# Deep learning for image analysis quick introduction

E. Decencière

MINES ParisTech
PSL Research University
Center for Mathematical Morphology

# Contents

# Contents

# A revolution in image analysis

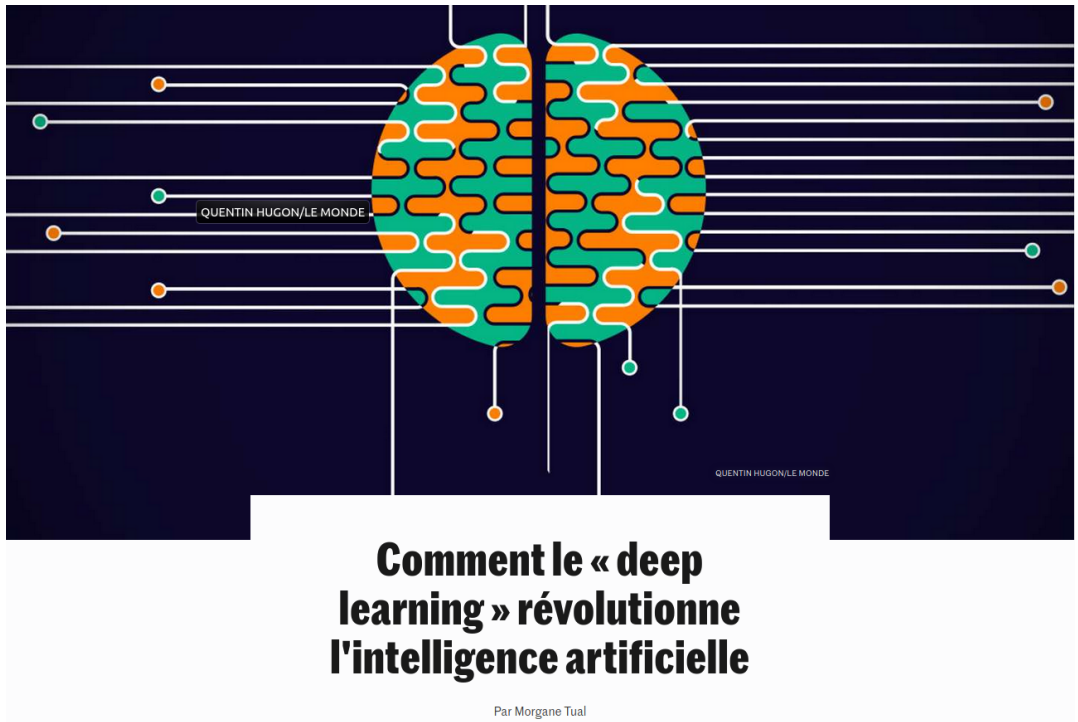



Figure: Le Monde, juillet 2015

# A revolution in image analysis


Figure: Nature, 2016

# A revolution in image analysis

## Le prix Turing récompense trois pionniers de l'intelligence artificielle (IA)

**L'association américaine ACM a remis son prestigieux prix aux chercheurs français, canadien et britannique : Yann LeCun, Yoshua Bengio et Geoffrey Hinton.**

Par David Larousserie • Publié le 27 mars 2019 à 11h01 - Mis à jour le 29 mars 2019 à 12h11

