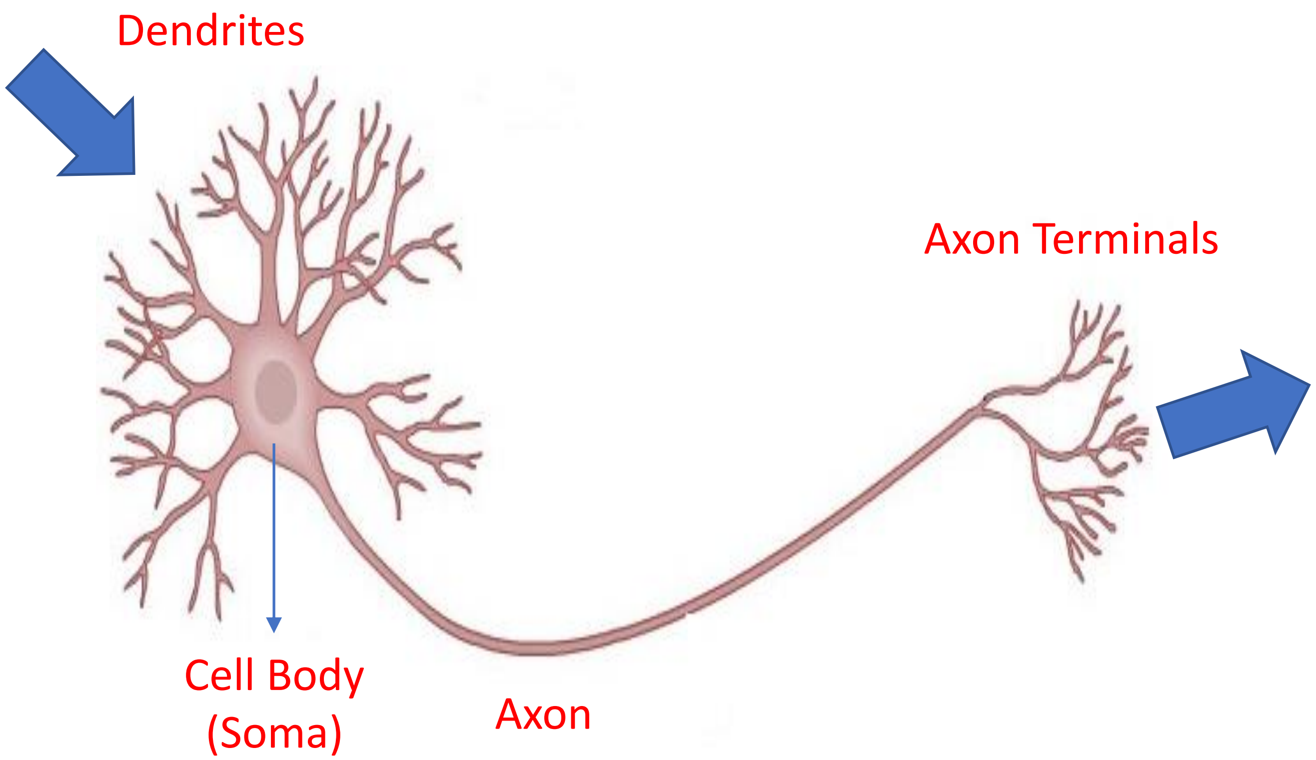
Deep Learning (Introduction)

Sadegh Eskandari
Department of Computer Science, University of Guilan
eskandari@guilan.ac.ir

Today …

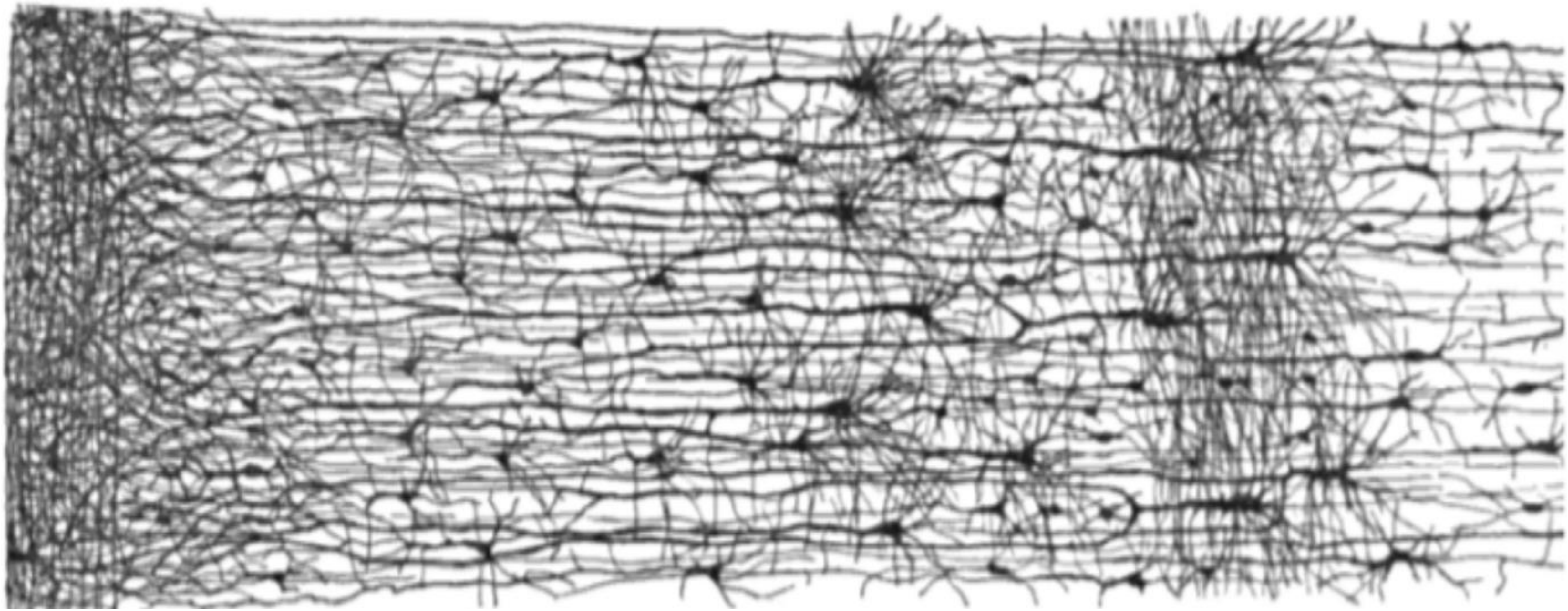
- Biological Neurons
- Artificial Neurons
- Multi-Layer Perceptron
- Training a Single Neuron
- Training an MLP (Backpropagation)

Biological Neurons



- Dendrites receive impulses (synapses) from other cells.
- The cell body fires (sends signal) if the received impulses are more than a threshold.
- Signal travels through the axon and is delivered to other cells via axon terminals

