbf{total}} = \sum \frac{1}{2}(\boldsymbol{target} - \boldsymbol{output})^2$$

$$E_{total} = Eo1 + Eo2$$

$$E_{o1} = \frac{1}{2}(0.01 - 0.75136507)^2 \Rightarrow \boldsymbol{0.274811083}$$

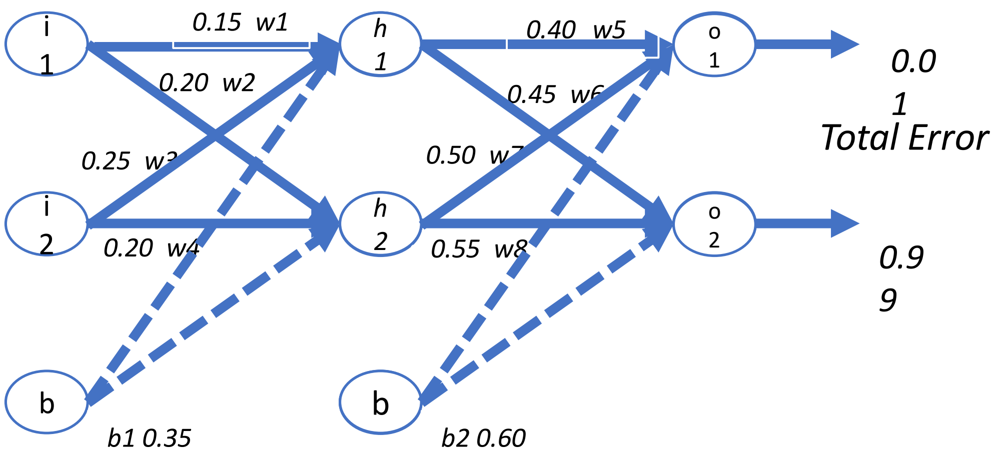
$$E_{o2} = \boldsymbol{0.023560026}$$

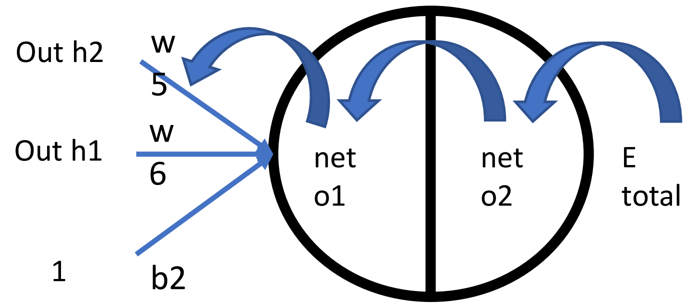
$$E\ total = Eo1 + Eo2 = \boldsymbol{0.298371109}$$

# Backward Propagation

## For output layer :

$$\frac{\partial Etotal}{\partial w5} = \frac{\partial Etotal}{\partial outo1} * \frac{\partial outo1}{\partial neto1} * \frac{\partial neto1}{\partial w5}$$



$$E\ total = \frac{1}{2}(target\ o1 - Out\ o1)^2 + \frac{1}{2}(target\ o2 - Out\ o2)^2$$

Derivative w.r.t Out o1

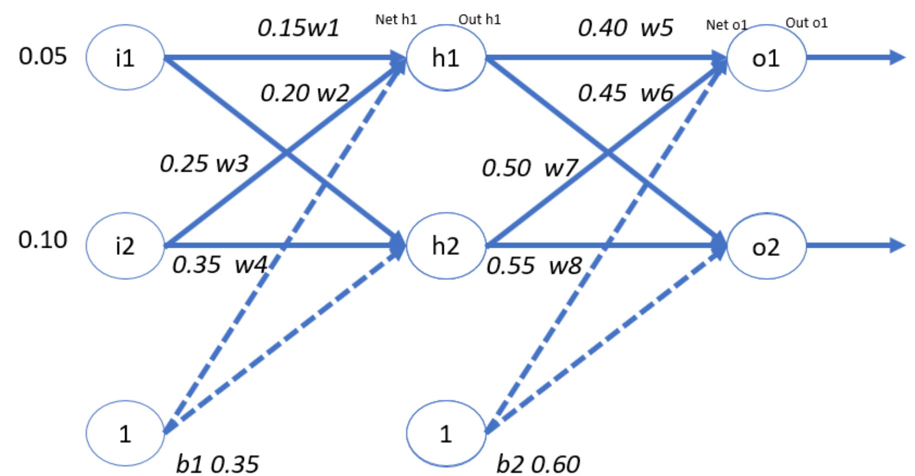
$$\frac{\partial Etotal}{\partial outo1} = -\ (target\ o1 - Out\ o1) + 0 = \mathbf{0.74136507}$$

$$Out\ o1 = \frac{1}{1 + e^{-net\ o1}}$$

$$\frac{\partial outo1}{\partial neto1} = Out\ o1\ (1 - Out\ o1) = \mathbf{0.18681560}$$

net o1 = w5 x out h1+ w6 x out h2 + b2 x 1

$$\frac{\partial neto1}{\partial w5} = Out\ h1 = \mathbf{0.5932699}$$

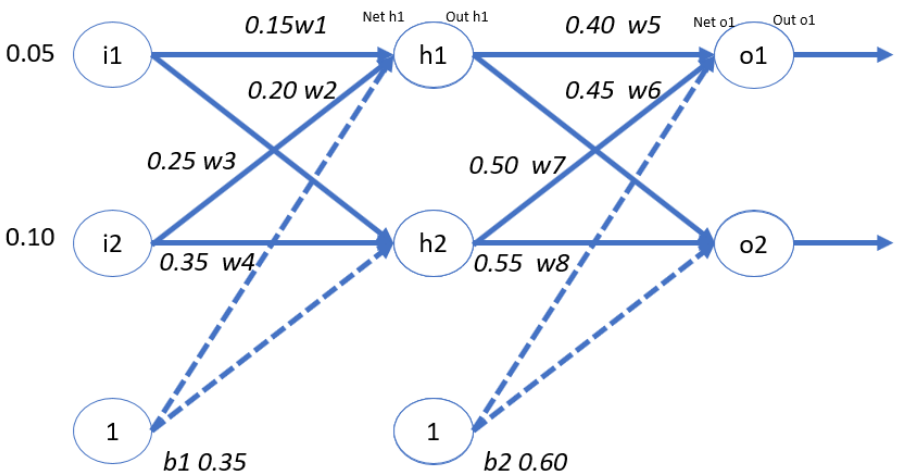



Constant are in RED color

# Backward Propagation

$$\frac{\partial Etotal}{\partial w5} = \frac{\partial Etotal}{\partial outo1} * \frac{\partial outo1}{\partial neto1} * \frac{\partial neto1}{\partial w5}$$

