Réseaux de neurones à convolution

Guillaume Bourmaud

PLAN

I. Couche de convolution

II. Réseaux de neurones à convolution

I) Couche de convolution

Limites d’une transformation affine générale (FC)

H=480 

W=640

$\mathbf{x} : 640 \times 480 \times 3 \approx 10^6$ éléments

Limites d’une transformation affine générale (FC)

H=480

W=640

$$\mathbf{x} : 640 \times 480 \times 3 \approx 10^6 \text{ éléments}$$

Exemple d’une seule couche FC préservant la résolution de l’image d’entrée

$$\mathbf{y} = \mathtt{W}\mathbf{x} + \mathbf{b}$$

10^6x1 $\quad$ 10^6x10^6 $\quad$ 10^6x1 $\quad$ 10^6x1

I)

Limites d'une transformation affine générale (FC)

H=480 

W=640

$\mathbf{x} : 640 \times 480 \times 3 \approx 10^6$ éléments

Exemple d'une seule couche FC préservant la résolution de l'image d'entrée

$$\mathbf{y} = \mathtt{W}\mathbf{x} + \mathbf{b}$$

10^6x1 10^6x10^6 10^6x1 10^6x1

Occupation mémoire de W : 4 octets (32 bits) x 10^6x10^6 = 4To.

Nombre de « multiplications+additions » également très élevé.

Opération de « convolution » = transformation affine spécifique

“Fully Connected”
=
Transformation affine générale

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & \mathtt{W}_{13} & \mathtt{W}_{14} \\ \mathtt{W}_{21} & \mathtt{W}_{22} & \mathtt{W}_{23} & \mathtt{W}_{24} \\ \mathtt{W}_{31} & \mathtt{W}_{32} & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Opération de « convolution » = transformation affine spécifique

“Fully Connected”
=
Transformation affine générale

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & \mathtt{W}_{13} & \mathtt{W}_{14} \\ \mathtt{W}_{21} & \mathtt{W}_{22} & \mathtt{W}_{23} & \mathtt{W}_{24} \\ \mathtt{W}_{31} & \mathtt{W}_{32} & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Localement connecté

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & 0 & 0 \\ 0 & \mathtt{W}_{22} & \mathtt{W}_{23} & 0 \\ 0 & 0 & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Opération de « convolution » = transformation affine spécifique

"Fully Connected"
=
Transformation affine générale

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & \mathtt{W}_{13} & \mathtt{W}_{14} \\ \mathtt{W}_{21} & \mathtt{W}_{22} & \mathtt{W}_{23} & \mathtt{W}_{24} \\ \mathtt{W}_{31} & \mathtt{W}_{32} & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Localement connecté

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & 0 & 0 \\ 0 & \mathtt{W}_{22} & \mathtt{W}_{23} & 0 \\ 0 & 0 & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Équivariance par translation
=
« Convolution »

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & 0 & 0 \\ 0 & \mathtt{W}_{11} & \mathtt{W}_{12} & 0 \\ 0 & 0 & \mathtt{W}_{11} & \mathtt{W}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix}$$

I)

Opération de « convolution » = transformation affine spécifique

“Fully Connected”
=
Transformation affine générale

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & \mathtt{W}_{13} & \mathtt{W}_{14} \\ \mathtt{W}_{21} & \mathtt{W}_{22} & \mathtt{W}_{23} & \mathtt{W}_{24} \\ \mathtt{W}_{31} & \mathtt{W}_{32} & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Localement connecté

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & 0 & 0 \\ 0 & \mathtt{W}_{22} & \mathtt{W}_{23} & 0 \\ 0 & 0 & \mathtt{W}_{33} & \mathtt{W}_{34} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

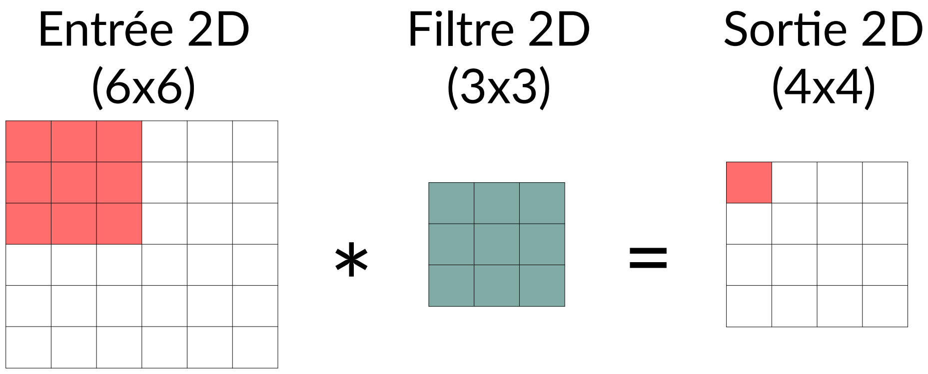
Équivariance par translation
=
« Convolution »

$$\begin{bmatrix} \mathtt{W}_{11} & \mathtt{W}_{12} & 0 & 0 \\ 0 & \mathtt{W}_{11} & \mathtt{W}_{12} & 0 \\ 0 & 0 & \mathtt{W}_{11} & \mathtt{W}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix}$$

Beaucoup moins de paramètres à stocker

Beaucoup moins de « multiplications+additions »

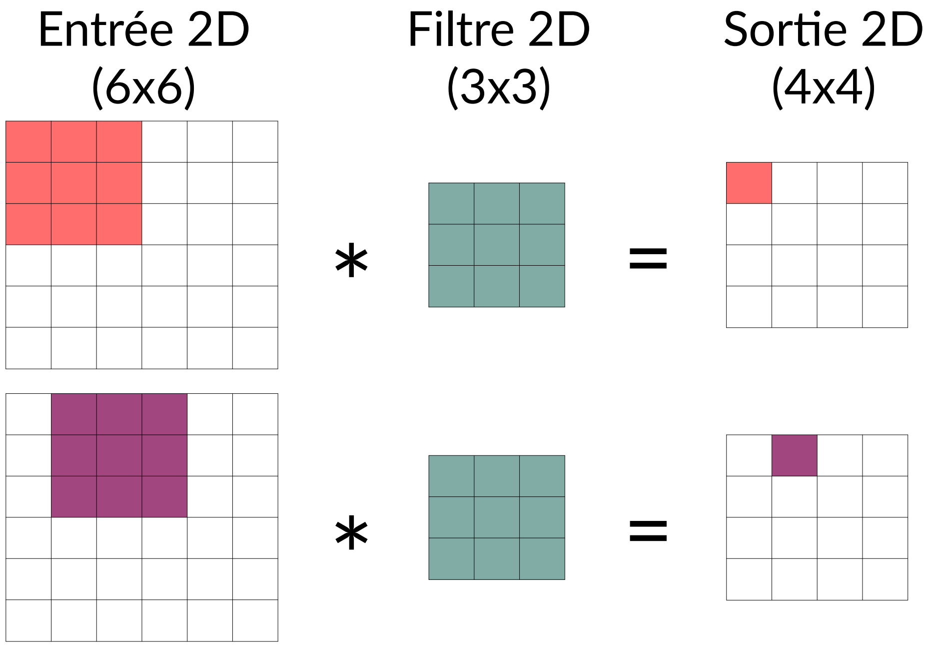
Opération de « convolution » en 2D



En fait, il s'agit d'une intercorrélation

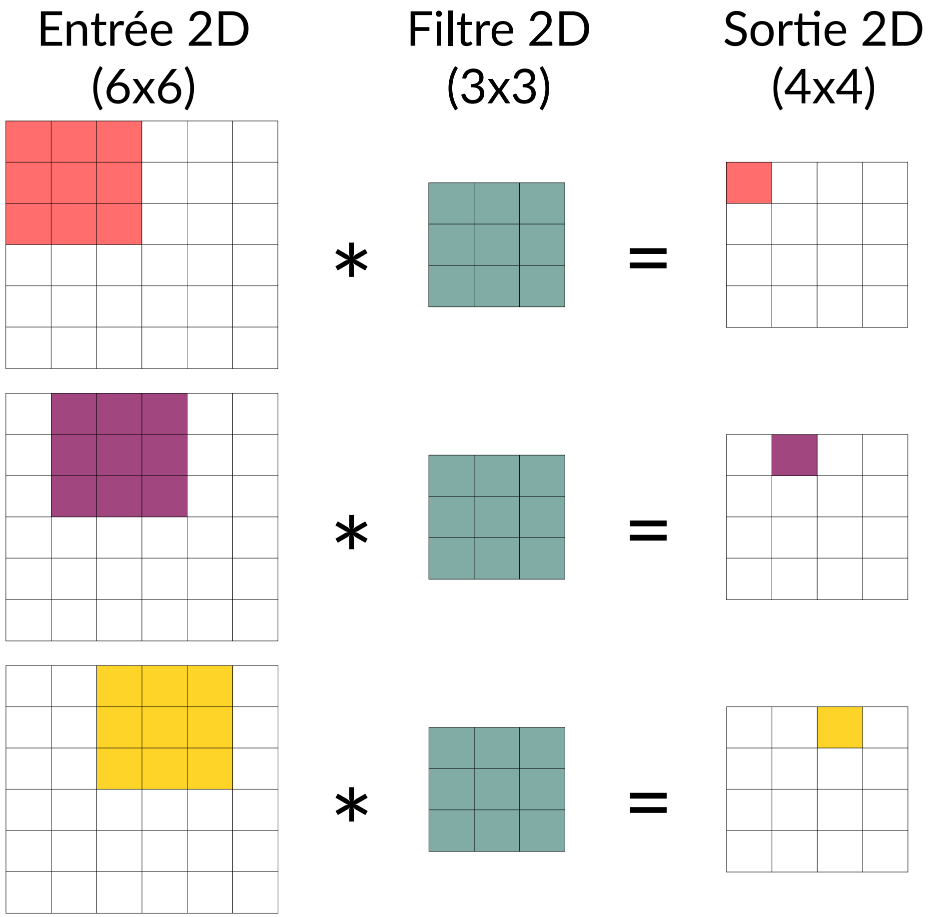
Opération de « convolution » en 2D

En fait, il s'agit d'une intercorrélation

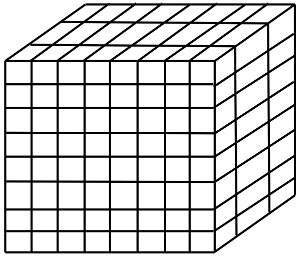


Opération de « convolution » en 2D

En fait, il s'agit d'une intercorrélation



Couche de convolution 2D : un seul filtre



$\mathtt{X}^{(0)}$

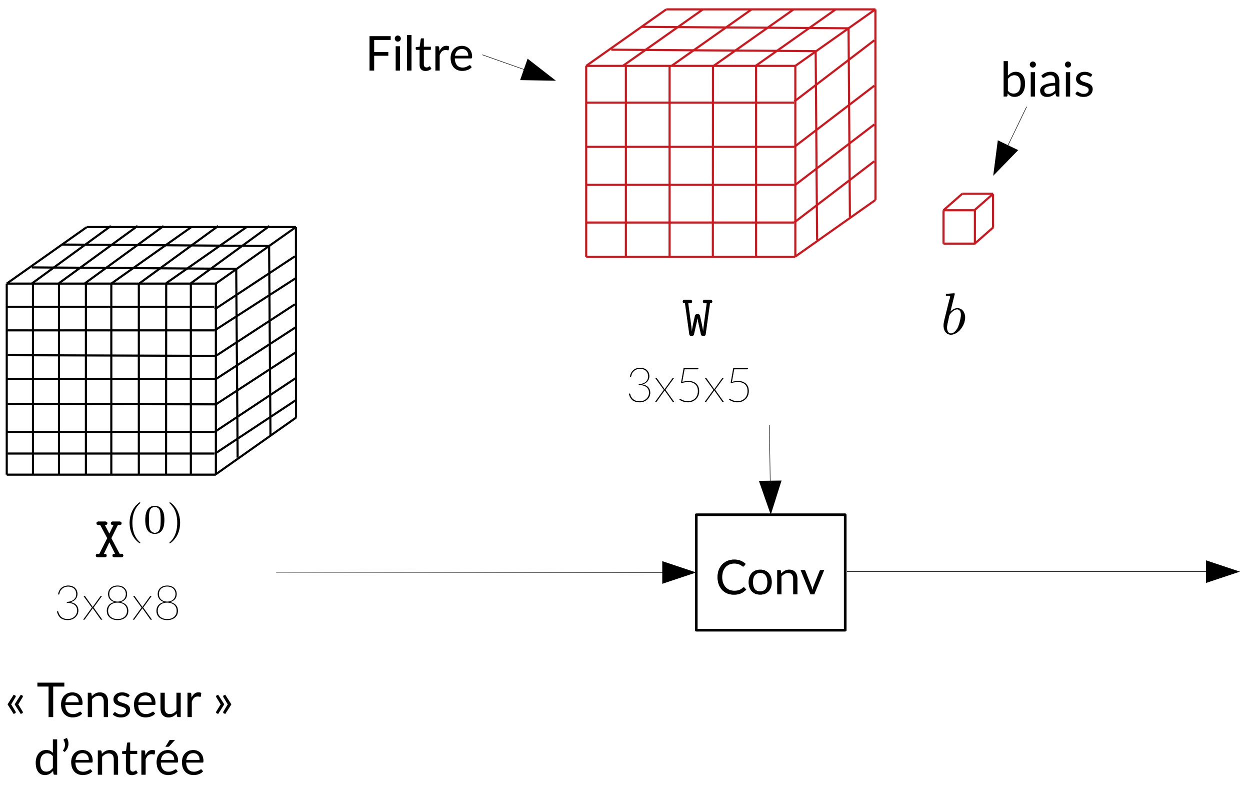
3x8x8

« Tenseur »
d'entrée

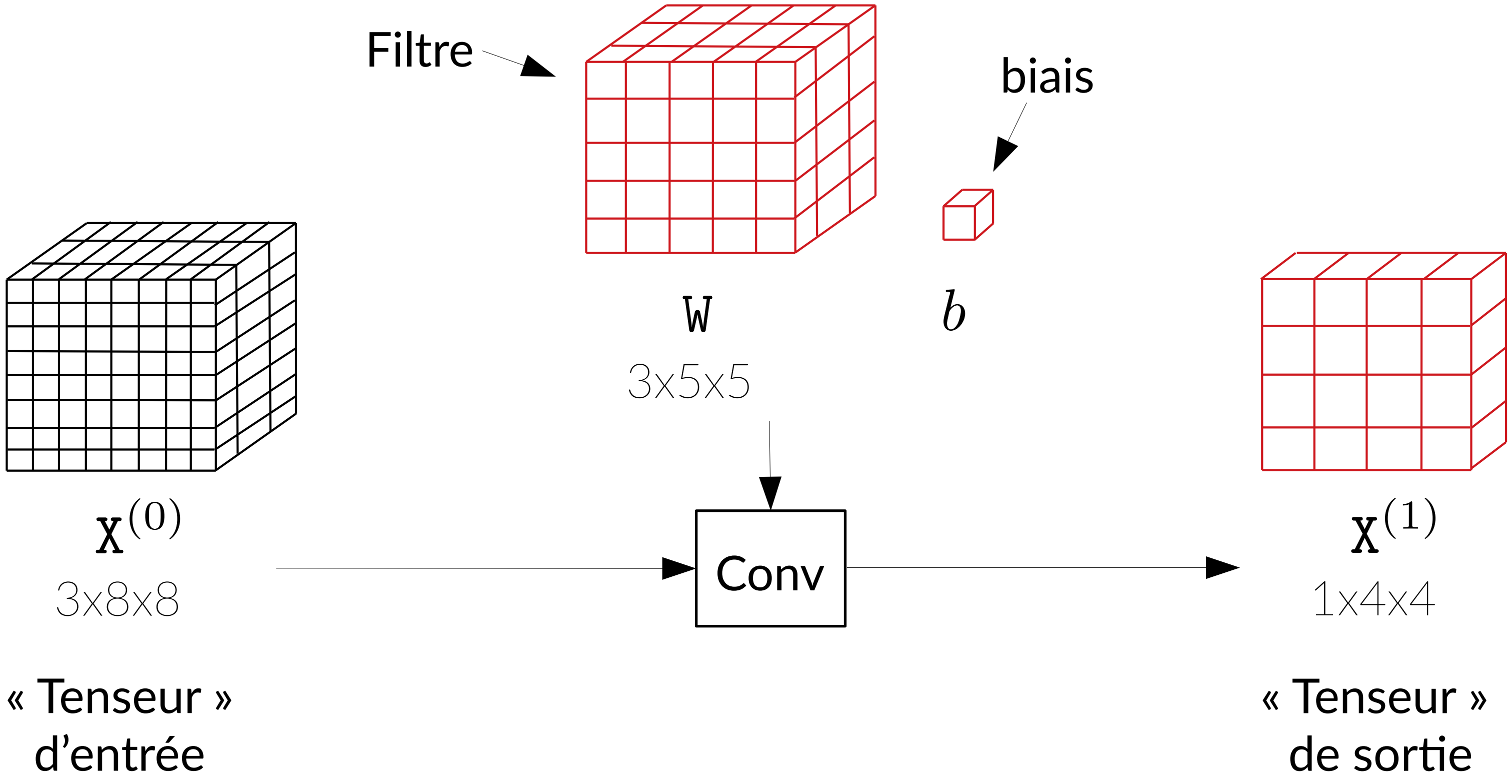
« Tenseur » = tableau multi-dimensionnel