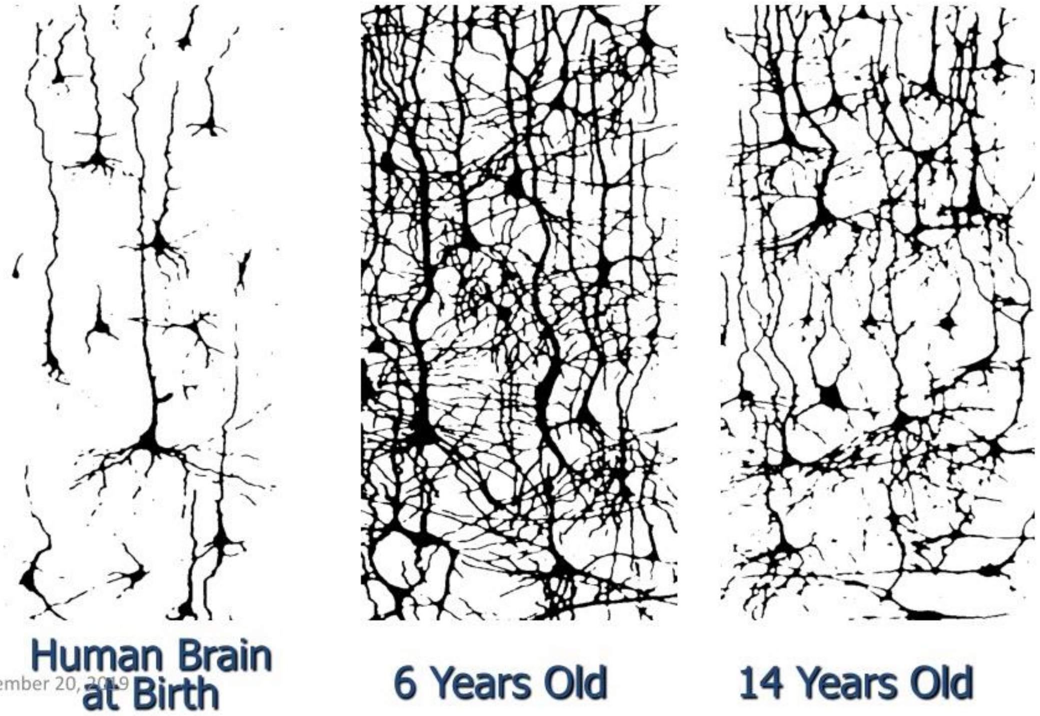
Biological Neurons



Multiple layers in a biological neural network of human cortex

Source: Montesinos López, O.A., Montesinos López, A. and Crossa, J., 2022. Fundamentals of Artificial Neural Networks and Deep Learning. In Multivariate Statistical Machine Learning Methods for Genomic Prediction (pp. 379-425). Springer, Cham.

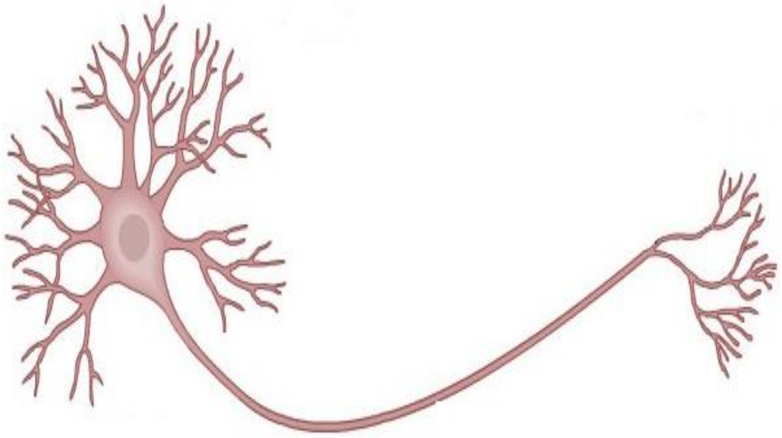
Biological Neurons

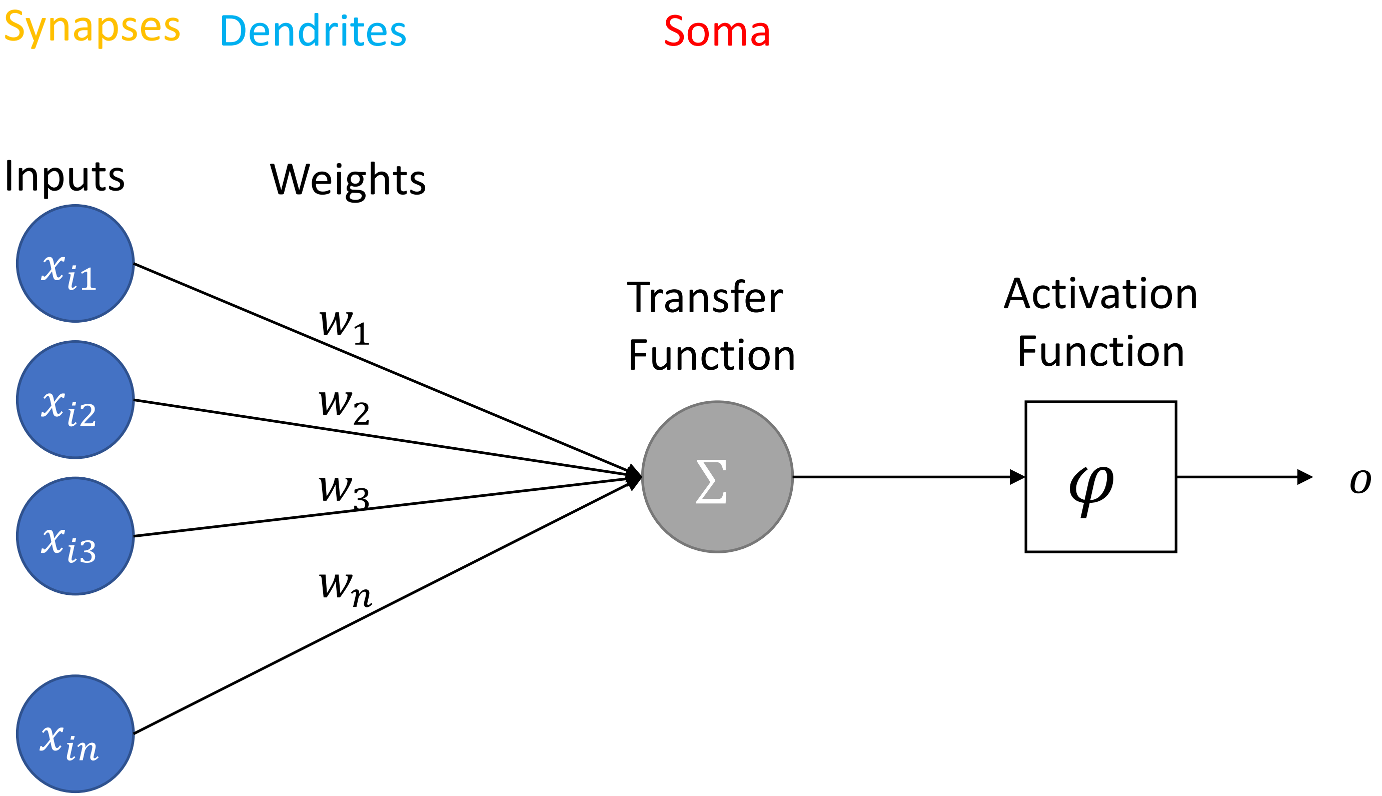


The postnatal development of the human cerebral cortex

Source: https://developingchild.harvard.edu/

Artificial Neurons

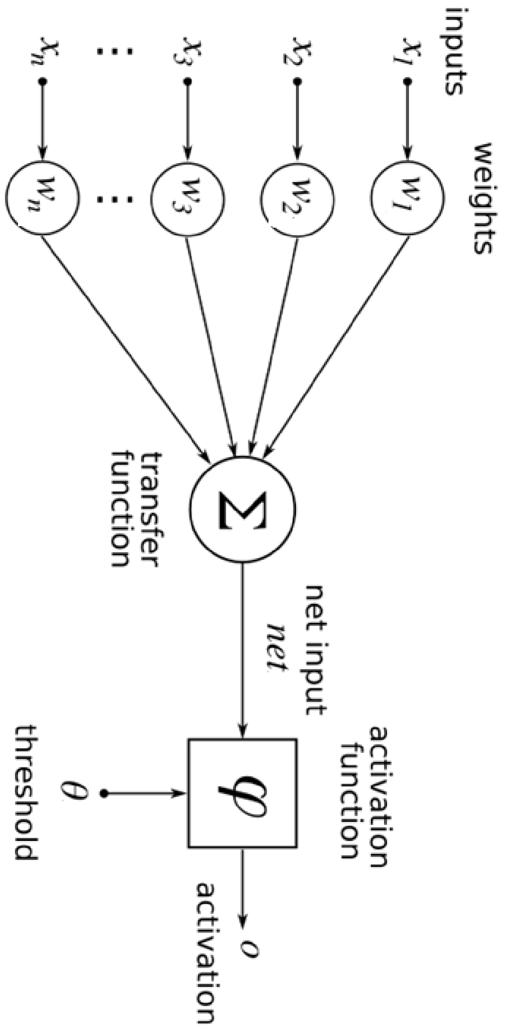




$$o = \varphi\left(\sum_{j=1}^{n} w_j x_j\right) = \varphi(\mathbf{w}^T \mathbf{x})$$

