Foundations Of Neural Networks and Deep Learning

Day-7

SME: Gagan P

Contact: gaganp000999@gmail.com

recap:

Which of the following best describes a perceptron?

A. A model that stores data like a database
B. A model that assigns weights to inputs, sums them, and applies an activation function
C. A statistical test for correlation
D. A clustering algorithm

recap:

Which of the following best describes a perceptron?

A. A model that stores data like a database

B. A model that assigns weights to inputs, sums them, and applies an activation function

C. A statistical test for correlation

D. A clustering algorithm

In the equation $y = f(\sum w_i x_i + b)$, what is f?

A. Cost function
B. Weight update function
C. Activation function
D. Bias adjustment function

In the equation $y = f(\sum w_i x_i + b)$, what is f?

A. Cost function
B. Weight update function
C. Activation function
D. Bias adjustment function

If all activation functions were linear, what would happen to a multi-layer perceptron?

A. It could approximate any nonlinear function
B. It would collapse into a single linear model
C. It would overfit
D. It would require less training data

If all activation functions were linear, what would happen to a multi-layer perceptron?

A. It could approximate any nonlinear function
B. It would collapse into a single linear model
C. It would overfit
D. It would require less training data

Which statement about ReLU is TRUE?

A. Outputs values between 0 and 1
B. Has vanishing gradients
C. Returns 0 for negative inputs and x for positive
D. Is only used for binary classification

Which statement about ReLU is TRUE?

A. Outputs values between 0 and 1
B. Has vanishing gradients
C. Returns 0 for negative inputs and x for positive
D. Is only used for binary classification

What do weights represent in a perceptron?

A. Random numbers
B. Importance or strength of each input feature
C. Bias for each neuron
D. Learning rate

What do weights represent in a perceptron?

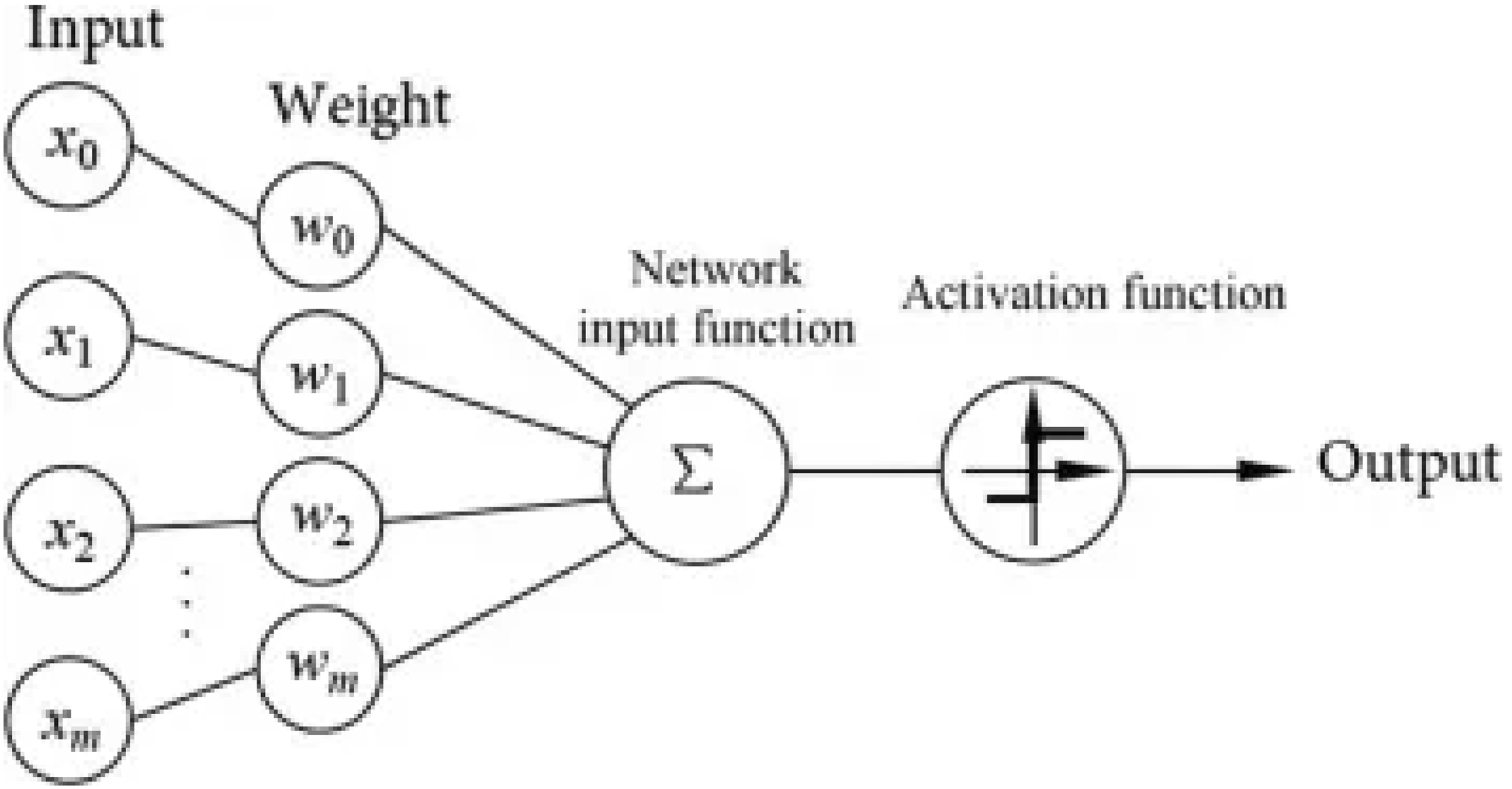
A. Random numbers

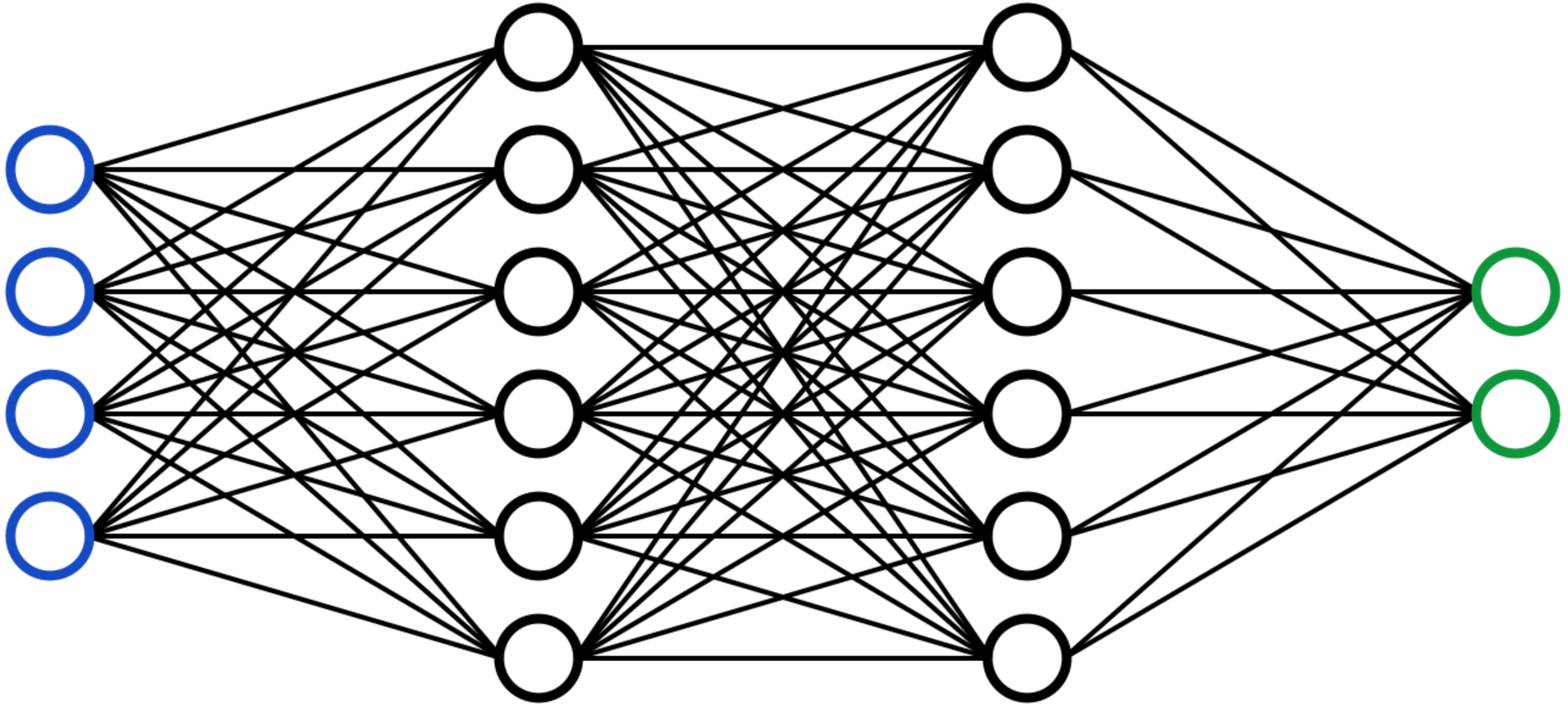
B. Importance or strength of each input feature

C. Bias for each neuron

D. Learning rate

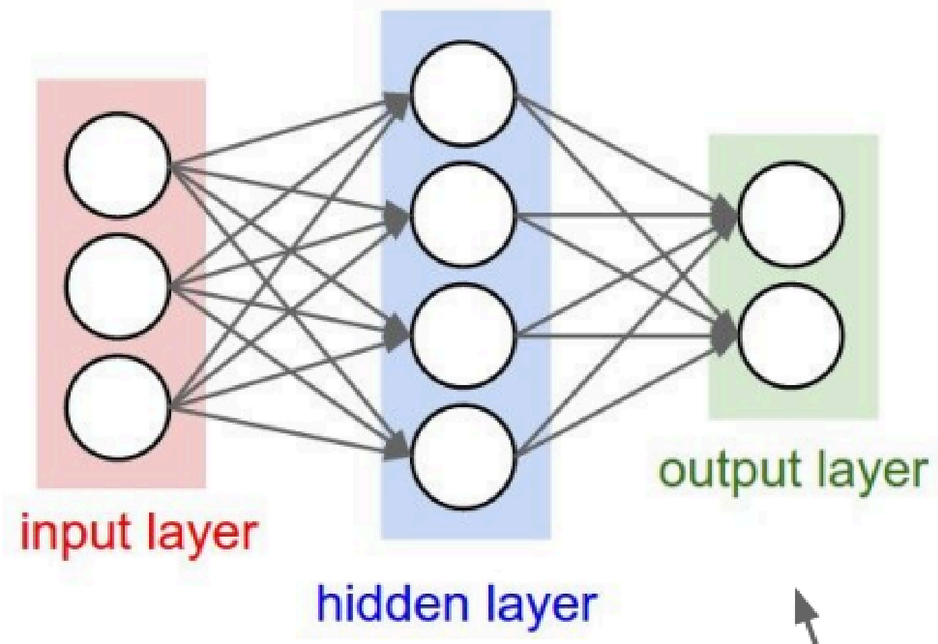
day 7 - Training A Simple Neural Network