Figure: Le Monde, mars 2019

# Contents

# Differentiable programming

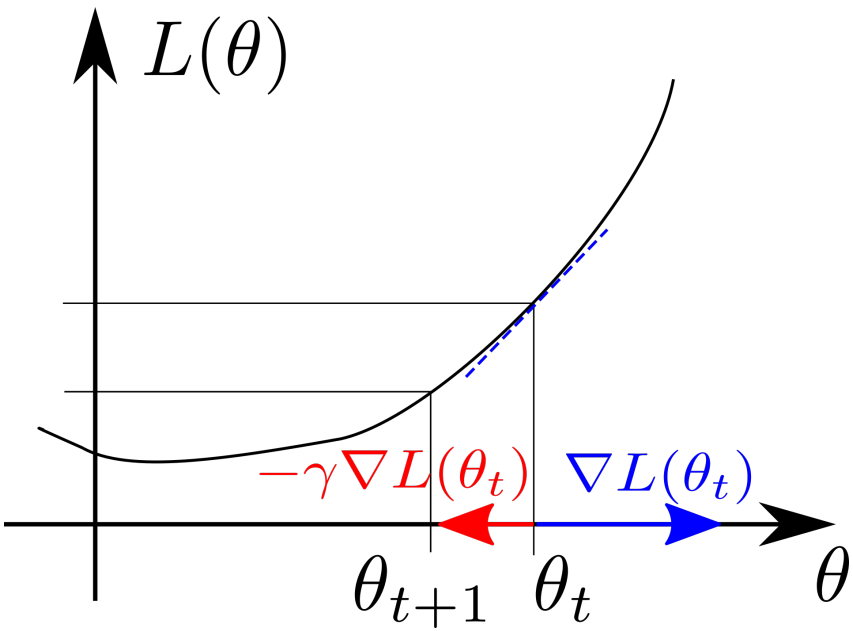
## Definition

Differentiable programming is an algorithmic framework where differentiable operators are combined together to build complex systems. The parameters of the system can then be optimized via gradient descent thanks to back-propagation.

- In our case, computational graphs will be used to compute a loss function $L$, which depends on the input of the system $\mathbf{X}$ and its parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_q\}$.
- Example:

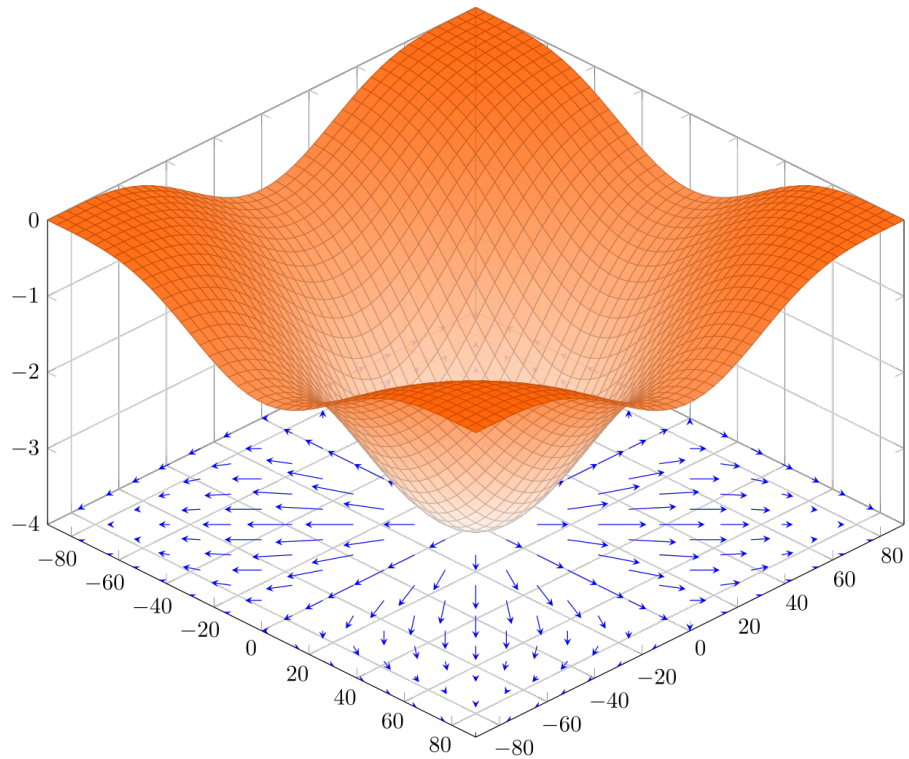
$$L(\mathbf{X}, \boldsymbol{\theta}) = (f_{\boldsymbol{\theta}_1} \circ g_{\boldsymbol{\theta}_2} + h_{\boldsymbol{\theta}_3})(\mathbf{X})$$

# Gradient descent in the scalar case




$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t)$$

# How to minimize a function?




## Definition: gradient

Let $L$ be a derivable function from $\mathbb{R}^n$ into $\mathbb{R}$. Its gradient $\nabla L$ is:

$$\nabla L(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial L}{\partial \boldsymbol{\theta}_1}(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{\theta}_n}(\boldsymbol{\theta}) \end{pmatrix}$$

Credits: By MartinThoma, CC0, https://commons.wikimedia.org/

# Computational graph

## Definition

A computational graph is a direct acyclic graph such that:

- A node is a mathematical operator
- To each edge is associated a value
- Each node can compute the values of its output edges from the values of its input edges
  - Nodes without input edges are *input nodes*. They represent the input values of the graph.
  - Similarly, output values can be held in the *output nodes*.

