CNN : Cool Neural Networks

« La découverte du feu pendant un hiver froid»

Computer Vision

➢ Transfert du style:

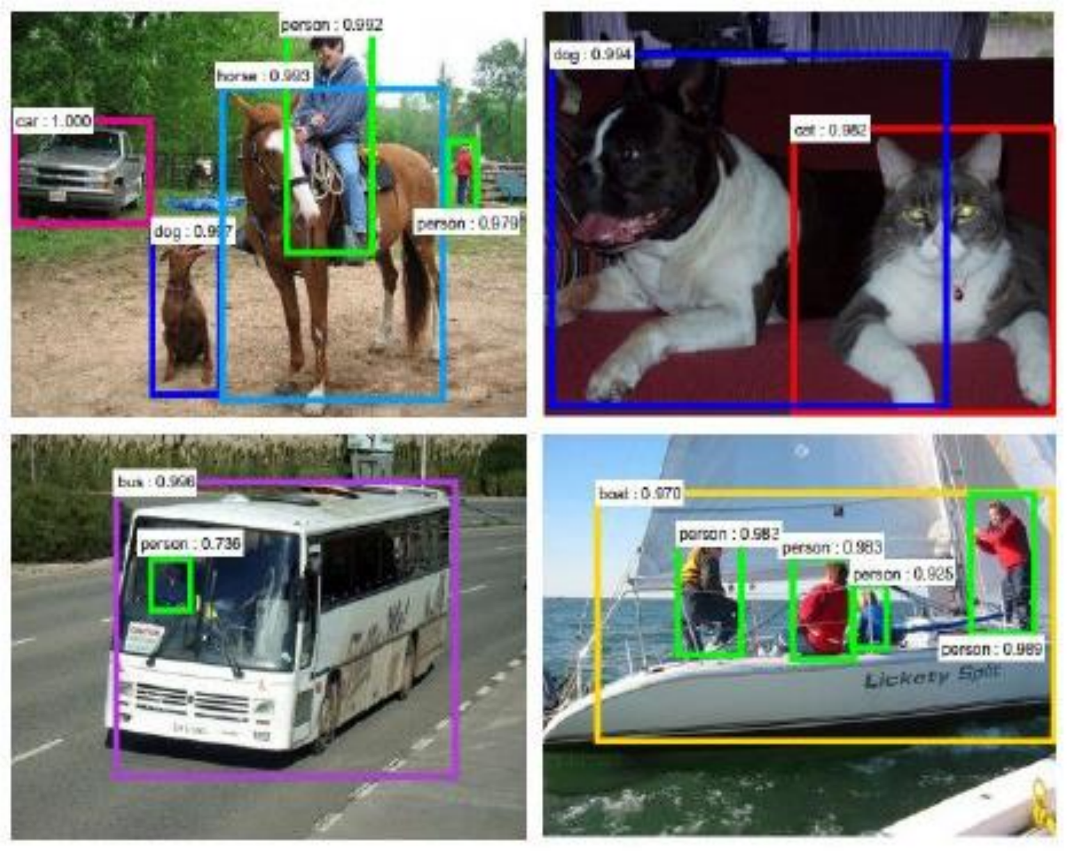
➢ Classification des images:

64x64

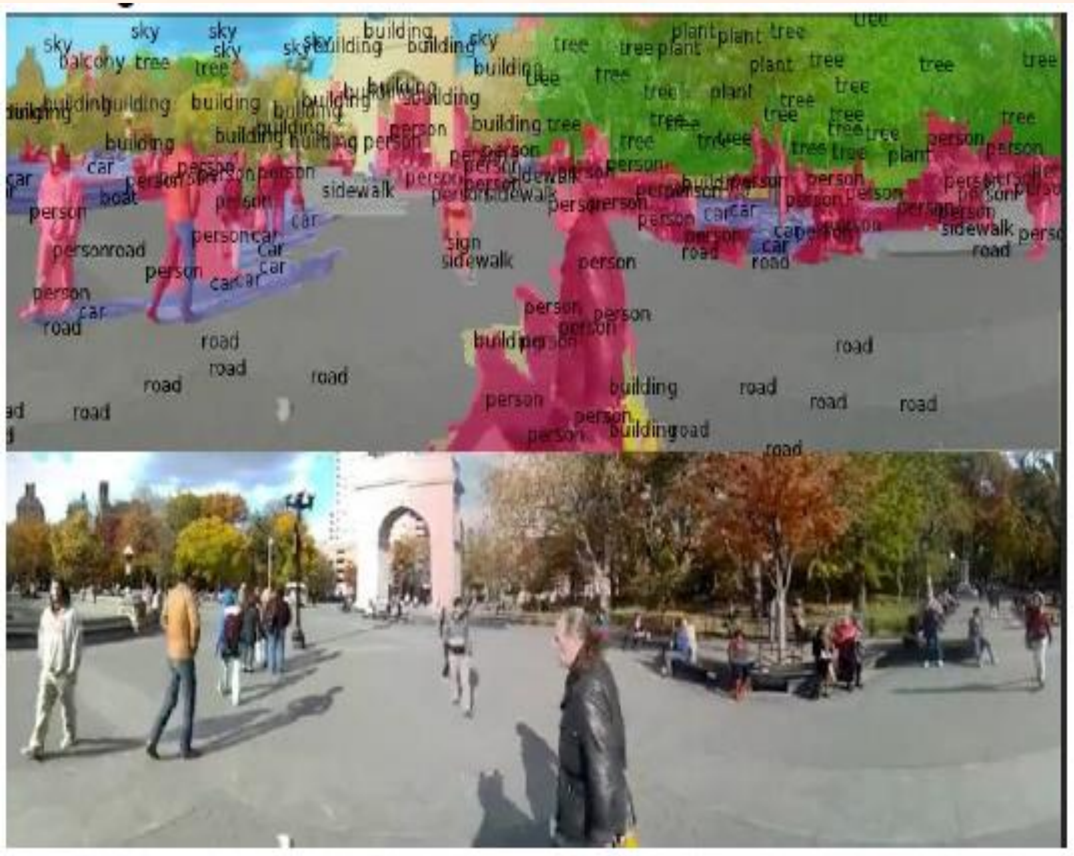
→ Cat? (0/1)

Encore plus de Computer Vision

➢ Détection d'objets:



➢ Segmentation sémantique:



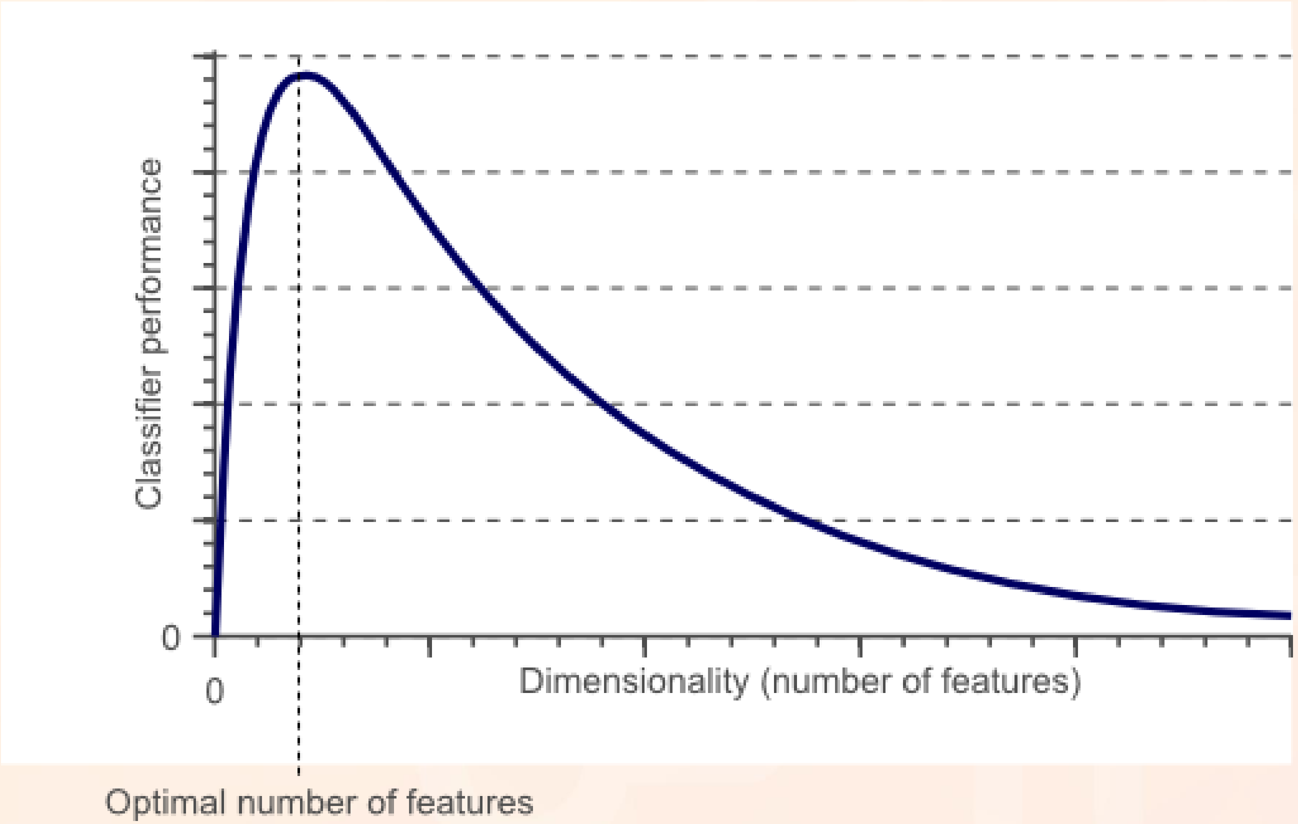
Encore plus…

- Estimation de posture

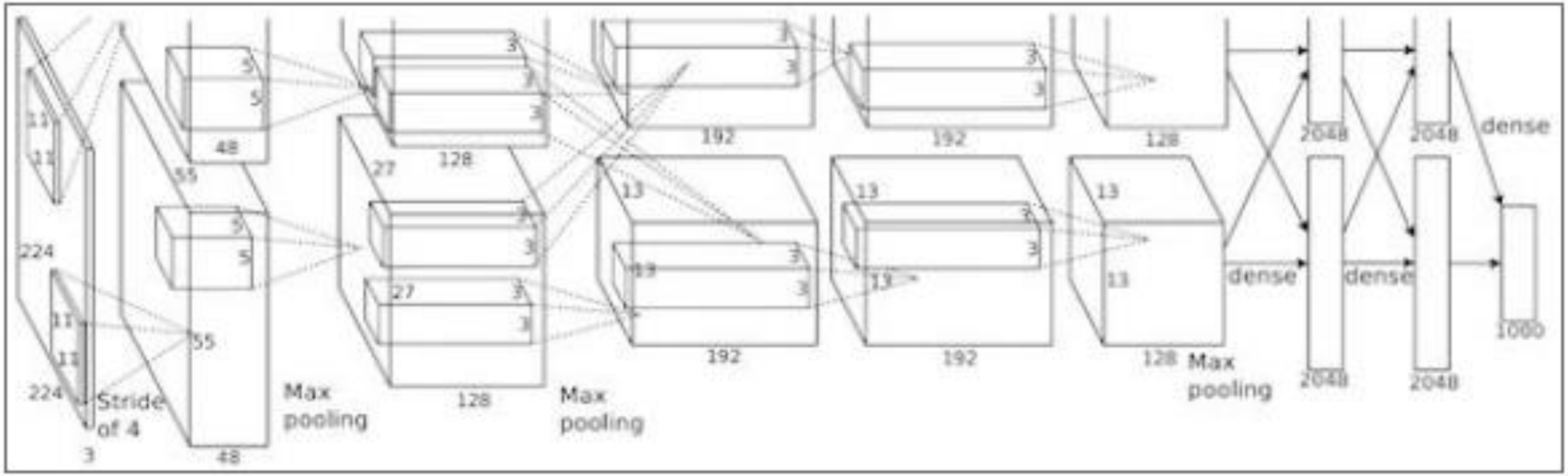
- Apprentissage par renforcement

Le problème avec les MLP

1000 x 1000 x 3

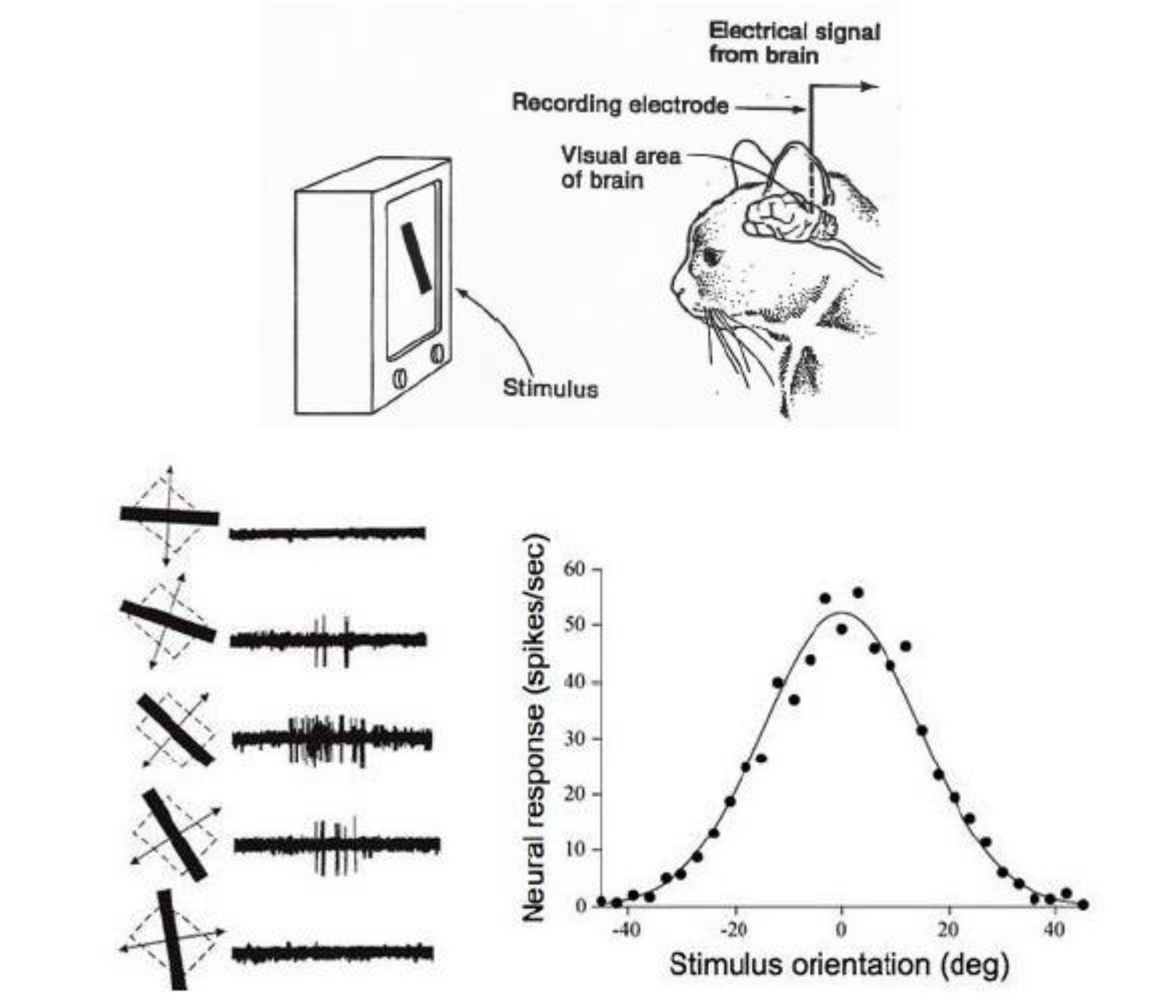


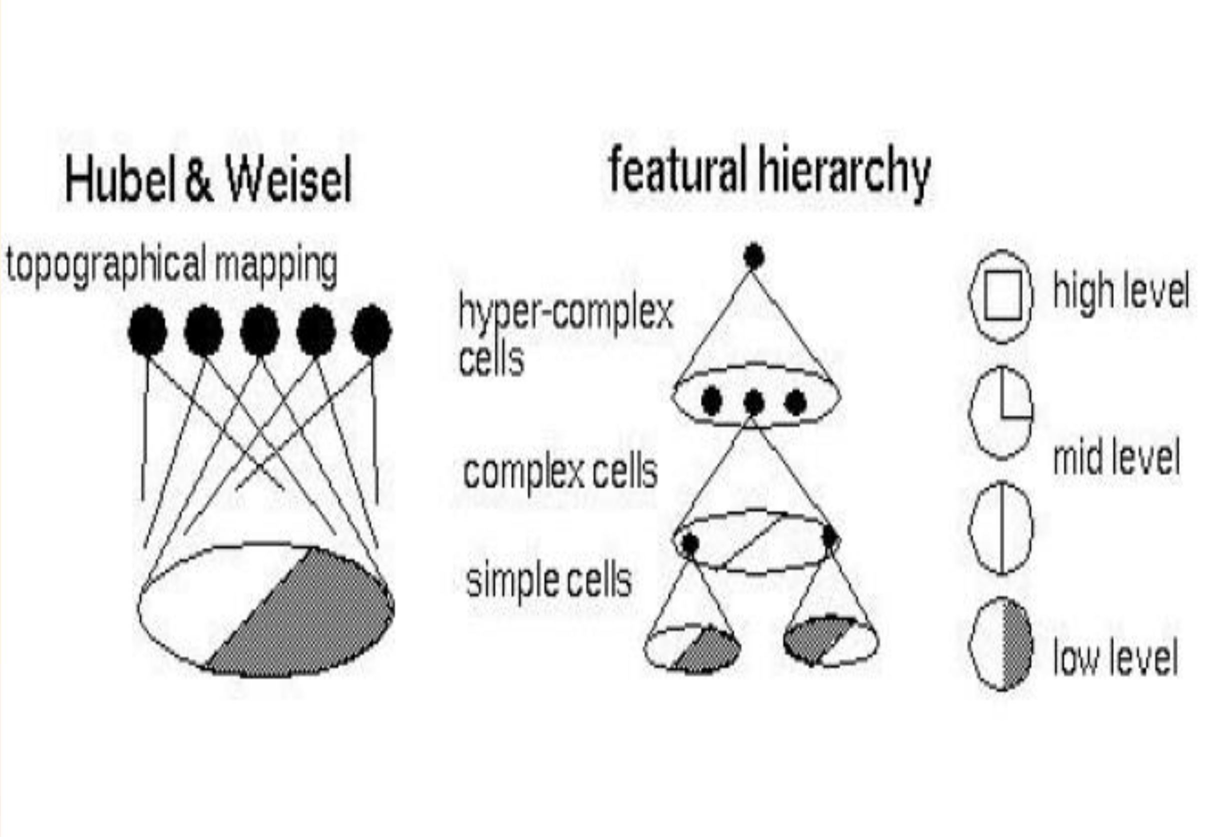
Le premier CNN: LeNet



"AlexNet"

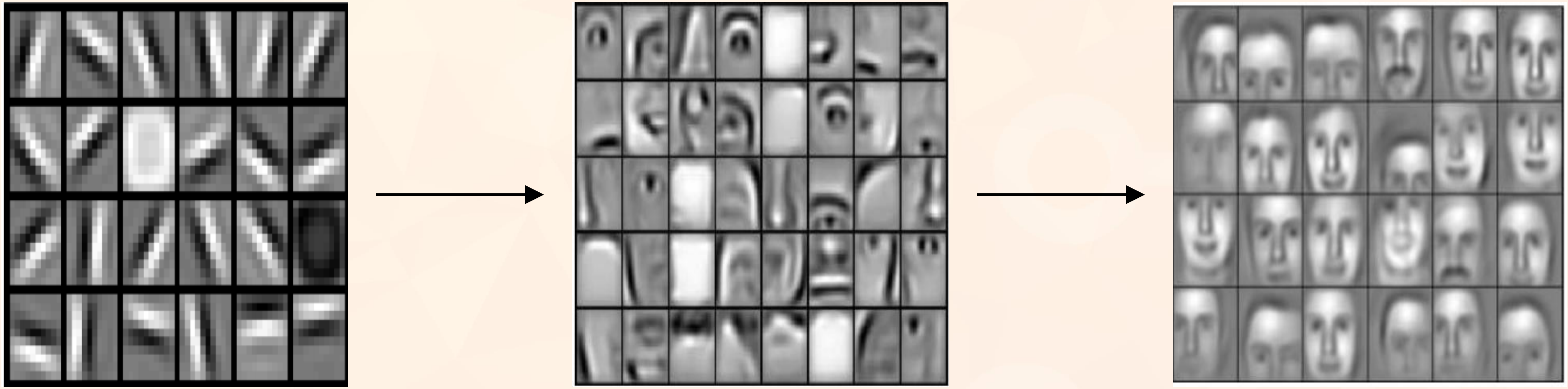
Tout a commencé avec l'histoire d'un chat



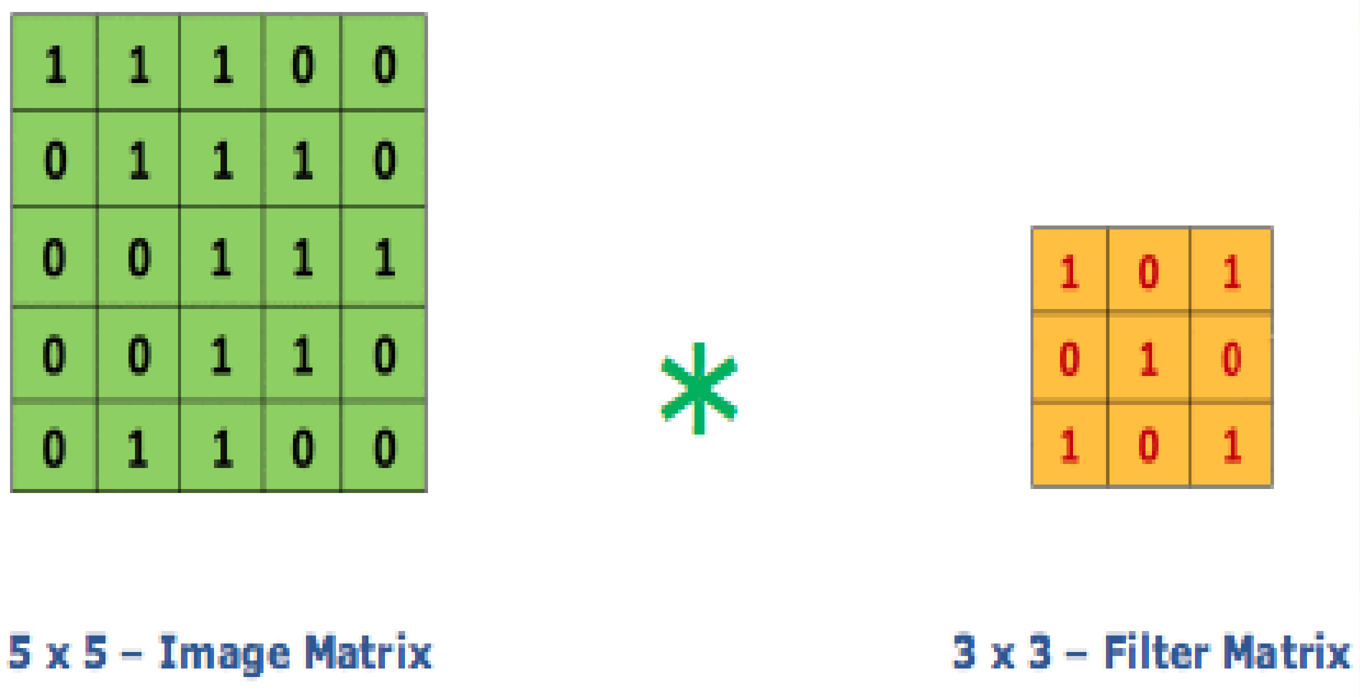


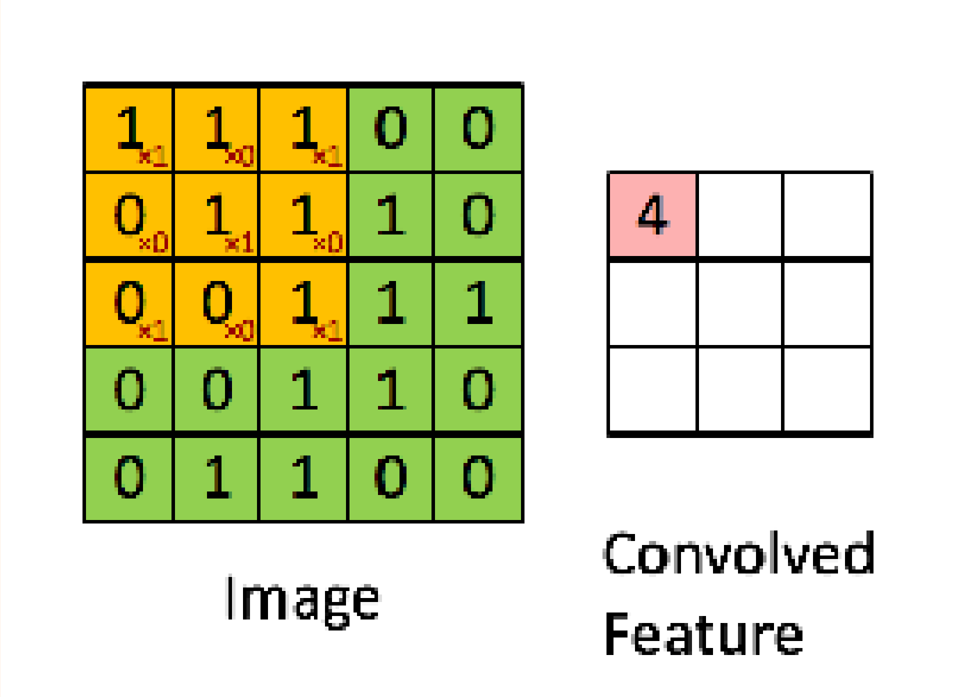
https://www.youtube.com/watch?v=8VdFf3egwfg&feature=youtu.be&t=1m10s&ab_channel=AliMoeeny