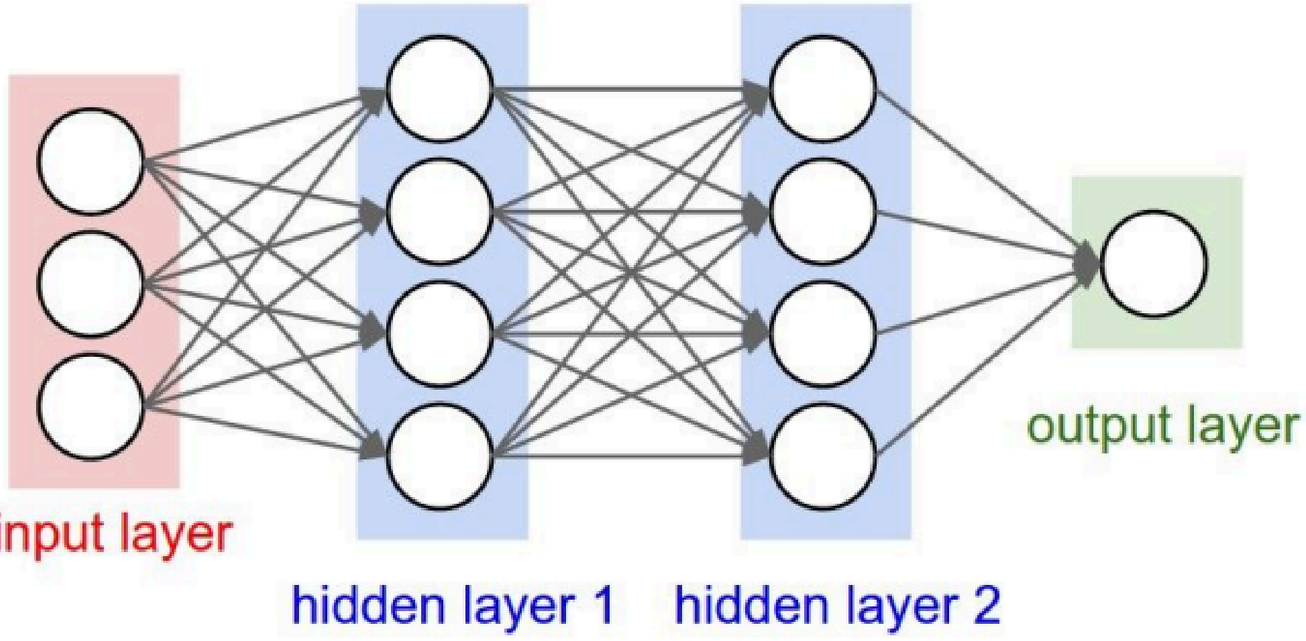

Input
Weight
x_0
x_1
x_2
x_m
w_0
w_1
w_2
w_m
Network
input function
Activation function
Σ
Output



Neural Network

Neural Networks: Architectures



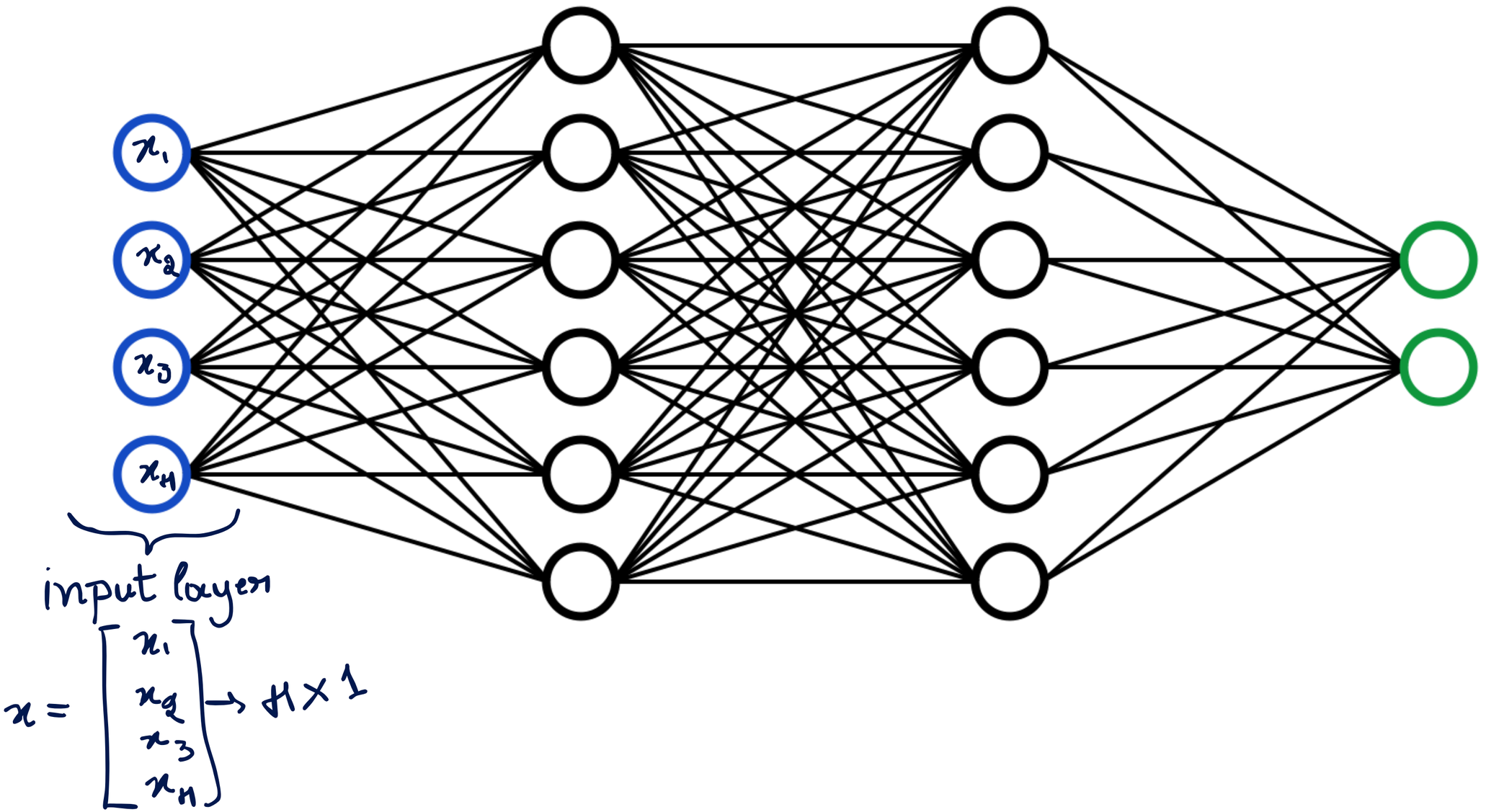


"2-layer Neural Net", or
"1-hidden-layer Neural Net"

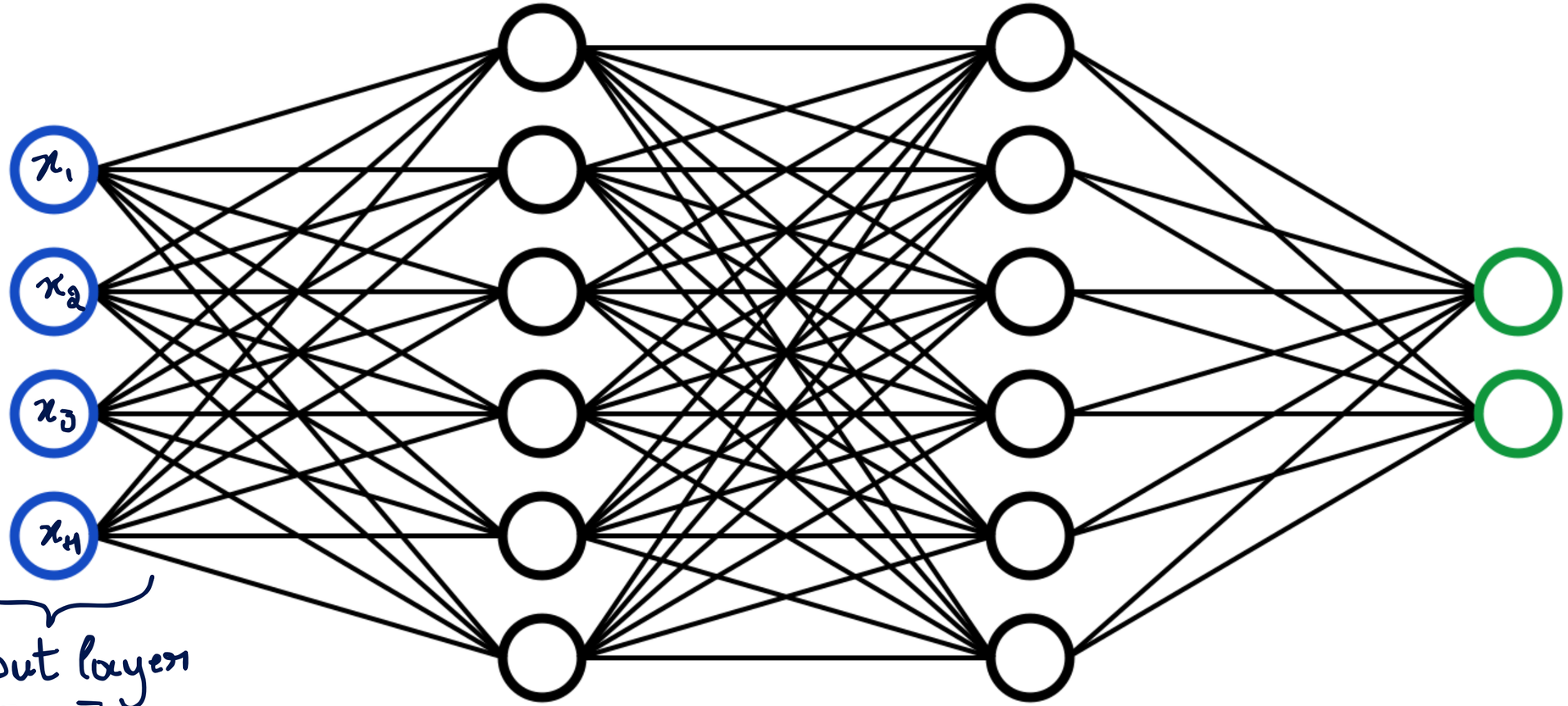
"3-layer Neural Net", or
"2-hidden-layer Neural Net"