Where $\boldsymbol{w} = [w_1, w_2, \dots, w_n]^T$ and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$

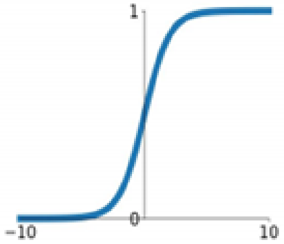
Artificial Neurons



Regular Activation Functions (φ)

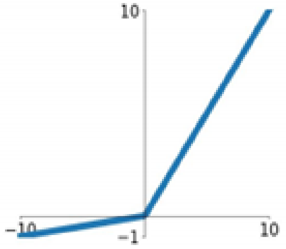
Sigmoid

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



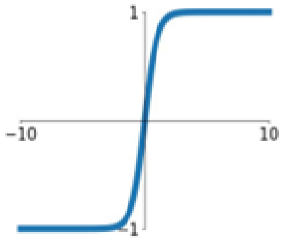
Leaky ReLU

$\max(0.1x, x)$



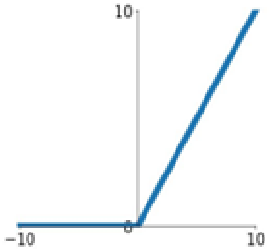
tanh

$\tanh(x)$



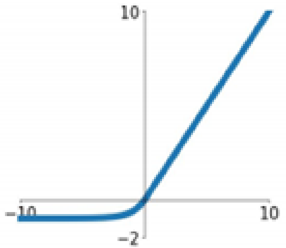
ReLU

$\max(0, x)$



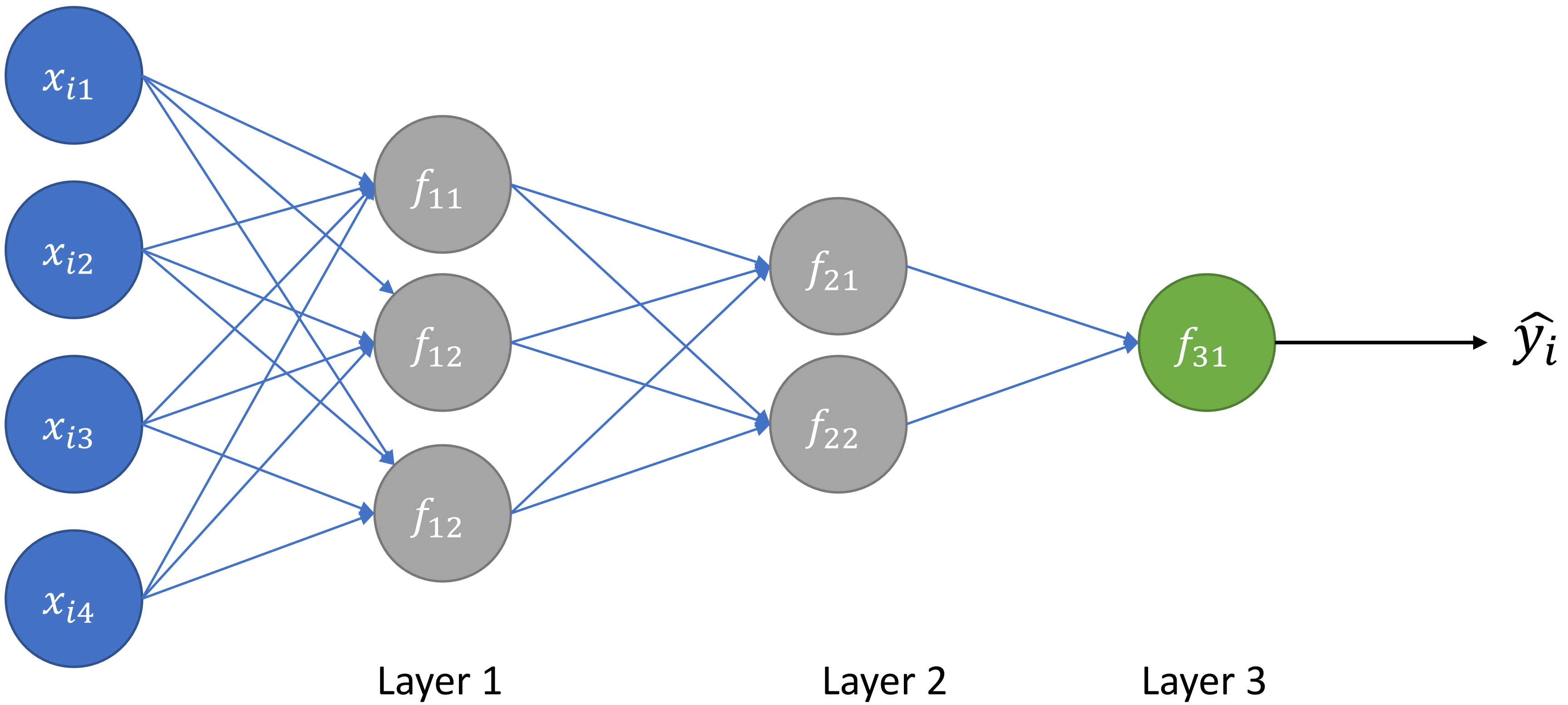
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