$$\frac{\partial Etotal}{\partial w5} = \mathbf{0.082167041}$$




Updation of ***weight w5 :***

w5_new = w5 – η x $\frac{\partial Etotal}{\partial w5}$

***W5_new*** = 0.40 – 0.5 x 0.082167 = .358916

η is learning rate here 0.5

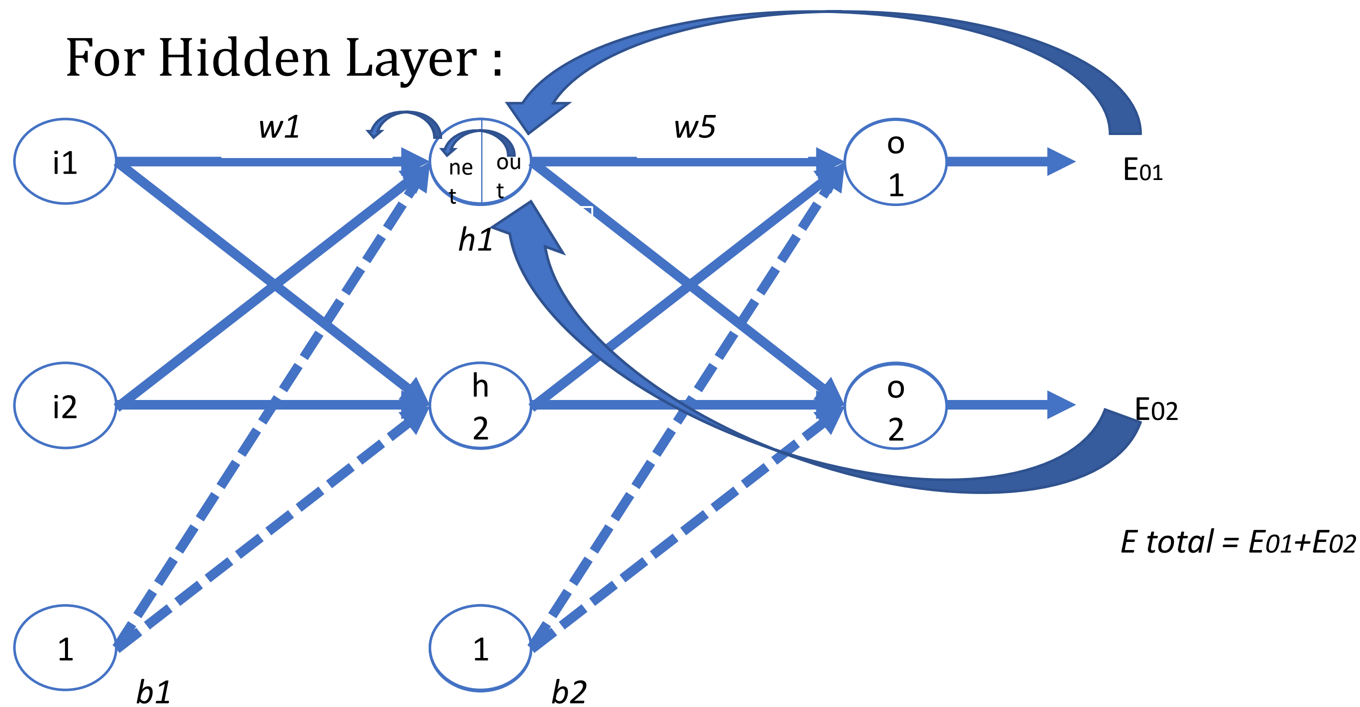
*w5* is now updated to *w5_new*
In next Forward pass w5_new is used

Find out updated values of weights w6, w7, w8 and bias b2 with the same procedure.

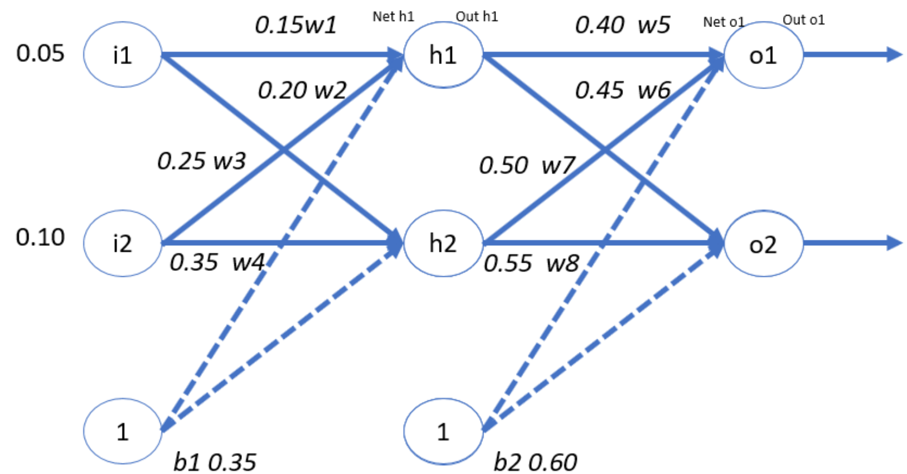
*w6_new = 0.408666186*
*w7_new = 0.511301270*
*w8_new = 0.561370121*

***Remember new values only considered in next Forward pass after complete updation of weights.***

**Next, we'll continue the backwards pass by calculating new values for w1**



$$\frac{\partial Etotal}{\partial w1} = \frac{\partial Etotal}{\partial outh1} * \frac{\partial outh1}{\partial neth1} * \frac{\partial neth1}{\partial w1}$$




*Etotal = Eo1 + Eo2*

*Eo1* = $\frac{1}{2}(target o1 - Out o1)^2$

*Eo2* = $\frac{1}{2}(target o2 - Out o2)^2$

*(Eo1 and Eo2 not directly depend on outh1)*

$$\frac{\partial Etotal}{\partial outh1} = \frac{\partial Eo1}{\partial outh1} + \frac{\partial Eo2}{\partial outh1}$$

*we will take both separately*

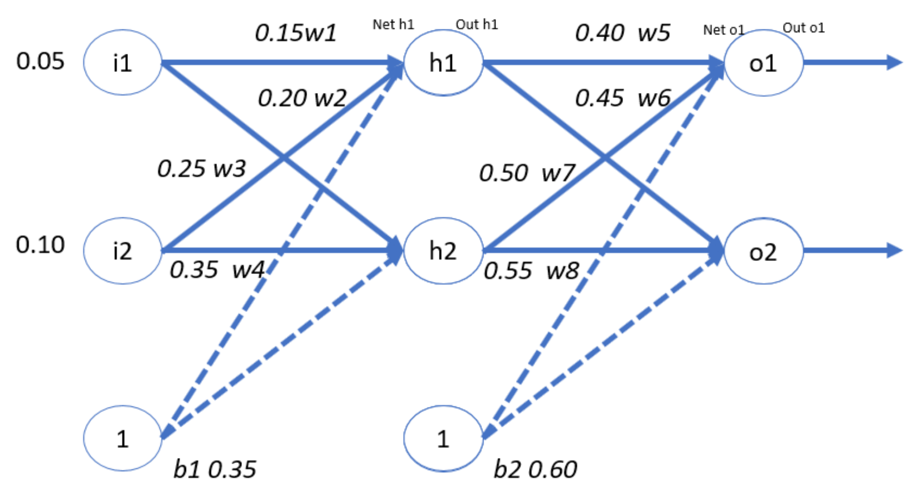
$$\frac{\partial Eo1}{\partial outh1} = \frac{\partial Eo1}{\partial neto1} * \frac{\partial neto1}{\partial outh1}$$

$$\frac{\partial Eo1}{\partial neto1} = \frac{\partial Eo1}{\partial outo1} * \frac{\partial outo1}{\partial neto1}$$