"Fully-connected" layers

Representation of Layers in a NN



Representation of Layers in a NN



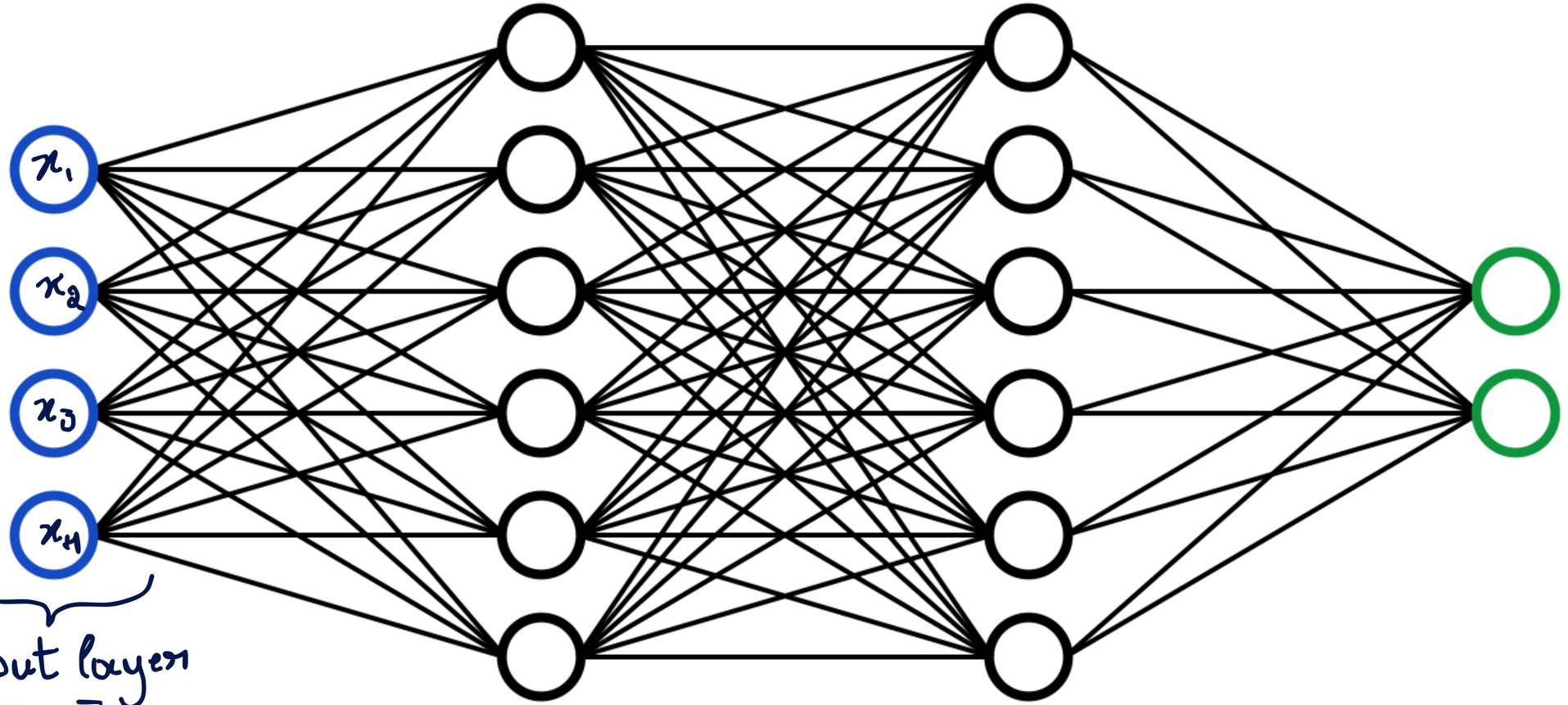
input layer

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_H \end{bmatrix} \rightarrow H \times 1$$

$$z_1 = xW_1 + b_1$$

$$a_1 = \sigma(z_1)$$

Representation of Layers in a NN



input layer

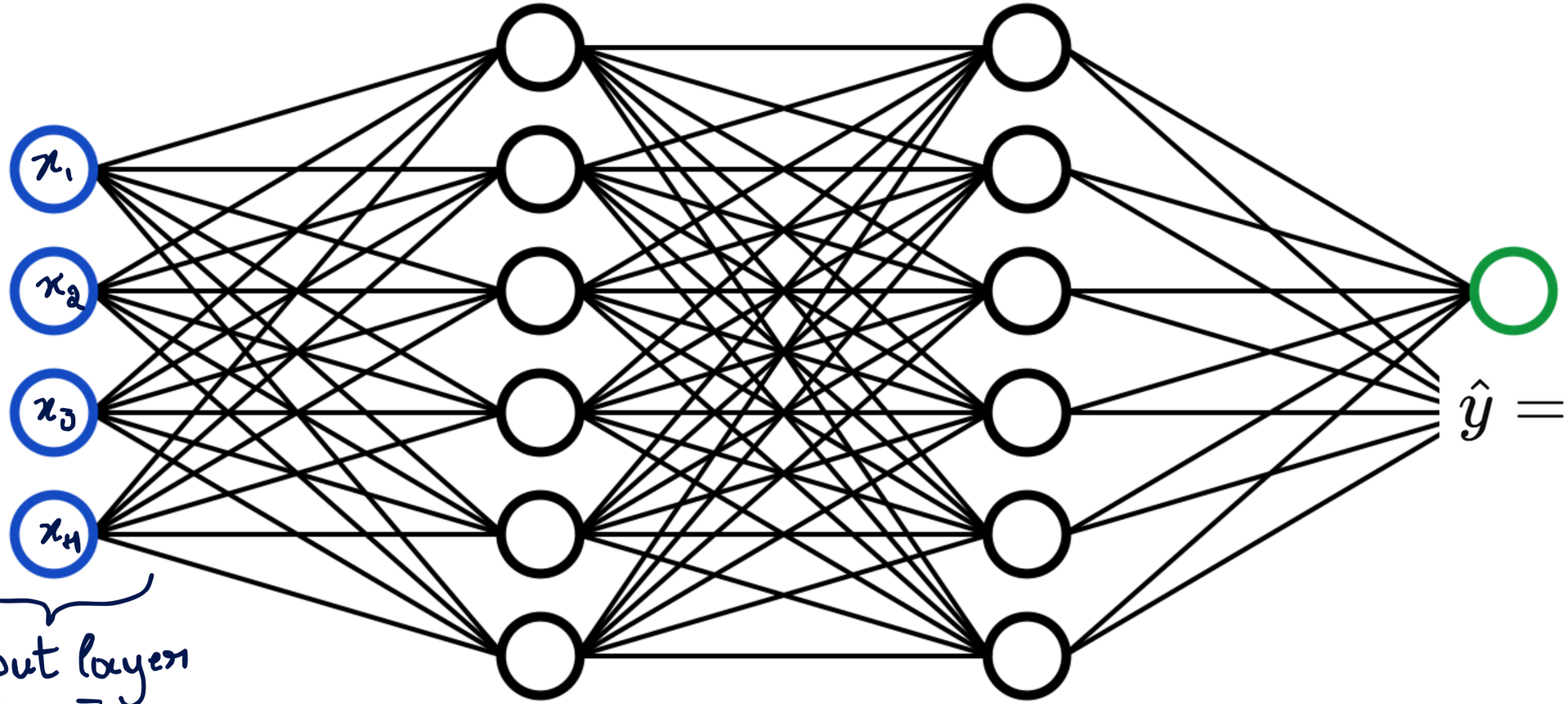
$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_H \end{bmatrix} \rightarrow H \times 1$$

$$z_1 = xW_1 + b_1$$

$$a_1 = \sigma(z_1)$$

$$z_2 = a_1 W_2 + b_2$$

Representation of Layers in a NN



input layer

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_H \end{bmatrix} \rightarrow H \times 1$$

$$z_1 = xW_1 + b_1$$

$$a_1 = \sigma(z_1)$$

$$z_2 = a_1 W_2 + b_2$$

$$\hat{y} = \sigma(z_2)$$

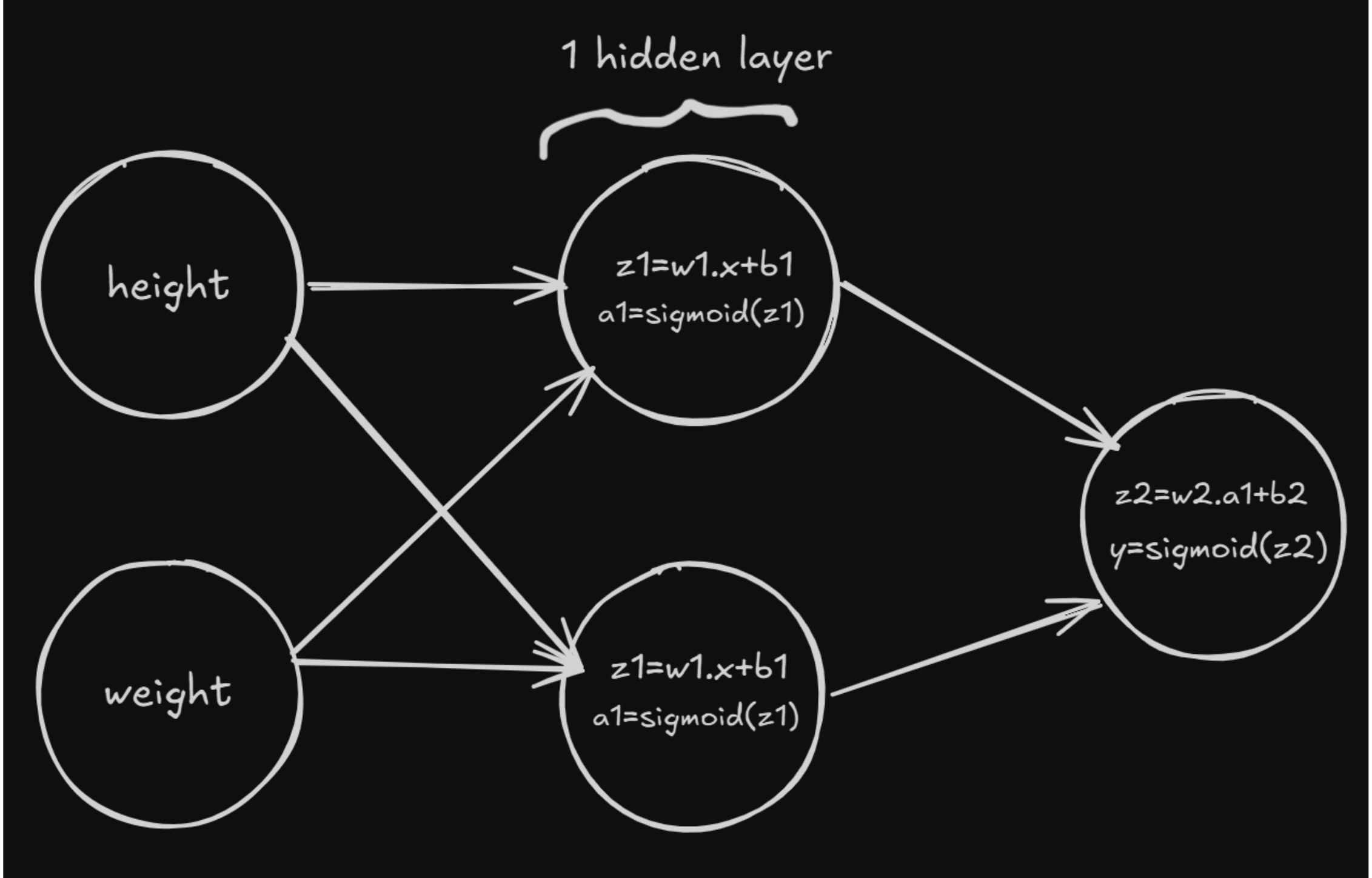
Training a Neural Network

- Forward Pass
- Backpropagation
- Loss Functions
- Gradient Descent
- Training Loop

Training Example

Let’s train our network to predict someone’s gender given their weight and height:

Name	Weight (lb)	Height (in)	Gender
Alice	133	65	F
Bob	160	72	M
Charlie	152	70	M
Diana	120	60	F


1 hidden layer
height
weight
z1=w1.x+b1
a1=sigmoid(z1)
z1=w1.x+b1
a1=sigmoid(z1)
z2=w2.a1+b2
y=sigmoid(z2)