Hypothèse d'Hiérarchie des caractéristiques



Convolution





Filtres

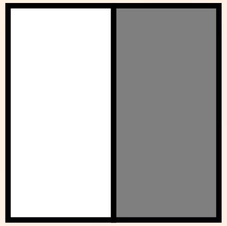
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

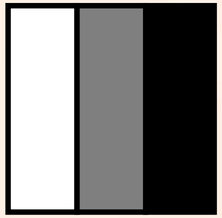
w11	**w12**	**w13**
w21	**w22**	**w23**
w31	**w32**	**w33**

=

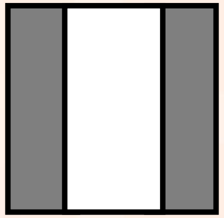
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



*

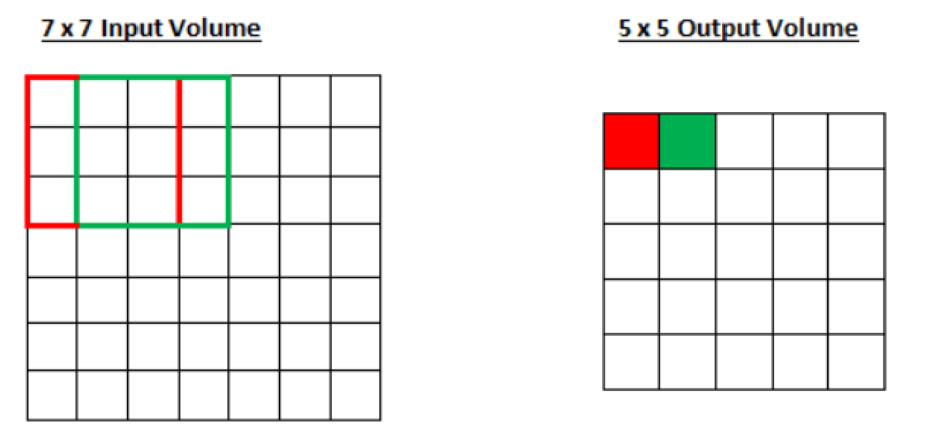


=

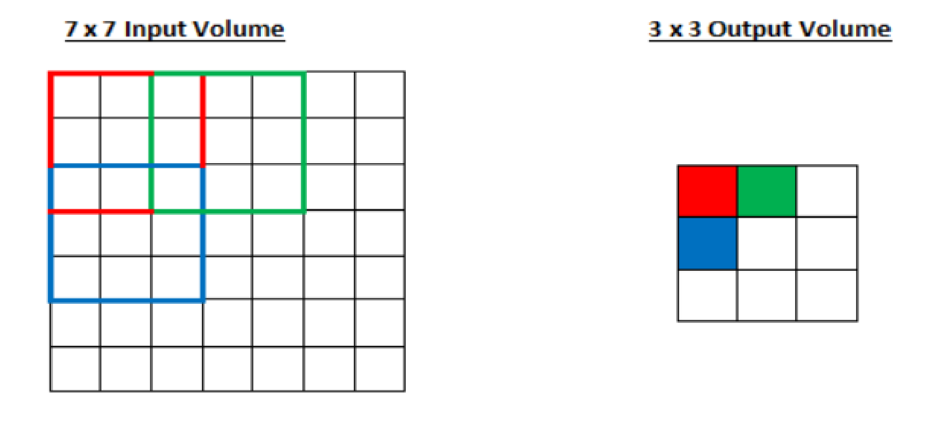


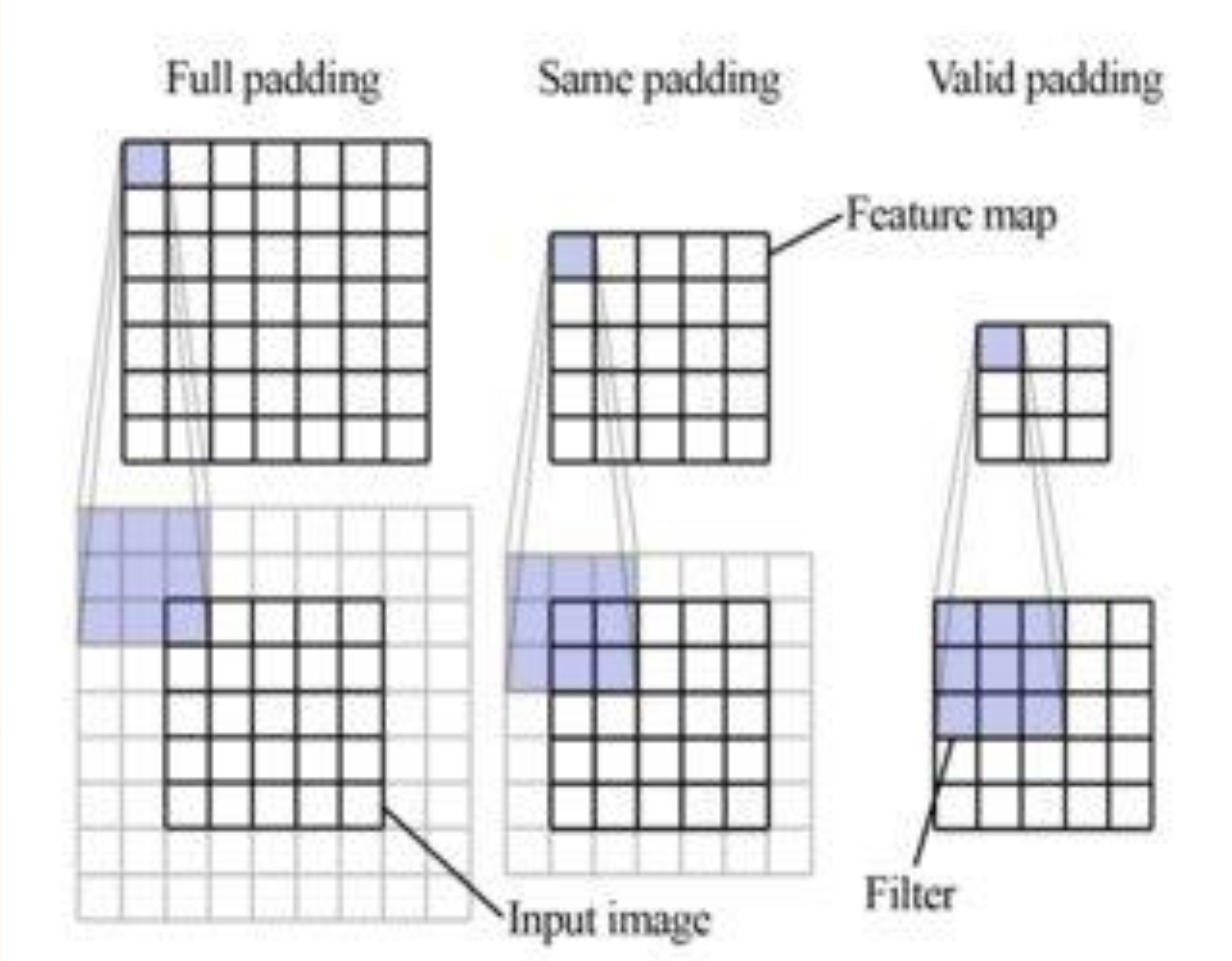
Taille de noyau, Stride et Padding

Stride de (1, 1) = 1



Stride de (2, 2) = 2





Exemple

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

Image 7x7

$*$

Stride (2, 2)

Padding 'VALID'