I)

Couche de convolution 2D : un seul filtre

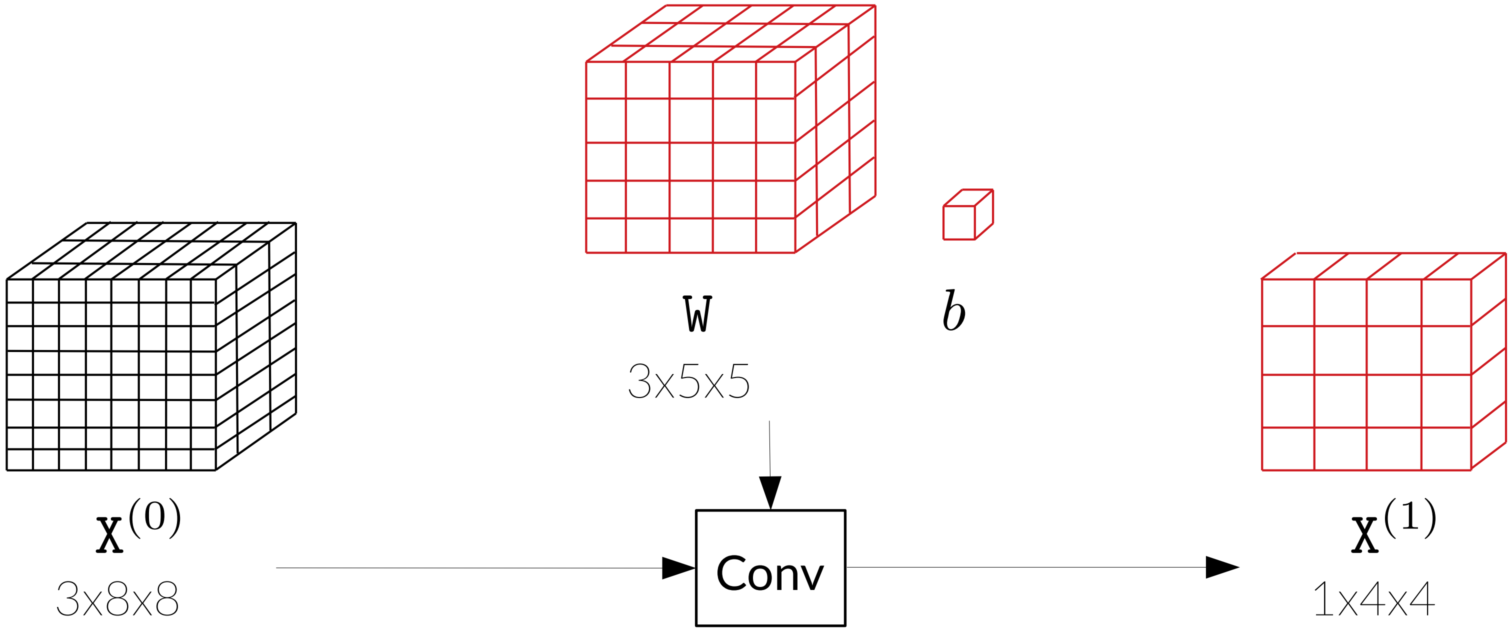


Couche de convolution 2D : un seul filtre



I)

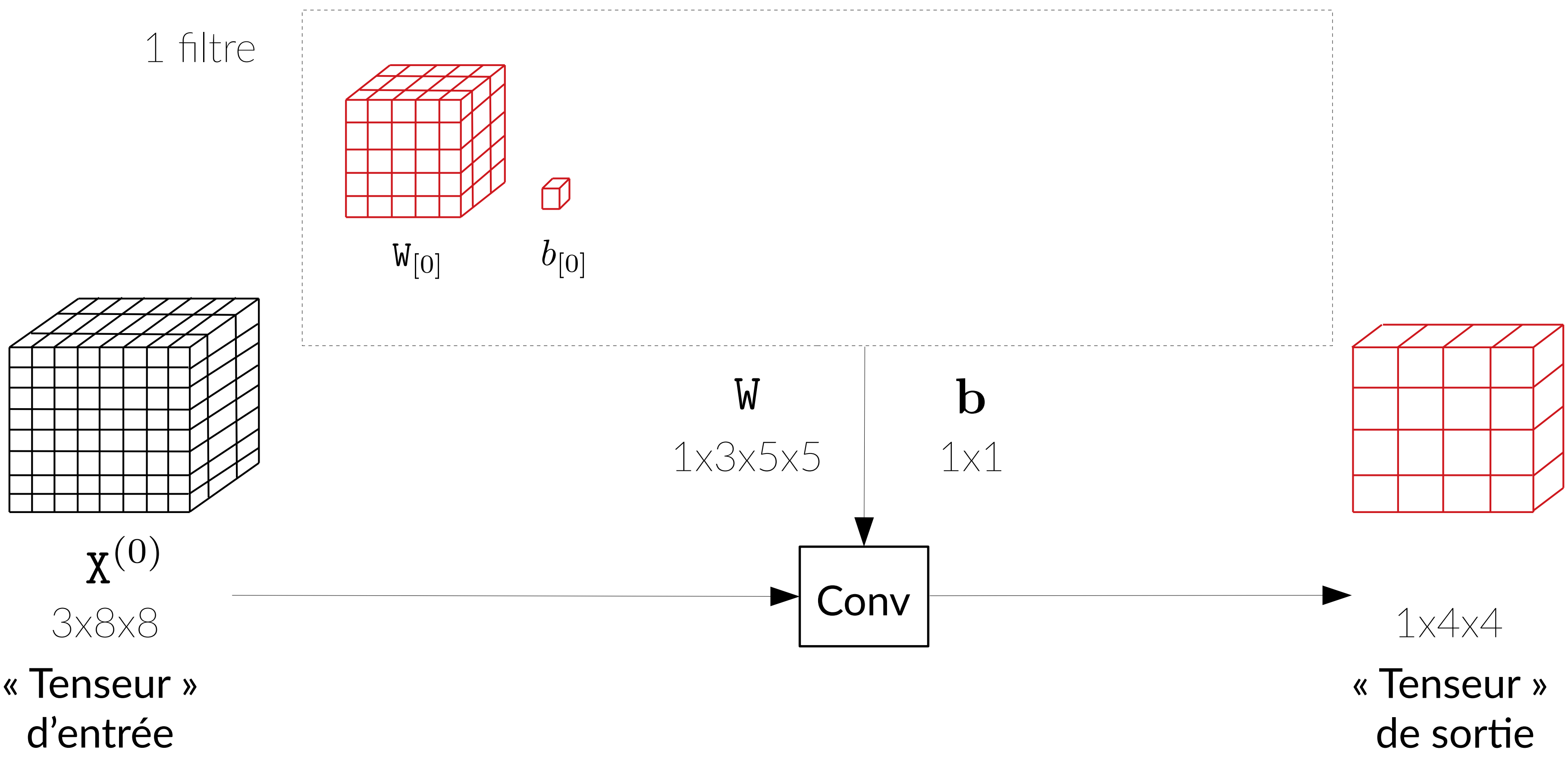
Couche de convolution 2D : un seul filtre



$$\mathtt{X}^{(1)}_{i,j} = \sum_{k=0}^{2} \sum_{m=0}^{4} \sum_{n=0}^{4} \mathtt{W}_{k,m,n} \mathtt{X}^{(0)}_{k,i+m,j+n} + b$$

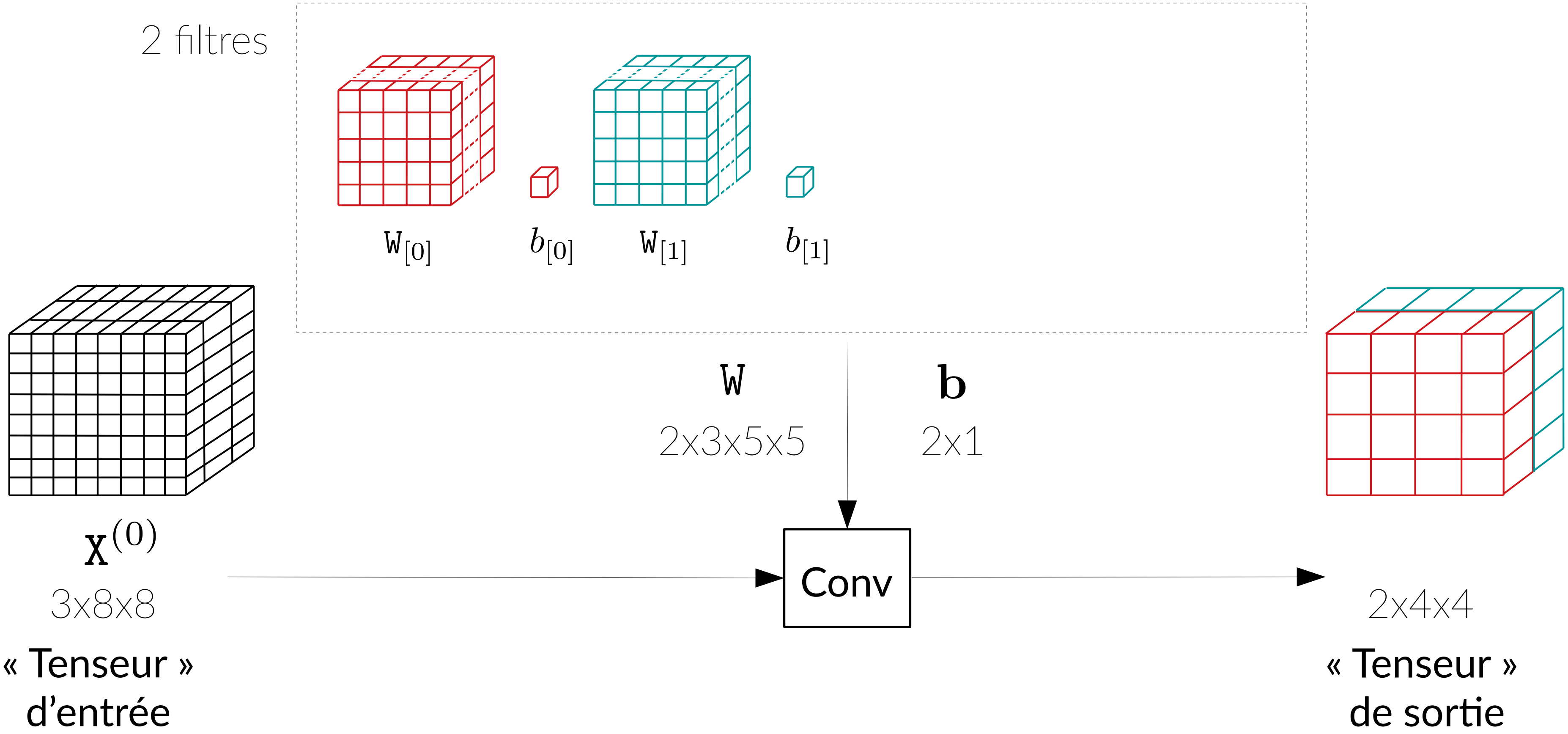
I)

Couche de convolution 2D : K filtres



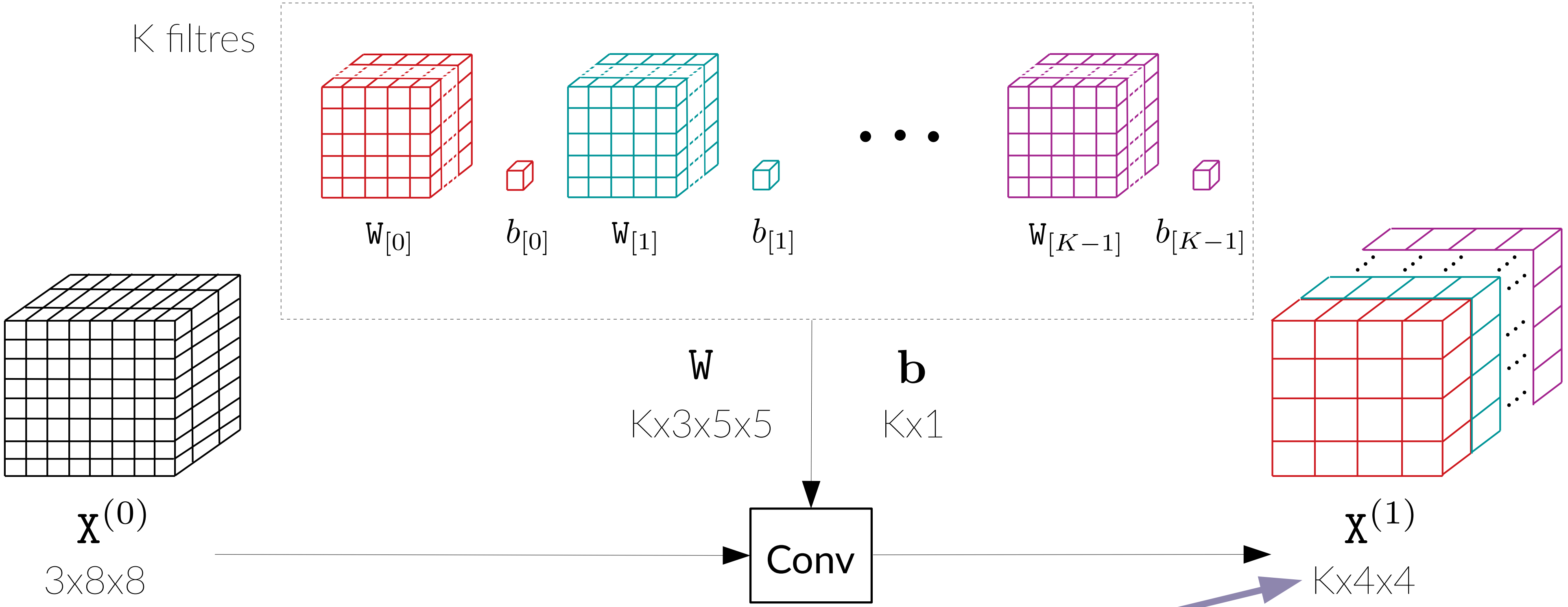
I)