Forward Pass

- The process through which data flows through the network
- Computation happens at each layer
- No logic, only calculating the weighted sum and applying activation
- Finally calculating the output at the final/output layer

$$z_1 = xW_1 + b_1$$
$$a_1 = \sigma(z_1)$$
$$z_2 = a_1W_2 + b_2$$
$$\hat{y} = \sigma(z_2)$$

finding the shape of weights and biases

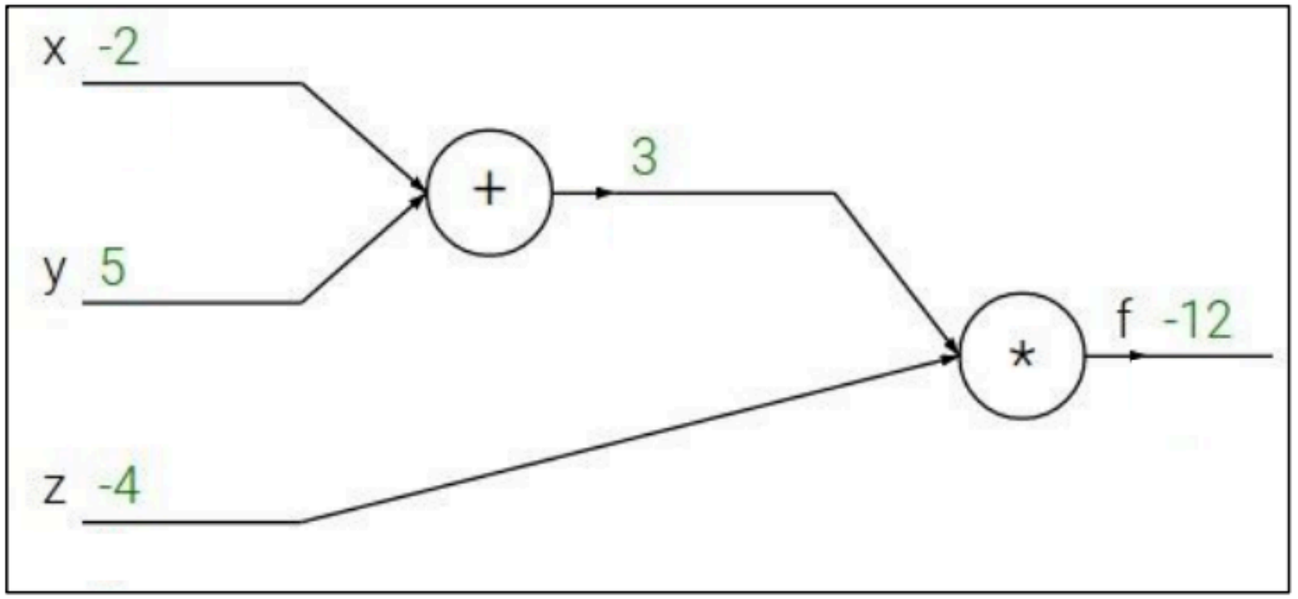
- remember that for matrix multiplications between input and weights or the activations and weights, the inner dimensions should always match.
  $(n, inner\ dimension)$, $(inner\ dimension, m)$
- for example in our case, the shape of our input is $(4, 2)$, so our weights in the hidden layer should be of the form $(2, number\ of\ neurons)$
- here is a simple trick to keep in mind:
  weights for the target layer from M neurons from the input layer to N neurons in the target layer should be of the shape .
- checking this for our case, we have 2 features in our input layer and 2 neurons in the hidden layer, so out weight should be of the shape $(2, 2)$.
- as for the shape of the biases, since we have 2 neurons in our hidden layer, each neuron has a bias, so the shape of the bias matrix is $(1, 2)$.

backpropagation/backprop

- this is by far the most important step in training a neural network
- in the feedforward neural network we used pre-set weights and biases, but we need to alter these weights and biases in such a way that our prediction is close to the true value as much as possible.
- to measure how close our predictions are from the true values, we use a standard known as loss functions, lower is the loss function, better is our Neural network