3	4	4
1	0	2
-1	0	3

Noyau 3x3

$=$

91	**100**	**83**
69	**91**	**127**
44	**72**	**74**

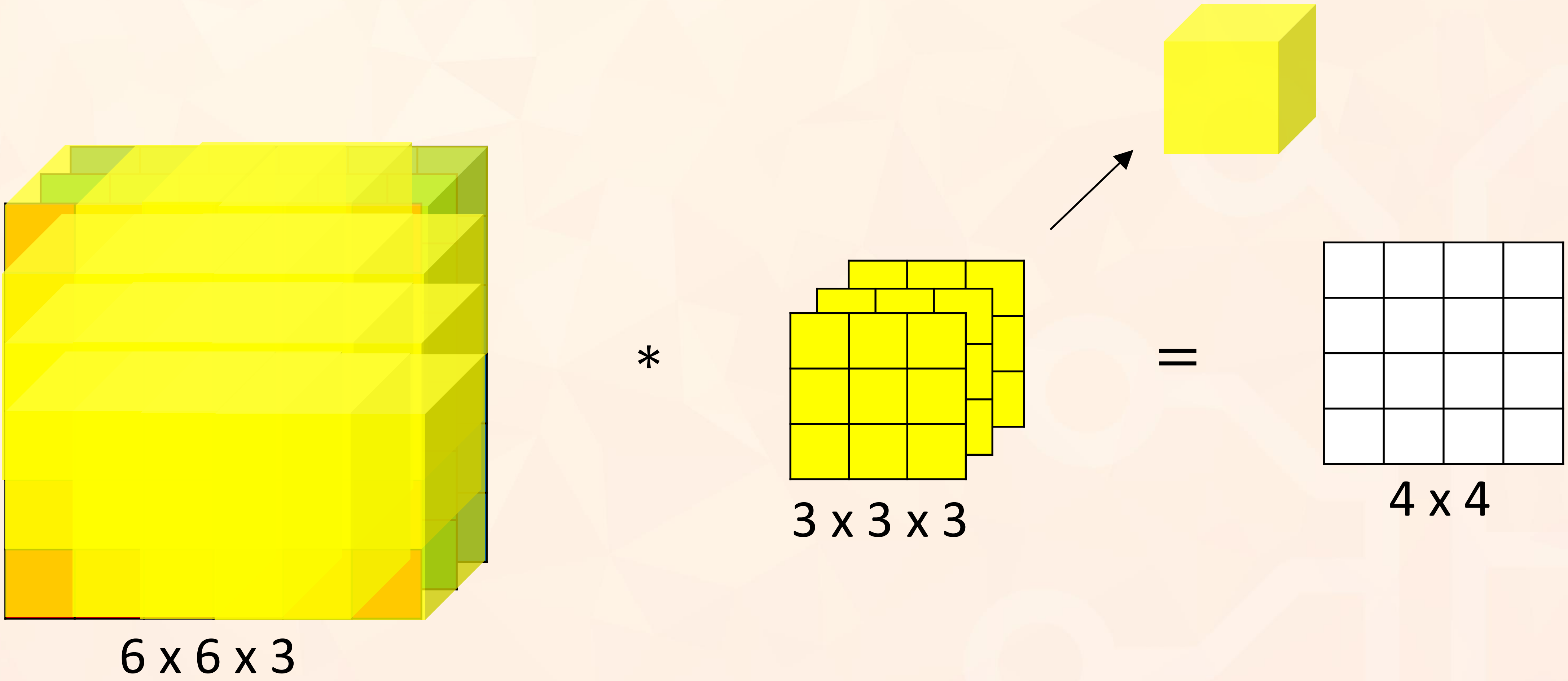
Feature map 3x3

$n \times n$ image $f \times f$ filter

padding p stride s

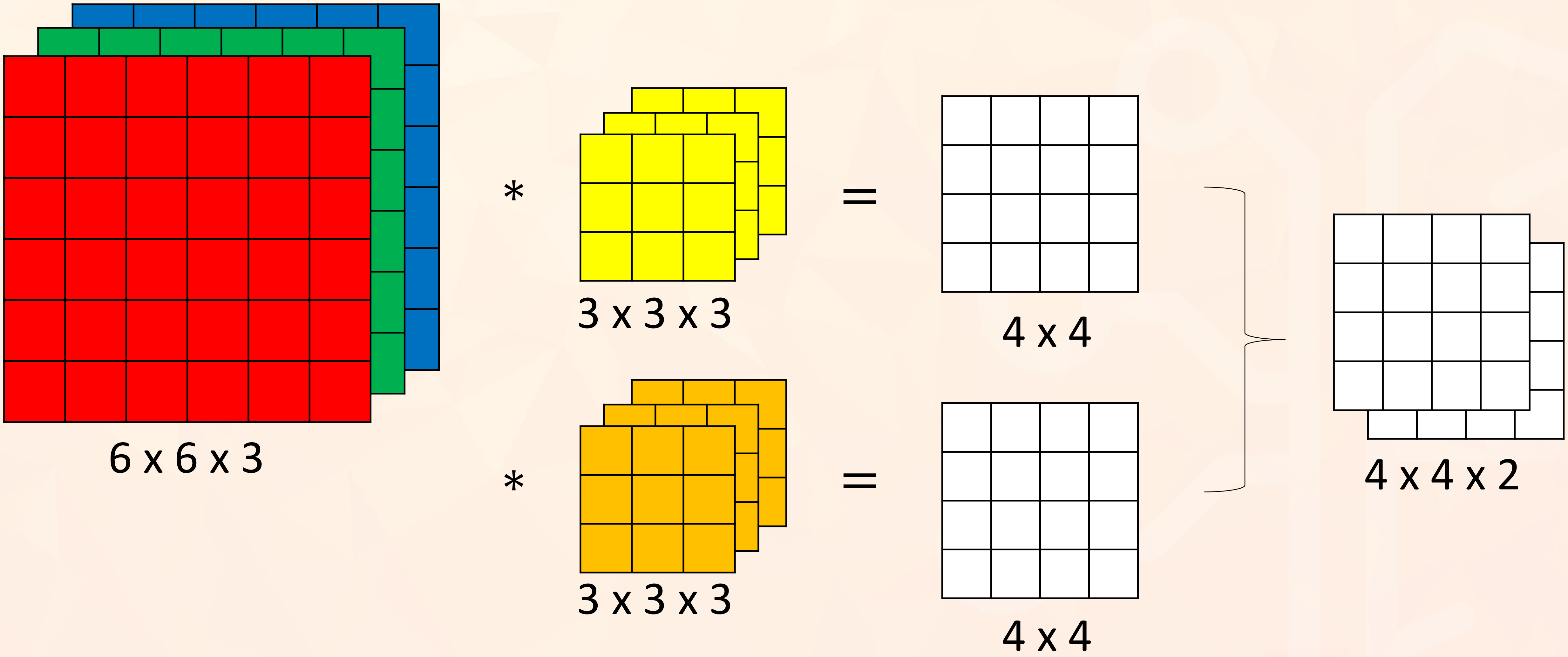
$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

Généralisation pour des volumes

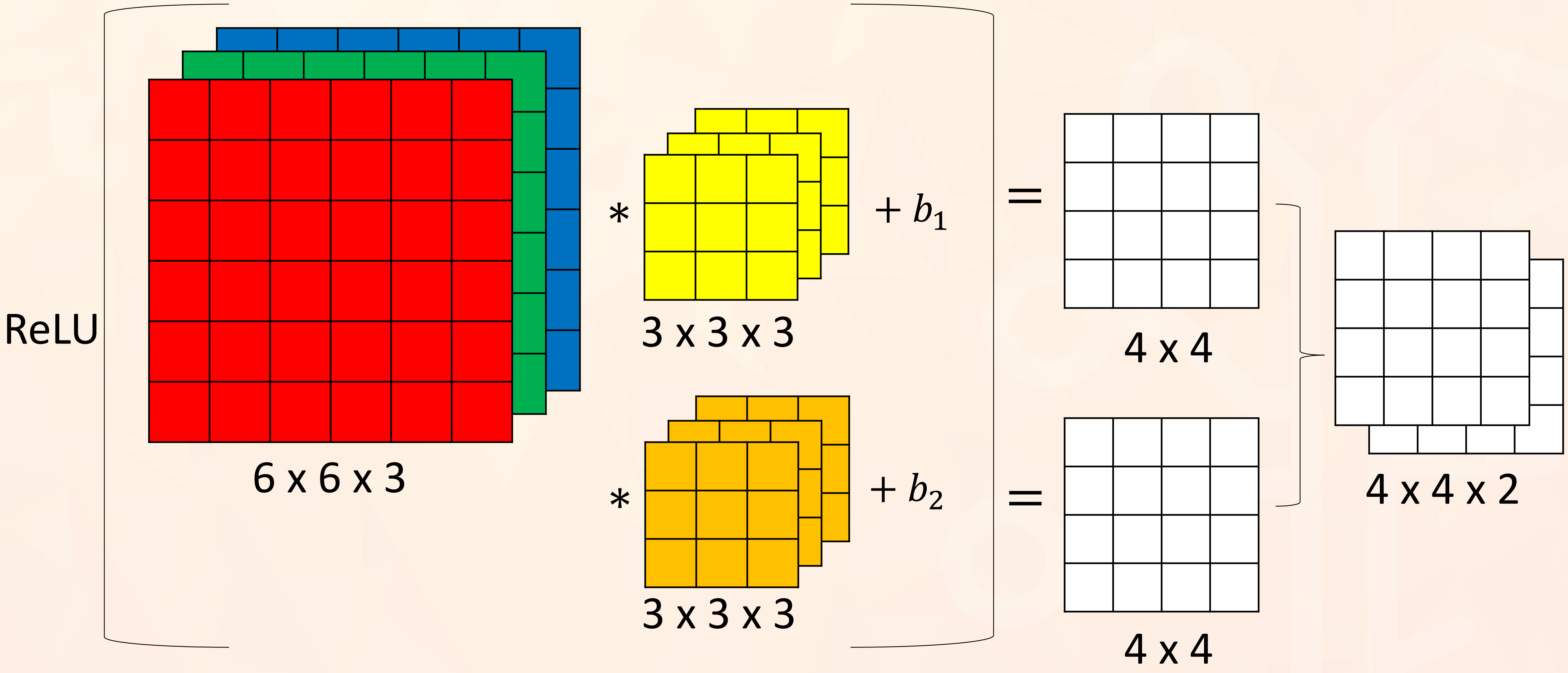


Important : Un noyau retourne toujours une seule ‘feature map’

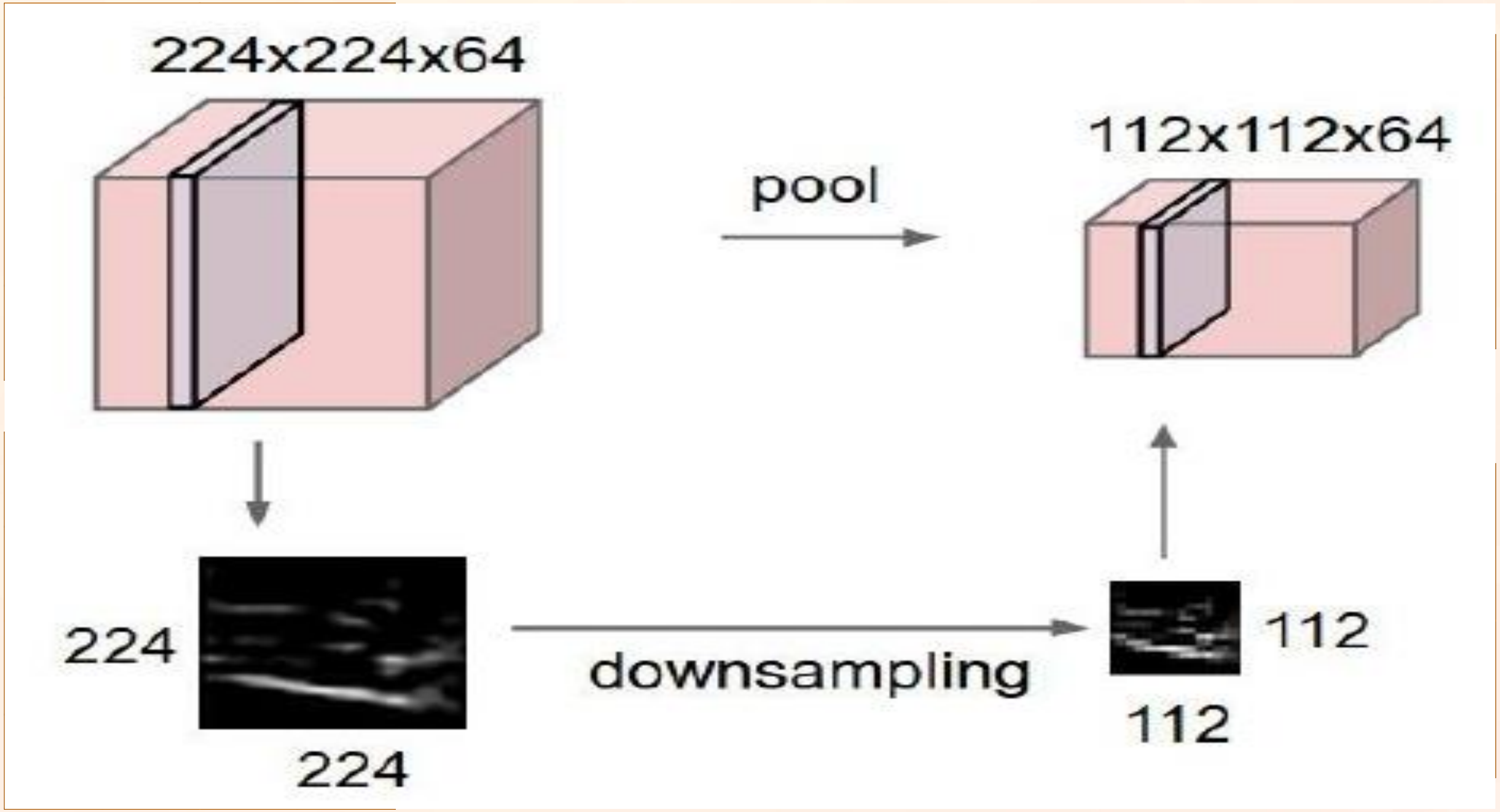
Plusieurs filtres



En fait…



Pooling: Réduction des dimensions



Exemple

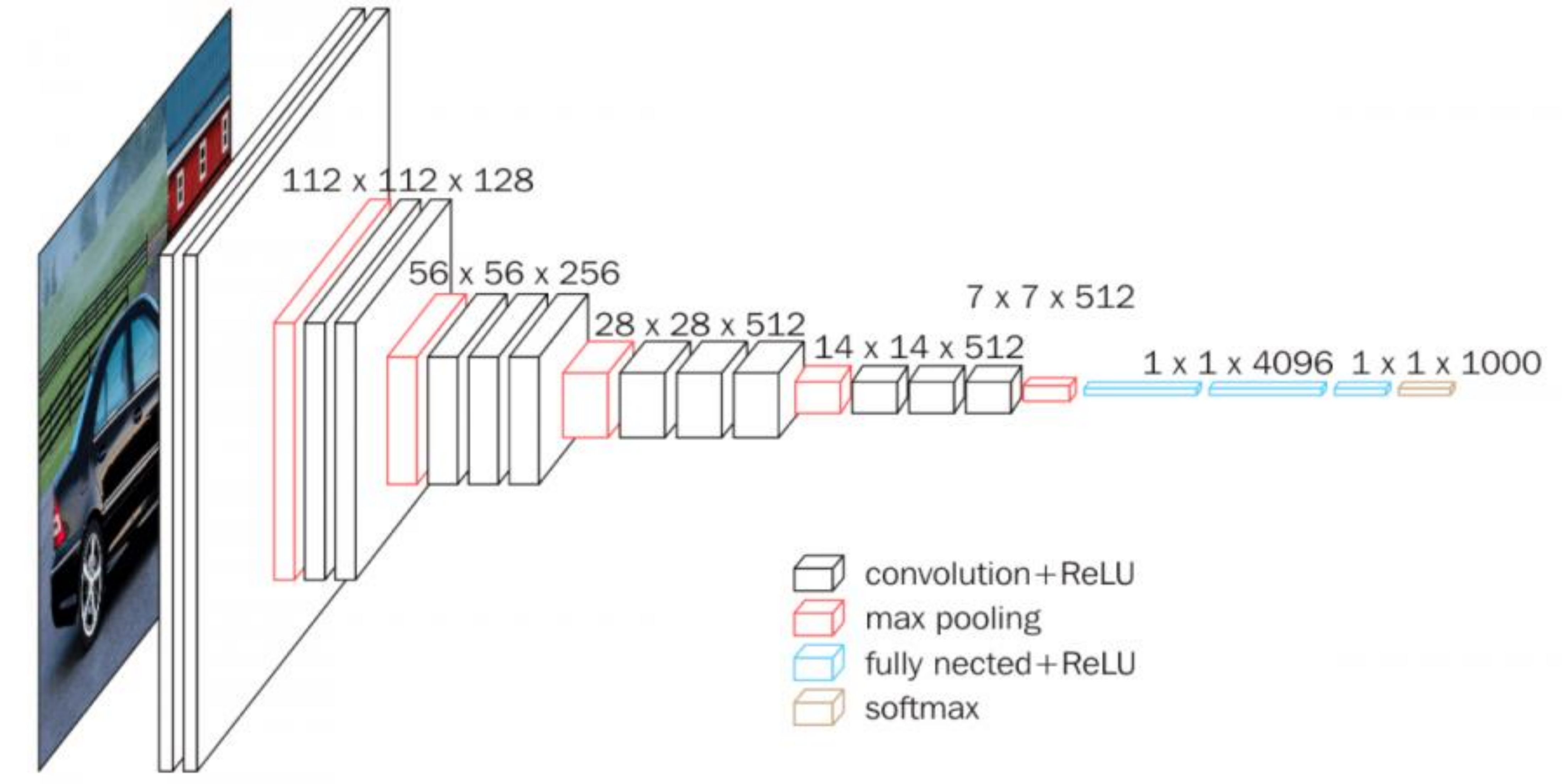
1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	6	1	0
5	6	1	2	9