Couche de convolution 2D : K filtres



I)

Couche de convolution 2D : K filtres



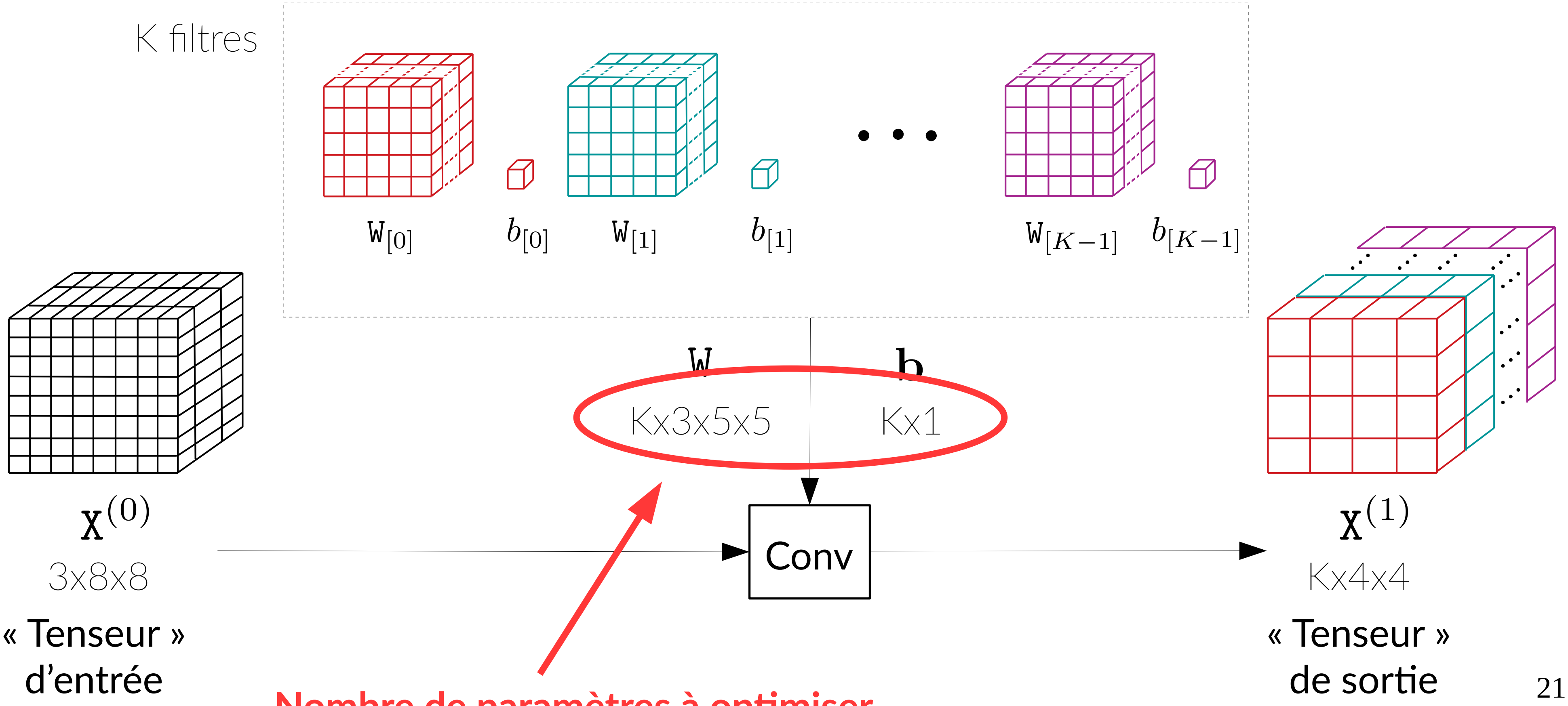
« Tenseur » d'entrée

Nombre de canaux
=
Nombre de filtres

« Tenseur » de sortie

I)

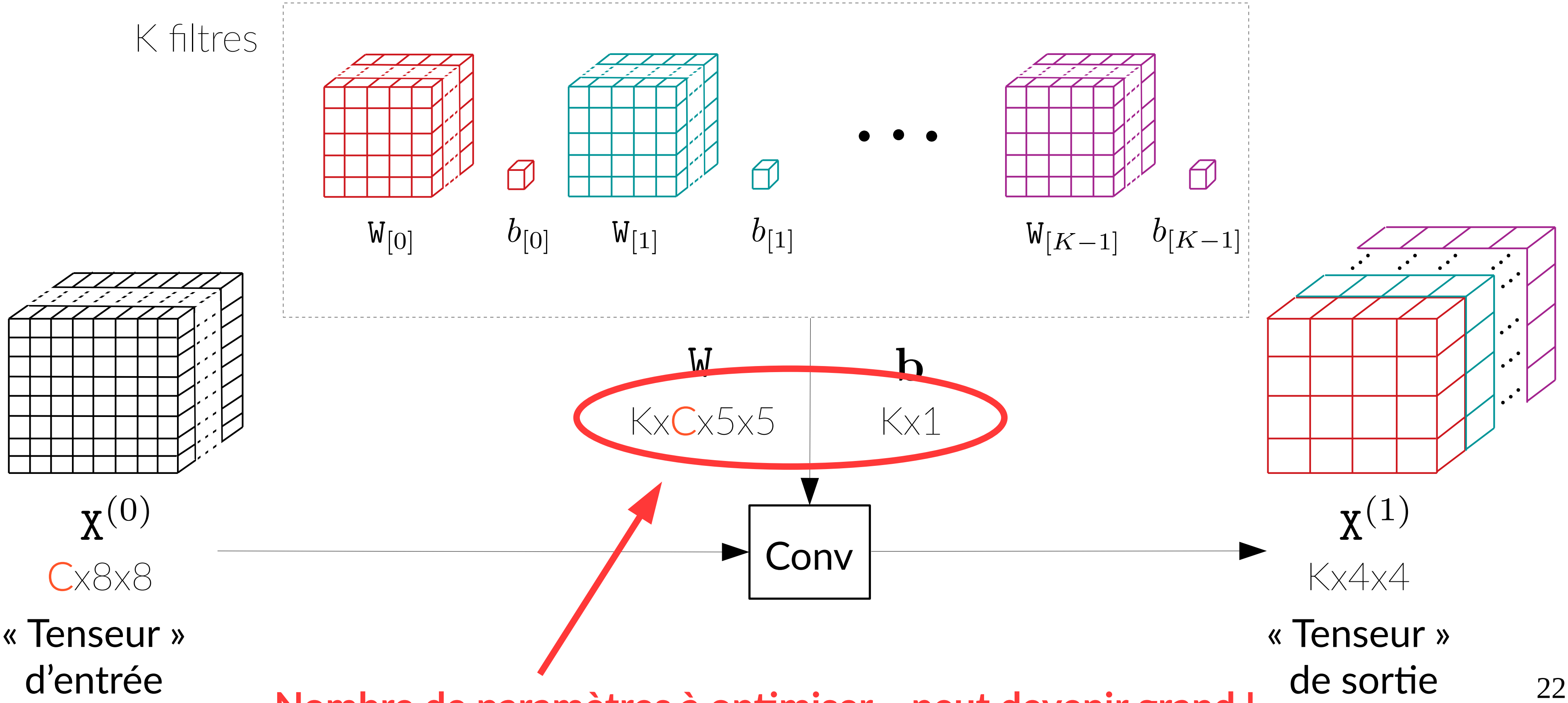
Couche de convolution 2D : K filtres



Nombre de paramètres à optimiser

I)

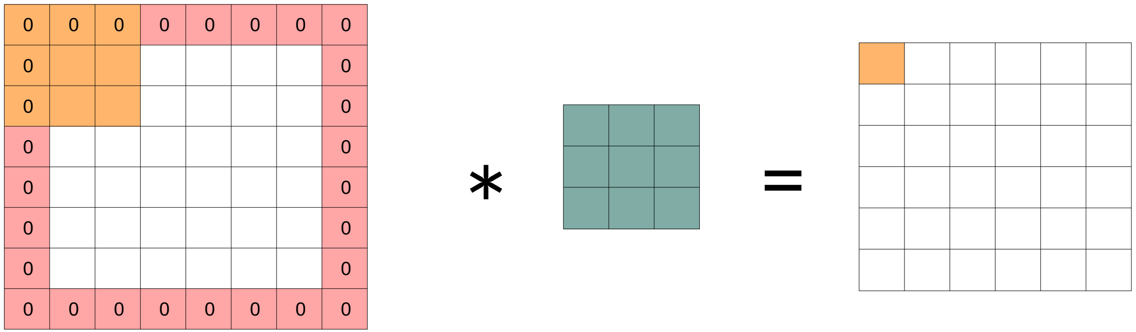
Couche de convolution 2D : K filtres



Nombre de paramètres à optimiser… peut devenir grand !

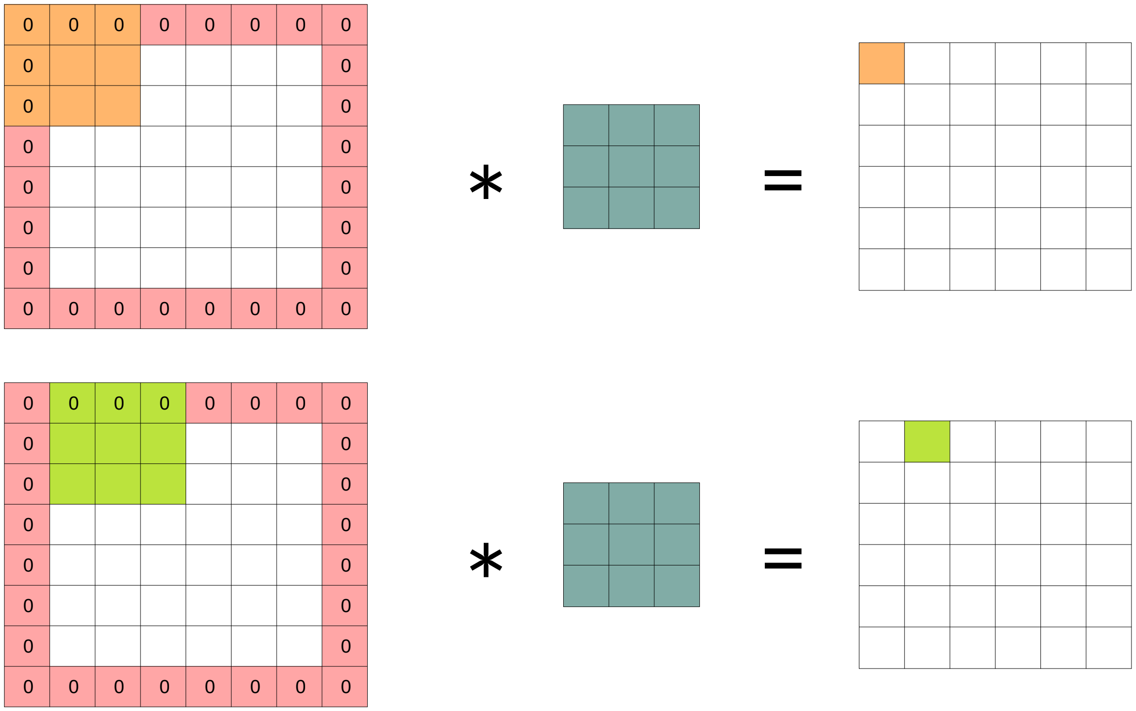
I)

« Zero padding »



I)

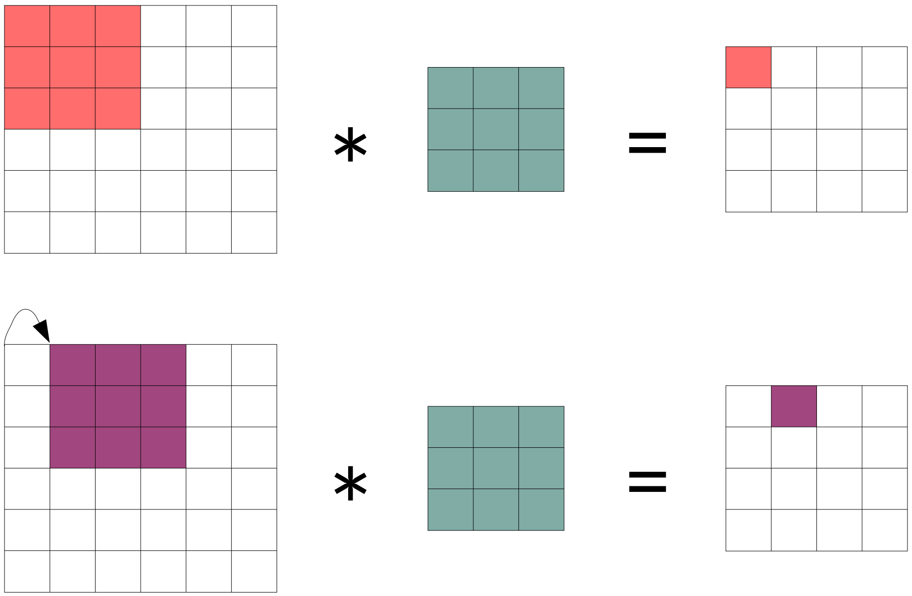
« Zero padding »



Permet de préserver la taille du tenseur d'entrée

« Stride »