Both already calculated

$$\frac{\partial Eo1}{\partial neto1} = 0.74136507 \text{ x } 0.18681560 = 0.1384985$$

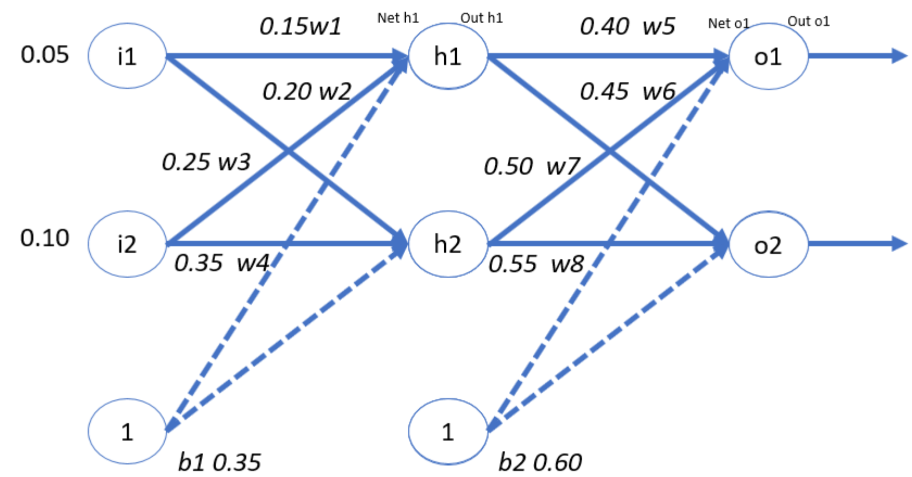
$$\frac{\partial neto1}{\partial outh1} = ?$$



neto1 = w5 $* outh1 +$ w6 $* outh2 + b2 * 1$

$\frac{\partial neto1}{\partial outh1}$ = w5 = 0.40

$\frac{\partial Eo1}{\partial outh1}$ = $\frac{\partial Eo1}{\partial neto1} * \frac{\partial neto1}{\partial outh1}$ = 0.1384985 x 0.40 = 0.0553994




$$\frac{\partial Eo2}{\partial outh1} = \frac{\partial Eo2}{\partial neto2} * \frac{\partial neto2}{\partial outh1}$$

$$\frac{\partial Eo2}{\partial neto2} = \frac{\partial Eo2}{\partial outo2} * \frac{\partial outo2}{\partial neto2}$$

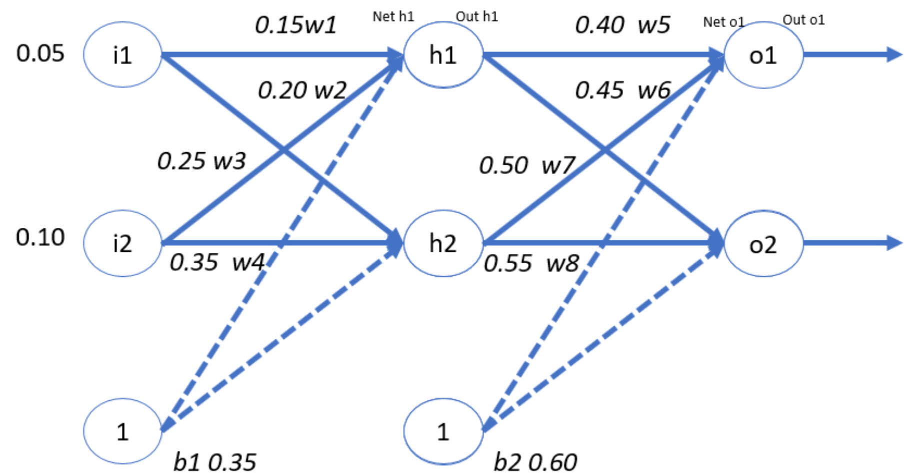
This time both not calculated

$$\text{Eo2} = \frac{1}{2}(\text{target o2} - \text{out o2})^2$$

$$\frac{\partial Eo2}{\partial outo2} = -(\text{target o2} - \text{out o2}) = -(0.99 - 0.772928)$$

$$\frac{\partial Eo2}{\partial outo2} = -0.217072$$

$$\frac{\partial Eo2}{\partial neto2} = \frac{\partial Eo2}{\partial outo2} * \frac{\partial outo2}{\partial neto2}$$




$$\text{Out o2} = \frac{1}{1+ e^{-neto2}}$$

$$\frac{\partial outo2}{\partial neto2} = \text{Out o2 } (1 - \text{Out o2})$$

$$= (0.7729284)(1 - 0.7729284) = 0.1755100$$

$$\frac{\partial Eo2}{\partial neto2} = (-0.217072) * (0.1755100) = -0.0380983$$

$$\frac{\partial Eo2}{\partial outh1} = \frac{\partial Eo2}{\partial neto2} * \frac{\partial neto2}{\partial outh1}$$

$$\frac{\partial Eo2}{\partial neto2} = -\ 0.0380983$$

neto2 = w7 $* outh1 +$ w8 $* outh2 + b2 * 1$

$$\frac{\partial neto2}{\partial outh1} = \text{w7} = 0.50$$

$$\frac{\partial Eo2}{\partial outh1} = \frac{\partial Eo2}{\partial neto2} * \frac{\partial neto2}{\partial outh1}$$

$$\frac{\partial Eo2}{\partial outh1} = -0.0380983 * 0.50 = -0.0190491$$

$$\frac{\partial Etotal}{\partial w1} = \frac{\partial Etotal}{\partial outh1} * \frac{\partial outh1}{\partial neth1} * \frac{\partial neth1}{\partial w1}$$

$$\frac{\partial Etotal}{\partial outh1} = \frac{\partial Eo1}{\partial outh1} + \frac{\partial Eo2}{\partial outh1}$$

$$\frac{\partial Etotal}{\partial outh1} = 0.0553994 + -0.0190491 = 0.0363503$$

$$\frac{\partial Etotal}{\partial w1} = \frac{\partial Etotal}{\partial outh1} * \frac{\partial outh1}{\partial neth1} * \frac{\partial neth1}{\partial w1}$$

outh1 = $\frac{1}{1+ e^{-net\,h1}}$

$\frac{\partial outh1}{\partial neth1}$ = outh1 x (1 - outh1) = 0.5932699 X (1- 0.5932699)