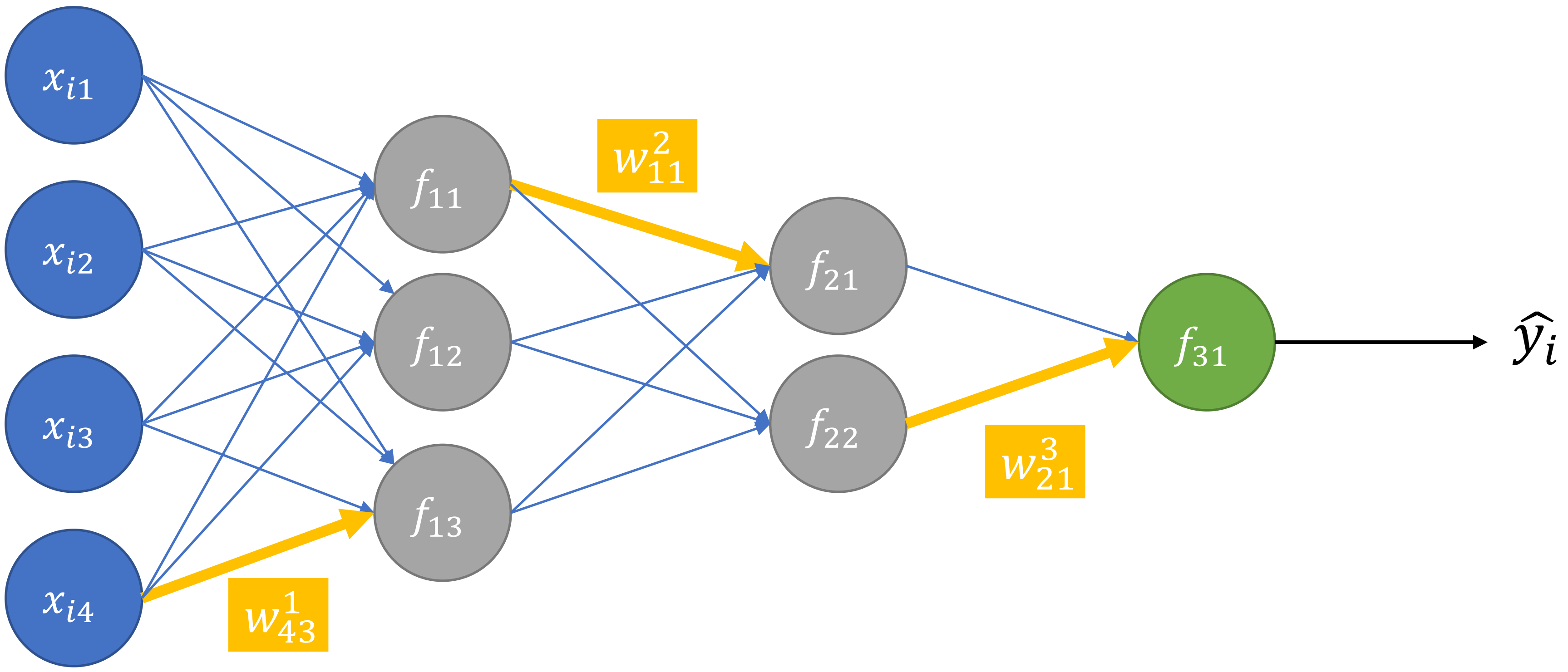
Multi-Layer Perceptron (MLP): Notations

3-Layer Fully-Connected Artificial Neural Network



Multi-Layer Perceptron (MLP): Notations

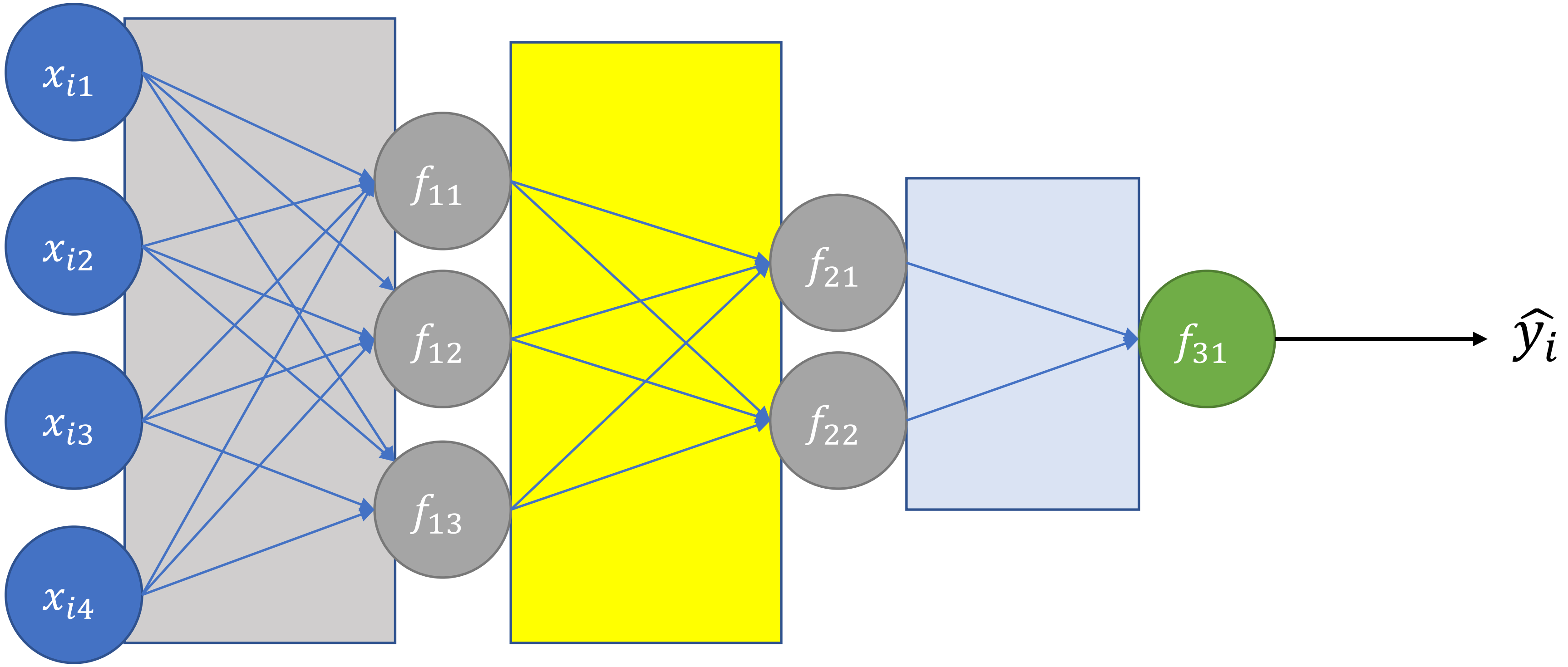
3-Layer Fully-Connected Artificial Neural Network



w_{ij}^{l} The weight of edge from neuron i of layer $l-1$ to neuron j of layer l

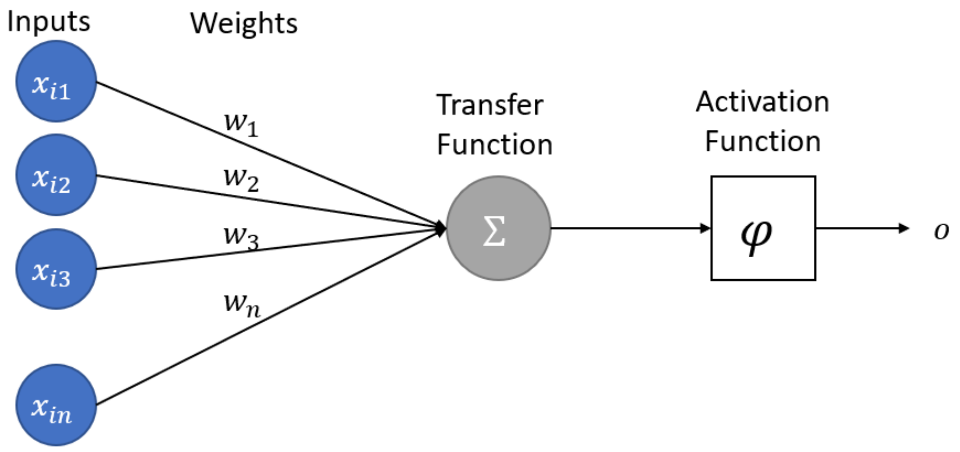
Multi-Layer Perceptron (MLP): Notations

3-Layer Fully-Connected Artificial Neural Network



$$w^1 = \begin{bmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 \\ w_{31}^1 & w_{32}^1 & w_{33}^1 \\ w_{41}^1 & w_{42}^1 & w_{43}^1 \end{bmatrix} \quad w^2 = \begin{bmatrix} w_{11}^2 & w_{12}^2 \\ w_{21}^2 & w_{22}^2 \\ w_{31}^2 & w_{32}^2 \end{bmatrix} \quad w^3 = \begin{bmatrix} w_{11}^3 \\ w_{21}^3 \end{bmatrix}$$

Training a Single Neuron



$$o = \varphi\left(\sum_{j=1}^{n} w_j x_j\right) = \varphi(\mathbf{w}^T \mathbf{x})$$

$$\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N)^{\boldsymbol{T}}$$

$$\mathbf{x_i} = (x_{i1}, x_{i2}, \dots, x_{in})^{\boldsymbol{T}}$$

$$\mathbf{y} = (y_1, y_2, \dots, y_N)^{\boldsymbol{T}}$$

Training data

N: num of data

n: dimension of each data point

Step 1: Loss function

$$l_i = (y_i - o_i)^2 = \left(y_i - \varphi(\boldsymbol{w}^T \boldsymbol{x}_i)\right)^2$$