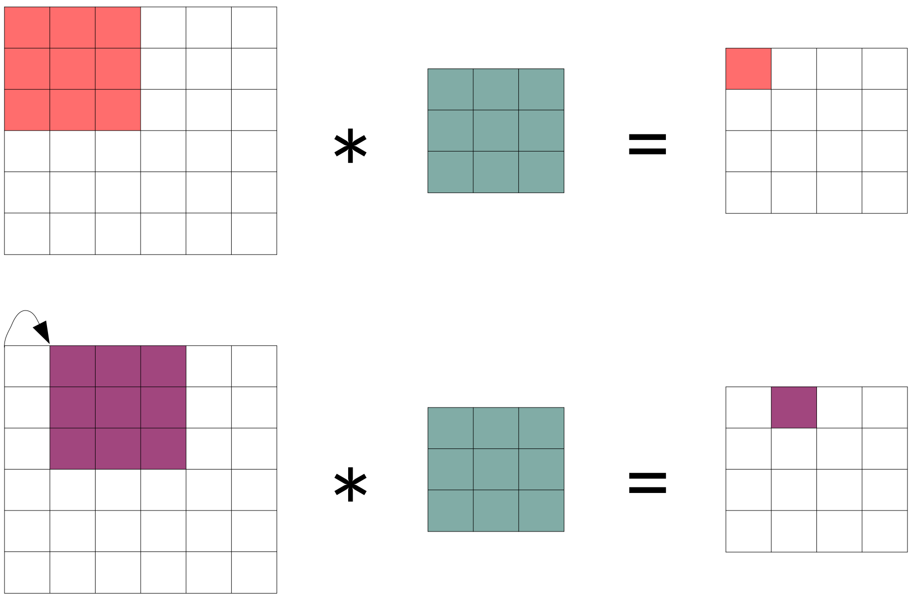
Exemple Stride = 1



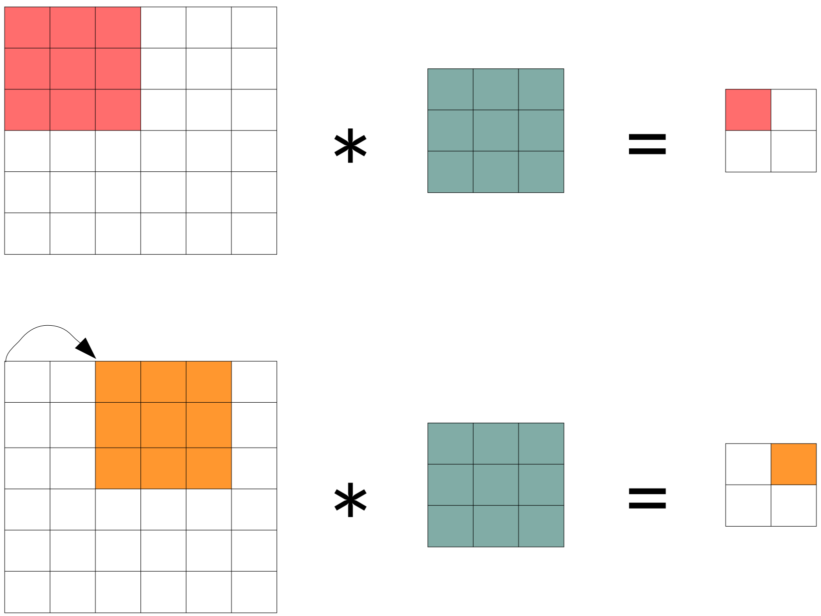
I)

« Stride »

Exemple Stride = 1



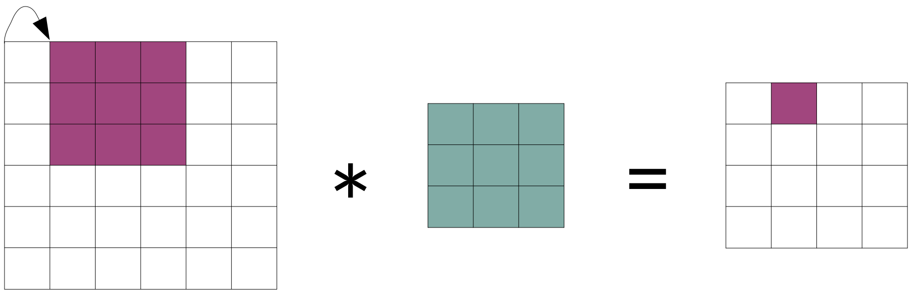
Exemple Stride = 2



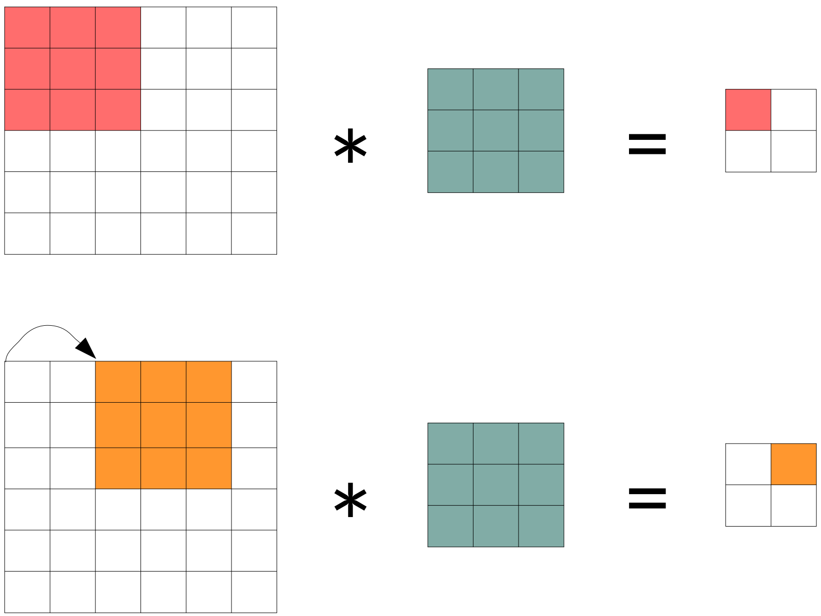
I)

« Stride »

Exemple Stride = 1



Exemple Stride = 2



Réduit la résolution de la sortie par « stride »

Principaux hyperparamètres d'une couche de convolution

- Taille des filtres
  - En pratique toujours impair
  - Souvent 1x1 ou 3x3, parfois 5x5 ou 7x7

Principaux hyperparamètres d'une couche de convolution

- Taille des filtres

  - En pratique toujours impair
  - Souvent 1x1 ou 3x3, parfois 5x5 ou 7x7

- Nombre de filtres

  = Nombre de canaux souhaité en sortie

Principaux hyperparamètres d’une couche de convolution

- Taille des filtres

  - En pratique toujours impair
  - Souvent 1x1 ou 3x3, parfois 5x5 ou 7x7

- Nombre de filtres

  = Nombre de canaux souhaité en sortie

- Quantité de zero-padding

  - Compense la taille du filtre si volonté de préserver la taille de l’entrée
  - 1x1 → padding = 0, 3x3 → padding = 1, 5x5 → padding = 2

Principaux hyperparamètres d'une couche de convolution

- Taille des filtres

  - En pratique toujours impair
  - Souvent 1x1 ou 3x3, parfois 5x5 ou 7x7

- Nombre de filtres

  = Nombre de canaux souhaité en sortie

- Quantité de zero-padding

  - Compense la taille du filtre si volonté de préserver la taille de l'entrée
  - 1x1 → padding = 0, 3x3 → padding = 1, 5x5 → padding = 2

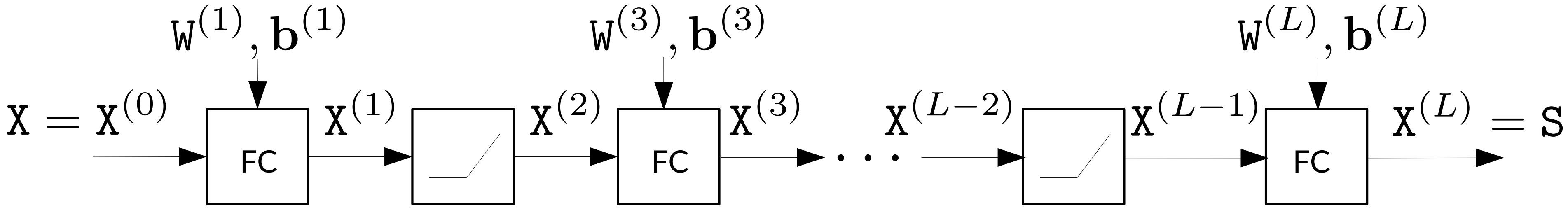
- Stride

  = 1 si volonté de préserver la résolution de l'entrée
  = 2 si volonté de réduire la résolution de l'entrée

II) Réseau de neurones à convolution

II)

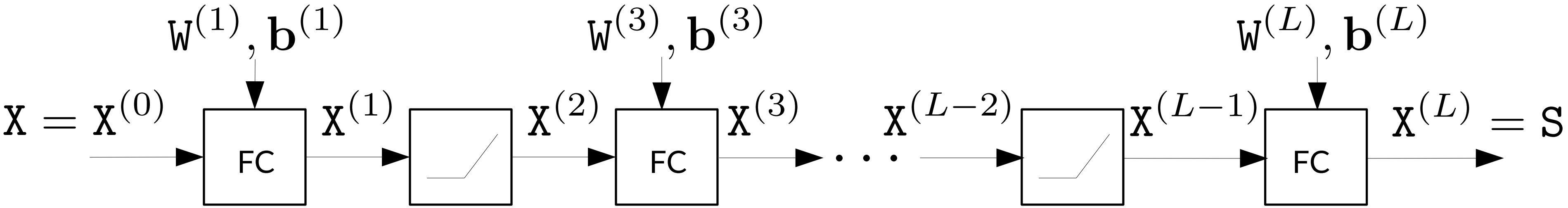
Réseau de neurones à convolution (CNN)



MLP

II)

Réseau de neurones à convolution (CNN)



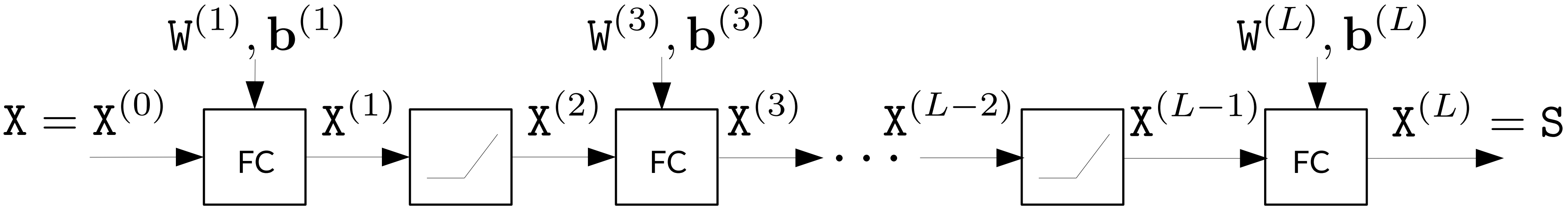
MLP

− FC (transformations affines générales)

+ Conv (transformations affines spécifiques)

= CNN

Réseau de neurones à convolution (CNN)



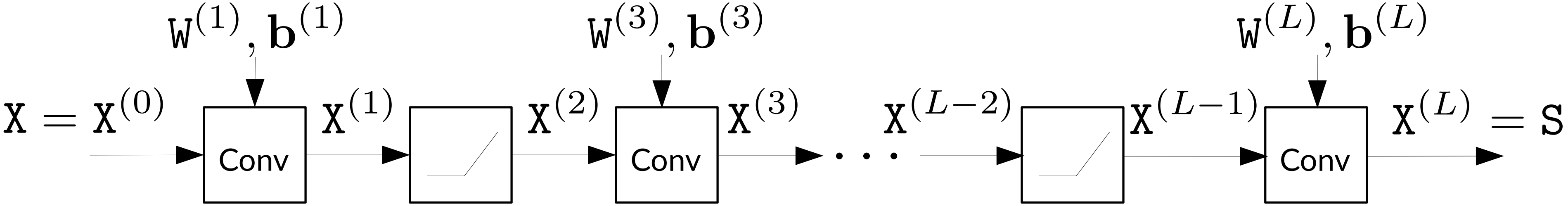
MLP

− FC (transformations affines générales)

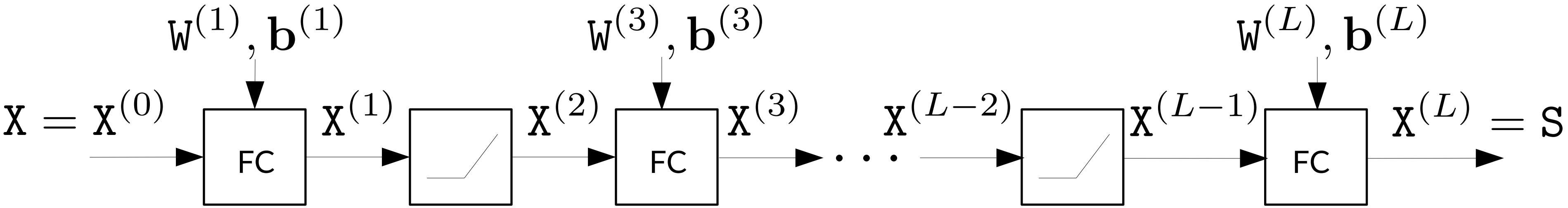
+ Conv (transformations affines spécifiques)

=

CNN



Réseau de neurones à convolution (CNN)

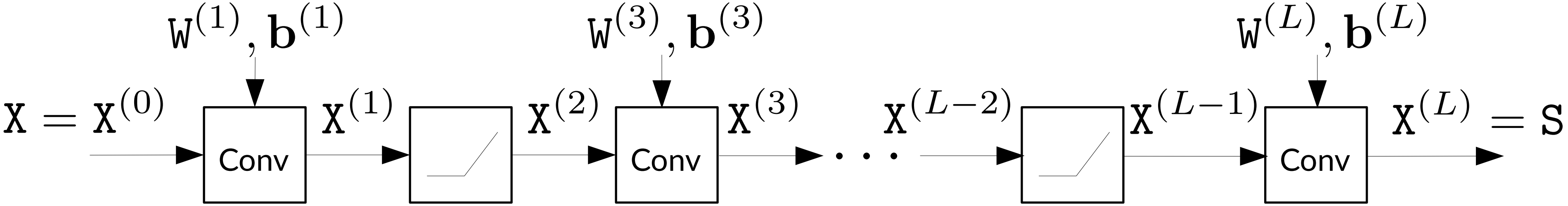


MLP

− FC (transformations affines générales)

+ Conv (transformations affines spécifiques)

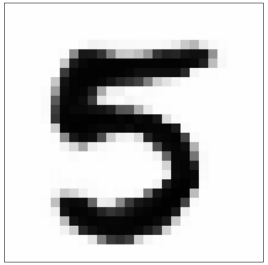
= CNN

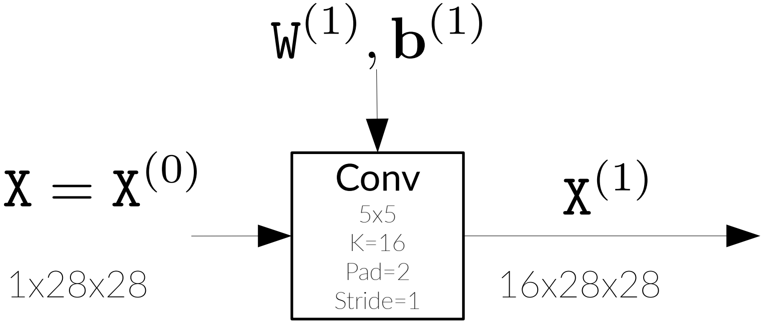


→ Initialisation des paramètres d’une couche de convolution identique à ceux d’une FC !