Computing a *forward pass* through the graph means choosing its input values, and then progressively computing the values of all edges.

# Computational graph example

Computational graph of:

$$\sigma(w_1 x + w_2 y + b)$$

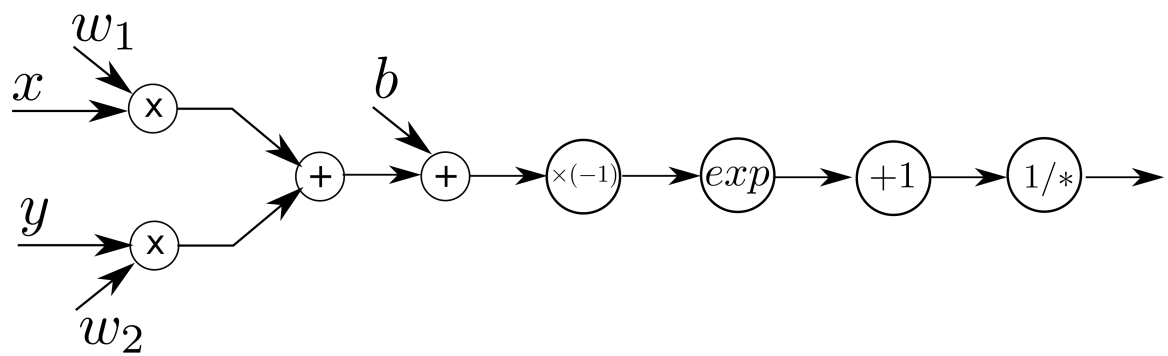
where $\sigma$ is the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$

# Computational graph example

Computational graph of:

$$\sigma(w_1 x + w_2 y + b)$$

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

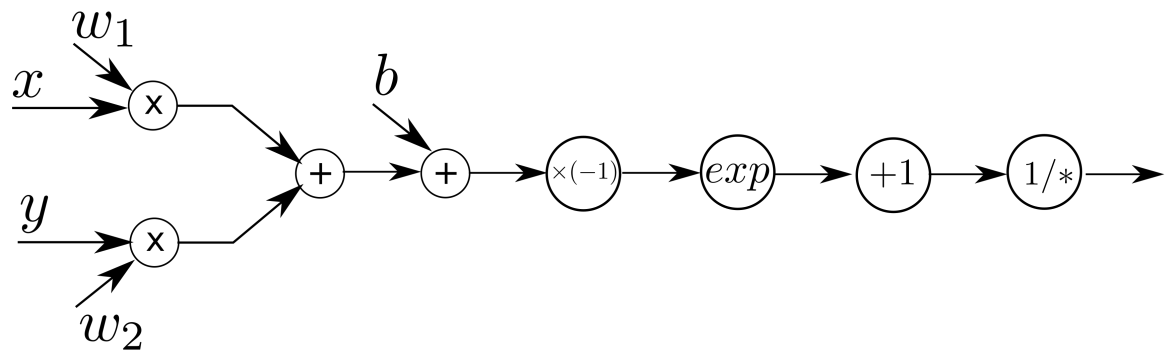


# Computational graph example

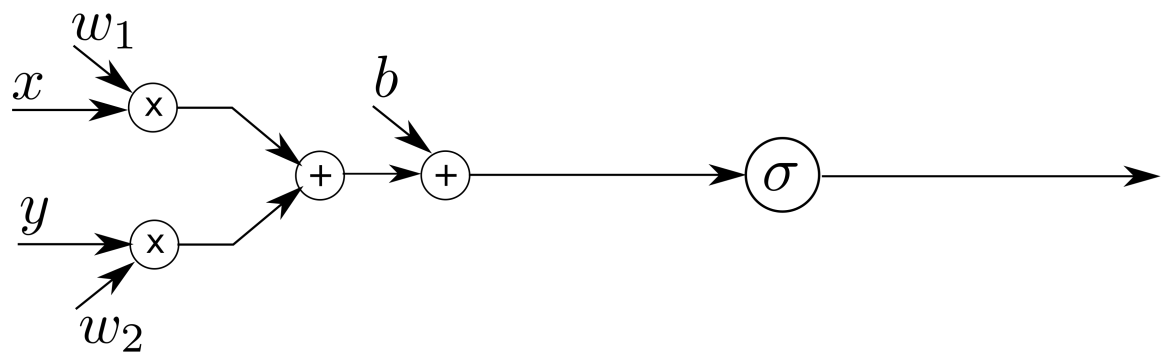
Computational graph of:

$$\sigma(w_1 x + w_2 y + b)$$

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




The graph can be represented at different levels of detail:



# Gradient descent applied to computational graphs

In the case of computational graphs, the loss $L$ depends on each parameter $\boldsymbol{\theta}_i$ via the composition of several simple functions. In order to compute the gradient $\nabla_{\boldsymbol{\theta}} L$ we will make extensive use of the chain rule theorem.

## Chain rule theorem

Let $f_1$ and $f_2$ be two derivable real functions ($\mathbb{R} \to \mathbb{R}$). Then for all $x$ in $\mathbb{R}$: :

$$(f_2 \circ f_1)'(x) = f_2'(f_1(x)).f_1'(x)$$

## Leibniz notation

Let us introduce variables $x$, $y$ and $z$:

$$x \xrightarrow{f_1} y \xrightarrow{f_2} z$$

Then:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}z}{\mathrm{d}y} \cdot \frac{\mathrm{d}y}{\mathrm{d}x}$$

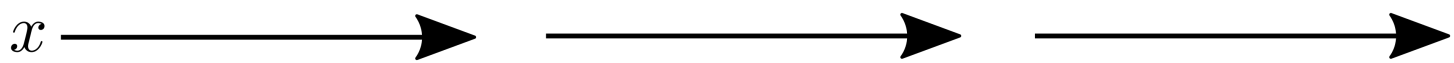
# The backpropagation algorithm

- The backpropagation algorithm is used in a neural network to efficiently compute the partial derivatives of the loss with respect to each parameter of the network.
- One can trace the origins of the method to the sixties
- It was first applied to NN in the eighties
  [Werbos, 1982, LeCun, 1985]

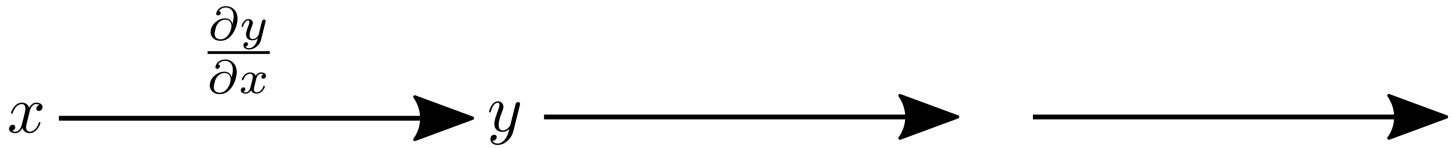
# The backpropagation algorithm: intuition

- Given a computational graph, the main idea is to compute the local partial derivatives during a forward pass
- Then, during a backward pass, the partial derivatives of the loss with respect to each parameter is computed