$\frac{\partial outh1}{\partial neth1}$ = 0.2413007

$$\frac{\partial Etotal}{\partial w1} = \frac{\partial Etotal}{\partial outh1} * \frac{\partial outh1}{\partial neth1} * \frac{\partial neth1}{\partial w1}$$

neth1 = i1 $*$ w1 + i2 $*$ w2 + b1 $*$ 1

$$\frac{\partial neth1}{\partial w1} = \text{i1} = 0.05$$

$$\frac{\partial Etotal}{\partial w1} = 0.0363503 * 0.2413007 * 0.05$$

$$\frac{\partial Etotal}{\partial w1} = \mathbf{0.00043856}$$

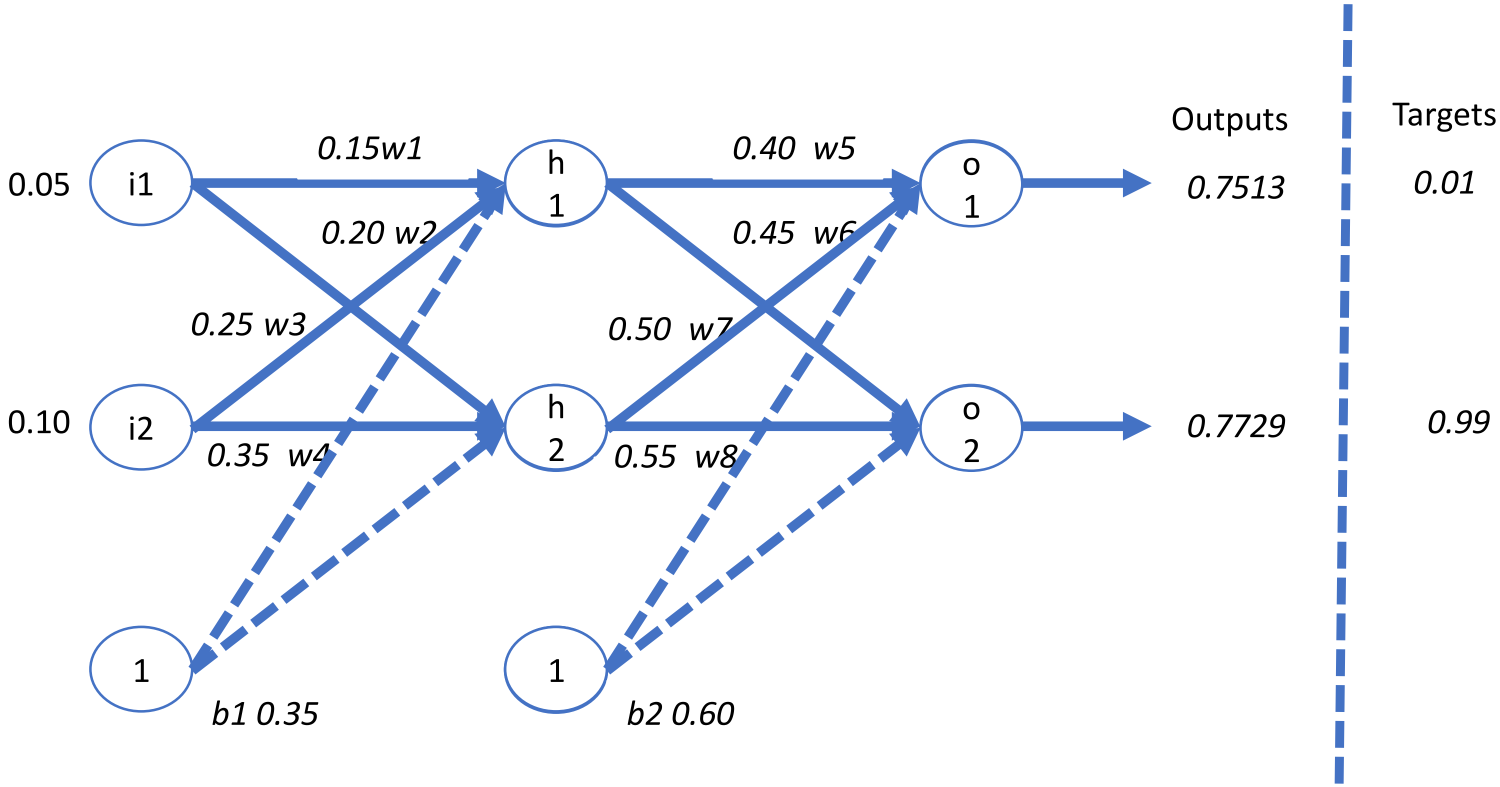
Updation of ***weight w1 :***

w1_new = w1 – η x $\frac{\partial Etotal}{\partial w1}$

w1_new = 0.15 – 0.5 $*$ 0.00043856 = 0.149780

With the same procedure weights **w2 w3 w4** and bias **b1** will be computed.

w1_new = 0.19956143
w2_new = 0.24975114
w3_new = 0.29950229

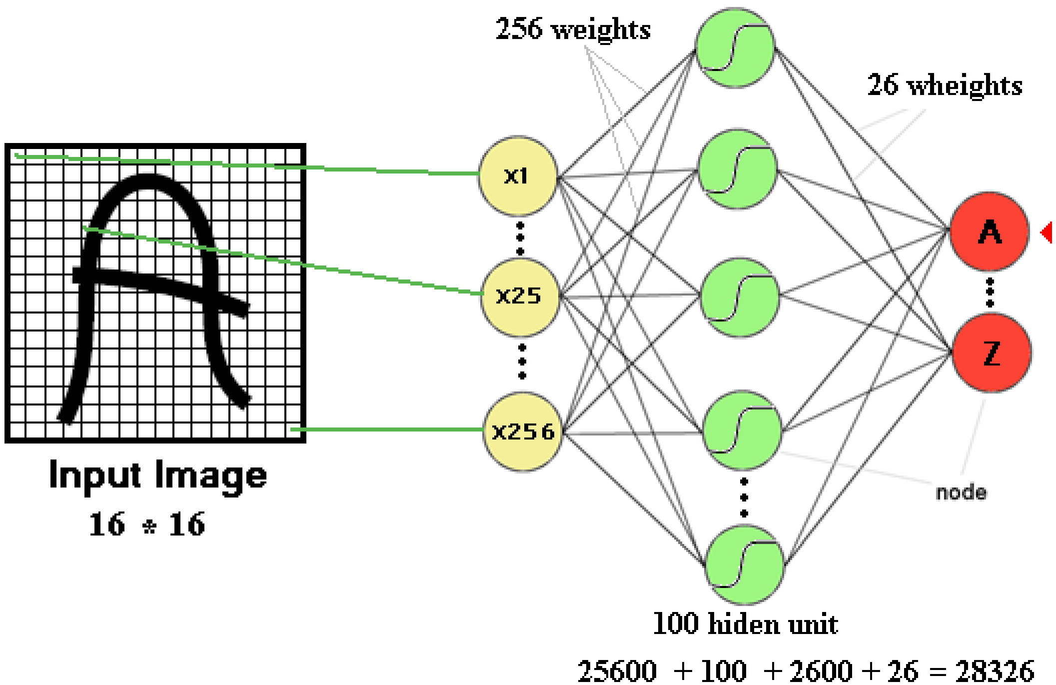



**Example of NN**

- Finally, we’ve updated all of our weights! When we fed forward the 0.05 and 0.1 inputs originally, the error on the network was 0.298371109.
- After this first round of backpropagation, the total error is now down to 0.291027924.
- It might not seem like much, but after repeating this process 10,000 times, for example, the error plummets to 0.0000351085.
- At this point, when we feed forward 0.05 and 0.1, the two outputs neurons generate 0.015912196 (vs 0.01 target) and 0.984065734 (vs 0.99 target).