II)

Exemple d’architecture de CNN pour MNIST

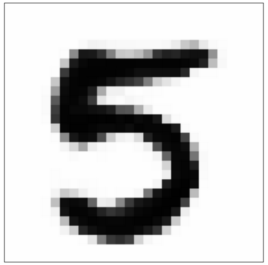


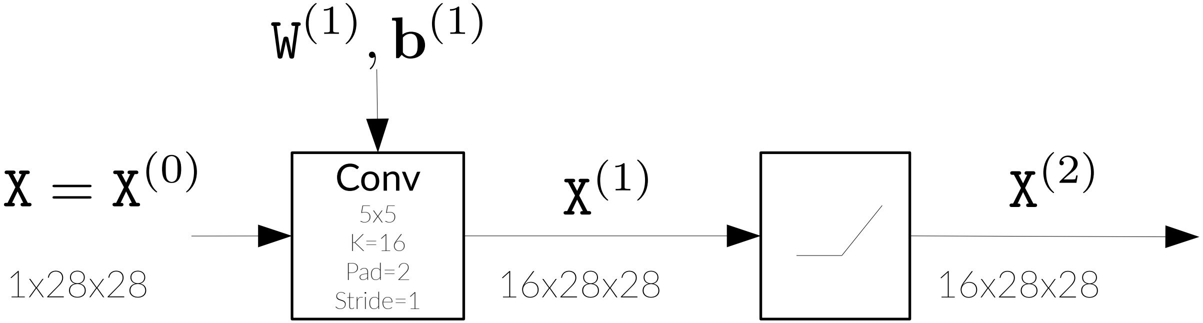


“0”
“1”
“2”
“3”
“4”
“5”
“6”
“7”
“8”
“9”

II)

Exemple d’architecture de CNN pour MNIST

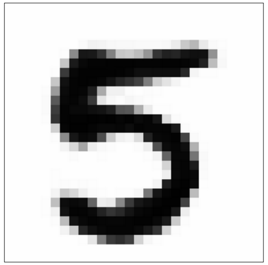


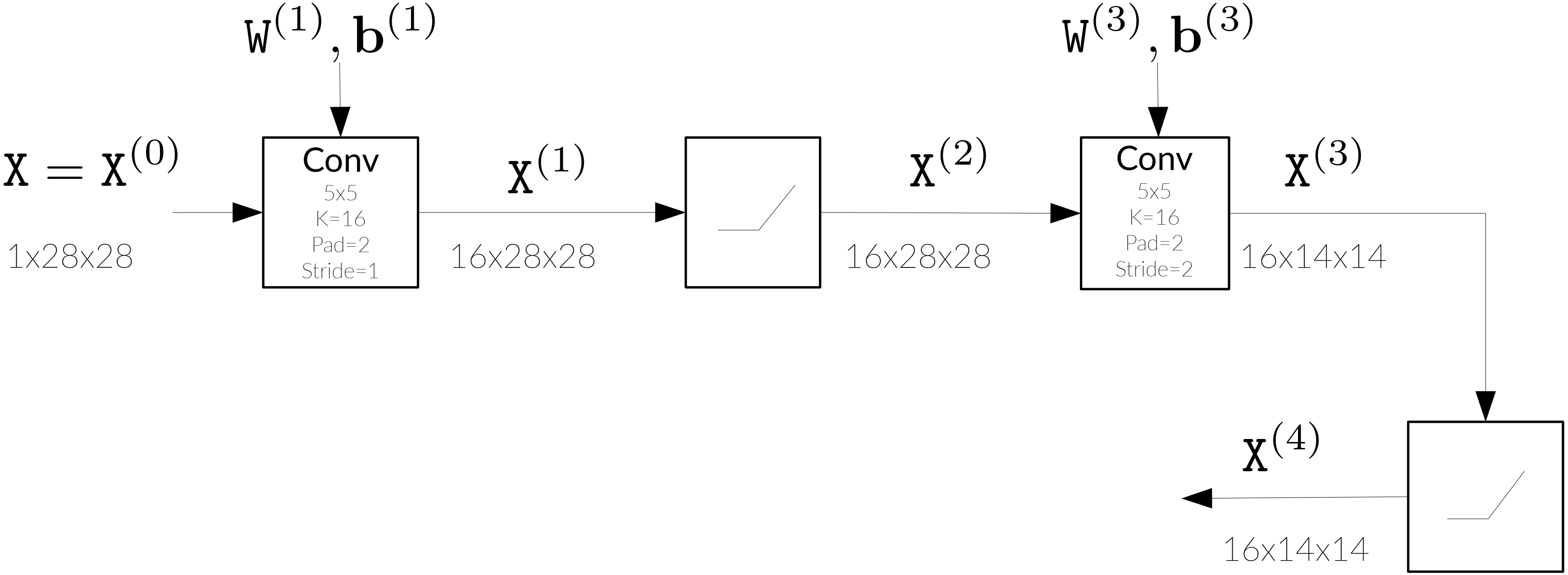


“0”
“1”
“2”
“3”
“4”
“5”
“6”
“7”
“8”
“9”

II)

Exemple d’architecture de CNN pour MNIST

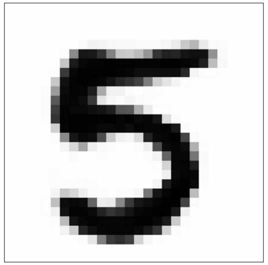


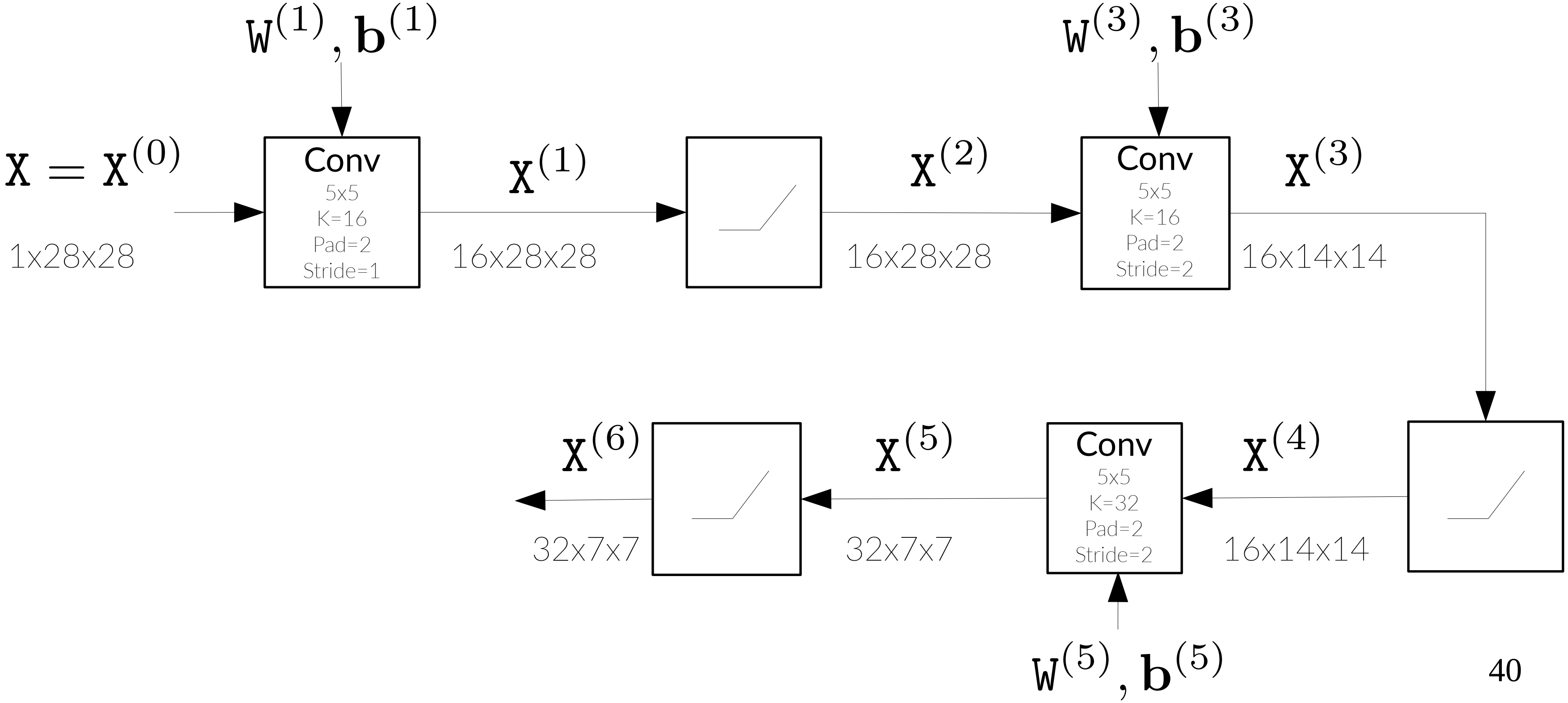


“0”
“1”
“2”
“3”
“4”
“5”
“6”
“7”
“8”
“9”

II)

Exemple d'architecture de CNN pour MNIST

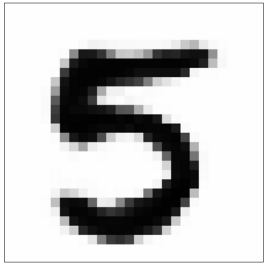


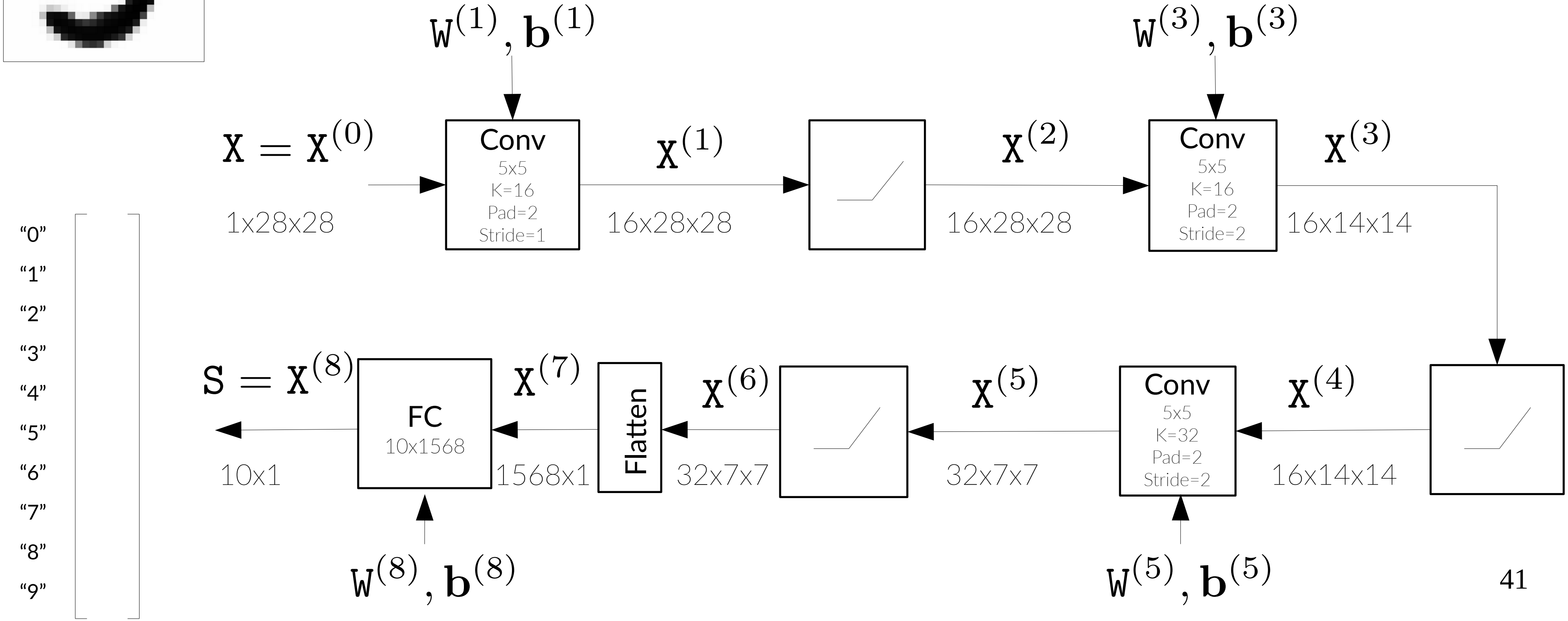


"0"
"1"
"2"
"3"
"4"
"5"
"6"
"7"
"8"
"9"

II)

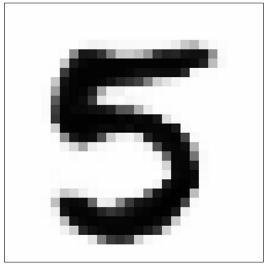
Exemple d'architecture de CNN pour MNIST

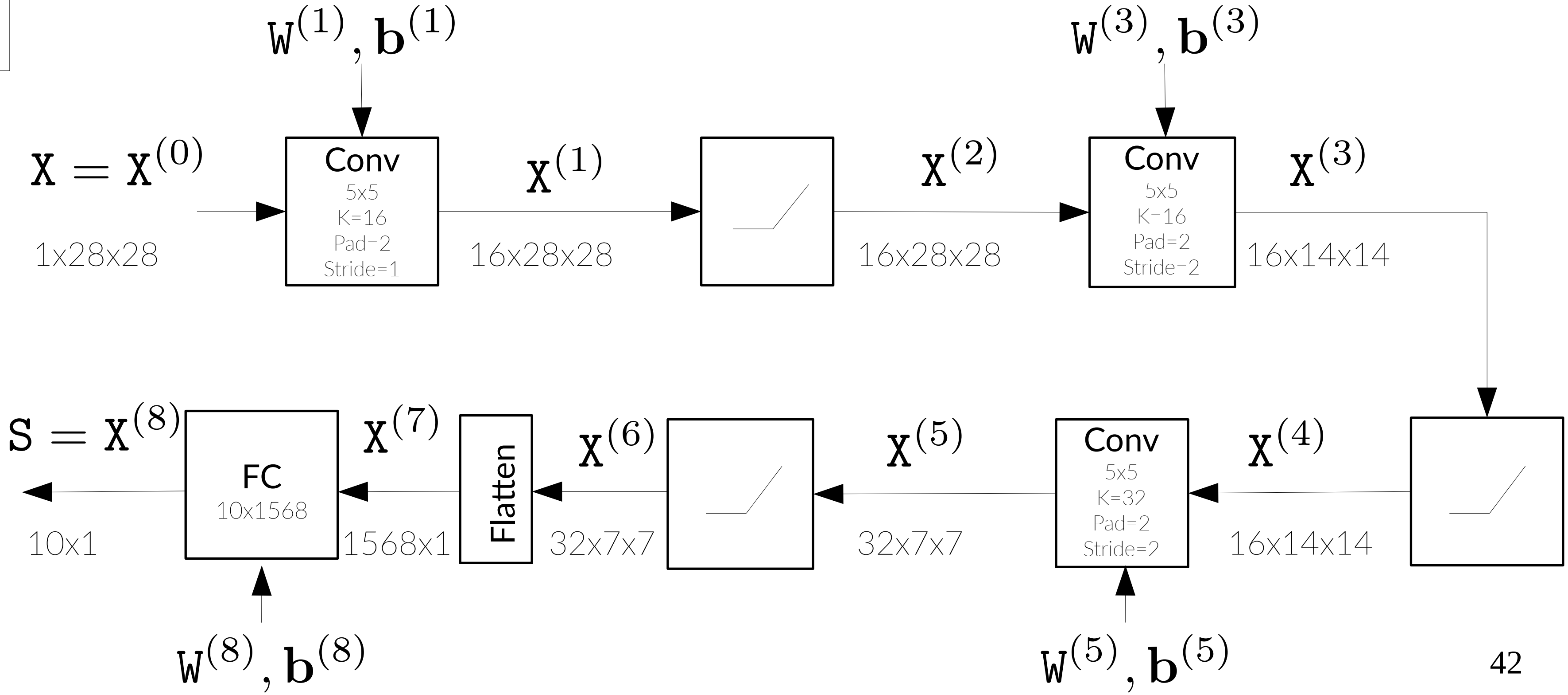




II)