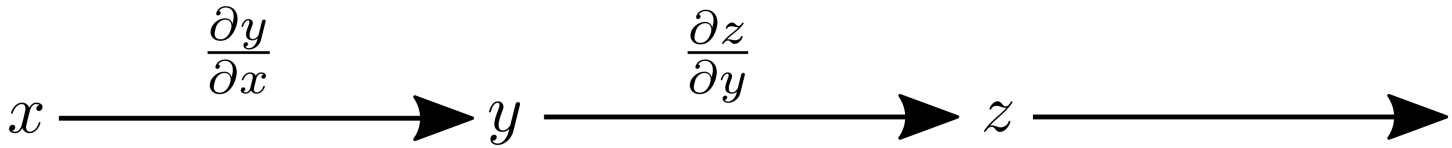
# Simple backpropagation example



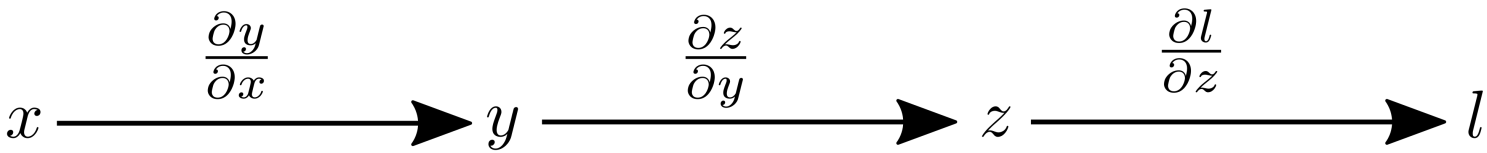
# Simple backpropagation example



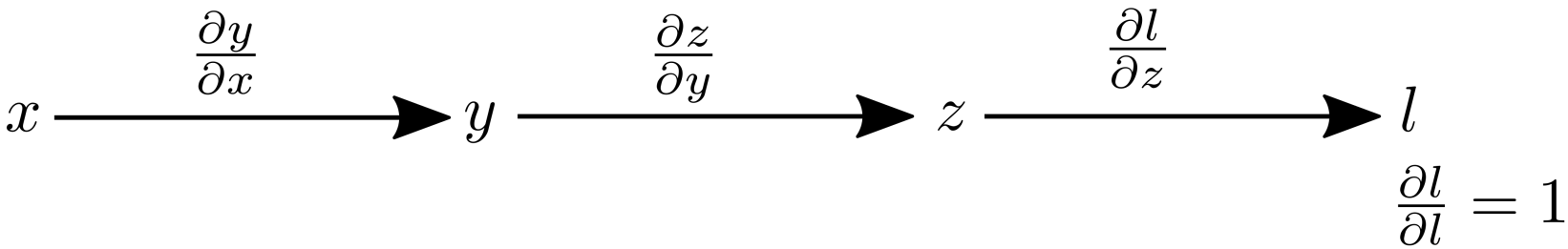
# Simple backpropagation example



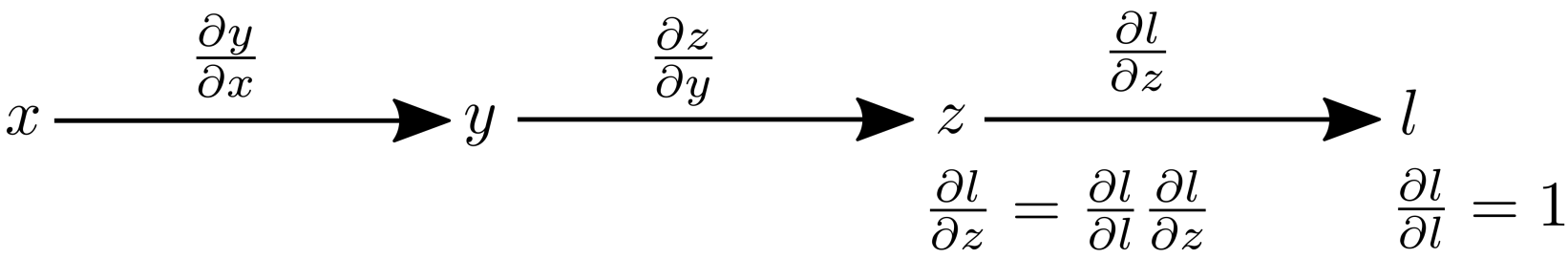