$$L = \sum_{i=1}^{N} l_i$$

Step 2: The objective function

$$w^* = \arg\min_{\boldsymbol{w}} L$$

Step 3: Optimization

Initialize $\boldsymbol{w}$

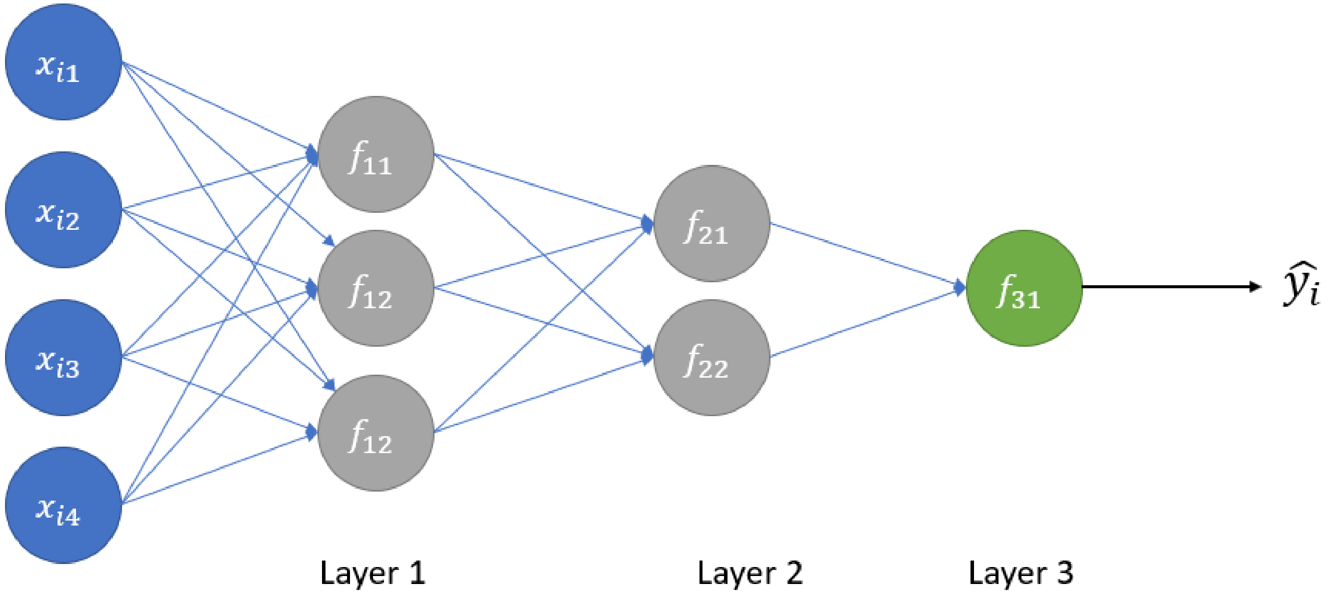
for $iter = 1$ to K:

$$\nabla_{\boldsymbol{w}} L = \left[\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \dots, \frac{\partial L}{\partial w_n}\right]$$

$$\boldsymbol{w}_{new} = \boldsymbol{w}_{old} - \gamma \nabla_{\boldsymbol{w}} L$$

Training an MLP (Backpropagation)

- As we know, $\nabla_W L$ plays a central role in finding the optimal parameters of model
- But calculating the gradients in multi-layer neural networks is not a trivial task!!

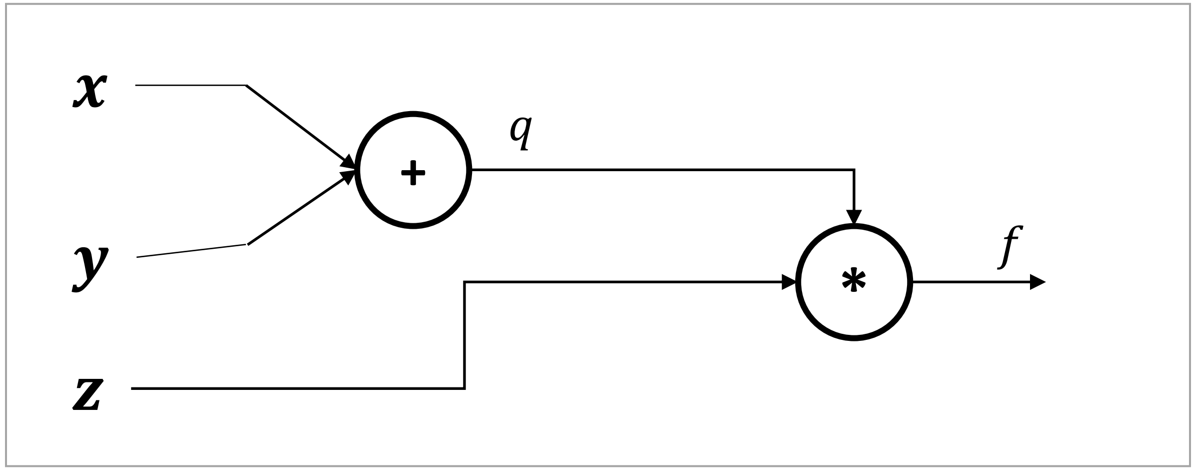


- Solution: Backpropagation (Rumelhart et al., 1986a)

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

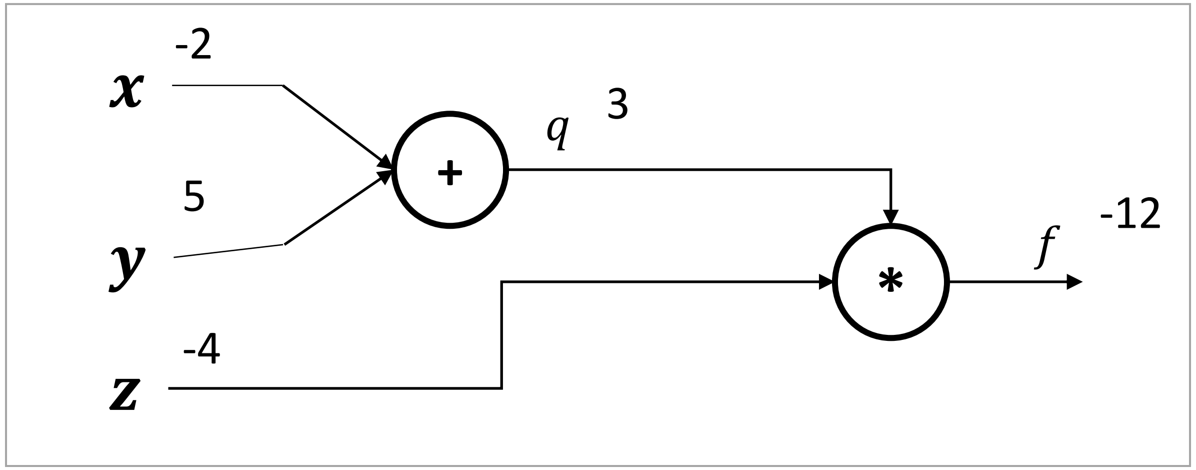


Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

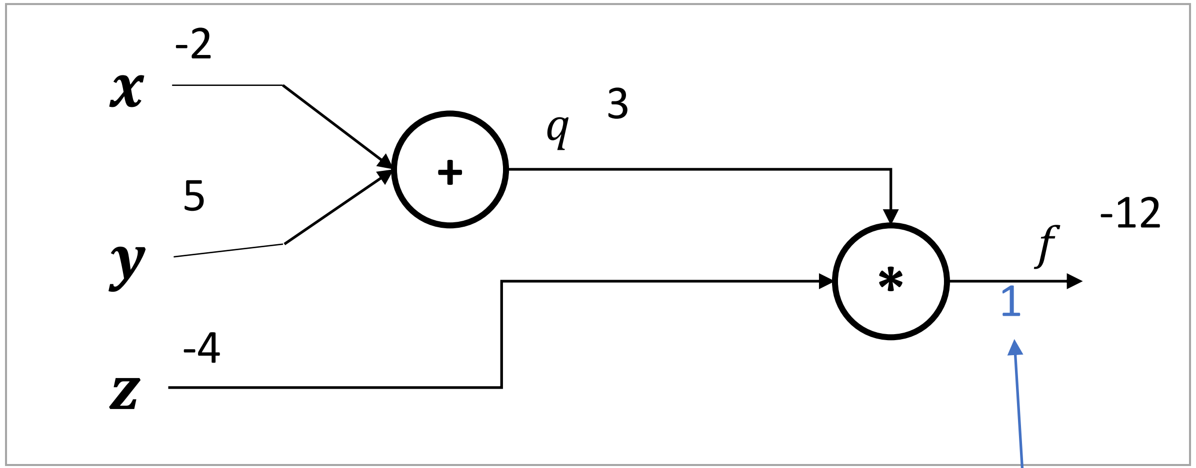
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