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

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



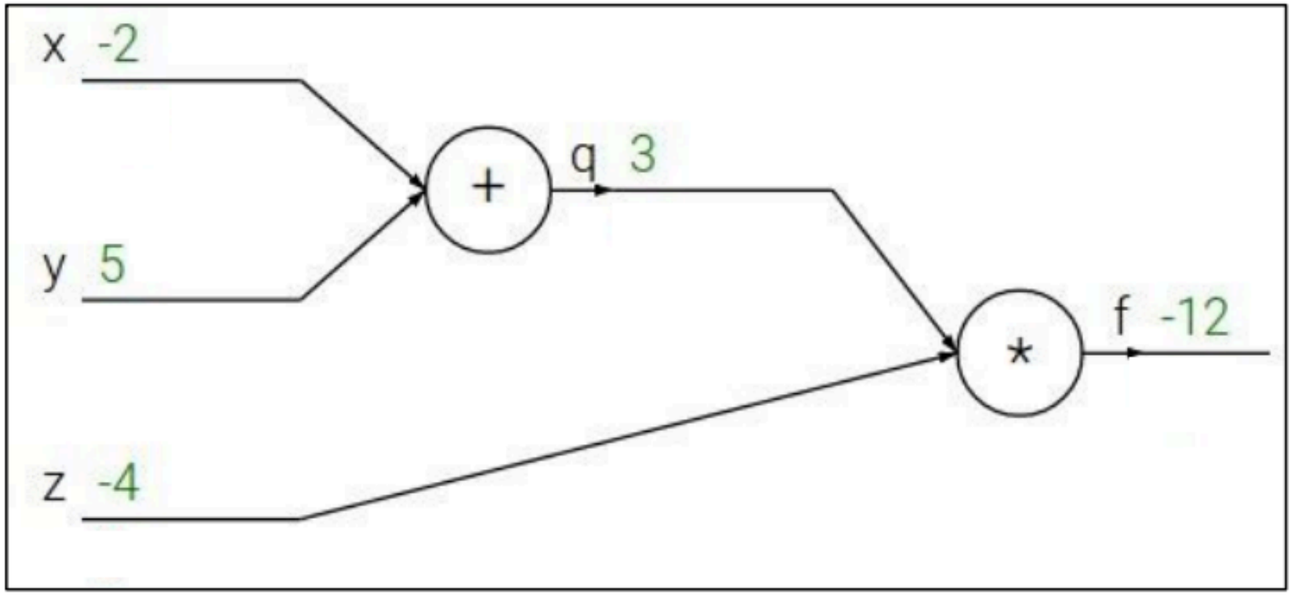
$$f(x, y, z) = (x + y)z$$

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



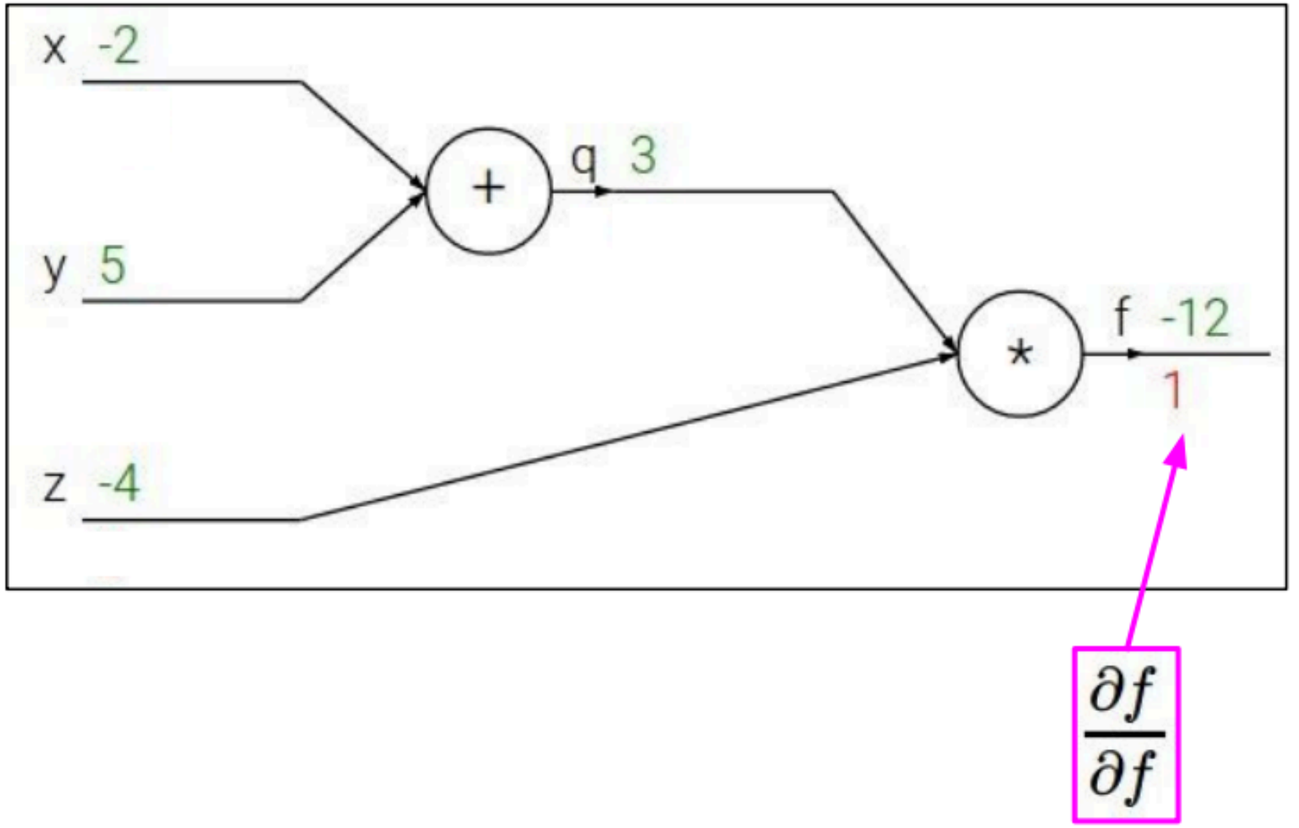
$$f(x, y, z) = (x + y)z$$

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



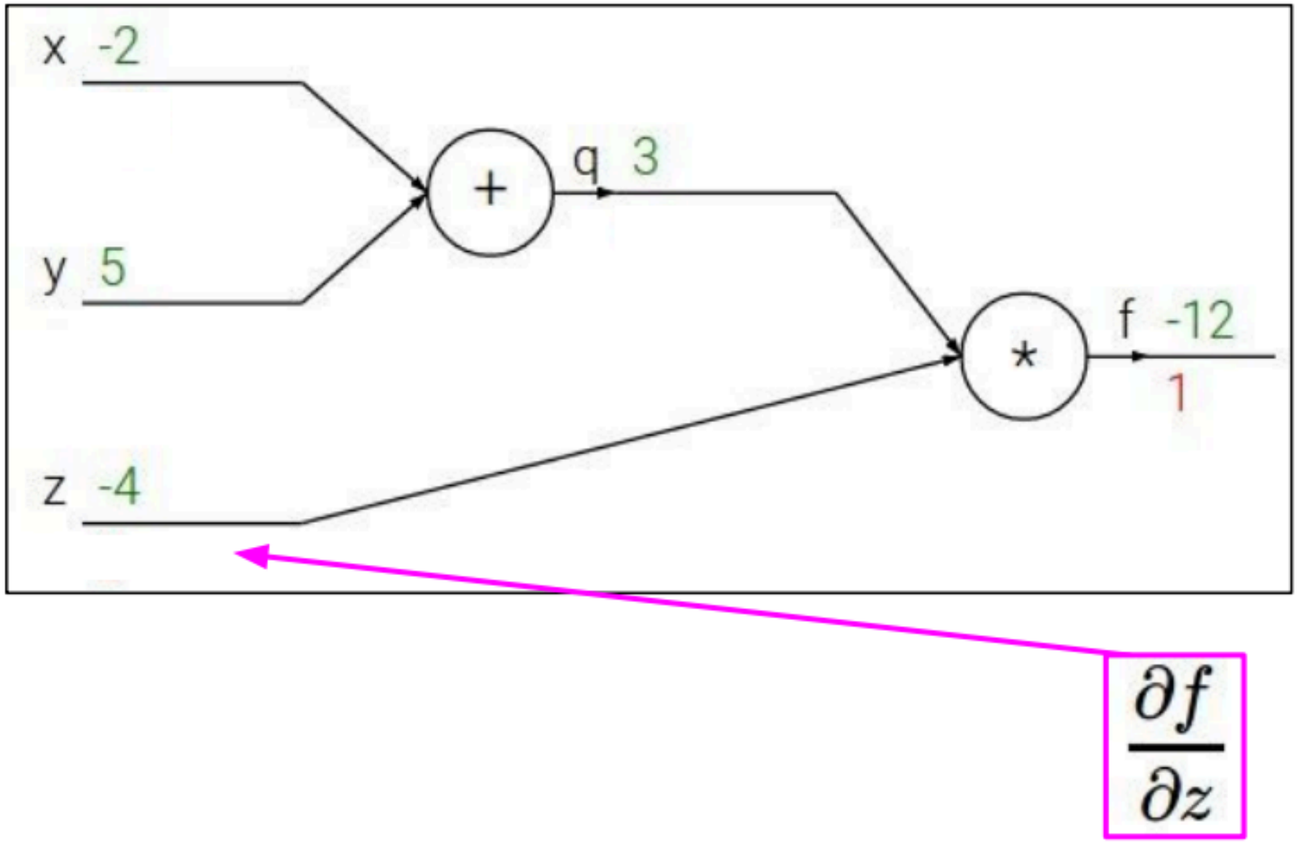
$$f(x, y, z) = (x + y)z$$

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



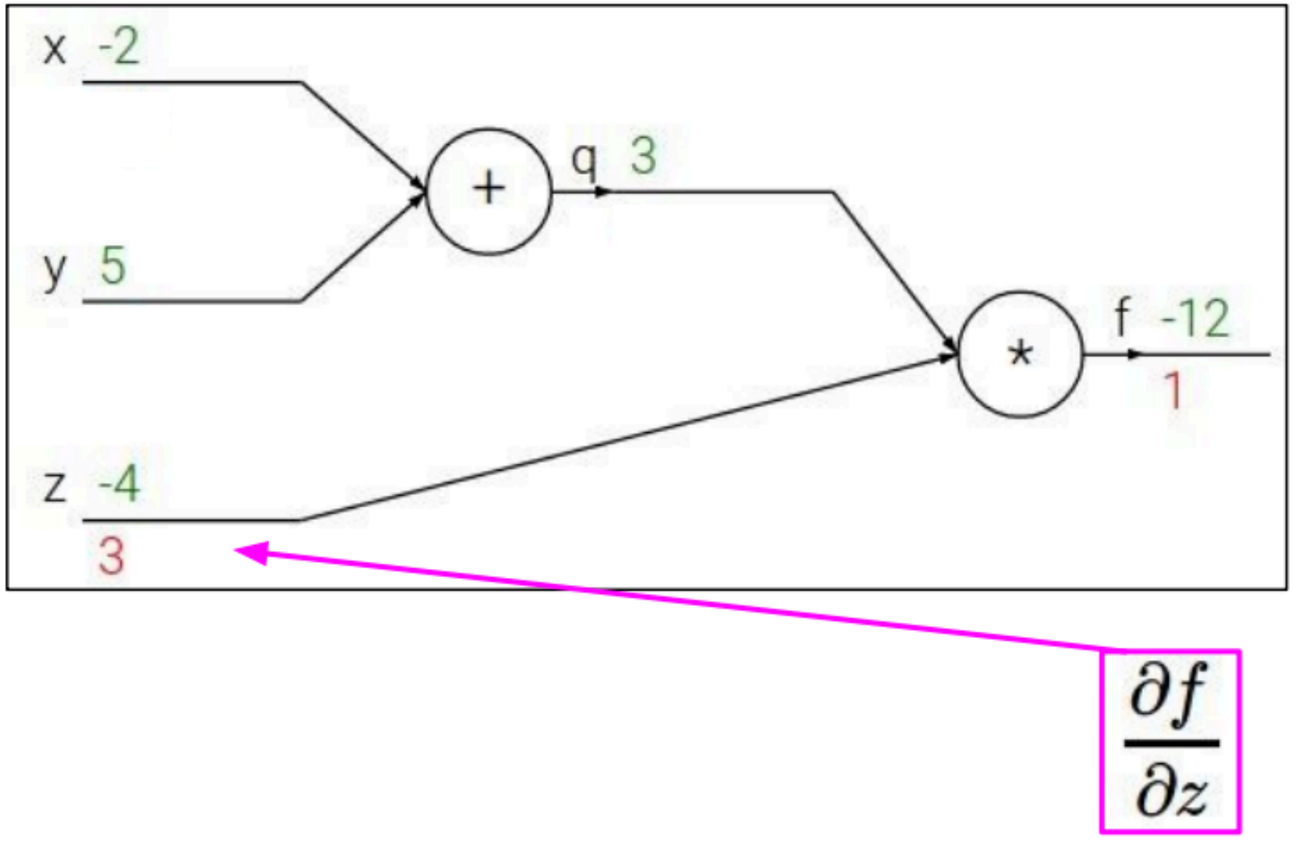
$$f(x, y, z) = (x + y)z$$

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