Exemple d'architecture de CNN pour MNIST

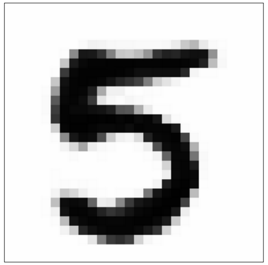




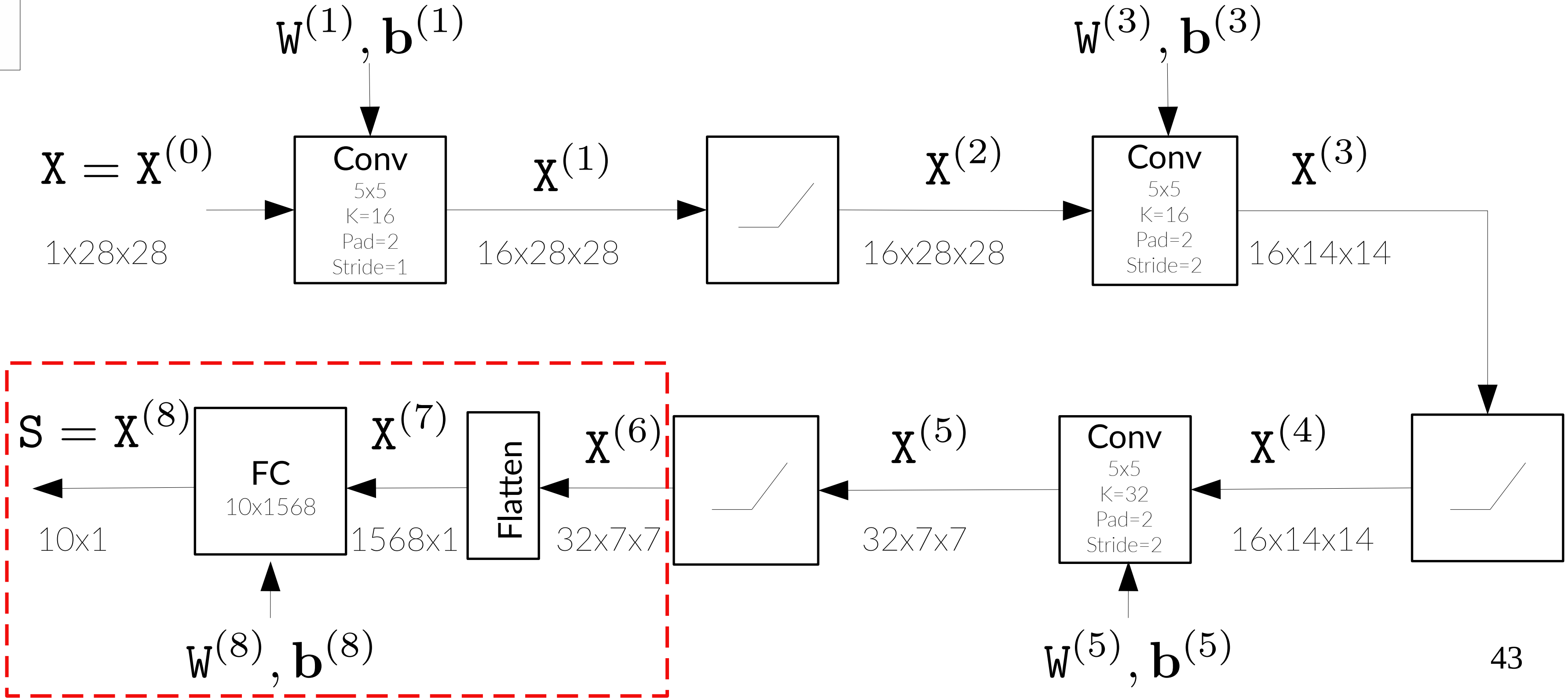
"0"	-13.1
"1"	-4.1
"2"	-2.4
"3"	1.8
"4"	6.8
"5"	1.7
"6"	-3.4
"7"	2.2
"8"	-1.1
"9"	-2.3

II)

Exemple d'architecture de CNN pour MNIST

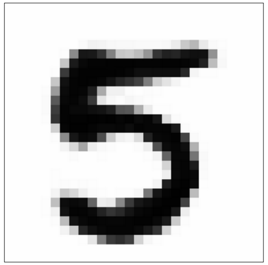


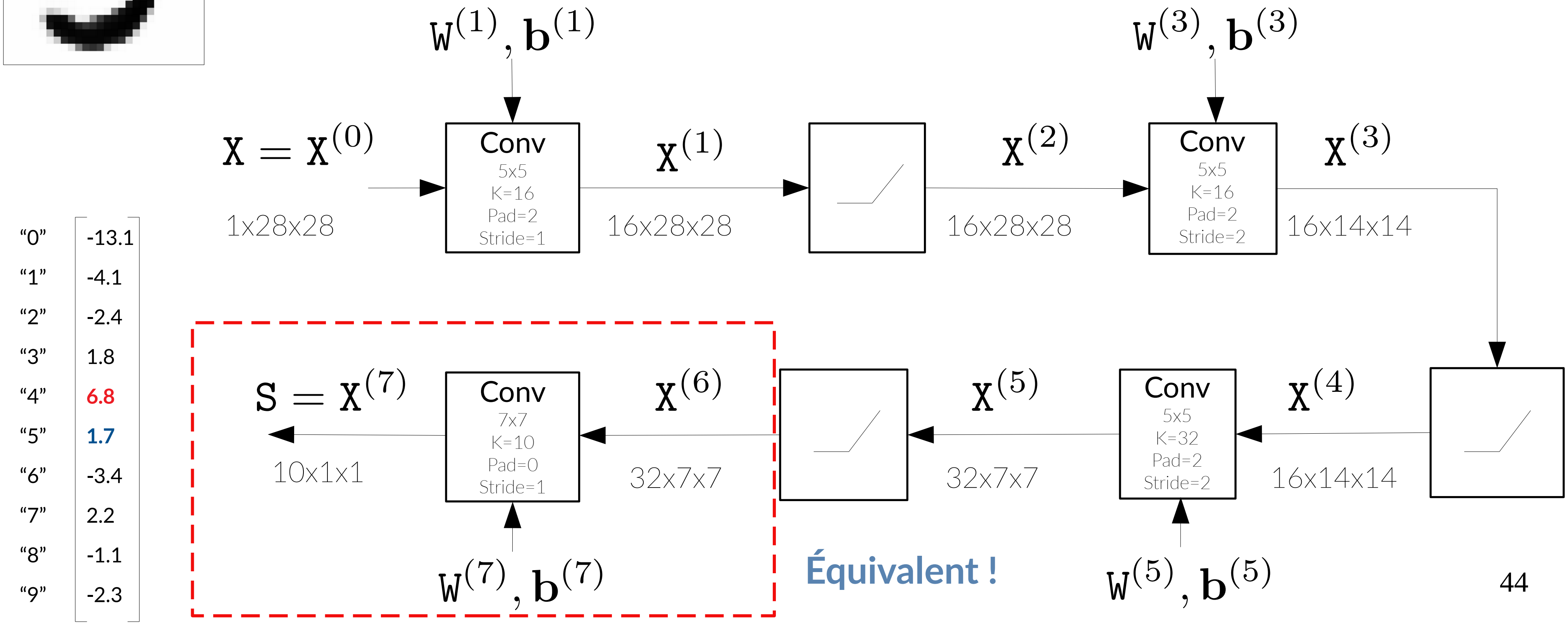
"0"	-13.1
"1"	-4.1
"2"	-2.4
"3"	1.8
"4"	6.8
"5"	1.7
"6"	-3.4
"7"	2.2
"8"	-1.1
"9"	-2.3



II)

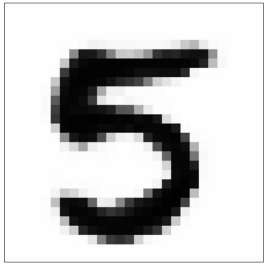
Exemple d'architecture de CNN pour MNIST

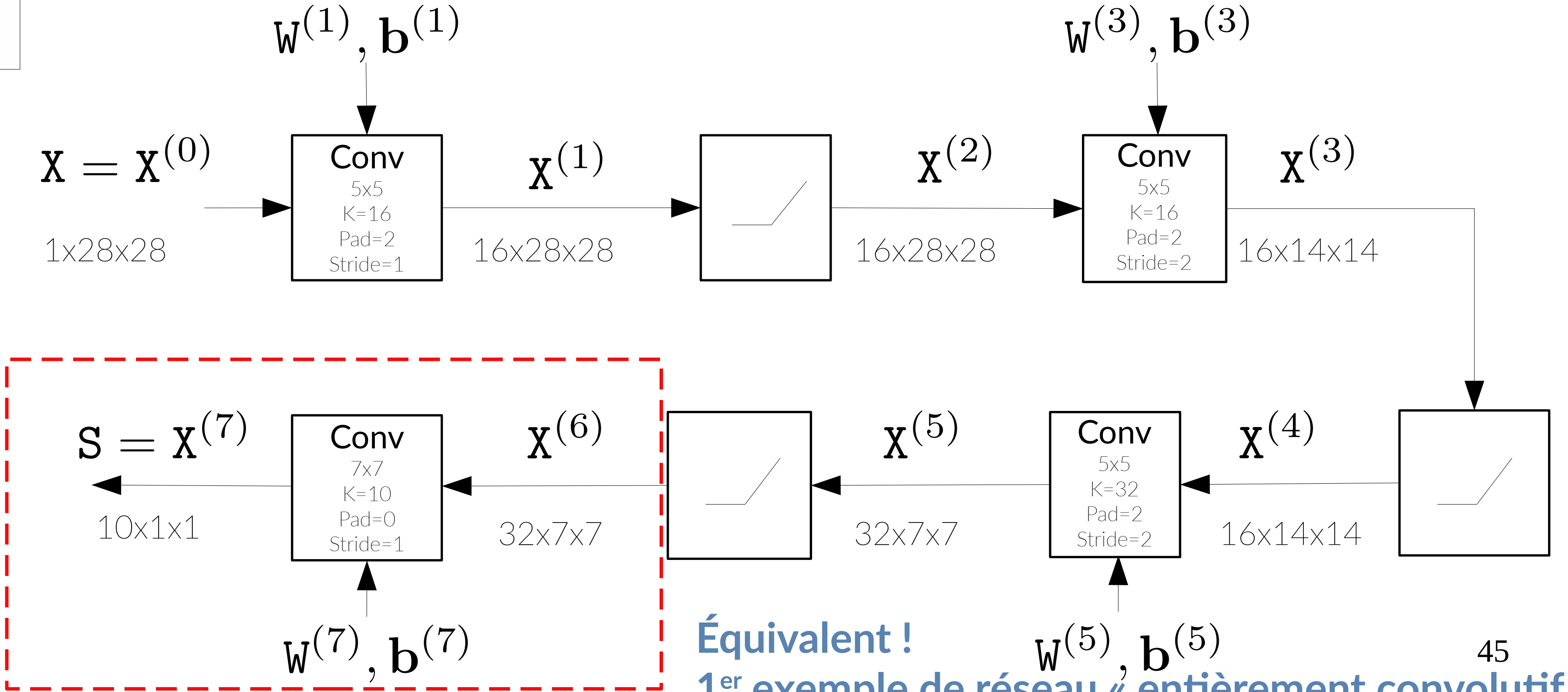




II)

Exemple d’architecture de CNN pour MNIST



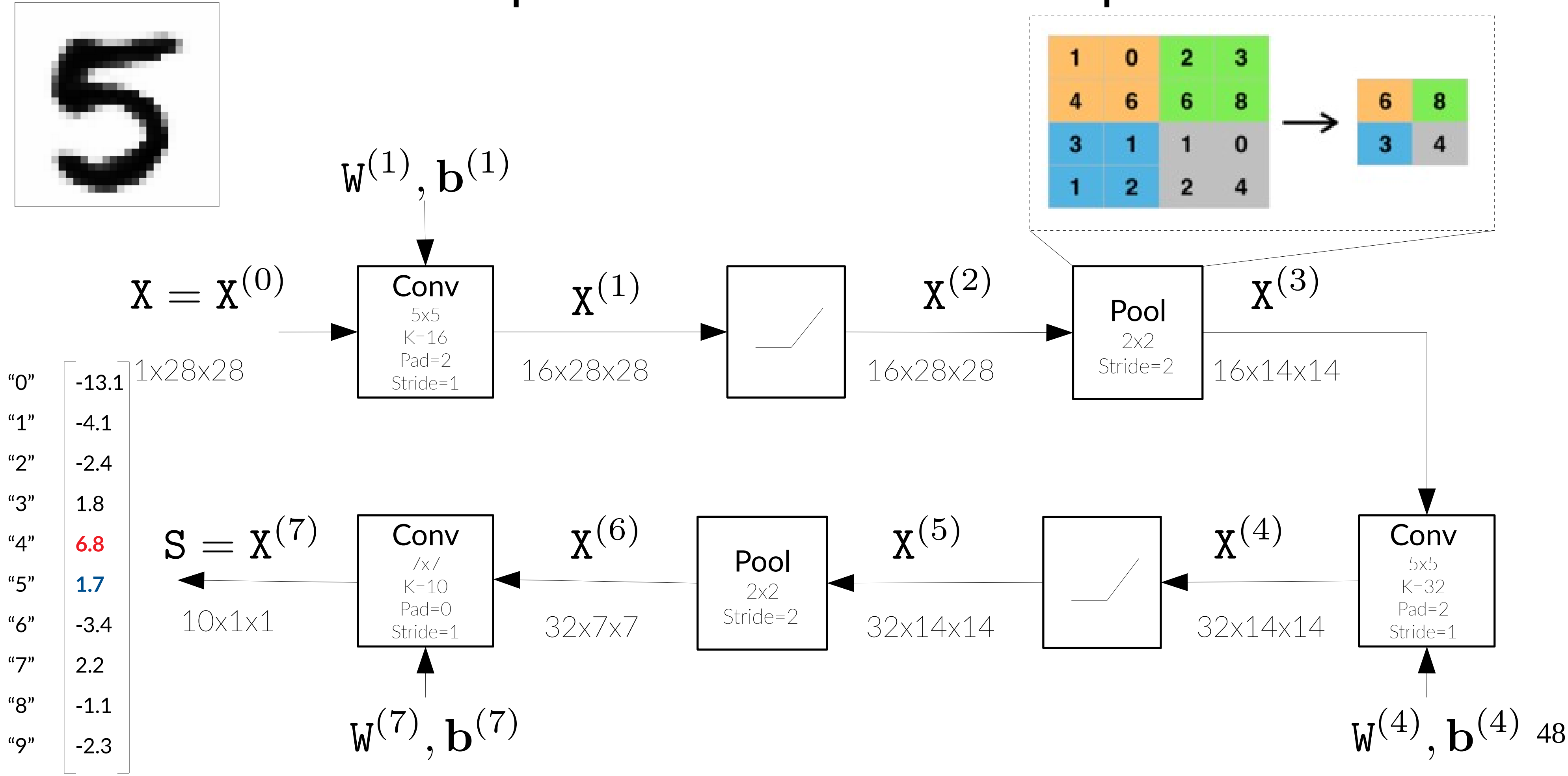


“0”	-13.1
“1”	-4.1
“2”	-2.4
“3”	1.8
“4”	6.8
“5”	1.7
“6”	-3.4
“7”	2.2
“8”	-1.1
“9”	-2.3

Équivalent !
1er exemple de réseau « entièrement convolutif »