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

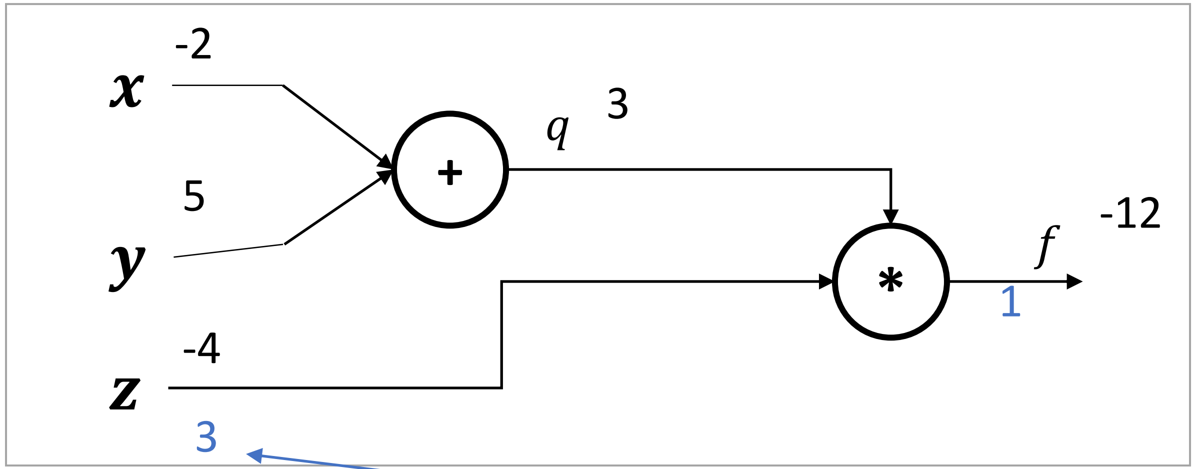
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

$\dfrac{\partial f}{\partial z}$

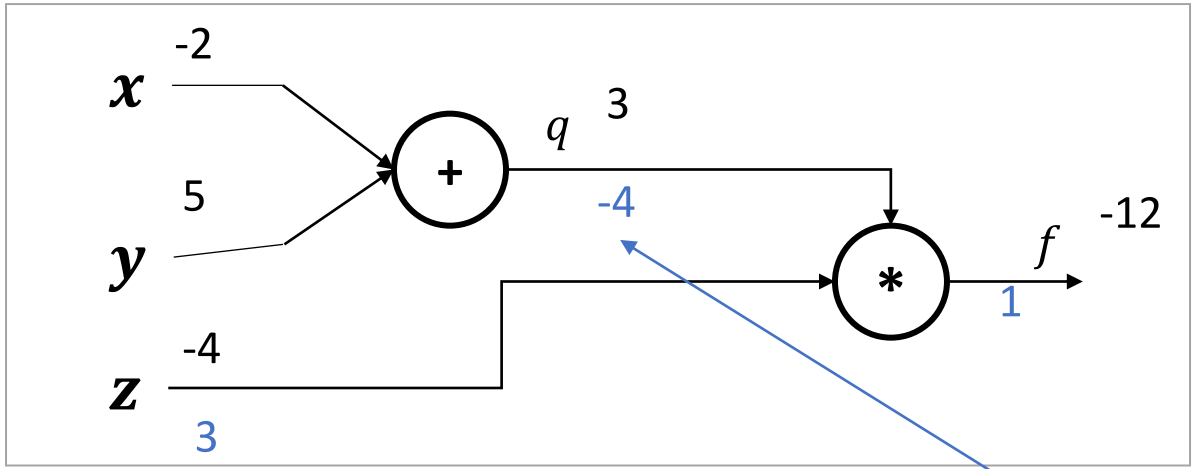
Want: $\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}, \dfrac{\partial f}{\partial z}$

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

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

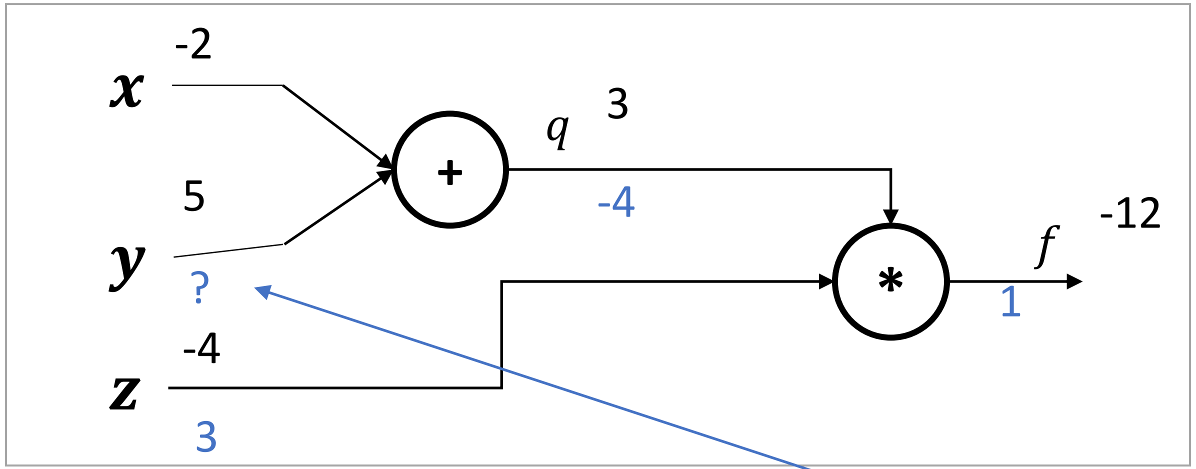
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

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

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

Chain Rule:

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

Upstream gradient

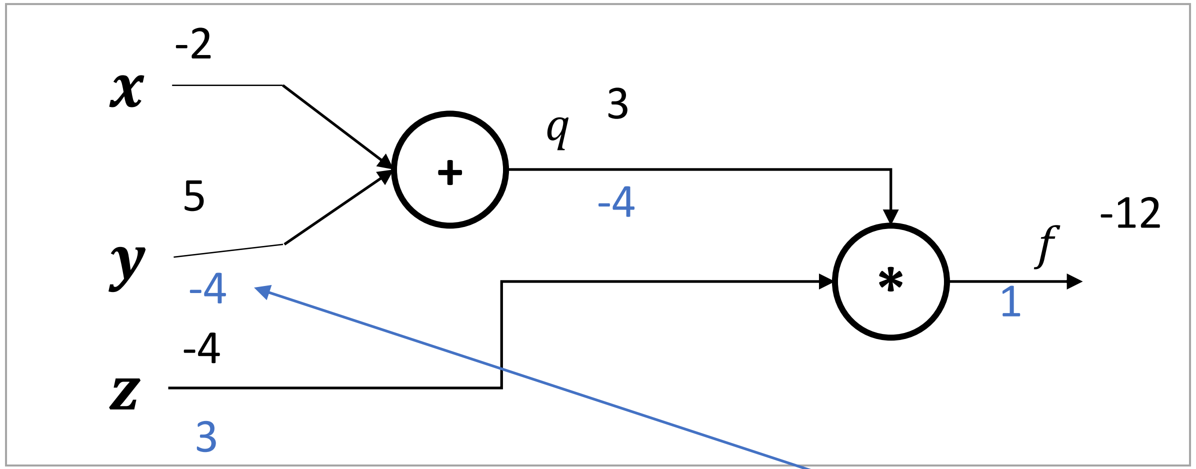
Local gradient

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

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

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

Chain Rule:

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

Upstream gradient

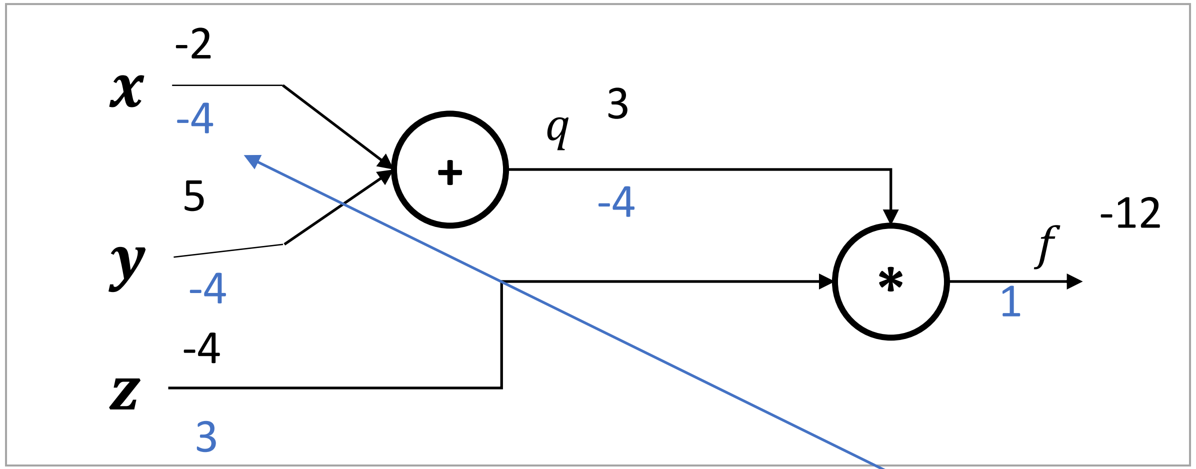
Local gradient

Training an MLP (Backpropagation)