$$\xrightarrow[\text{Stride (1, 1)}]{\text{Max Pooling}}$$

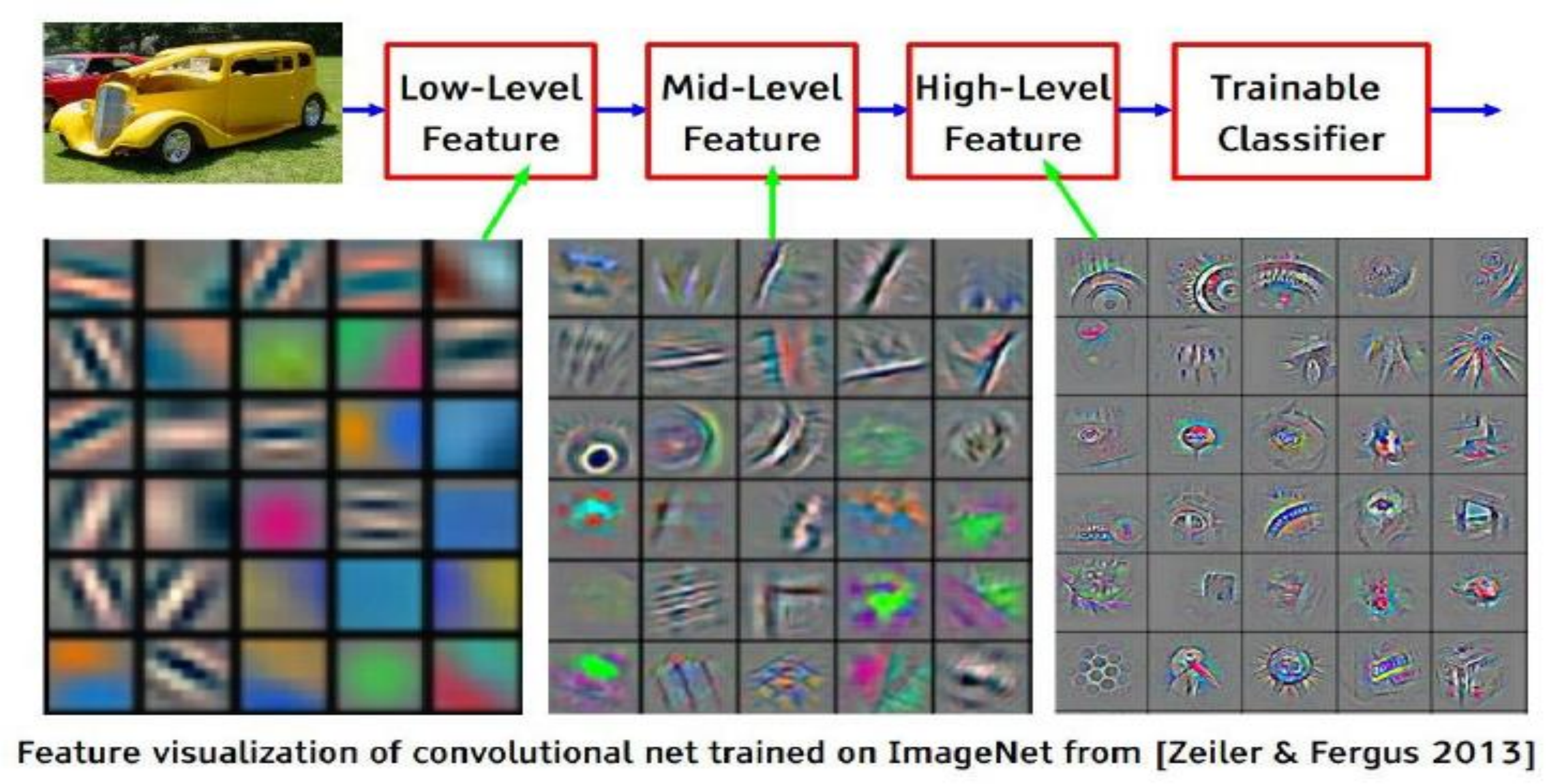
9	9	5
9	9	5
8	6	9

$$\left\lfloor \frac{n-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n-f}{s} + 1 \right\rfloor$$

Etude de cas: VGG19

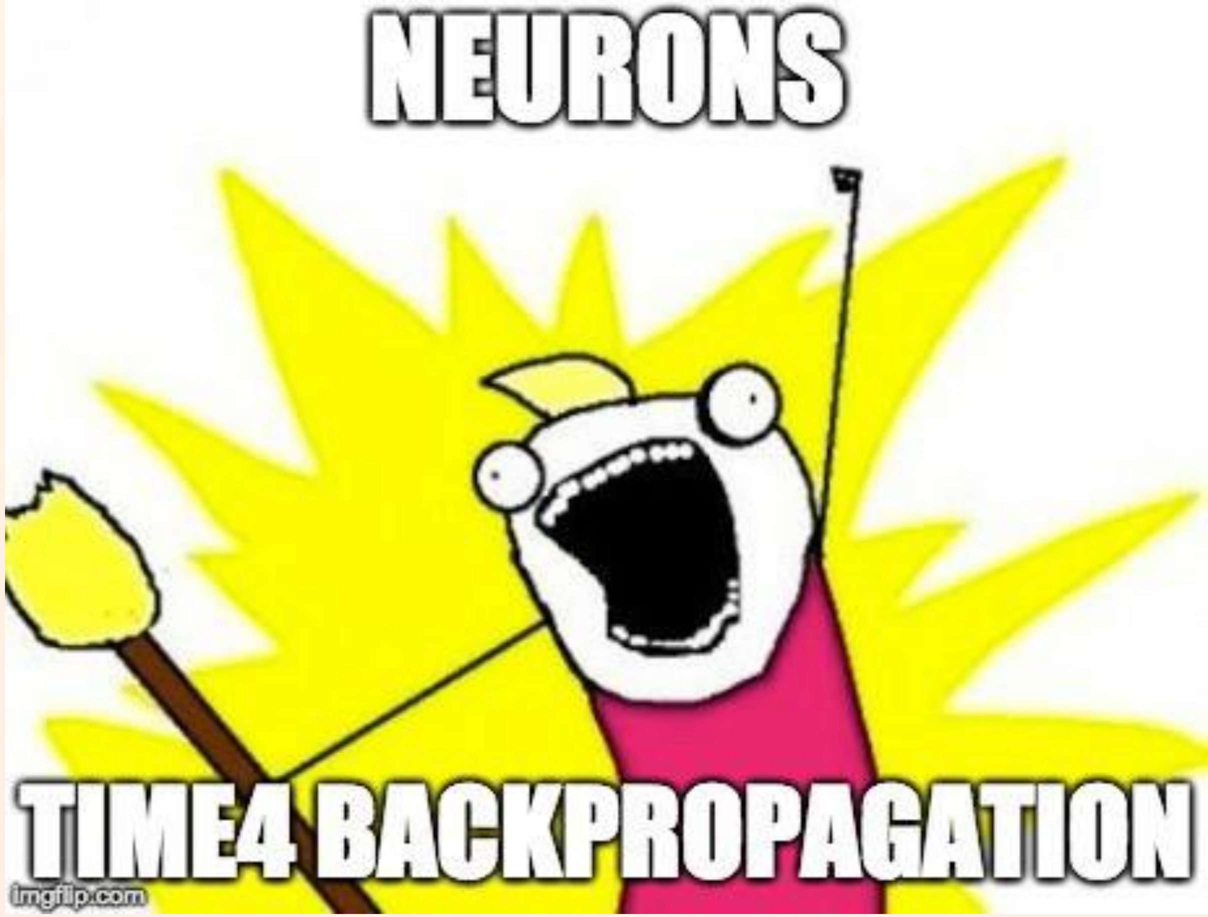


Vérification des hypothèses



https://www.youtube.com/watch?v=AgkfIQ4IGaM&ab_channel=JasonYosinski

Now backpropagate



- Gradients de la sortie/cout par rapport aux poids/paramètres
- Graphe de computation
  - Règle de la chaine
  - Récursivité

Backpropagation 101: Notations

- Etant donné une fonction f et sa valeur en un point x, on veut « calculer » le gradient de f en x

$$f(x,y) = xy \quad \rightarrow \quad \frac{\partial f}{\partial x} = y \quad \frac{\partial f}{\partial y} = x$$

x=4,y=−3 f(x,y)=−12

$$f(x,y) = x + y \quad \rightarrow \quad \frac{\partial f}{\partial x} = 1 \quad \frac{\partial f}{\partial y} = 1$$

x=4,y=−3 f(x,y)=−1

$$f(x,y) = \max(x,y) \quad \rightarrow \quad \frac{\partial f}{\partial x} = 1(x >= y) \quad \frac{\partial f}{\partial y} = 1(y >= x)$$

Backpropagation 101: Graphe de computation

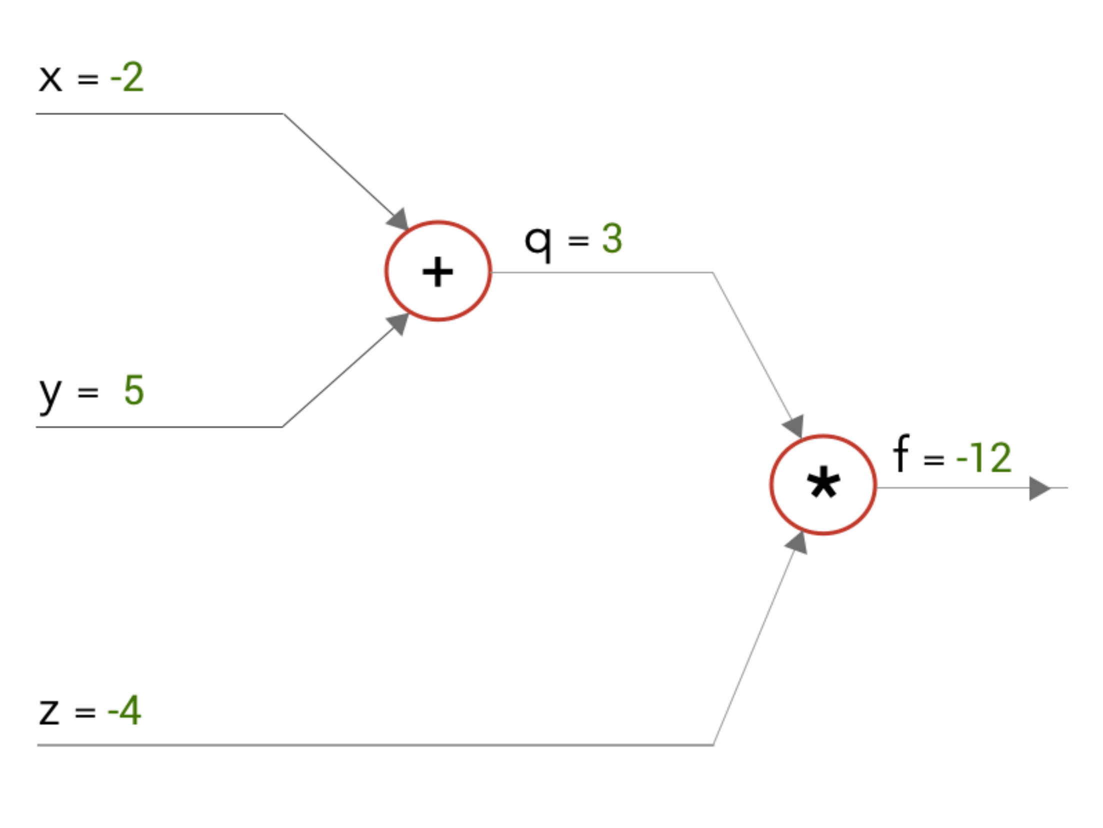
- Un exemple simple: $f(x, y, z) = (x + y) * z$