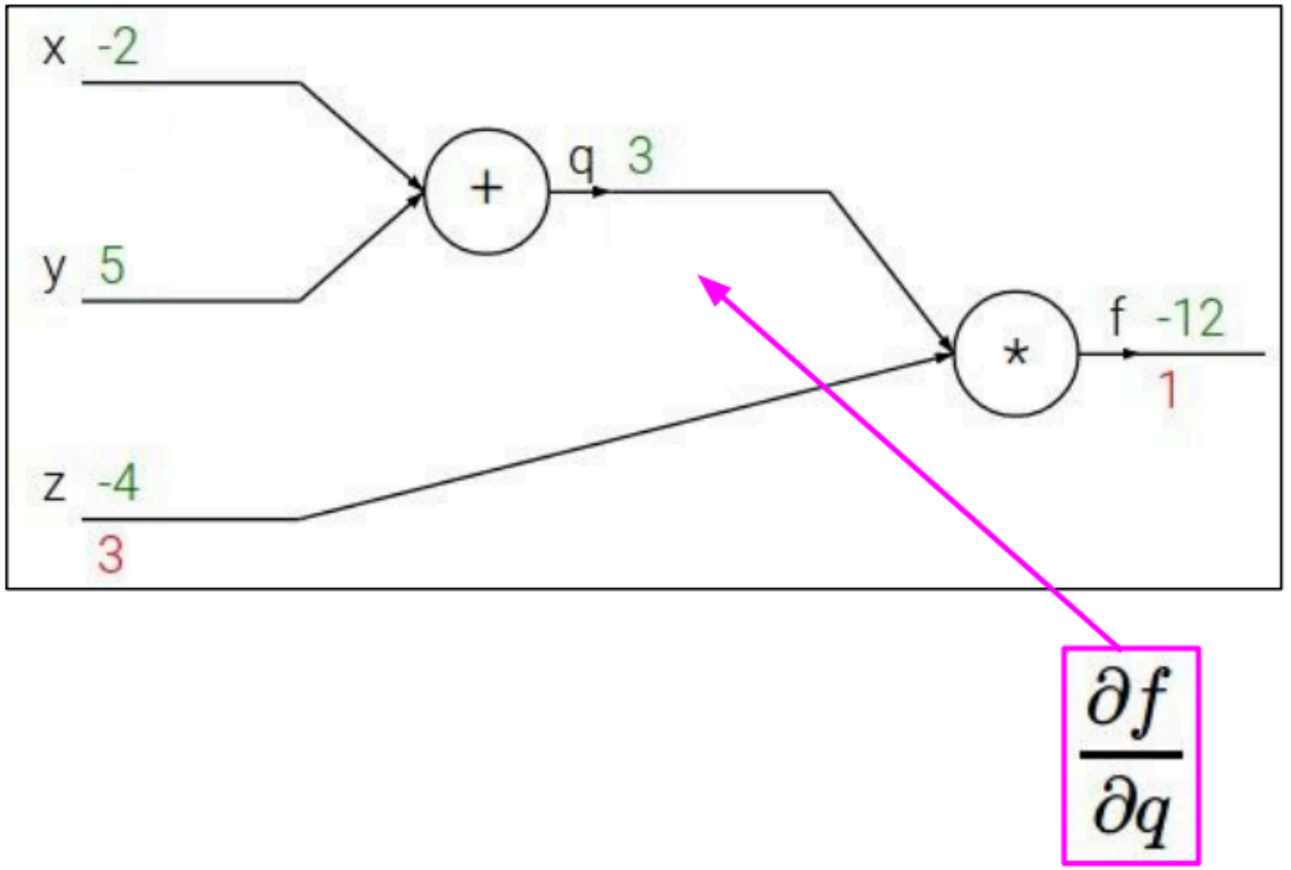
$$f(x, y, z) = (x + y)z$$

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



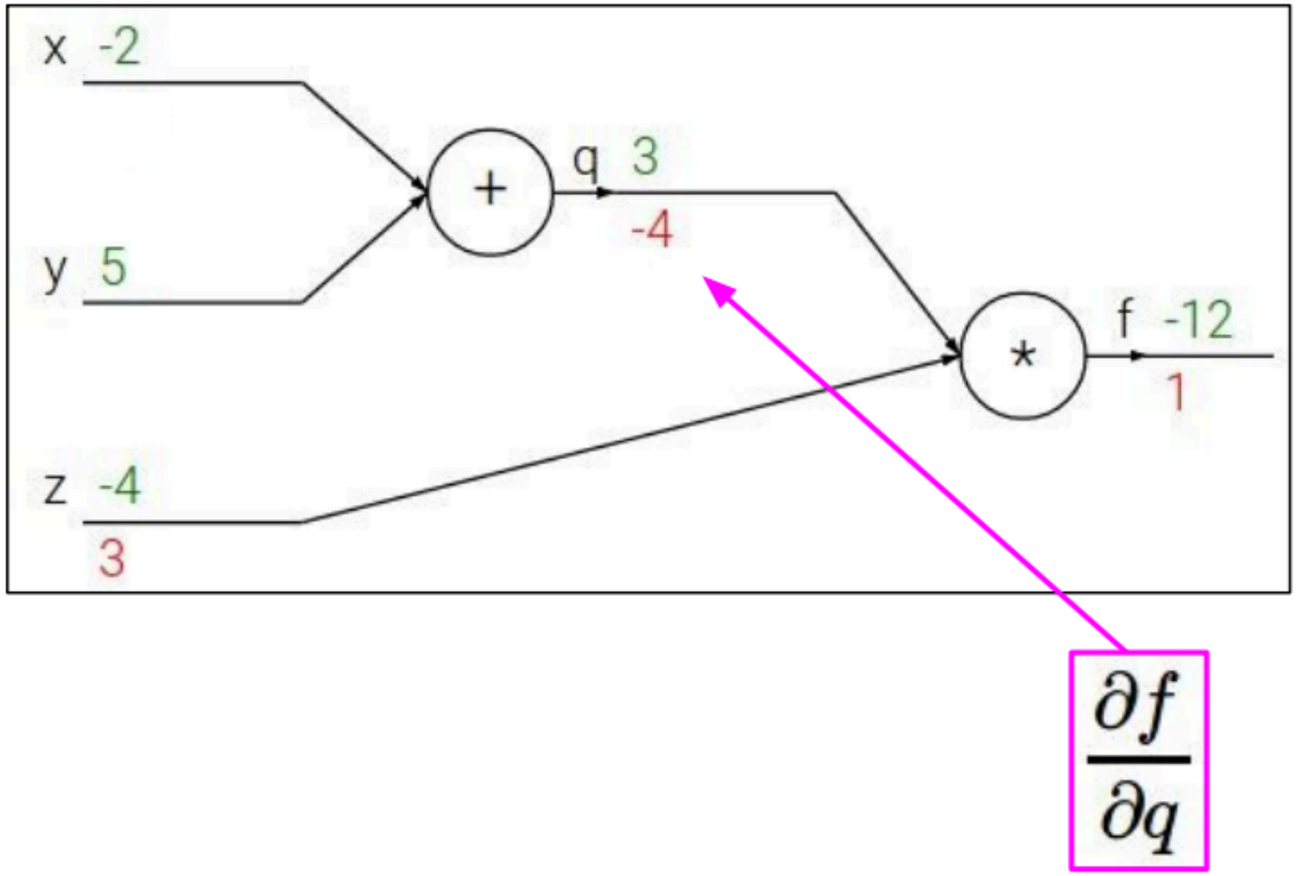
$$f(x, y, z) = (x + y)z$$

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



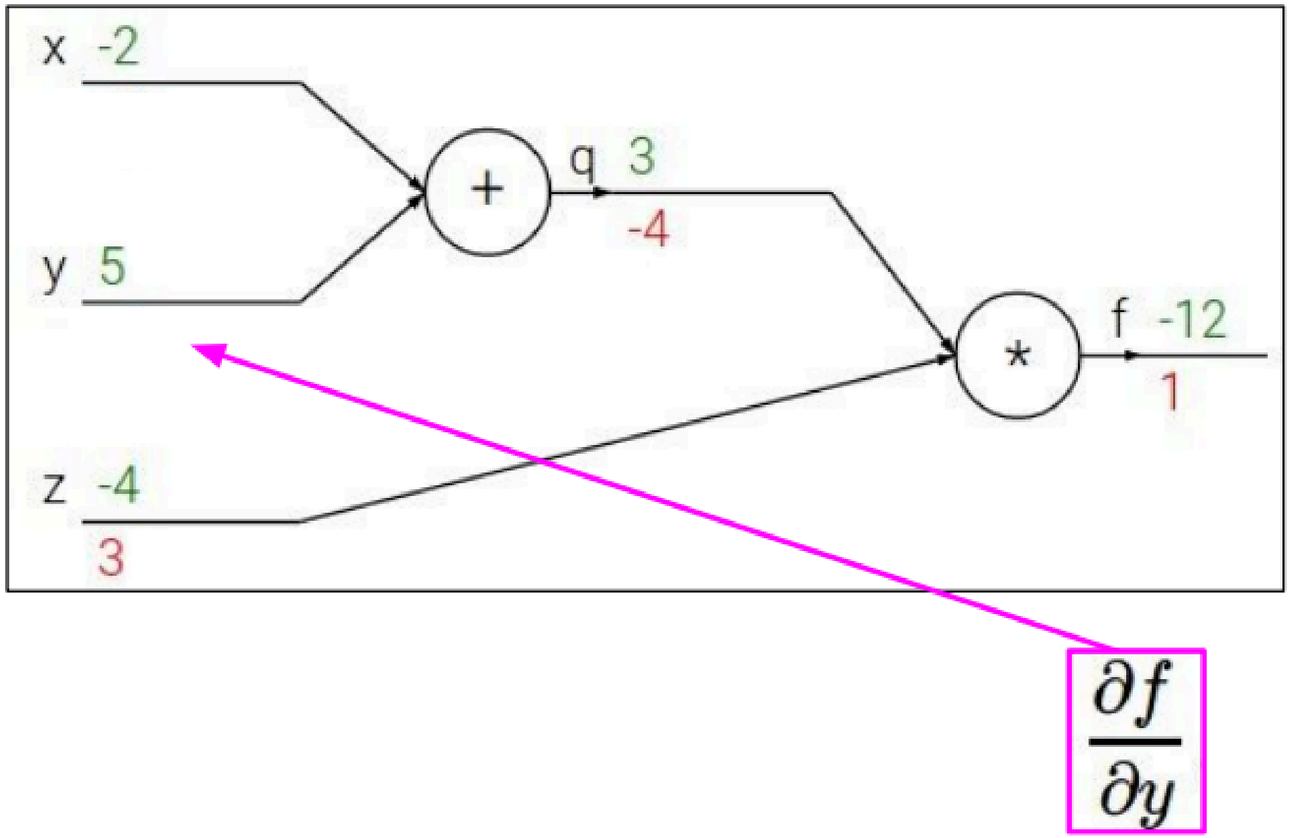
$$f(x, y, z) = (x + y)z$$

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



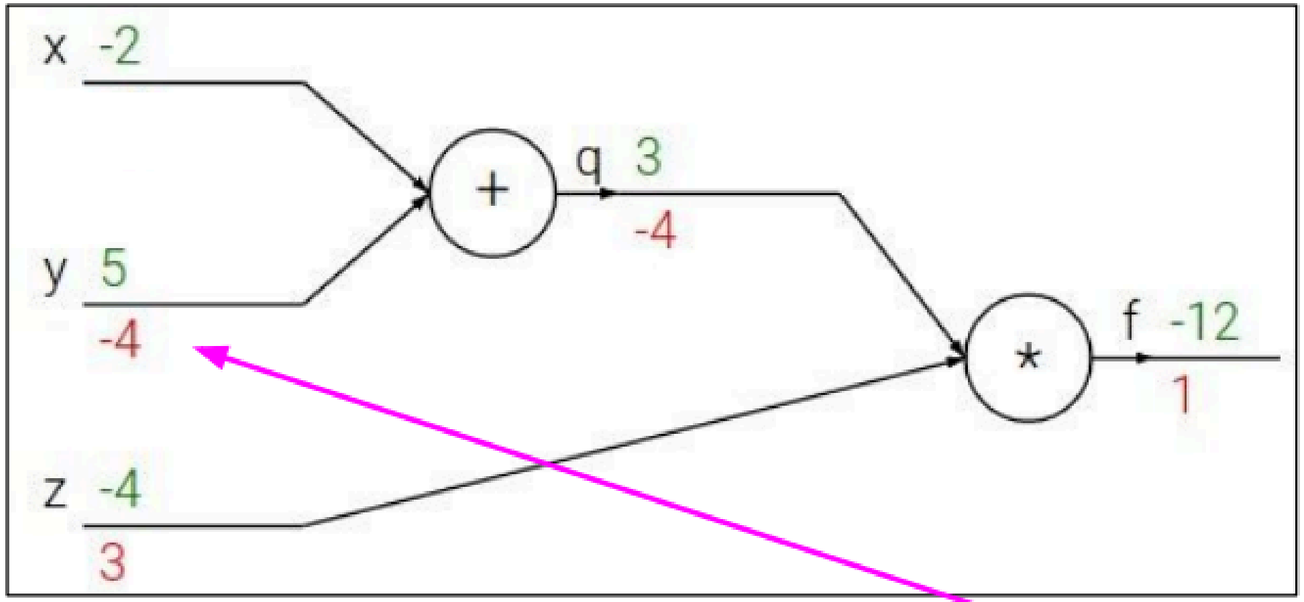
$$f(x, y, z) = (x + y)z$$

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



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

Chain rule:

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

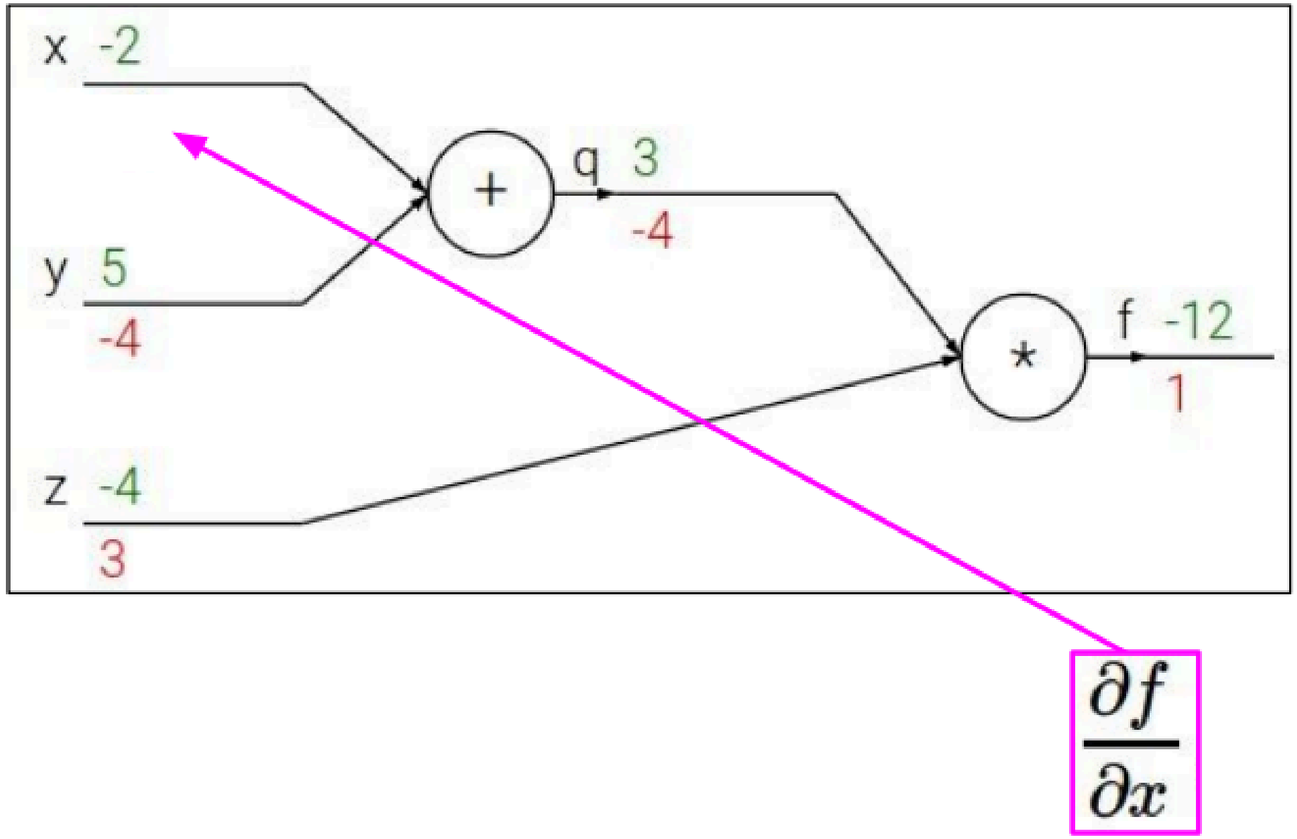
$$f(x, y, z) = (x + y)z$$

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



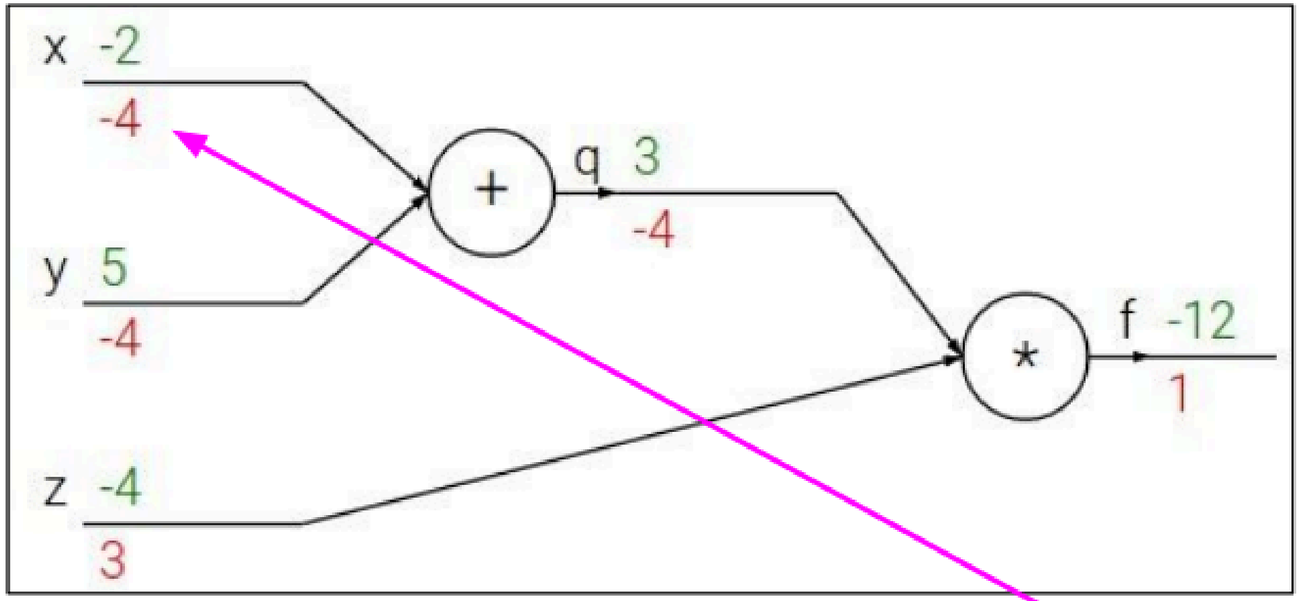
$$f(x, y, z) = (x + y)z$$

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

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

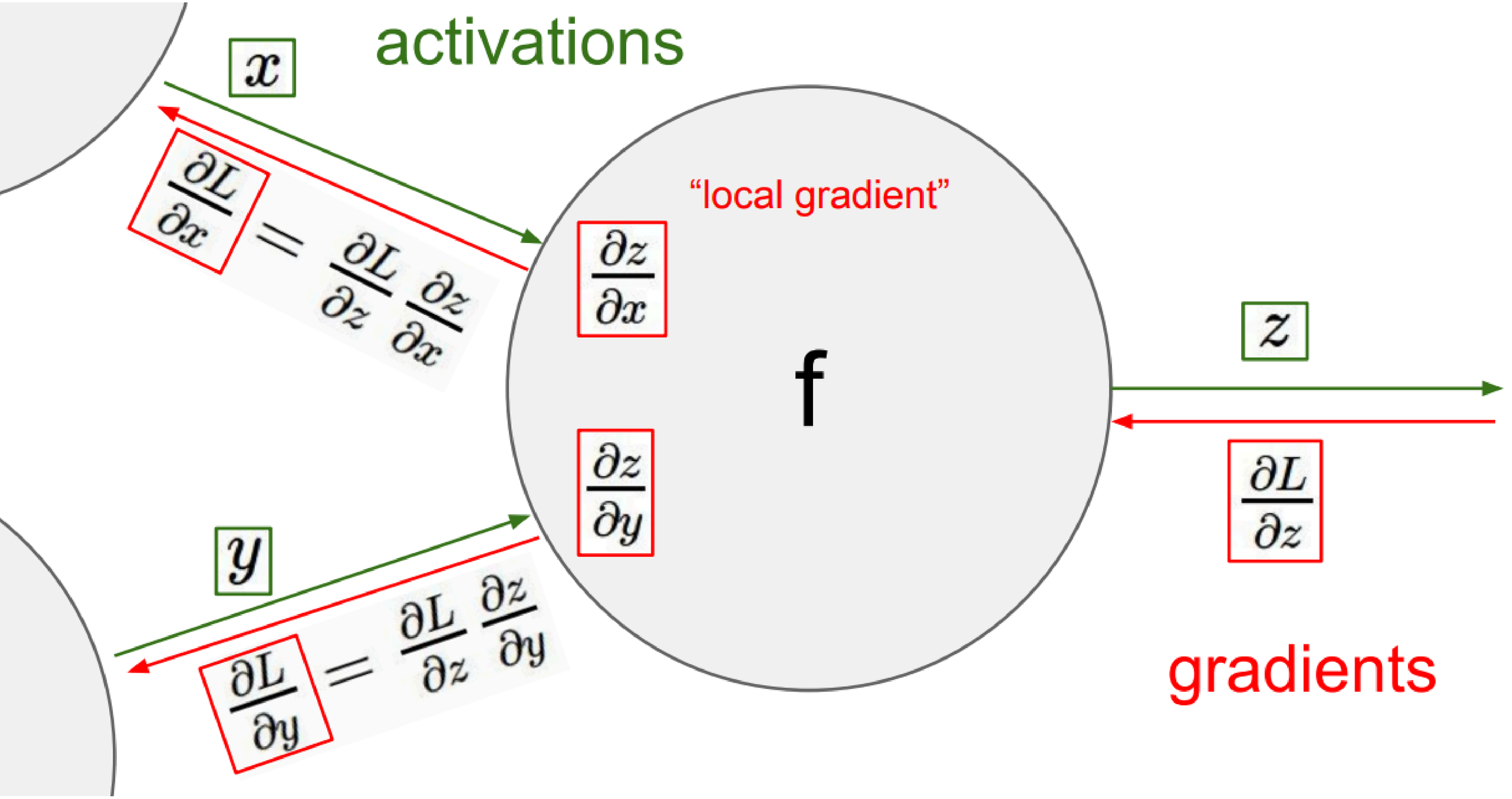
$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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



Chain rule:

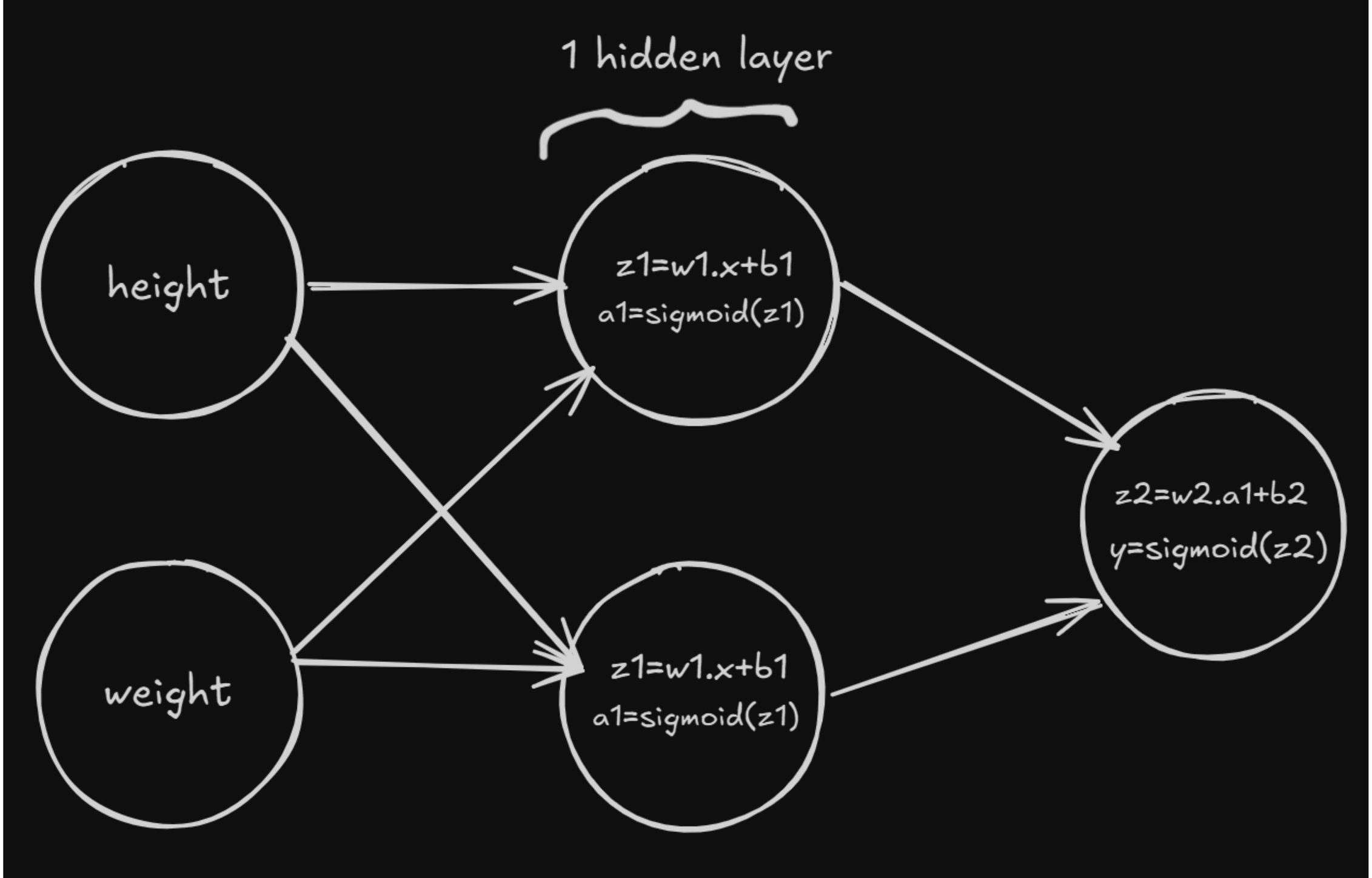
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$


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

Tranining the Neural Network

Let’s train our network to predict someone’s gender given their weight and height:

Name	Weight (lb)	Height (in)	Gender
Alice	133	65	F
Bob	160	72	M
Charlie	152	70	M
Diana	120	60	F


1 hidden layer
height
weight
z1=w1.x+b1
a1=sigmoid(z1)
z1=w1.x+b1
a1=sigmoid(z1)
z2=w2.a1+b2
y=sigmoid(z2)

Loss Function

Why a Loss Function?

Measures how far predictions are from the true values.
Goal: Minimize loss

Mean Squared Error (MSE):

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

Loss Function

Why a Loss Function?