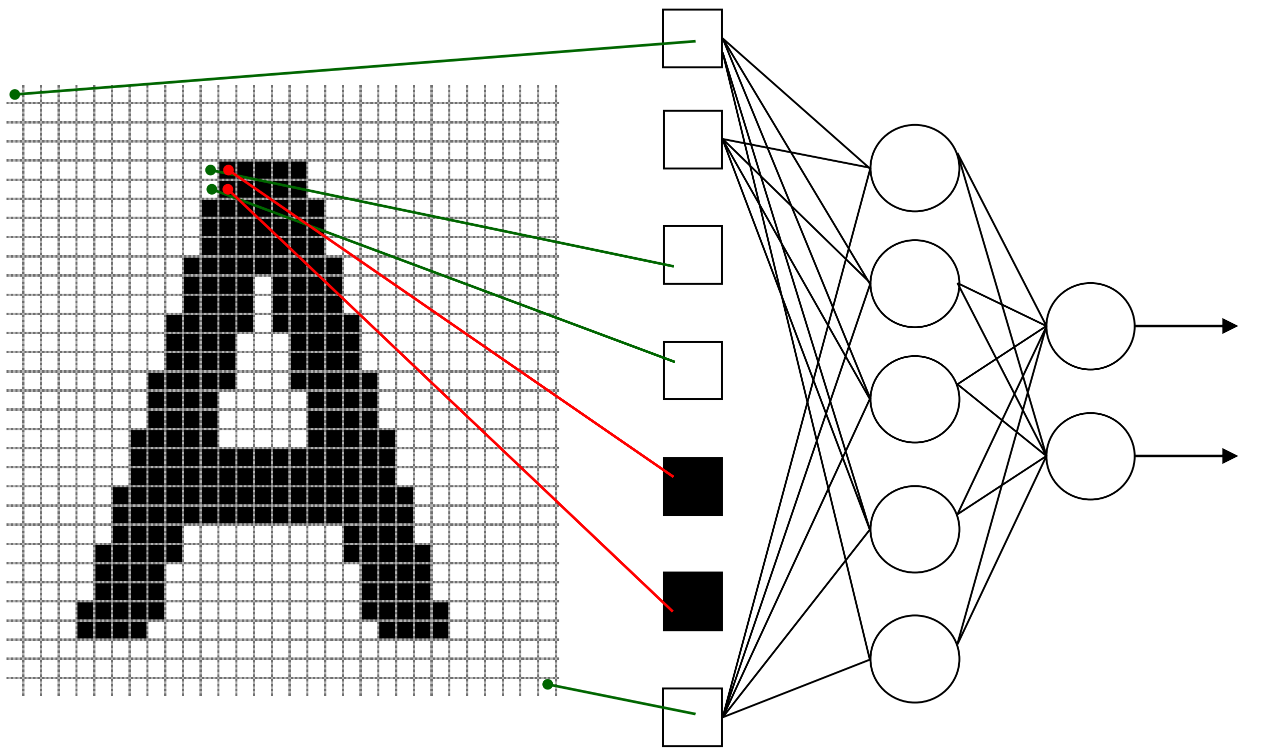
## Drawbacks of Neural Networks

- The number of trainable parameters becomes extremely large.



## Drawbacks of Neural Networks

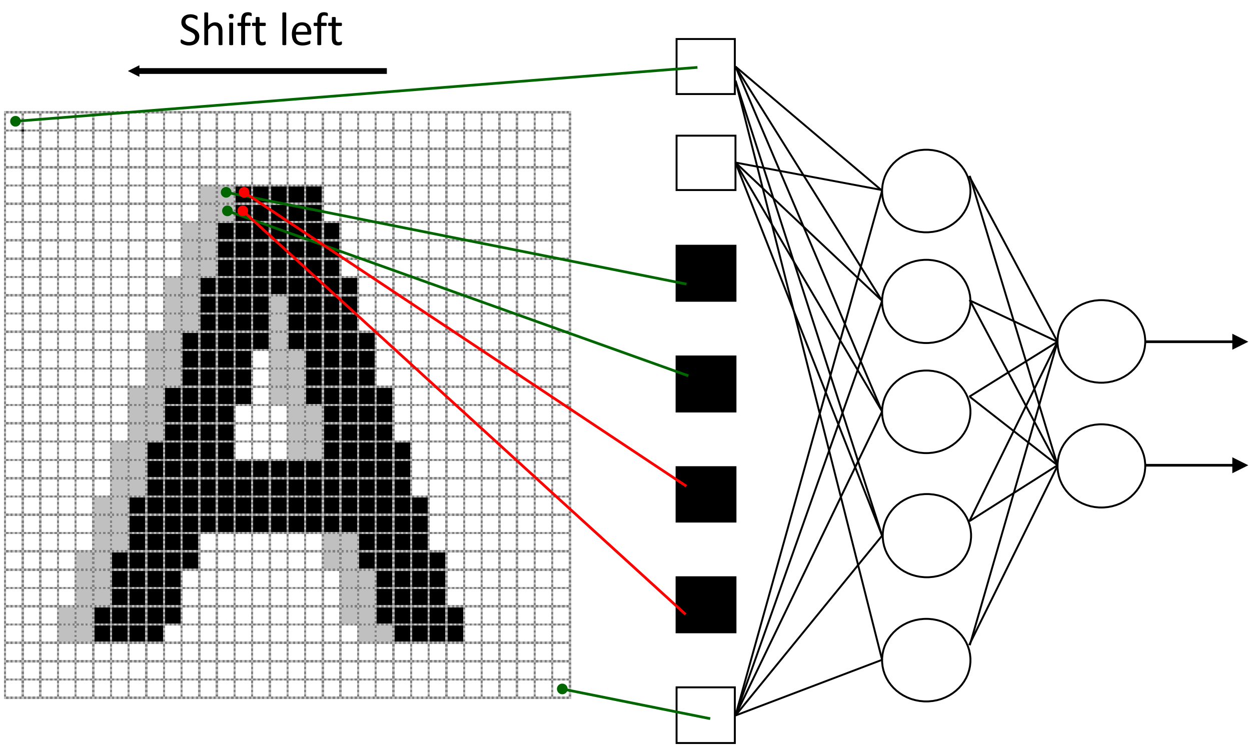
- ❑ Little or no invariance to shifting, scaling, and other forms of distortion