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

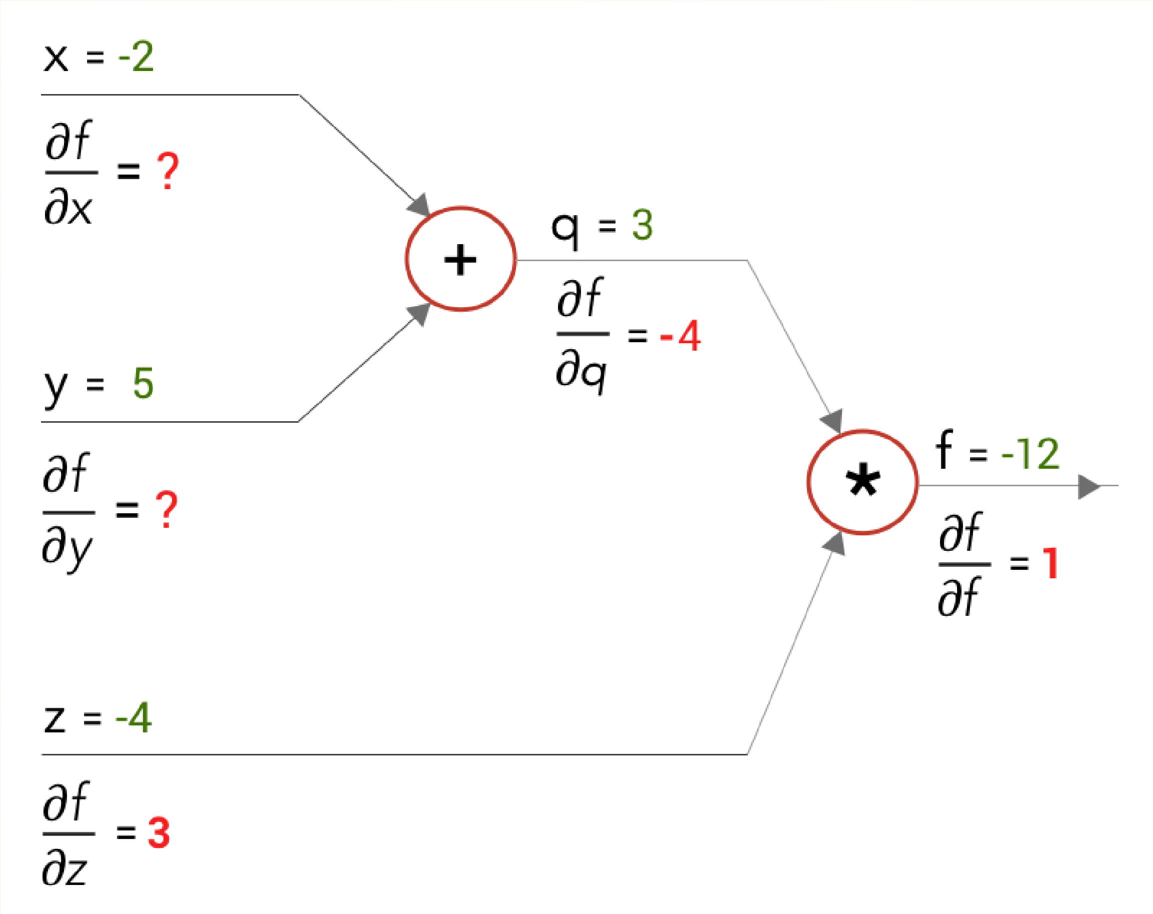
$$q = x + y$$

$$f = q * z$$

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



Backpropagation 101: calcul des gradients



$f = q * z$

$\frac{\partial f}{\partial q} = z \mid z = -4$

$\frac{\partial f}{\partial z} = q \mid q = 3$

$q = x + y$

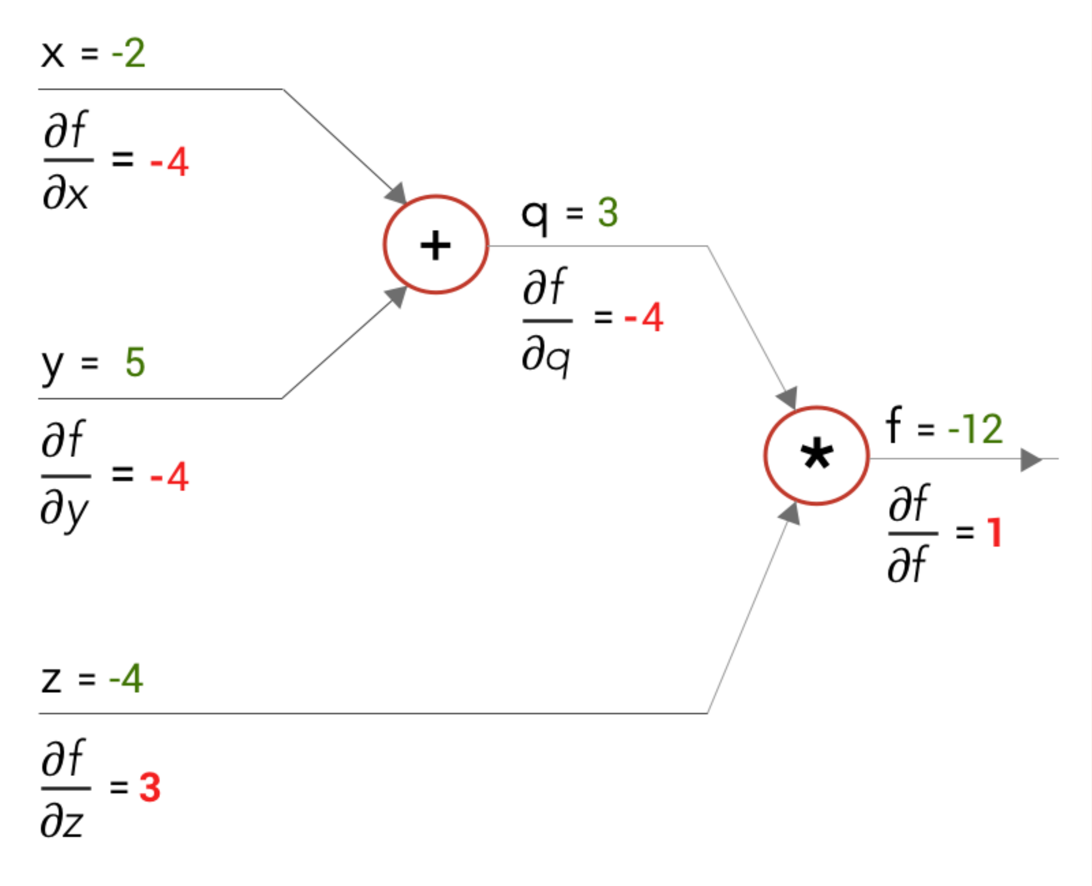
$\frac{\partial q}{\partial x} = 1 \qquad \frac{\partial q}{\partial y} = 1$

Backpropagation 101: Règle de la chaîne

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

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x} = -4 * 1 = -4$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial y} = -4 * 1 = -4$$



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

Backpropagation 101: réseau de neurones

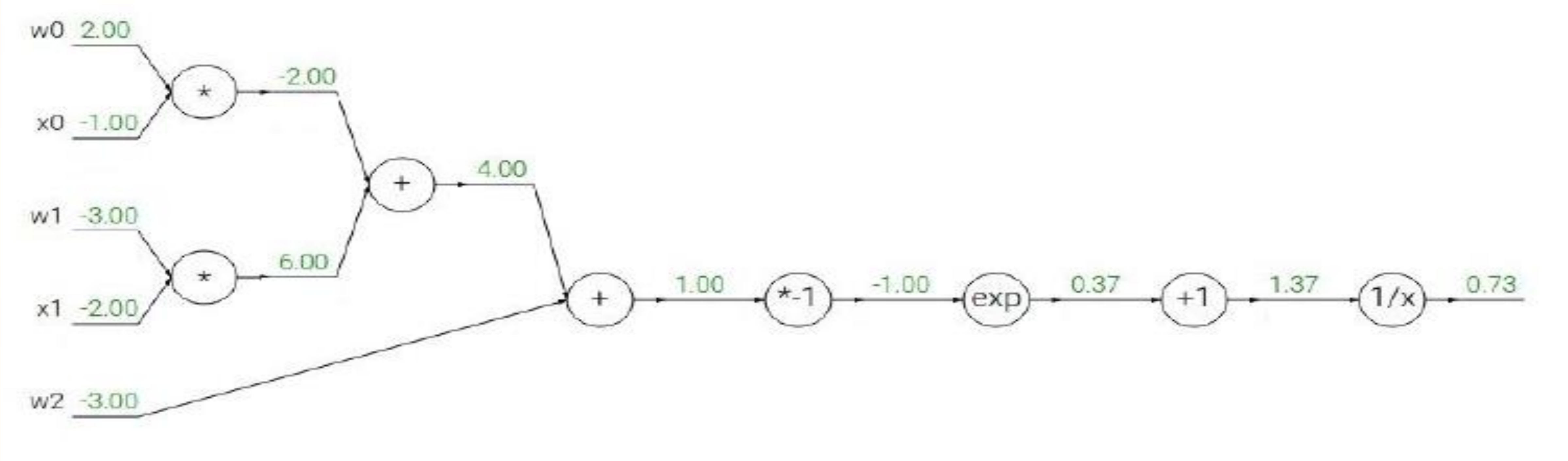
$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

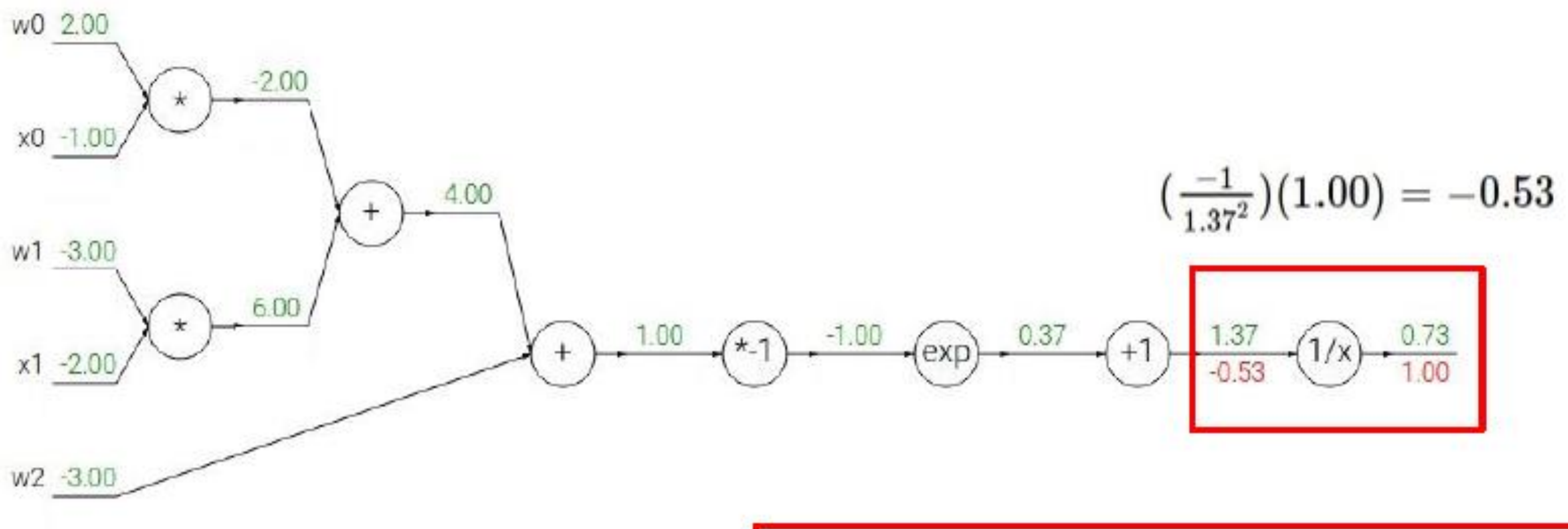
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