Measures how far predictions are from the true values.
Goal: Minimize loss

Mean Squared Error (MSE):

good for regression

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

Loss Function

Binary Cross Entropy (BCE)

$$L = -\frac{1}{n}\sum[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

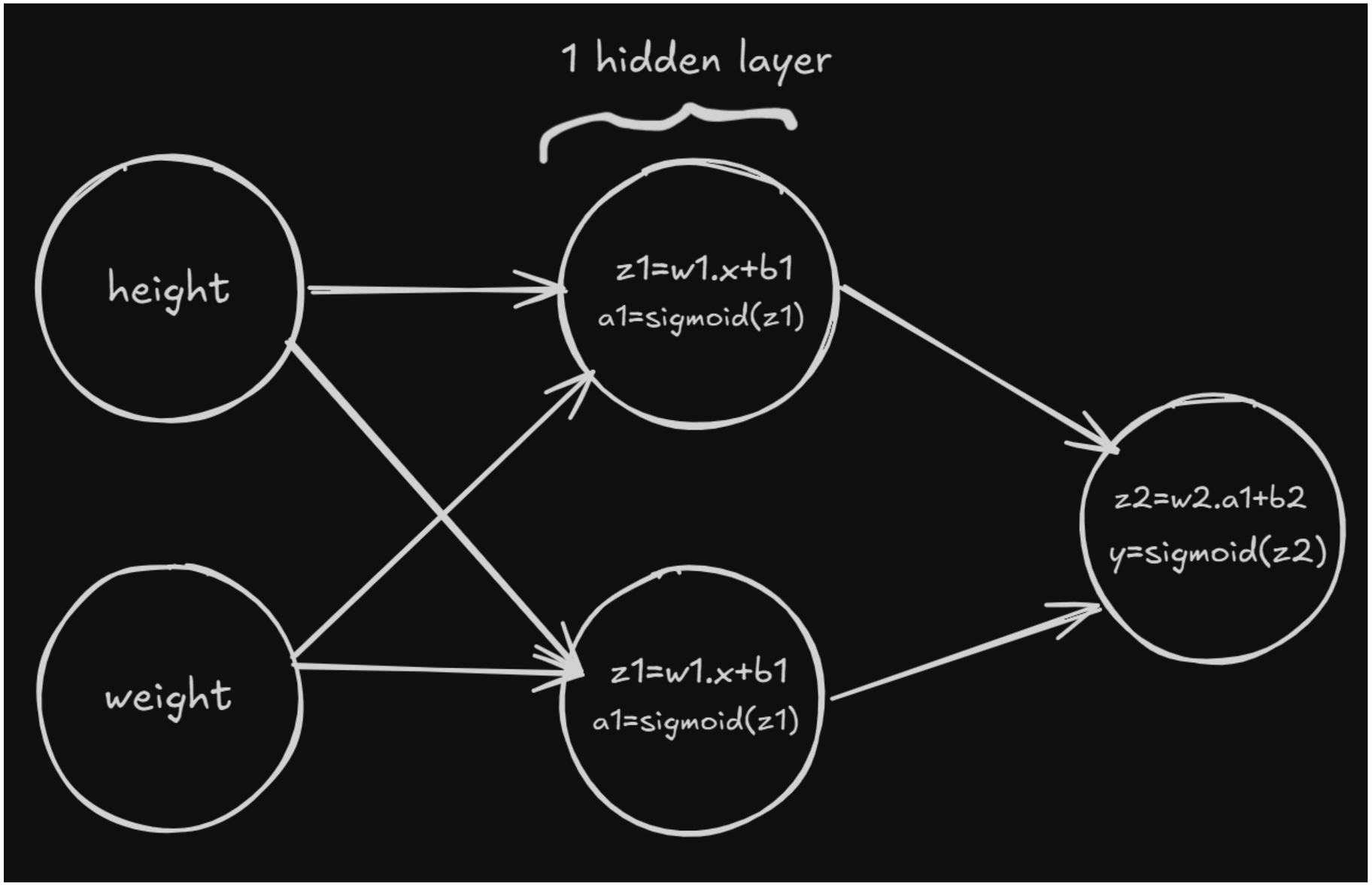
Punishes confident wrong predictions more strongly.

ideal for binary classification

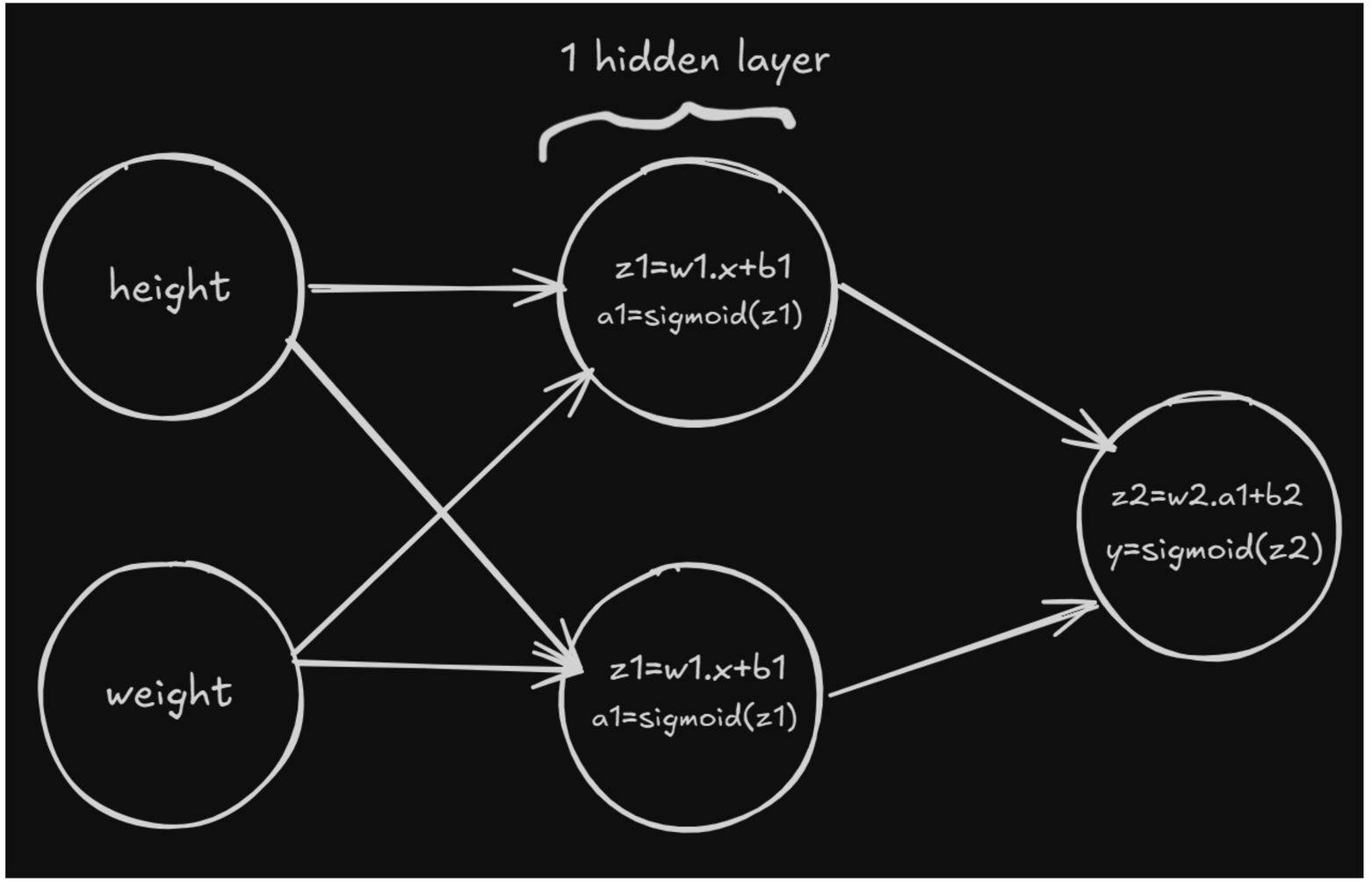
$$L = -\frac{1}{n}\sum[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

y_true	y_pred	Loss (approx)
1	0.95	0.05
1	0.05	3
0	0.95	3

forward pass



backpropagation



our aim:

to find

- $\frac{\partial L}{\partial w_2}$
- $\frac{\partial L}{\partial b_2}$
- $\frac{\partial L}{\partial w_1}$
- $\frac{\partial L}{\partial b_1}$

$$z_2 = a_1 \cdot w_2 + b_2$$

$$\hat{y} = \sigma(z_2) = \frac{1}{1 + e^{-z_2}}$$

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

what is dL/dz2

$$z_2 = a_1 \cdot w_2 + b_2$$

$$\hat{y} = \sigma(z_2) = \frac{1}{1 + e^{-z_2}}$$

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

what is dL/dz2

$$\frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2}$$

chain rule!!

find the derivative of loss wrt y

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

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}}$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{\hat{y} - y}{\hat{y}(1 - \hat{y})}$$

Derivative of the sigmoid

$$\hat{y} = \sigma(z_2) \Rightarrow \frac{\partial \hat{y}}{\partial z_2} = \hat{y}(1 - \hat{y})$$

Applying Chain Rule

$$\frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2}$$

$$\frac{\partial L}{\partial z_2} = \frac{\hat{y} - y}{\hat{y}(1 - \hat{y})} \cdot \hat{y}(1 - \hat{y})$$

$$\boxed{\frac{\partial L}{\partial z_2} = \hat{y} - y}$$

now how to find dL/dw2 ??

we already have: $$\boxed{\frac{\partial L}{\partial z_2} = \hat{y} - y}$$

and since: $$z_2 = a_1 \cdot w_2 + b_2 \Rightarrow \frac{\partial z_2}{\partial w_2} = a_1$$

applying chain rule

$$\frac{\partial L}{\partial w_2} = a_1^T . dz_2$$

3. $\frac{\partial L}{\partial b_2}$

b_2 is added directly to z_2, so the gradient is just:

$$\frac{\partial L}{\partial b_2} = \sum dz_2$$

4. $\frac{\partial L}{\partial a_1}$

we now move backward into the hidden layer.

we need the gradient w.r.t. a_1, because it affects the loss through z_2.

$\frac{\partial L}{\partial a_1} = dz_2 \cdot w_2^T$

$$z_2 = a_1 \cdot w_2 + b_2 \Rightarrow \frac{\partial z_2}{\partial w_2} = a_1$$

5. $\frac{\partial L}{\partial z_1}$

now we apply the derivative of the sigmoid function at the hidden layer:

$$\frac{\partial a_1}{\partial z_1} = \sigma(z_1) \cdot (1 - \sigma(z_1)) = a_1 \cdot (1 - a_1)$$

because derivative of $\sigma(x) = \sigma(x).(1 - \sigma(x))$

using chain rule again:

$$\frac{\partial L}{\partial z_1} = \frac{\partial L}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \Rightarrow dz_1 = \frac{\partial L}{\partial a_1} \cdot a_1 \cdot (1 - a_1)$$

6. $\frac{\partial L}{\partial w_1}$

same logic as we did with w_2:

$$\frac{\partial L}{\partial w_1} = x^T \cdot dz_1$$

7. $\frac{\partial L}{\partial b_1}$

bias gradients are just summed across the batch:

$$\frac{\partial L}{\partial b_1} = \sum dz_1$$

hands on session