Slide credit: Fei-Fei Li

## Drawbacks of Neural Networks

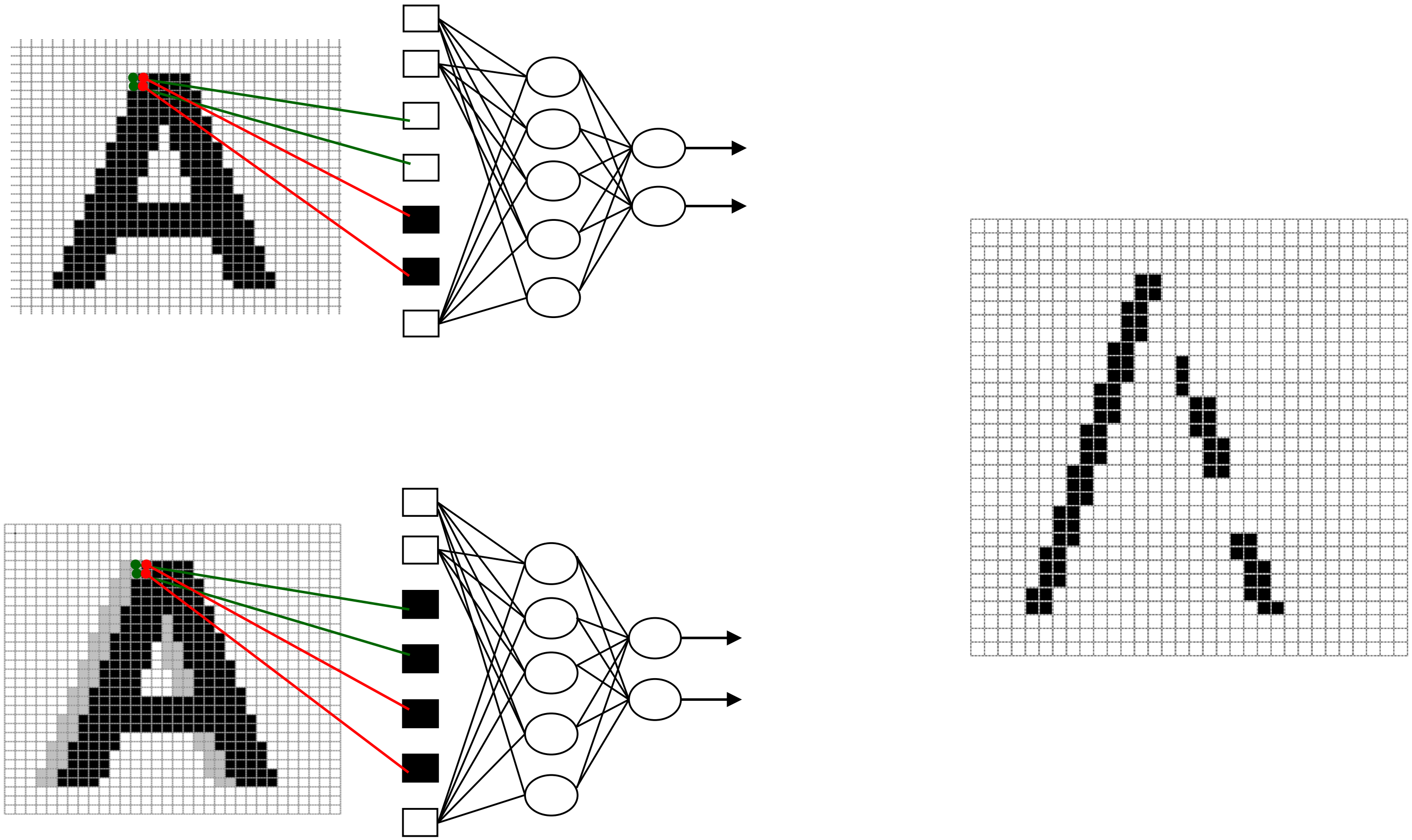
- Little or no invariance to shifting, scaling, and other forms of distortion




Slide credit: Fei-Fei Li

## Drawbacks of Neural Networks

- Little or no invariance to shifting, scaling, and other forms of distortion



Slide credit: Fei-Fei Li

# Definition of Loss

In a supervised deep learning context the **loss function** measures the **quality** of a particular set of parameters based on how well the output of the network **agrees** with the ground truth labels in the training data.

# Nomenclature

**loss** function

=

**cost** function

=

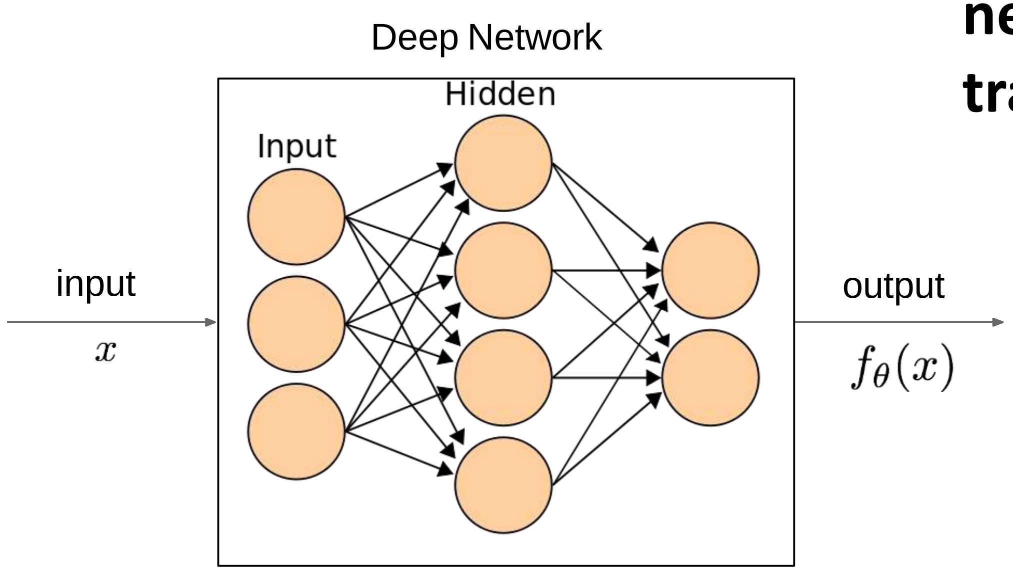
**objective** function

=

**error** function

# Loss function (1)

**How good does our network with the training data?**




labels (ground truth)

input

$$\mathcal{L}(w) = distance(f_\theta(x), y)$$

error

parameters (weights, biases)

# Common types of loss functions (1)

- Loss functions depen on the type of task:
  - Regression: the network predicts **continuous**, **numeric** variables
    - Example: Length of fishes in images, temperature from latitude/longitud
    - Absolute value, square error