Computational Graphs: A simple example

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

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



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

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

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

Chain Rule:

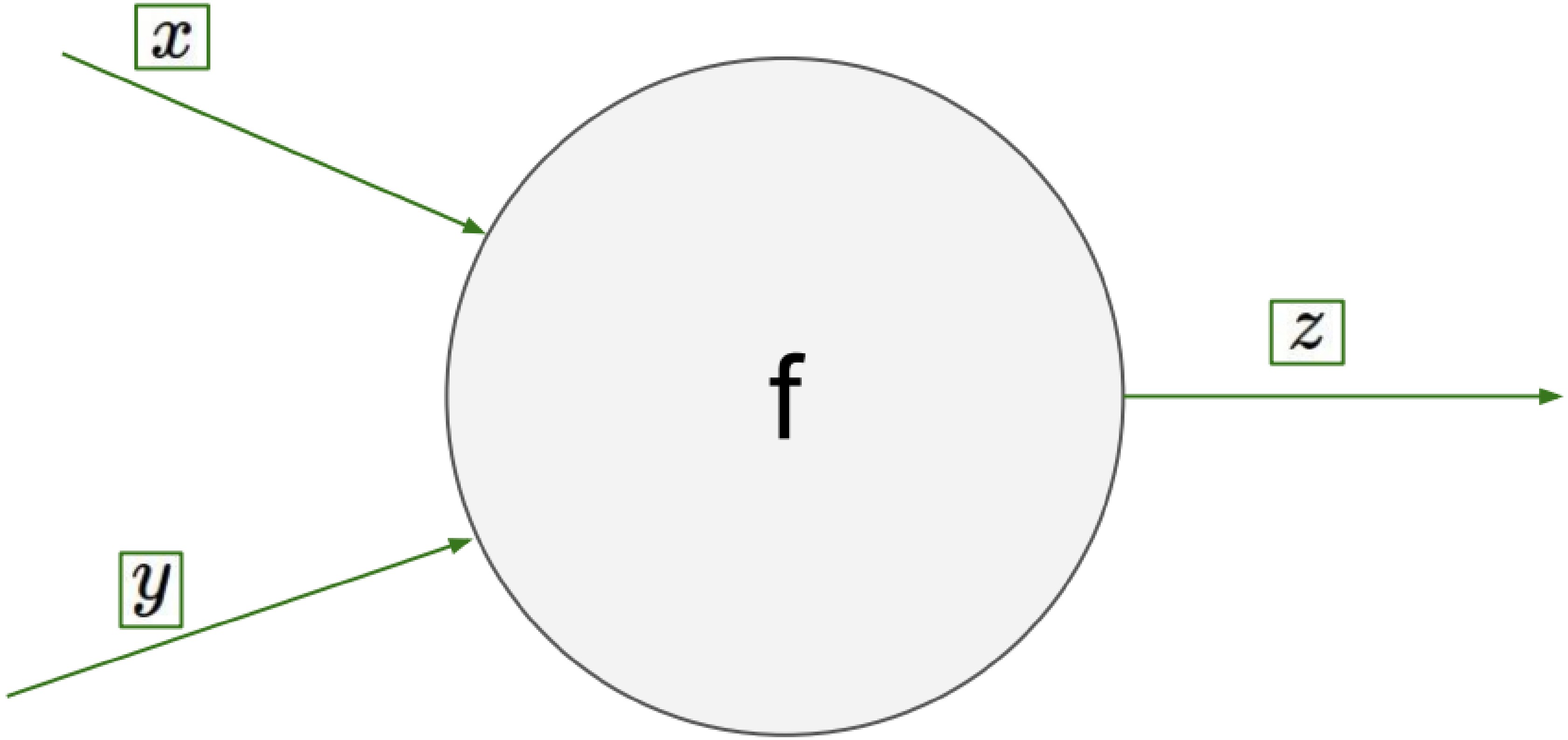
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