# Simple backpropagation example



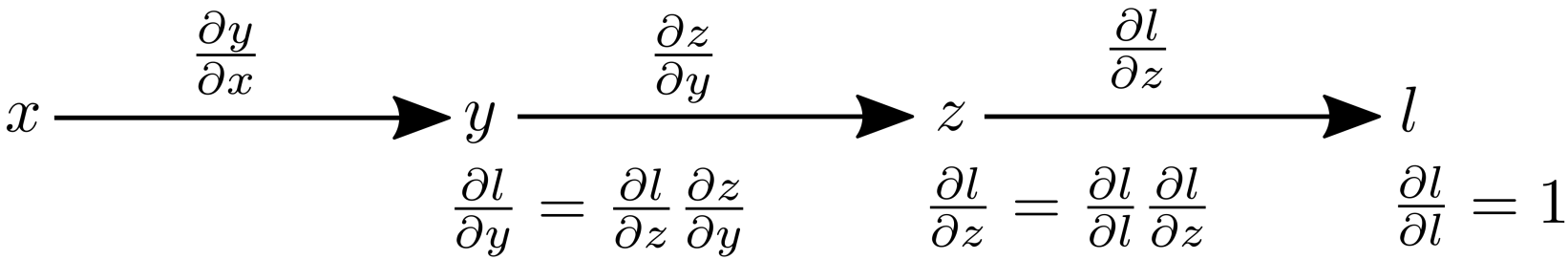
# Simple backpropagation example



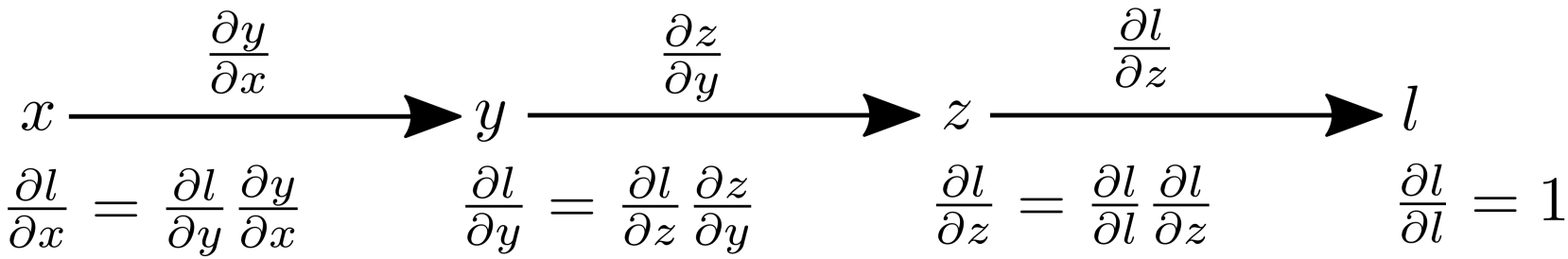
# Simple backpropagation example



# Simple backpropagation example



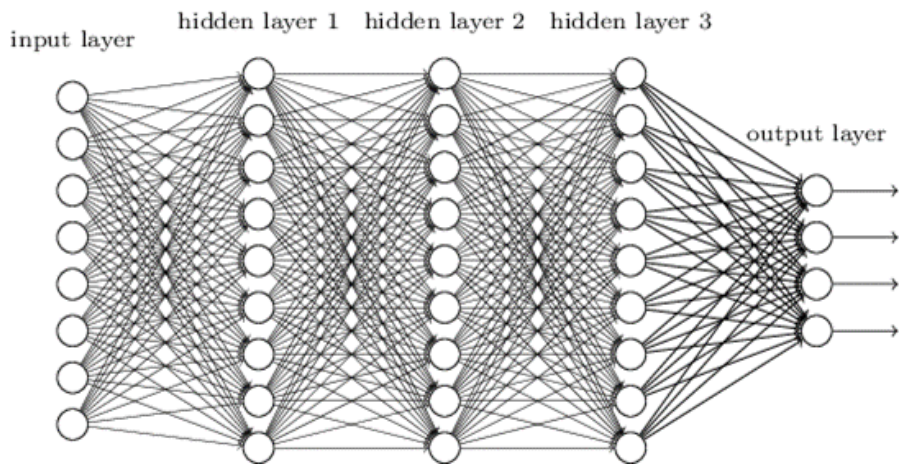
# Simple backpropagation example



# Feed-forward