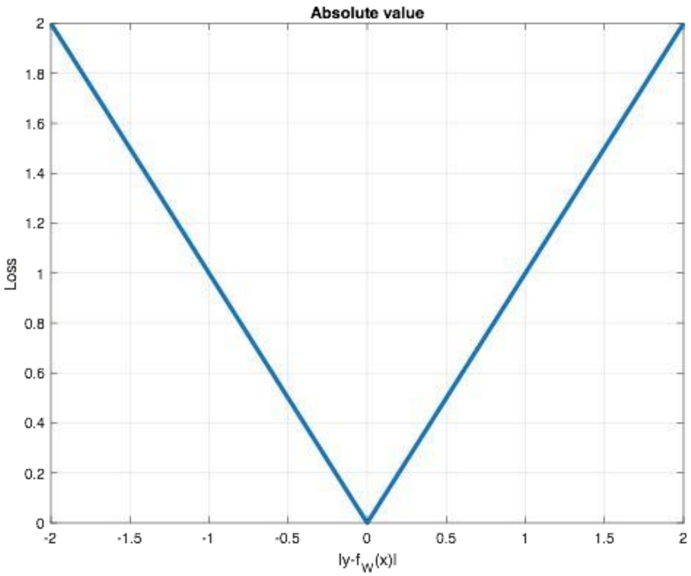
# Common types of loss functions (2)

- Loss functions depen on the type of task:
  - Classification: the network predicts **categorical** variables (fixed number of classes)
    - Example: classify email as spam, predict student grades from essays.
    - hinge loss, Cross-entropy loss

# Absolute value, L1-norm

- Very intuitive loss function
  - produces sparser solutions
    - good in high dimensional spaces
    - prediction speed
  - less sensitive to outliers

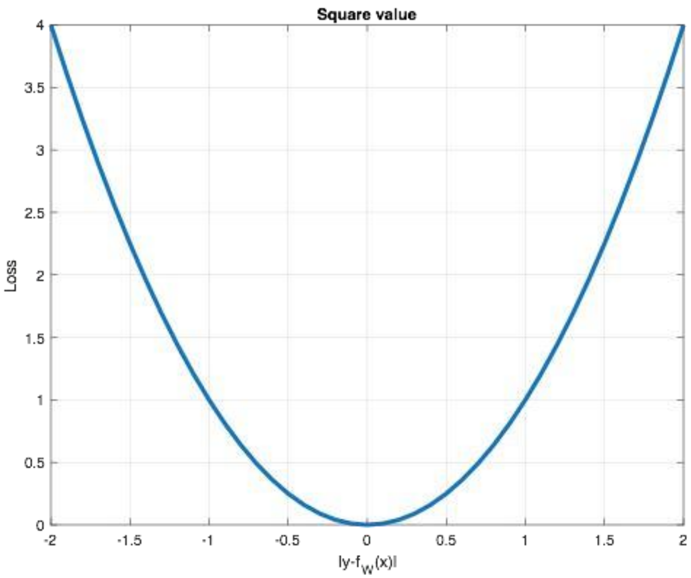
$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_\theta(x_i)|$$



# Square error, Euclidean loss, L2-norm

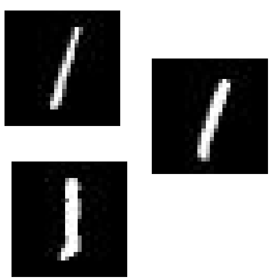
- Very common loss function
  - More precise and better than L1-norm
  - Penalizes large errors more strongly
  - Sensitive to outliers

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2$$

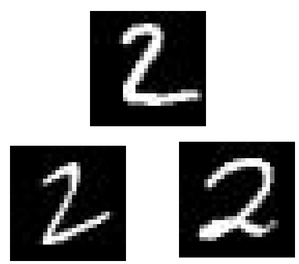


# Classification (1)

We want the network to classify the input into a  fixed number of classes



class “1”

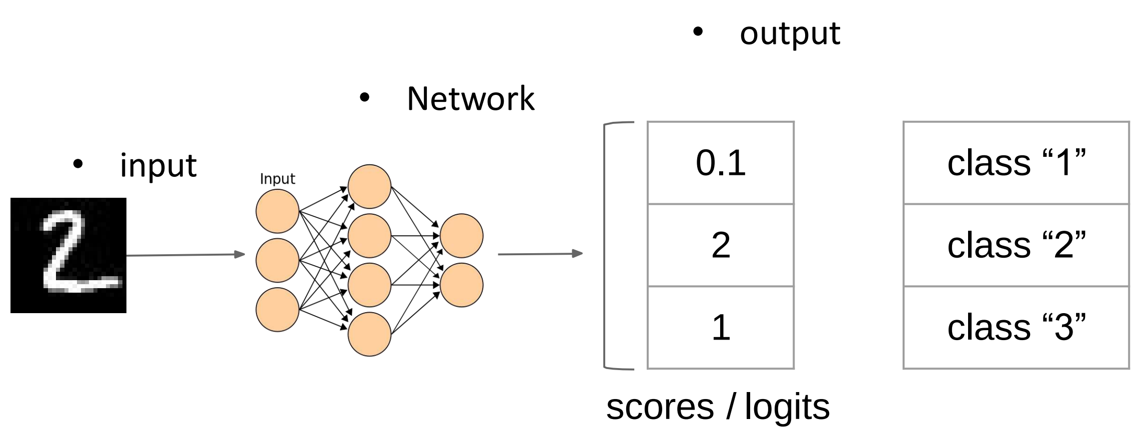


class “2”

class “3”

# Classification (2)

- Each input can have only one label
    - One prediction per output class
        - The network will have “k” outputs (number of classes)

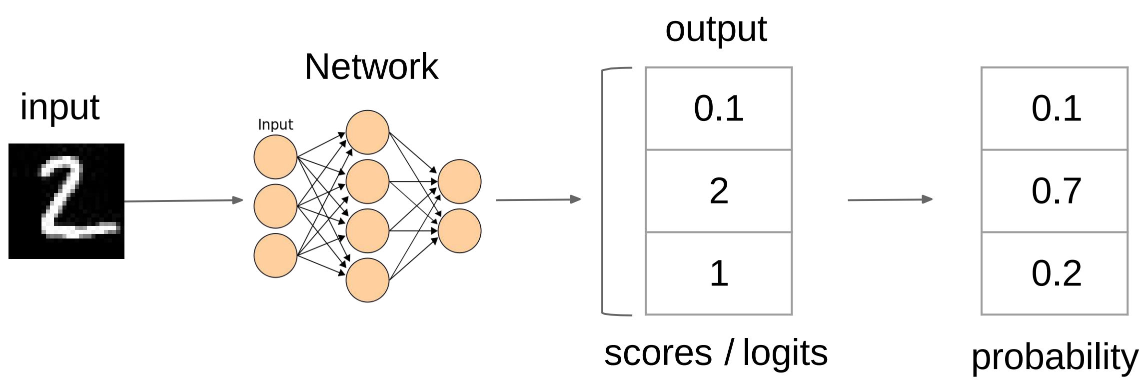


# Classification (3)

- How can we create a loss function to improve the scores?
  - Somehow write the labels (ground truth of the data) into a vector → One-hot encoding
  - Non-probabilistic interpretation → **hinge loss**
  - Probabilistic interpretation: need to transform the scores into a probability function → Softmax

# Softmax (1)

- Convert scores into probabilities
  - From 0.0 to 1.0
  - Probability for all classes adds to 1.0