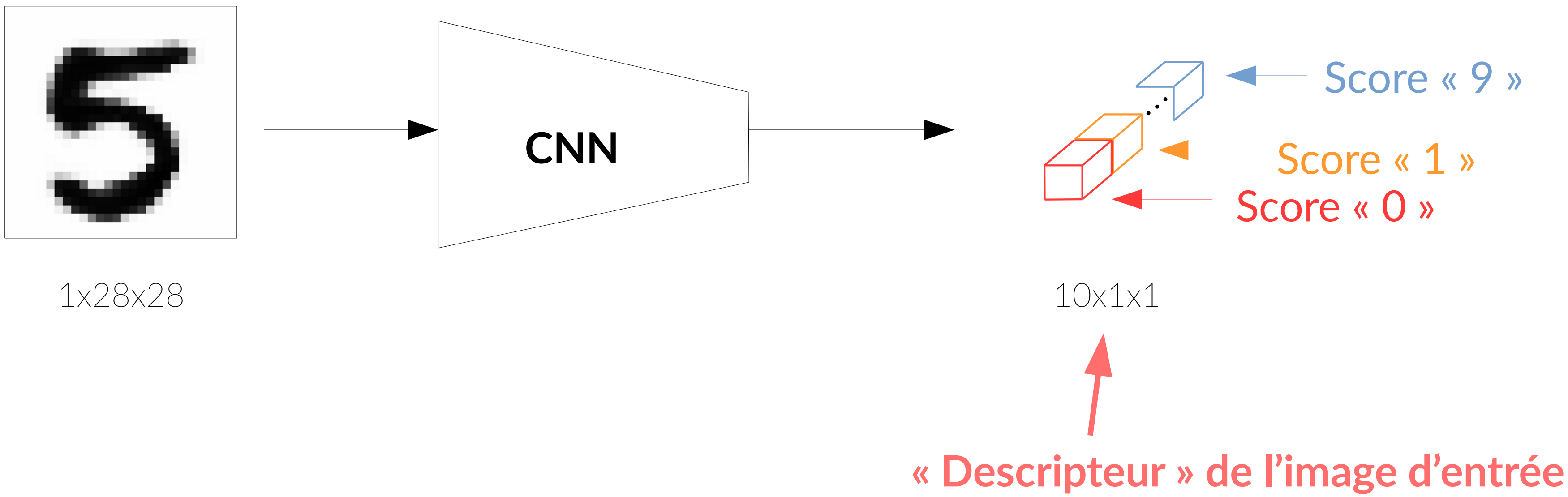
II)

Autre exemple d'architecture : CNN pour MNIST



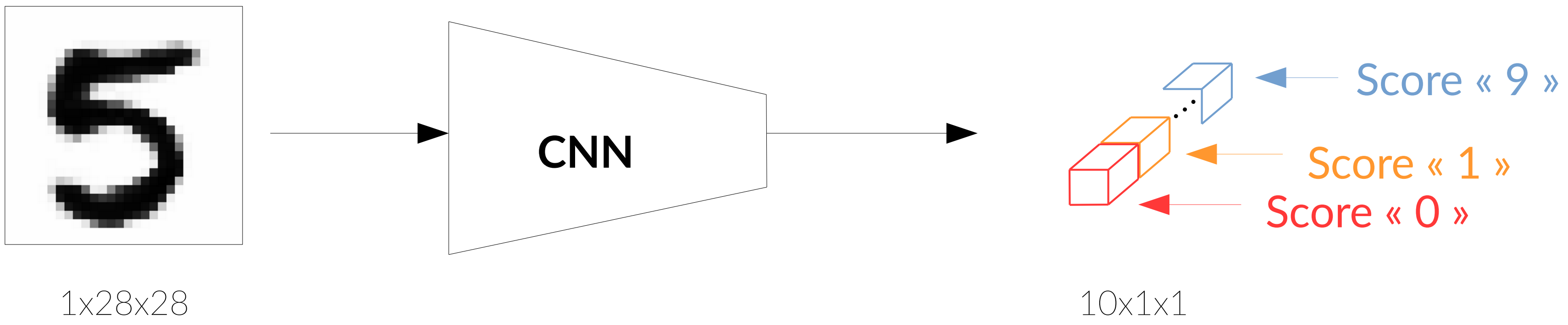
Architectures de CNN : Deux cas extrêmes

Cas 1 : Extraire une information **globale** présente dans l'image d'entrée



Architectures de CNN : Deux cas extrêmes

Cas 1 : Extraire une information **globale** présente dans l'image d'entrée



Réduction **progressive** de la résolution

↓

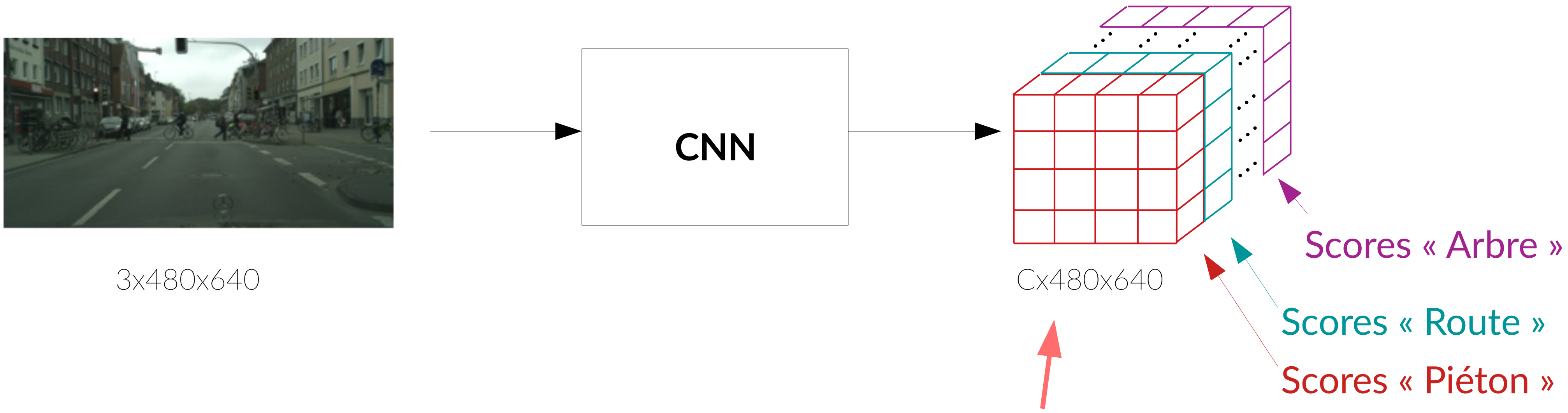
Utilisation de couches de conv ou pooling avec stride = 2

II)

Architectures de CNN : Deux cas extrêmes

Cas 1 : Extraire une information **globale** présente dans l'image d'entrée

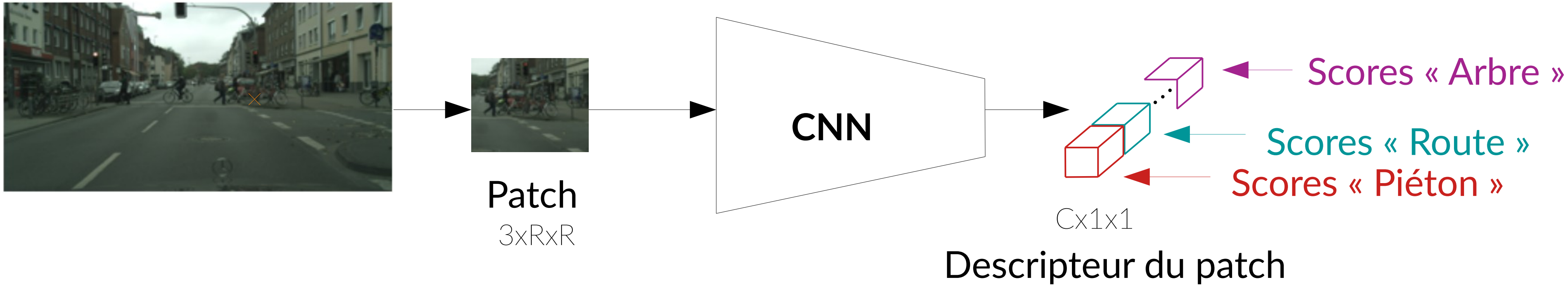
Cas 2 : Extraire une information **pour chaque pixel** de l'image d'entrée



Un « descripteur » par pixel de l'image d'entrée

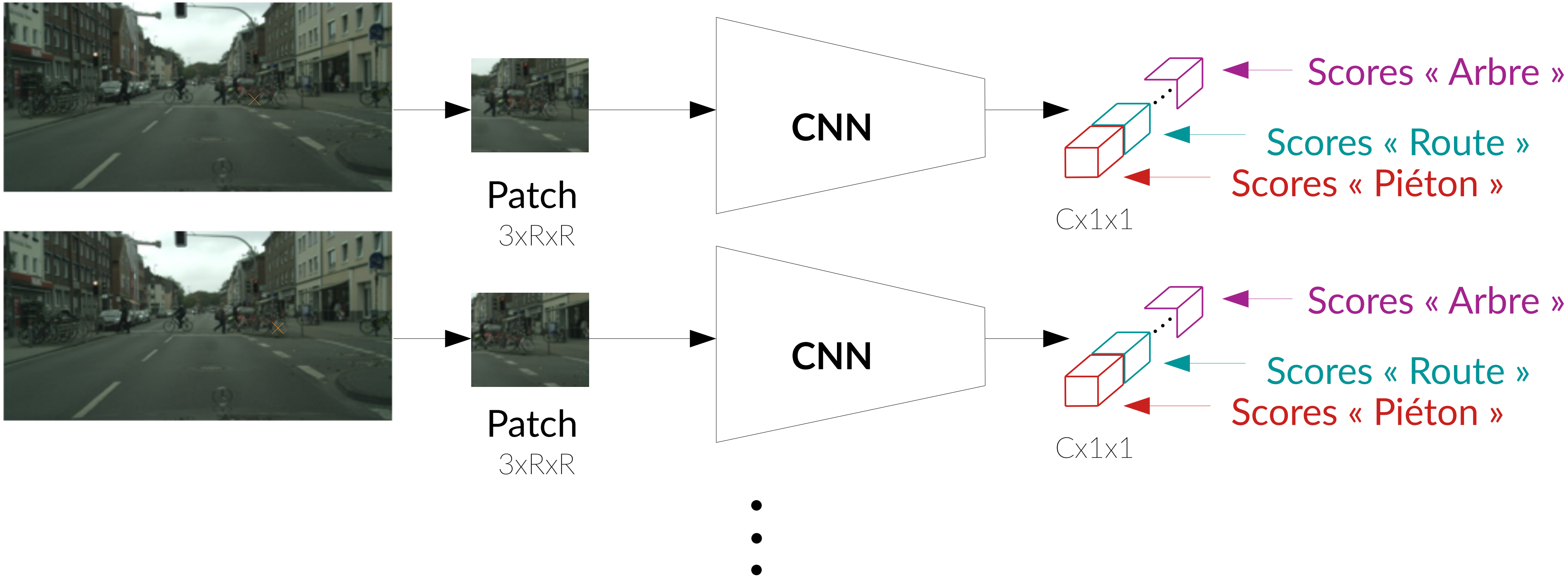
II)

Comment obtenir un descripteur pour chaque pixel ?

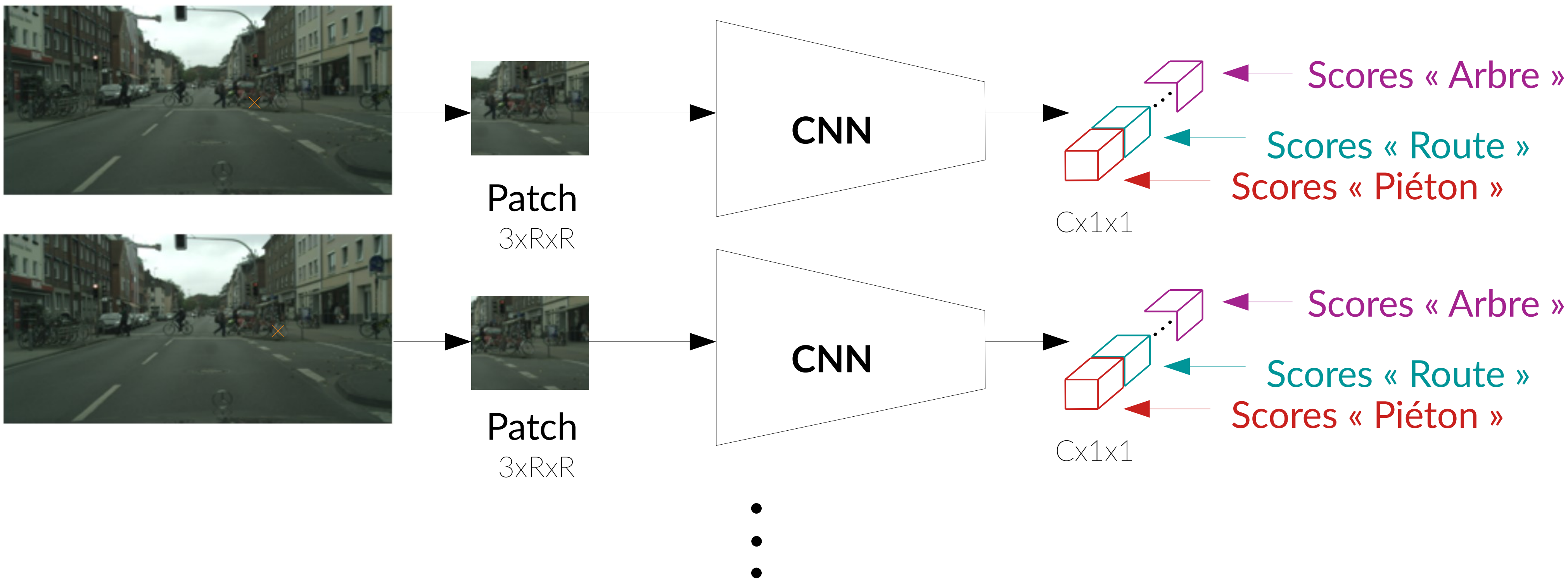


II)

Comment obtenir un descripteur pour chaque pixel ?



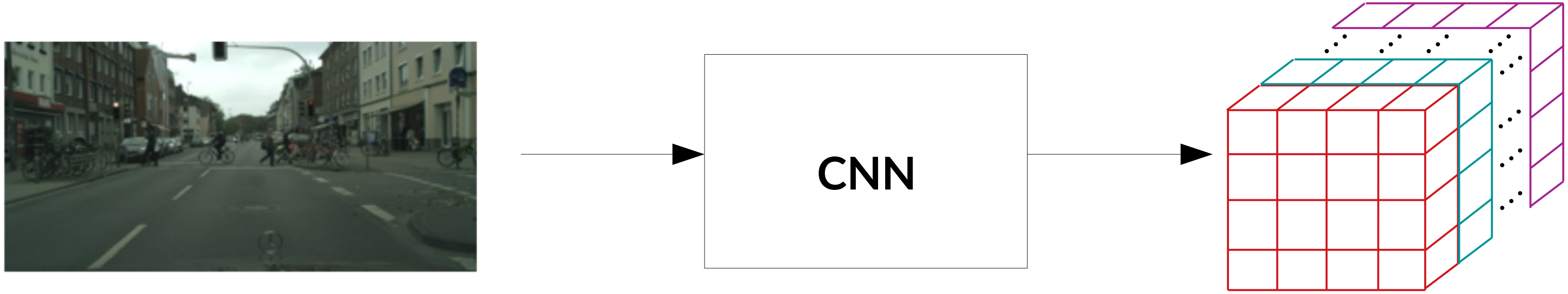
Comment obtenir un descripteur pour chaque pixel ?