neural networks

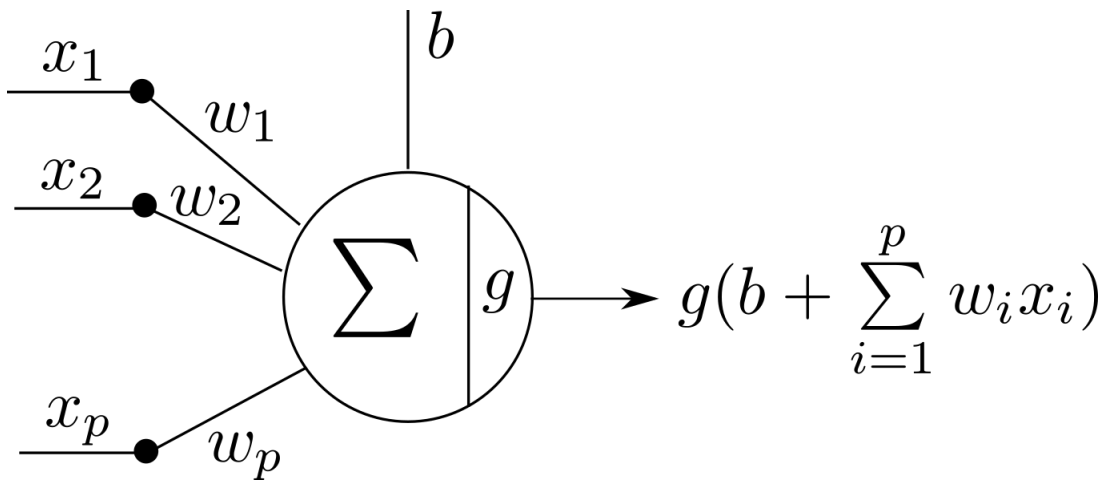
## Definition

- A feed-forward neural networks is a computational graph without cycles
- Its computing units, the neurons, are organized in layers



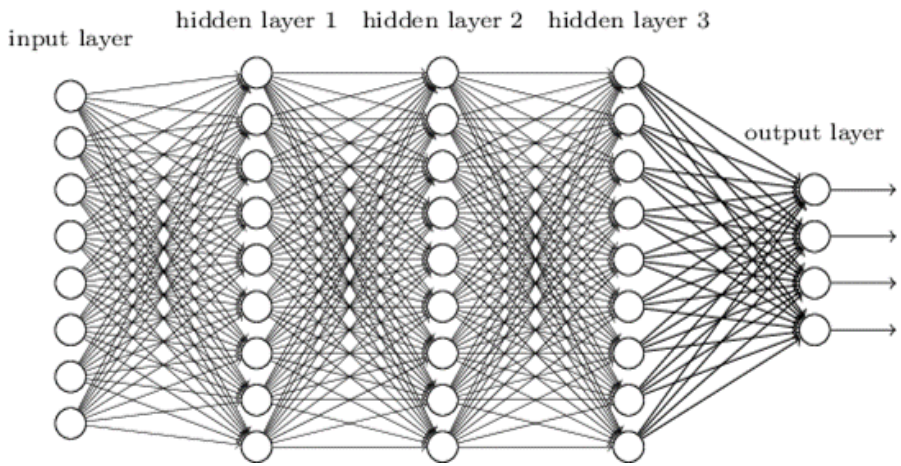

Credits: (from http://www.jtoy.net)