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

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



Backpropagation 101: réseau de neurones



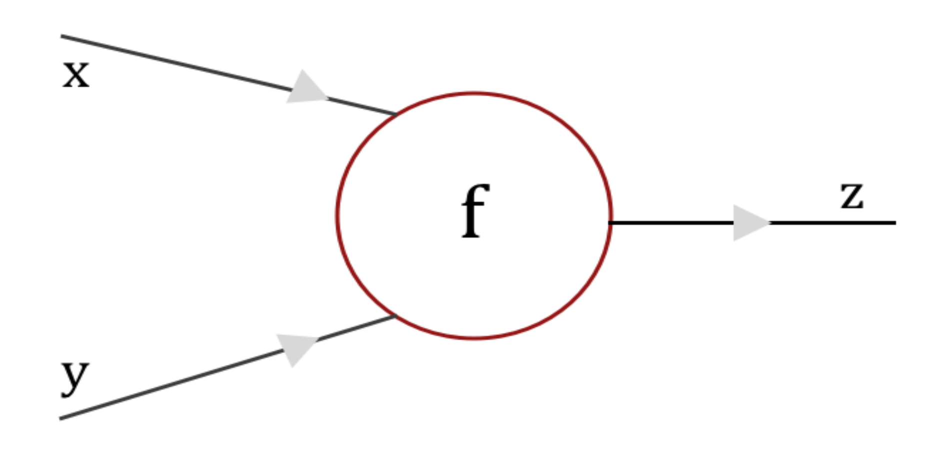
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

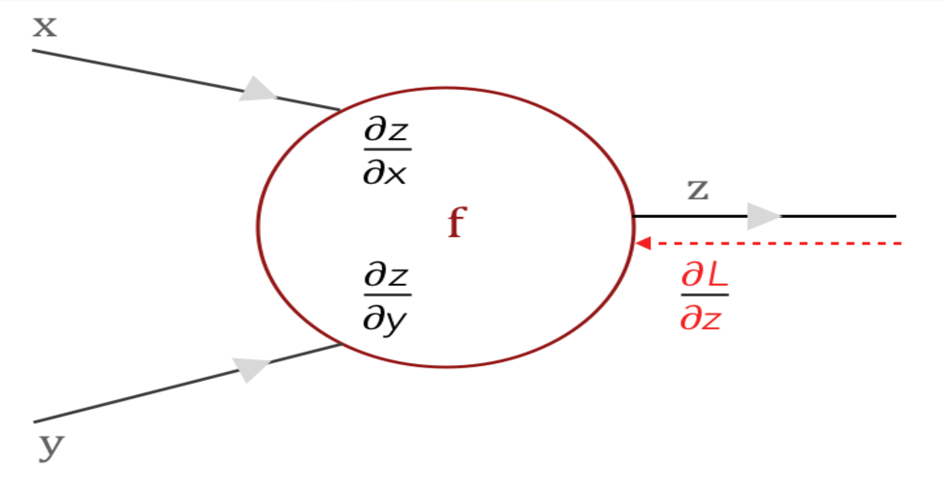
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

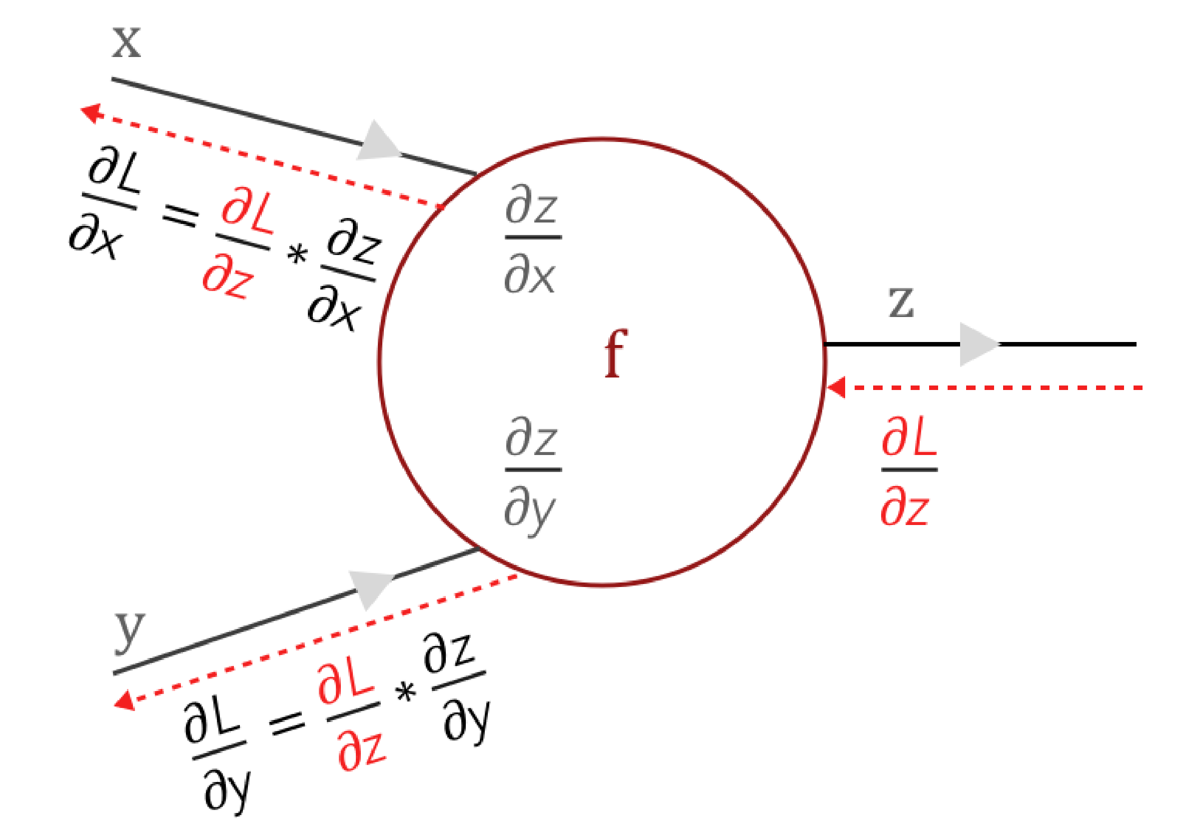
$$\boxed{f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2}$$

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

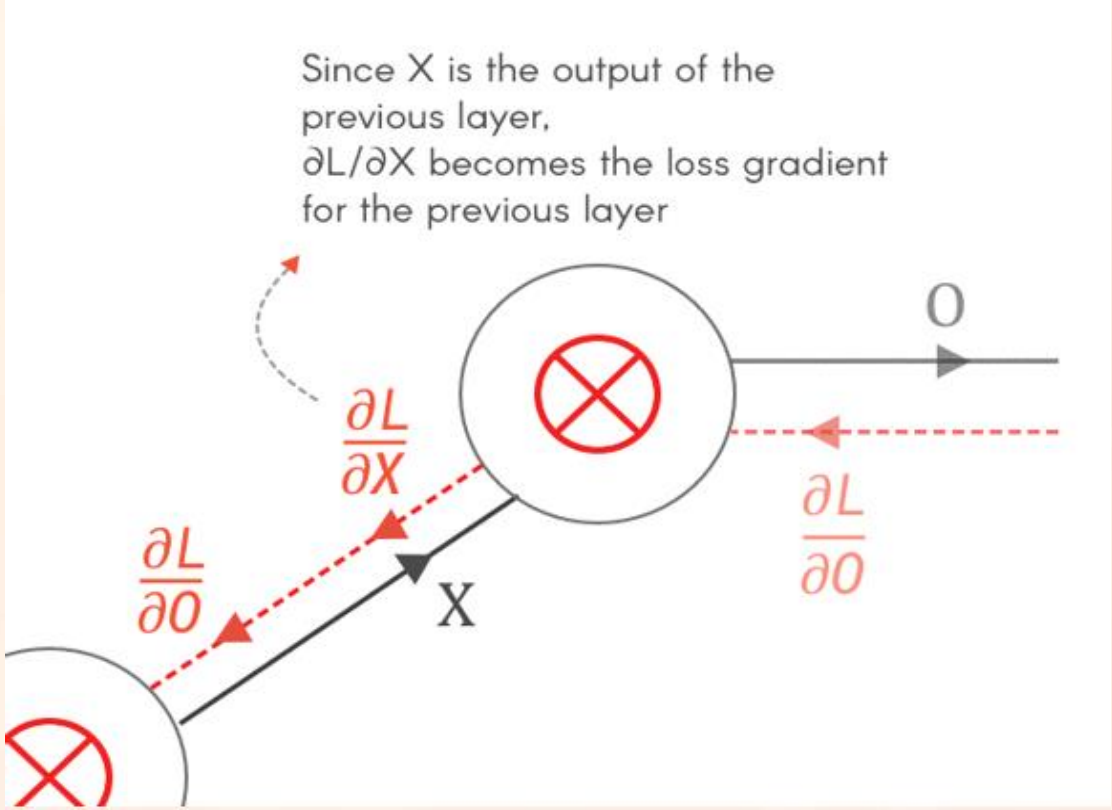
Backpropagation 101: Plus généralement

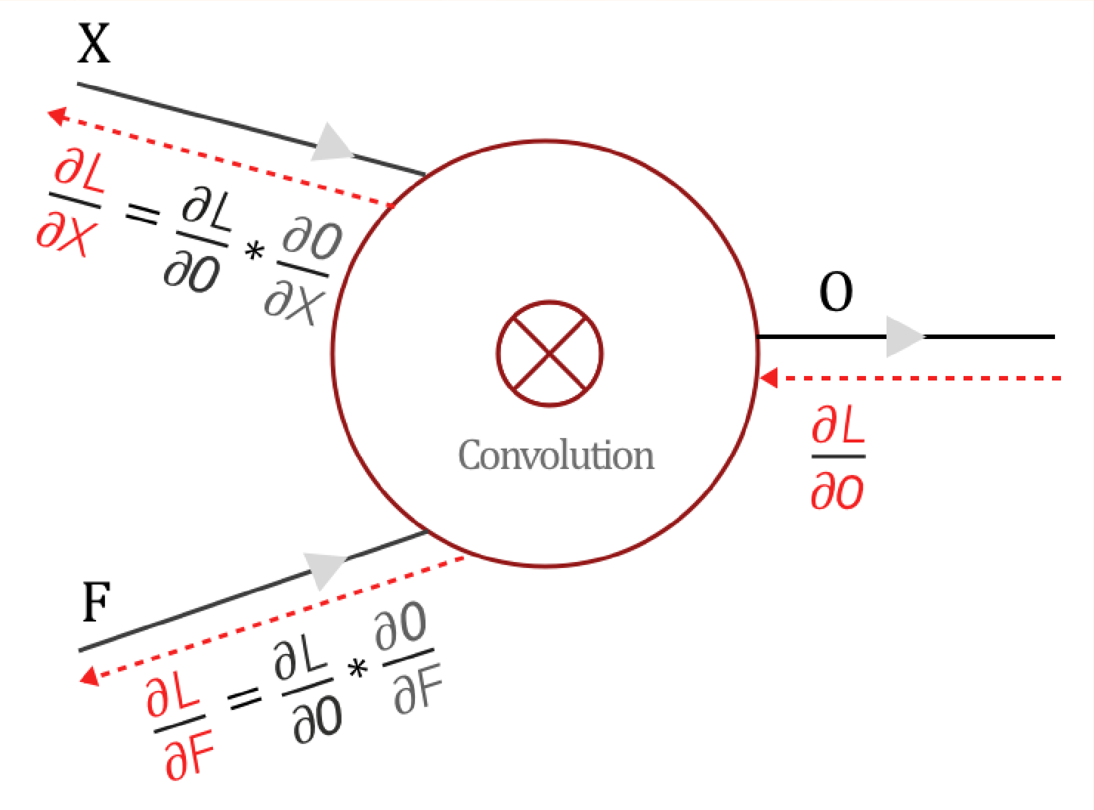






Graph de convolution





Back-(to convolution)-propagation

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} \qquad \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$$

Input X **Filter F**

$$\underbrace{\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}}_{\text{Output } O} = \text{Convolution}\left(\underbrace{\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}}_{\text{Input } X}, \underbrace{\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}}_{\text{Filter } F} \right)$$

$$\underbrace{\begin{bmatrix} \boxed{\begin{matrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{matrix}} & \begin{matrix} X_{13} \\ X_{23} \end{matrix} \\ \begin{matrix} X_{31} & X_{32} \end{matrix} & X_{33} \end{bmatrix}}_{\text{Input } X} \otimes \underbrace{\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}}_{\text{Filter } F}$$

$$\begin{bmatrix} X_{11}F_{11} & X_{12}F_{12} & X_{13} \\ X_{21}F_{21} & X_{22}F_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}$$

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

Gradients du filtre

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

$$\frac{\partial O_{11}}{\partial F_{11}} = X_{11} \quad \frac{\partial O_{11}}{\partial F_{12}} = X_{12} \quad \frac{\partial O_{11}}{\partial F_{21}} = X_{21} \quad \frac{\partial O_{11}}{\partial F_{22}} = X_{22}$$

$$\frac{\partial L}{\partial F_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial F_i}$$