Upstream gradient

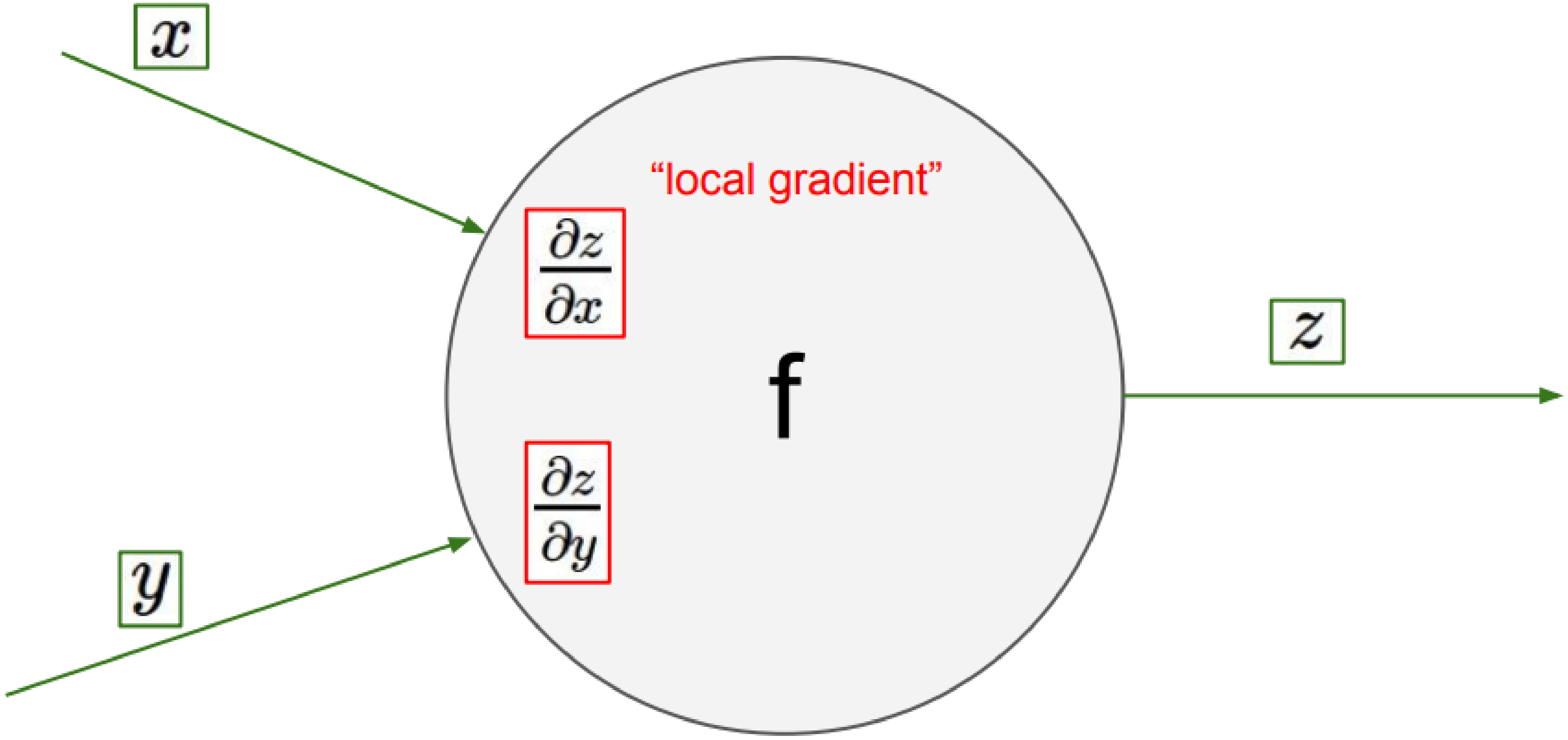
Local gradient

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

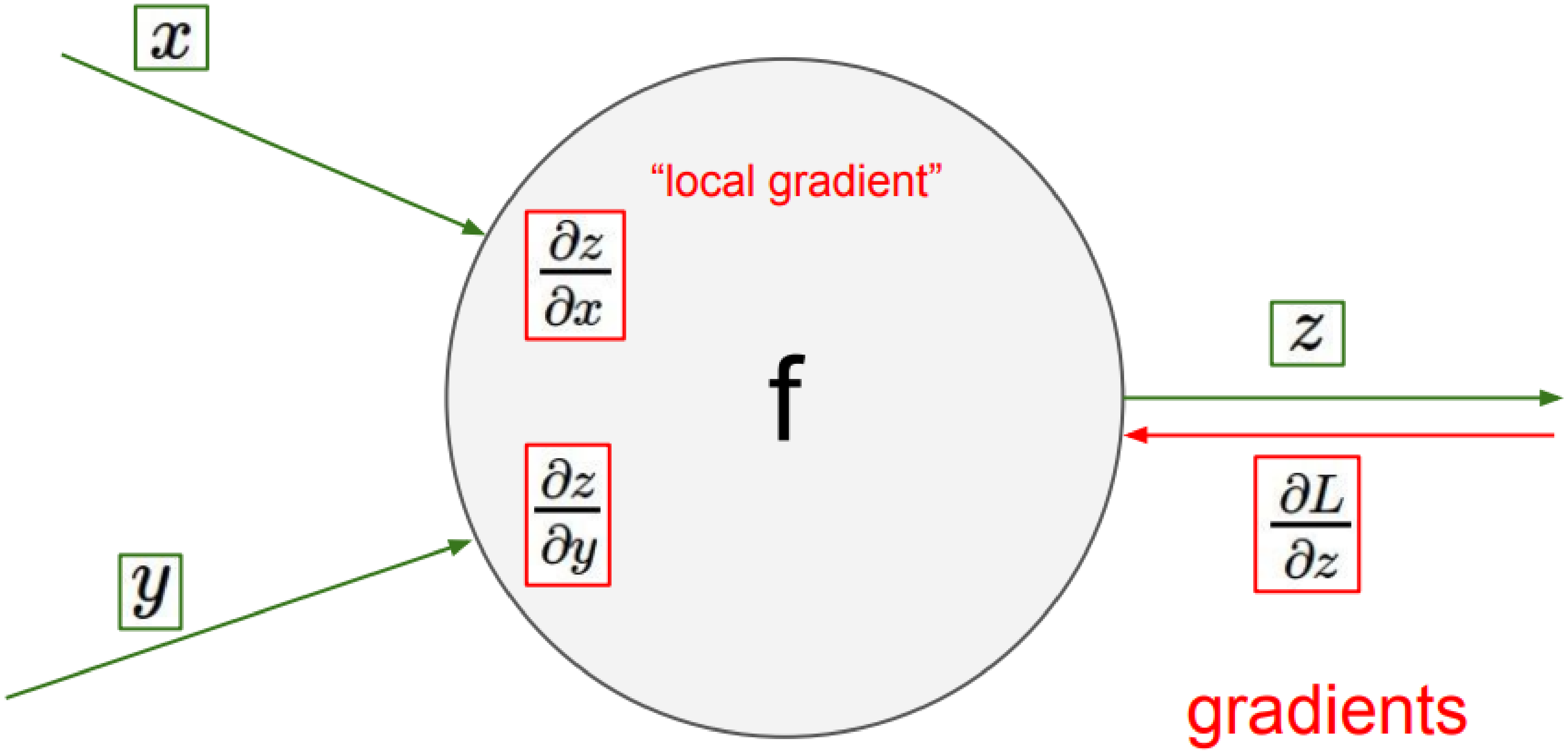
Training an MLP (Backpropagation)



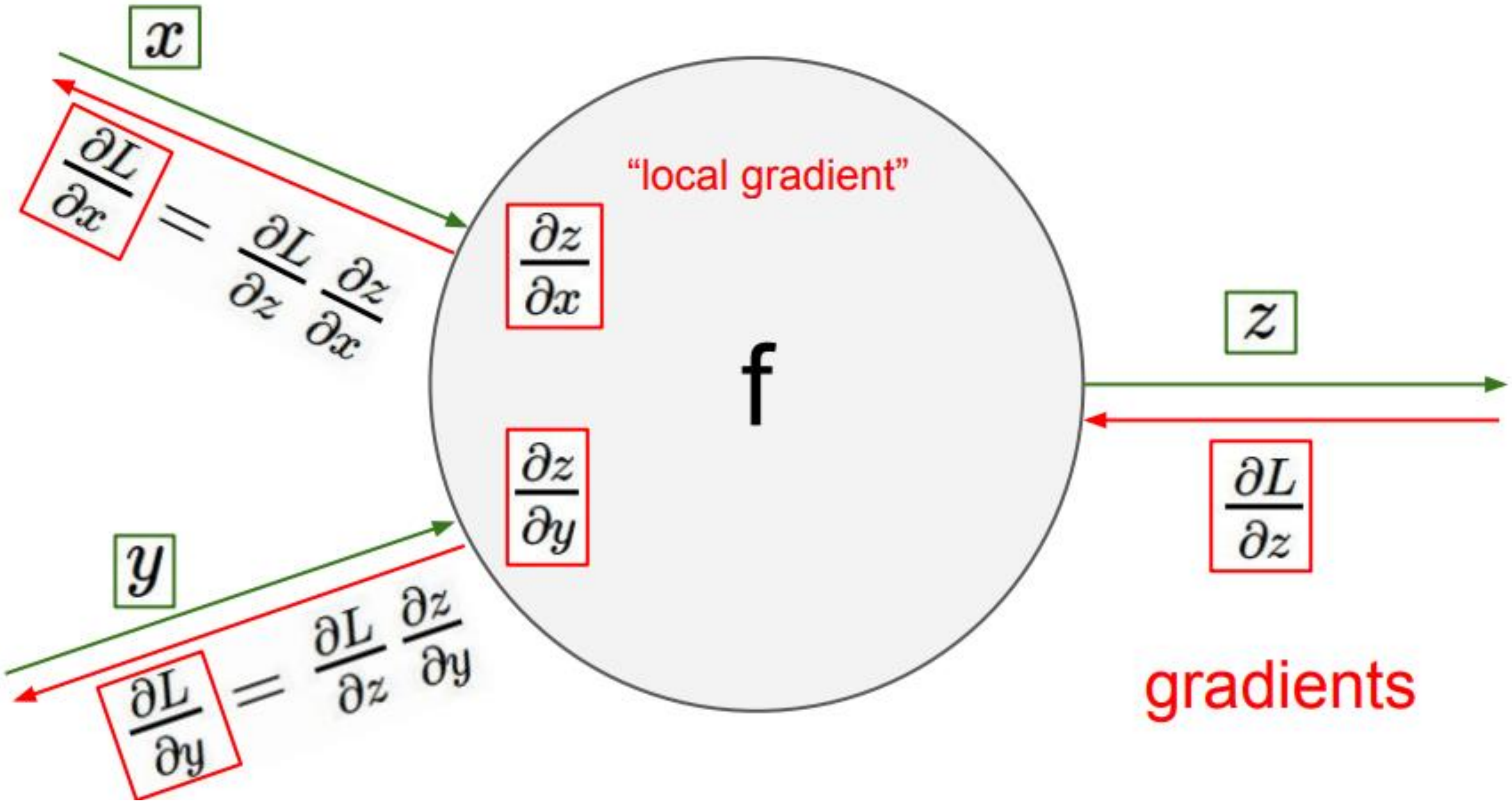
Training an MLP (Backpropagation)



Training an MLP (Backpropagation)



Training an MLP (Backpropagation)



Training an MLP (Backpropagation)