# Artificial neuron

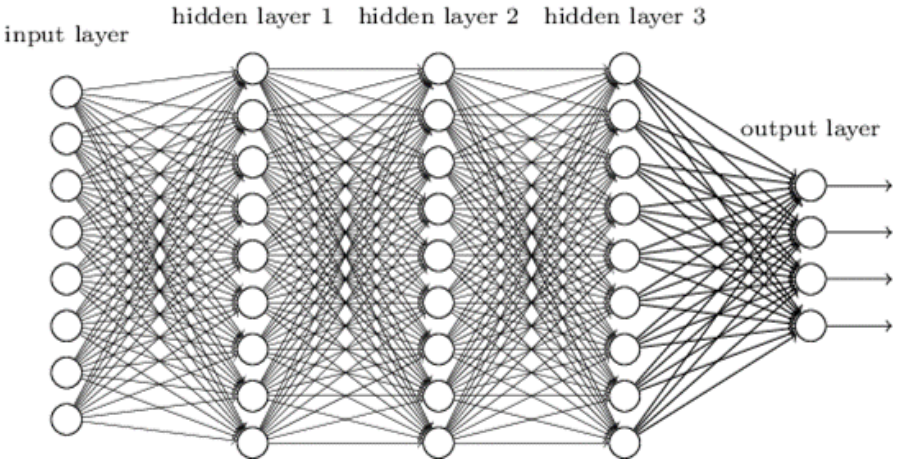


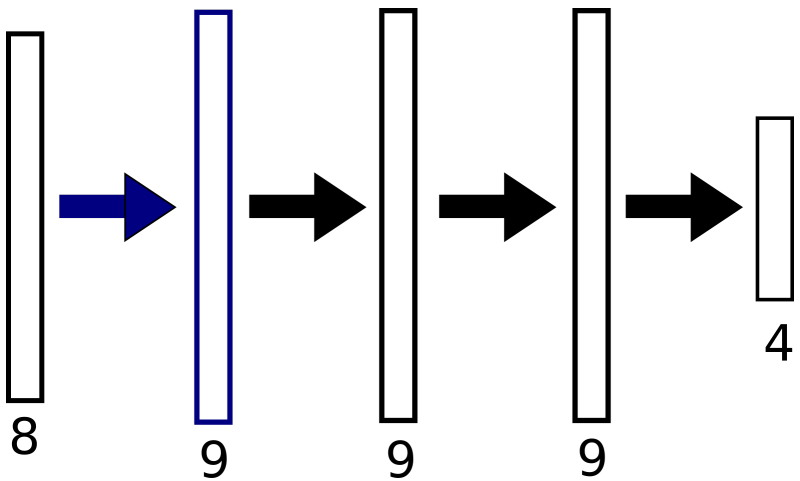
# Fully-connected layer

- A layer is said to be fully-connected (FC) if each of its neurons is connected to all the neurons of the previous layer
- If a FC layer contains $r$ neurons, and the previous layer $q$, then its weights are a 2D dimensional array (a matrix) of size $q \times r$



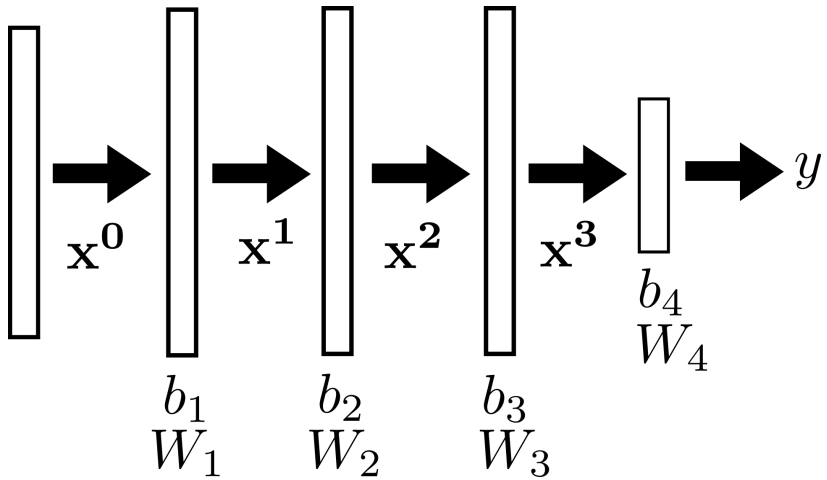
# Graphical representation of NNs




- Data is organized into arrays, linked with operators
- A layer corresponds to an operator between arrays.

# The equations of fully-connected layers




$$\mathbf{x}^i = \mathbf{g}_i(\mathbf{W}_i\mathbf{x}^{i-1} + \mathbf{b}_i),\ i = 1, 2, 3$$

$$y = \mathbf{g}_4(\mathbf{W}_4\mathbf{x}^4 + \mathbf{b}_4)$$

# References I

[LeCun, 1985] LeCun, Y. (1985). Une procedure d'apprentissage pour reseau a seuil asymmetrique (A learning scheme for asymmetric threshold networks). In *proceedings of Cognitiva 85*.

[Werbos, 1982] Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In *SpringerLink*, pages 762–770. Springer Berlin Heidelberg.