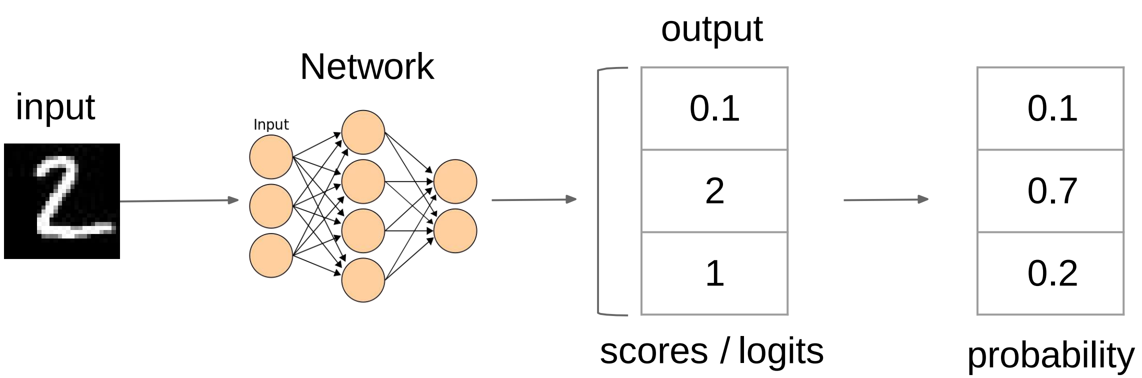


# Softmax (2)

- Softmax function

scores (logits)

$$S(l_i) = \frac{e^{l_i}}{\sum_k e^{l_k}}$$



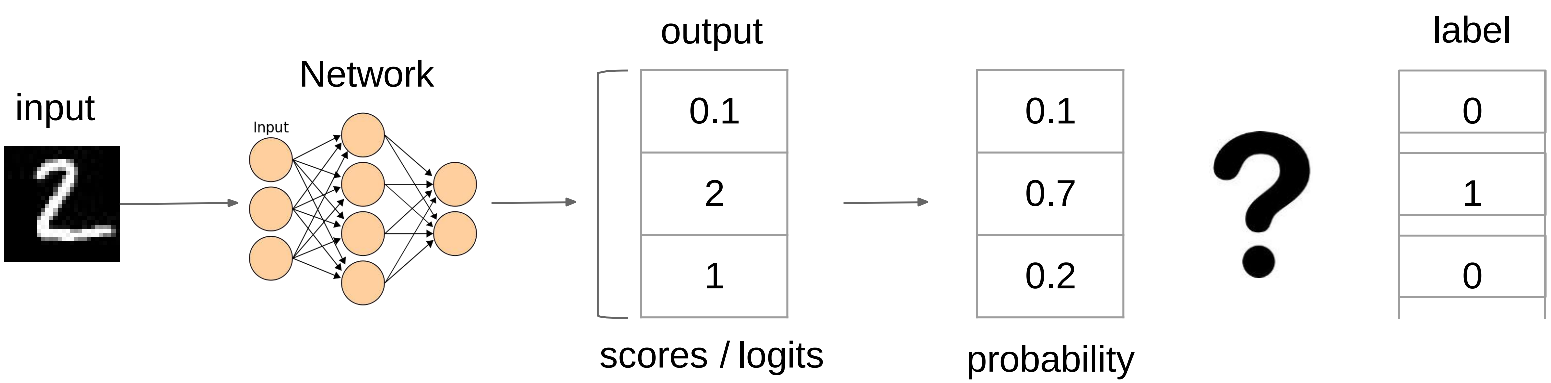

Neural Networks and Deep Learning ([softmax](softmax))

# One-hot encoding

- Transform each label into a vector (with only 1  and 0)
  - Length equal to the total number of classes “k”
  - Value of 1 for the correct class and 0 elsewhere

| class “1” | class “2” | class “3” |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

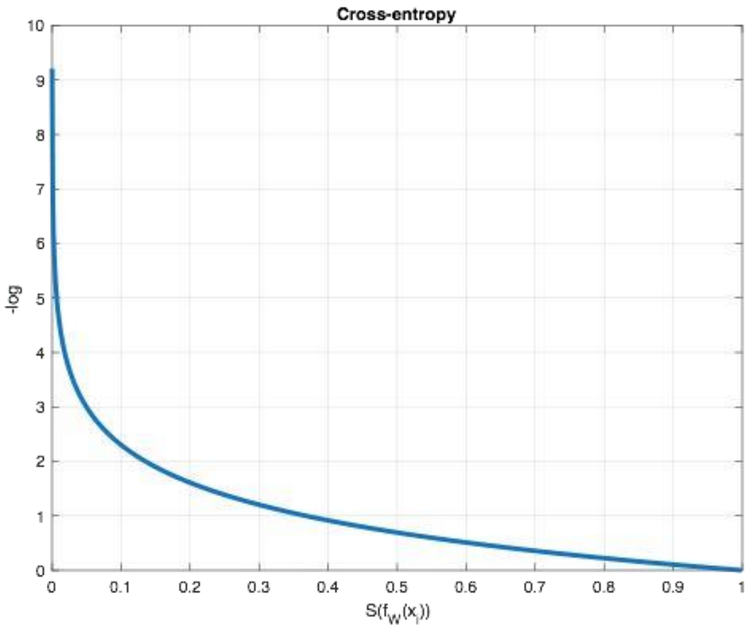
# Cross-entropy loss (1)




$$\mathcal{L}_i = -\sum_k y_k \log(S(l_k)) = -\log(S(l))$$

# Cross-entropy loss (2)

$$\mathcal{L}_i = -\sum_k y_k \log(S(l_k)) = -\log(S(l))$$



# Cross-entropy loss (3)

For a set of n inputs

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i$$

labels (one-hot)

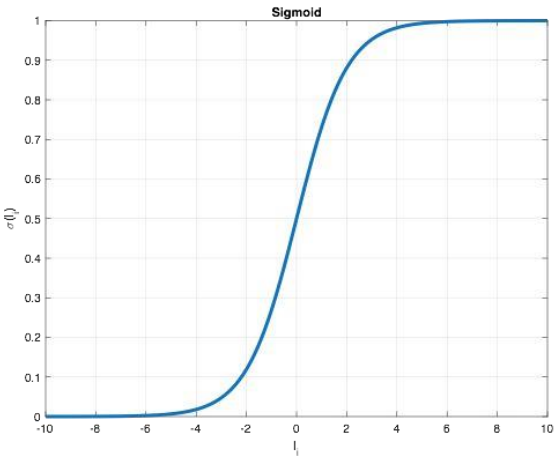
$$\mathcal{L} = -\sum_{i=1}^{n}\mathbf{y}_i \log(S(f_\theta(\mathbf{x}_i)))$$

Softmax

# Multi-label classification (1)

- Outputs can be matched to more than one label
  - “car”, “automobile”, “motor vehicle” can be applied to a same image of a car.
- Use sigmoid at each output independently instead of softmax

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



# Multi-label classification (2)

- Cross-entropy loss for multi-label classification:

$$\mathcal{L}_i = -\sum_k y_k \log(\sigma(l_i)) + (1 - y_k) \log(1 - \sigma(l_i))$$

Thanks! Questions?