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * X_{11} + \frac{\partial L}{\partial O_{12}} * X_{12} + \frac{\partial L}{\partial O_{21}} * X_{21} + \frac{\partial L}{\partial O_{22}} * X_{22}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * X_{12} + \frac{\partial L}{\partial O_{12}} * X_{13} + \frac{\partial L}{\partial O_{21}} * X_{22} + \frac{\partial L}{\partial O_{22}} * X_{23}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * X_{21} + \frac{\partial L}{\partial O_{12}} * X_{22} + \frac{\partial L}{\partial O_{21}} * X_{31} + \frac{\partial L}{\partial O_{22}} * X_{32}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * X_{22} + \frac{\partial L}{\partial O_{12}} * X_{23} + \frac{\partial L}{\partial O_{21}} * X_{32} + \frac{\partial L}{\partial O_{22}} * X_{33}$$

Gradient du filtre

$$\begin{bmatrix} \frac{\partial L}{\partial F_{11}} & \frac{\partial L}{\partial F_{12}} \\ \frac{\partial L}{\partial F_{21}} & \frac{\partial L}{\partial F_{22}} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

where

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} = \text{Input } X$$

$$\begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} = \frac{\partial L}{\partial O}$$

Loss gradient from previous layer

Gradient de la couche précédente

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

$$\frac{\partial O_{11}}{\partial X_{11}} = F_{11} \quad \frac{\partial O_{11}}{\partial X_{12}} = F_{12} \quad \frac{\partial O_{11}}{\partial X_{21}} = F_{21} \quad \frac{\partial O_{11}}{\partial X_{22}} = F_{22}$$

$$\frac{\partial L}{\partial X_i} = \sum_{k=1}^{M} \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial X_i}$$

$$\frac{\partial L}{\partial X_{11}} = \frac{\partial L}{\partial O_{11}} * F_{11}$$

$$\frac{\partial L}{\partial X_{12}} = \frac{\partial L}{\partial O_{11}} * F_{12} + \frac{\partial L}{\partial O_{12}} * F_{11}$$

$$\frac{\partial L}{\partial X_{13}} = \frac{\partial L}{\partial O_{12}} * F_{12}$$

$$\frac{\partial L}{\partial X_{21}} = \frac{\partial L}{\partial O_{11}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{11}$$

$$\frac{\partial L}{\partial X_{22}} = \frac{\partial L}{\partial O_{11}} * F_{22} + \frac{\partial L}{\partial O_{12}} * F_{21} + \frac{\partial L}{\partial O_{21}} * F_{12} + \frac{\partial L}{\partial O_{22}} * F_{11}$$

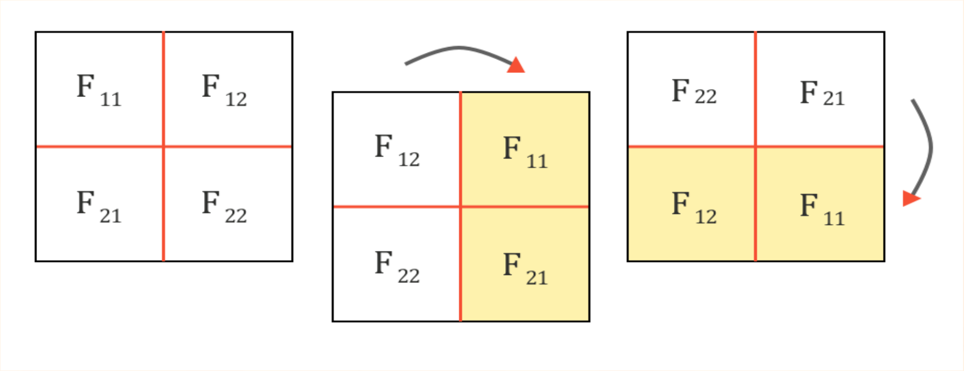
$$\frac{\partial L}{\partial X_{23}} = \frac{\partial L}{\partial O_{12}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{12}$$

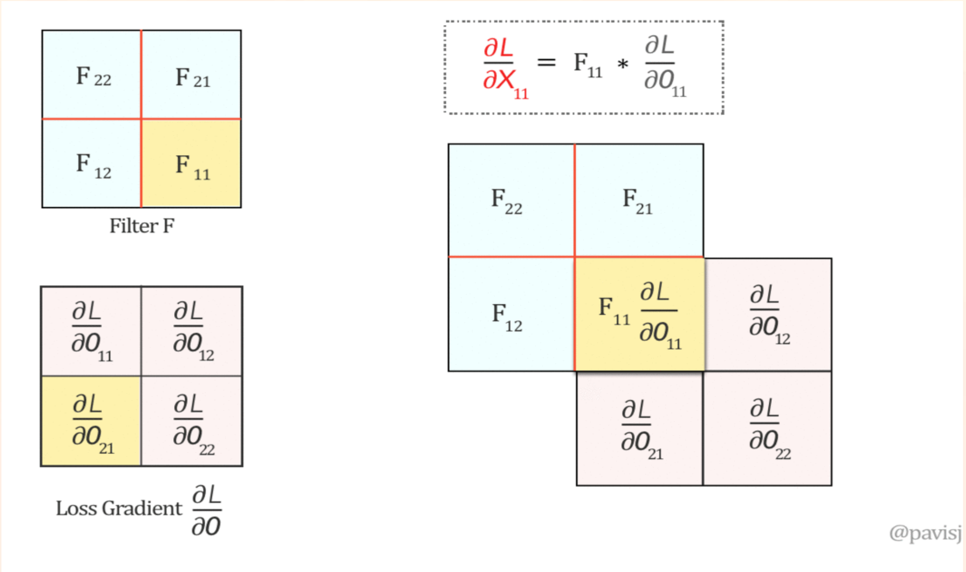
$$\frac{\partial L}{\partial X_{31}} = \frac{\partial L}{\partial O_{21}} * F_{21}$$

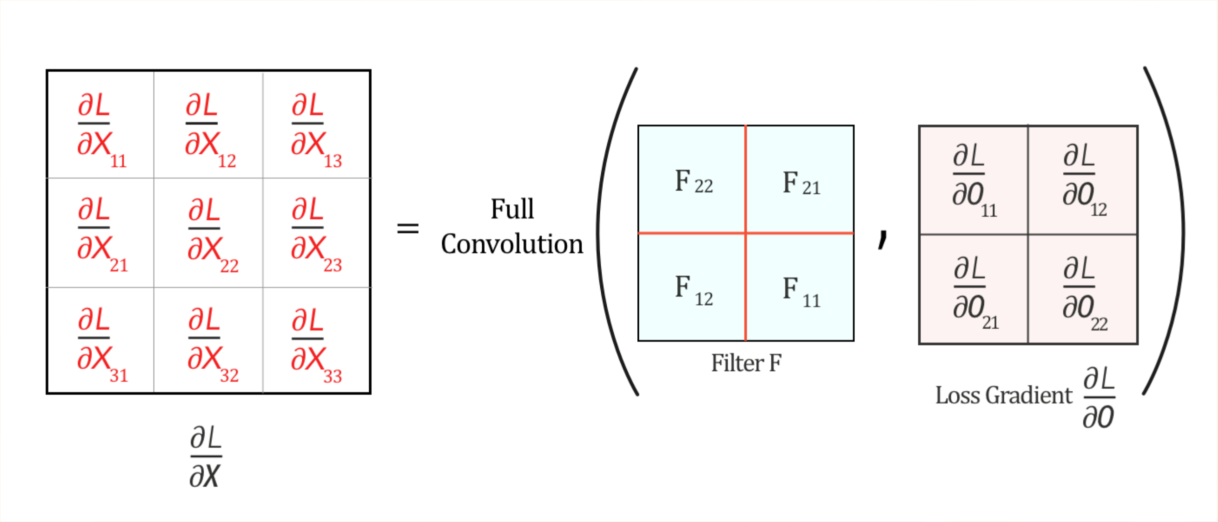
$$\frac{\partial L}{\partial X_{32}} = \frac{\partial L}{\partial O_{21}} * F_{22} + \frac{\partial L}{\partial O_{22}} * F_{21}$$

$$\frac{\partial L}{\partial X_{33}} = \frac{\partial L}{\partial O_{22}} * F_{22}$$

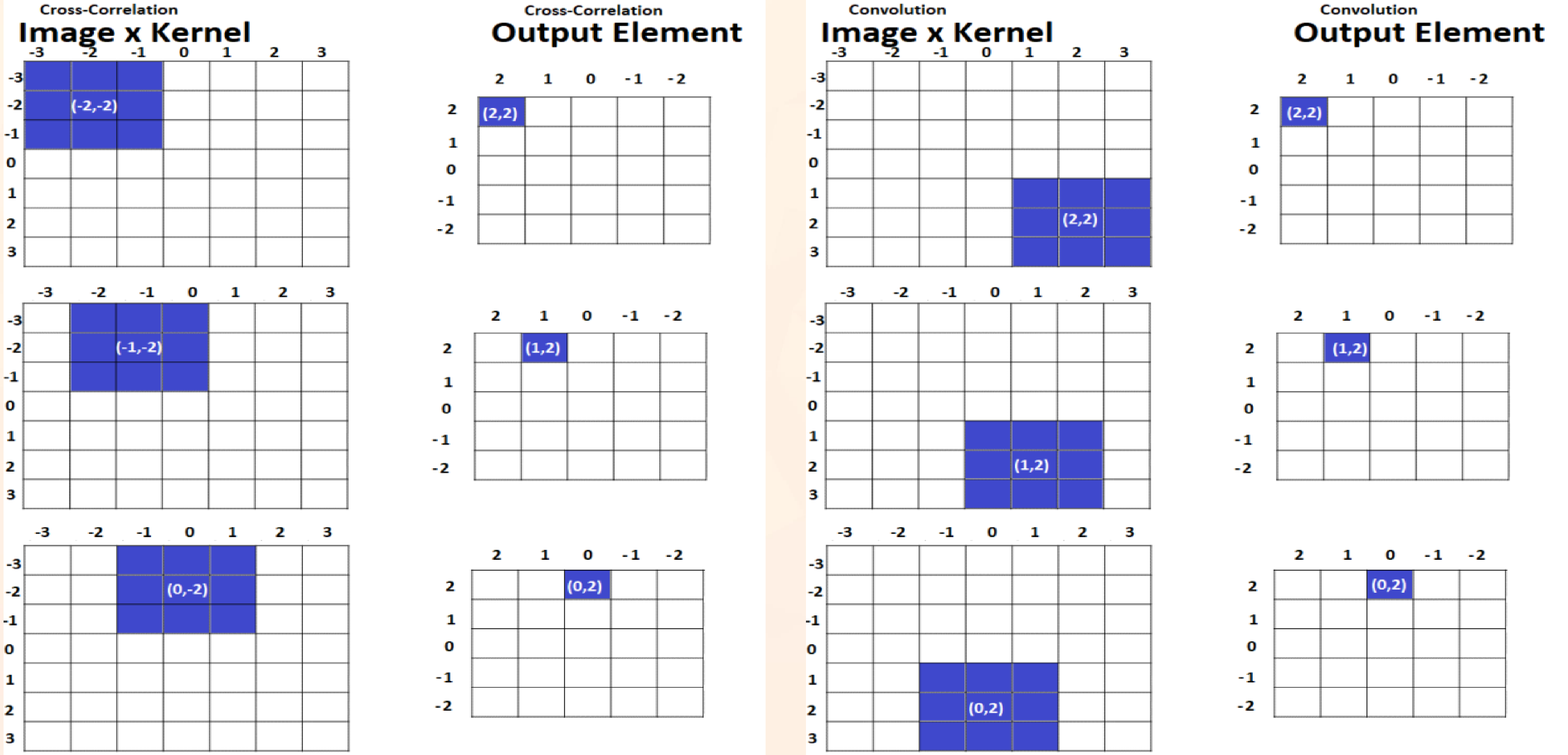
Gradient de la couche précédente







Convolution vs Cross-Correlation



Finalement

$$\frac{\partial L}{\partial F} = \text{Convolution}\left(\text{Input } X \text{ , Loss gradient } \frac{\partial L}{\partial O}\right)$$

$$\frac{\partial L}{\partial X} = \begin{matrix}\text{Full}\\ \text{Convolution}\end{matrix}\left(\begin{matrix}180^{\circ}\text{rotated}\\ \text{Filter } F\end{matrix} \text{ , } \begin{matrix}\text{Loss}\\ \text{Gradient}\end{matrix} \frac{\partial L}{\partial O}\right)$$


CHEERS!!
YOU DID IT
imgflip.com