Les patchs voisins ont beaucoup de pixels en commun → calculs redondants

→ Utilisation d'une architecture entièrement convolutive

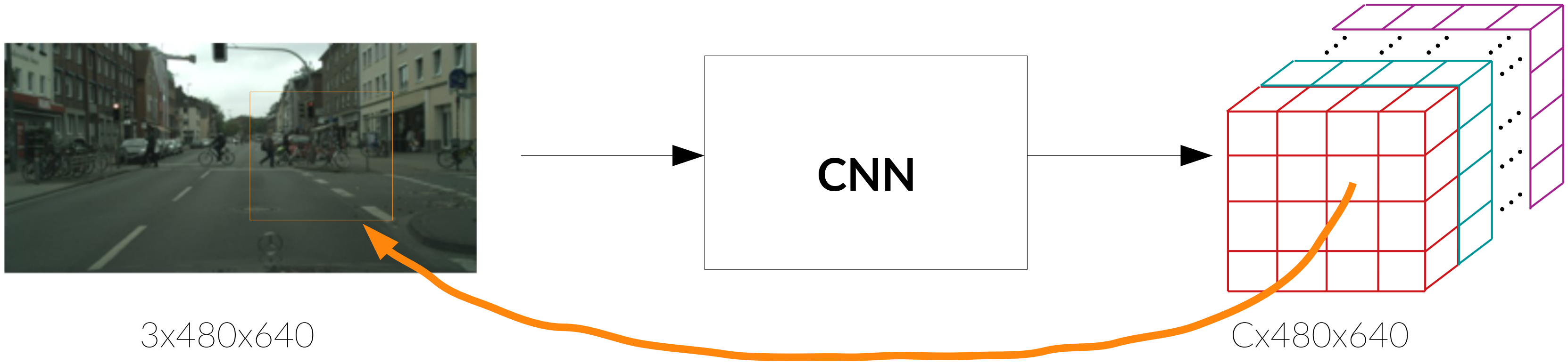
Architecture entièrement convolutive



3x480x640

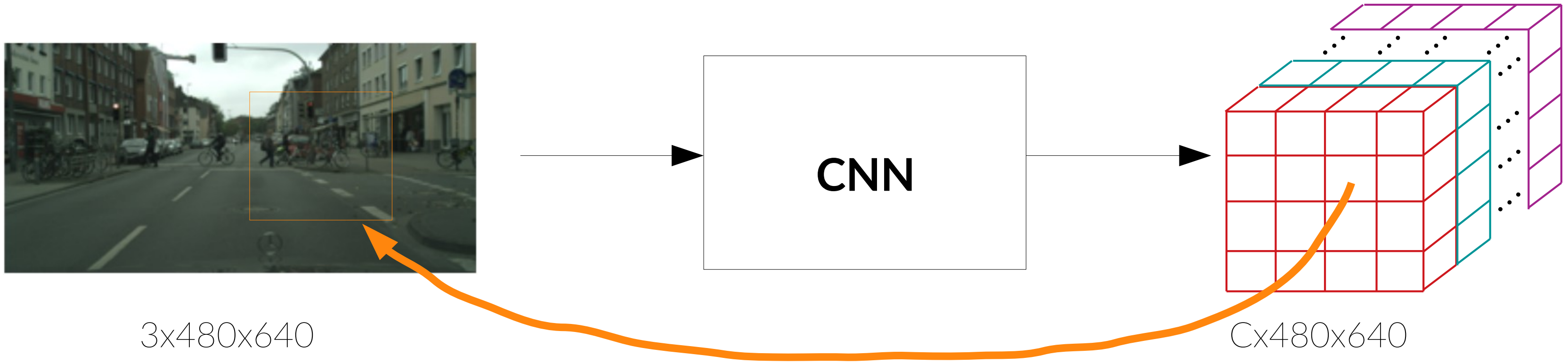
Cx480x640

Architecture entièrement convolutive



Descripteur Cx1x1 d’un patch de taille RxR,
où RxR s’appelle le Champ Récepteur du CNN (« Receptive Field »)

Architecture entièrement convolutive

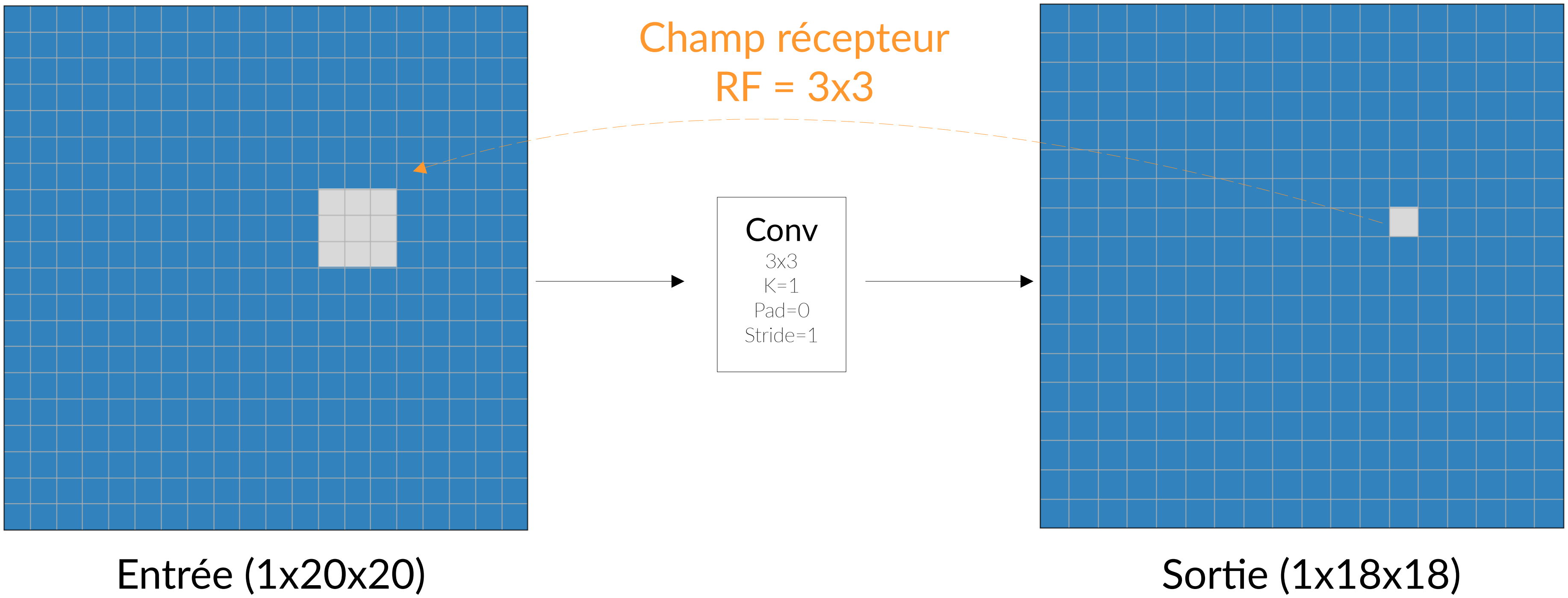


Descripteur Cx1x1 d'un patch de taille RxR,
où RxR s'appelle le Champ Récepteur du CNN (« Receptive Field »)

Comment avoir un grand champ récepteur tout en restant raisonnable en mémoire et temps de calculs ?

II)

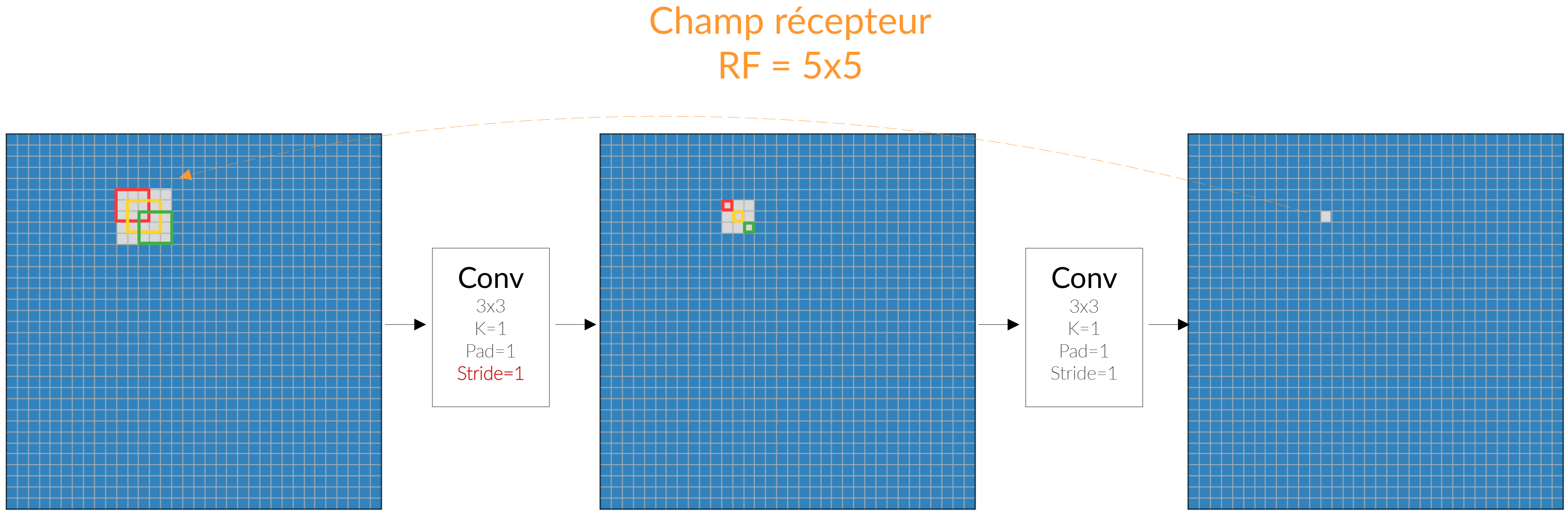
Champ récepteur d’une couche de convolution



Pour une couche → champ récepteur = taille du filtre

II)

Champ récepteur de deux couches de convolution
Stride = 1



$\mathtt{X}^{(0)}$ (1x34x34)

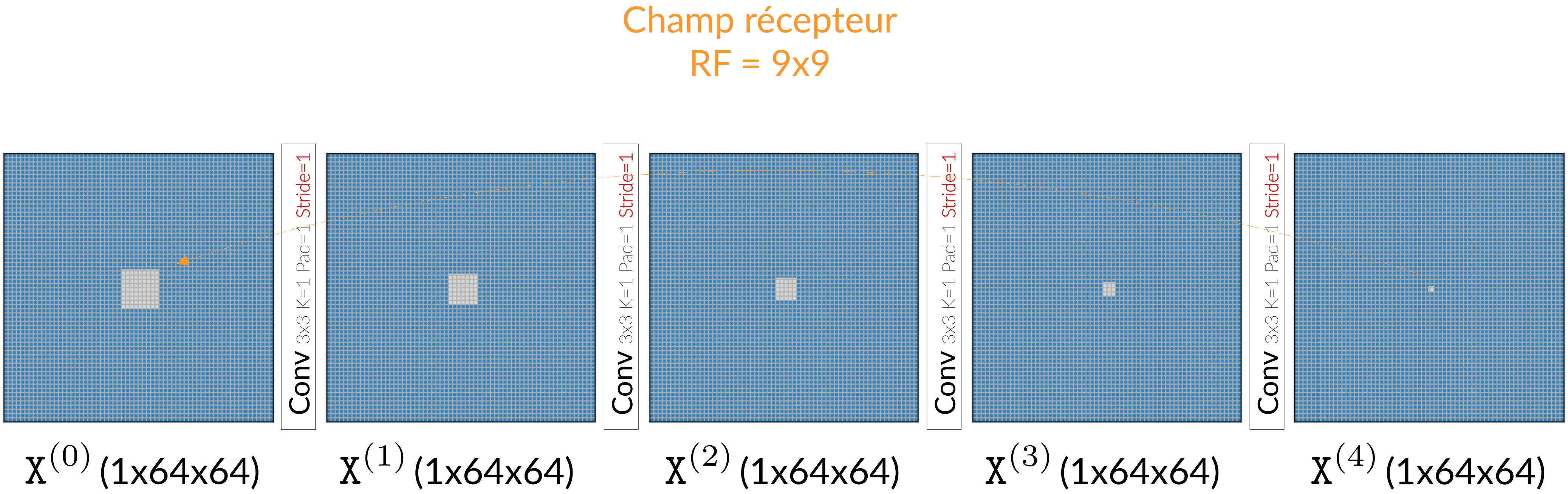
$\mathtt{X}^{(1)}$ (1x34x34)

$\mathtt{X}^{(2)}$ (1x34x34)

Remarque : les ReLU n'affectent pas le RF donc on ne les représente pas ici.

II)

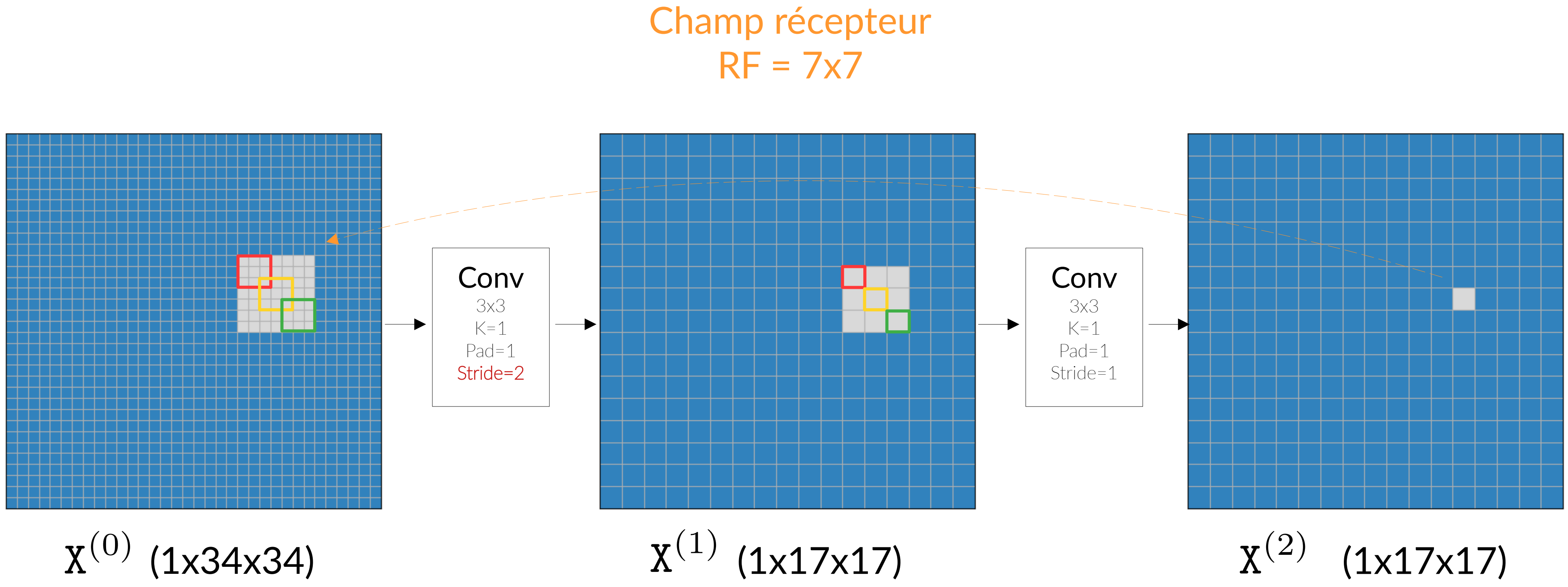
Champ récepteur de quatre couches de convolution
Stride = 1



→ Croissance du RF très lente ...

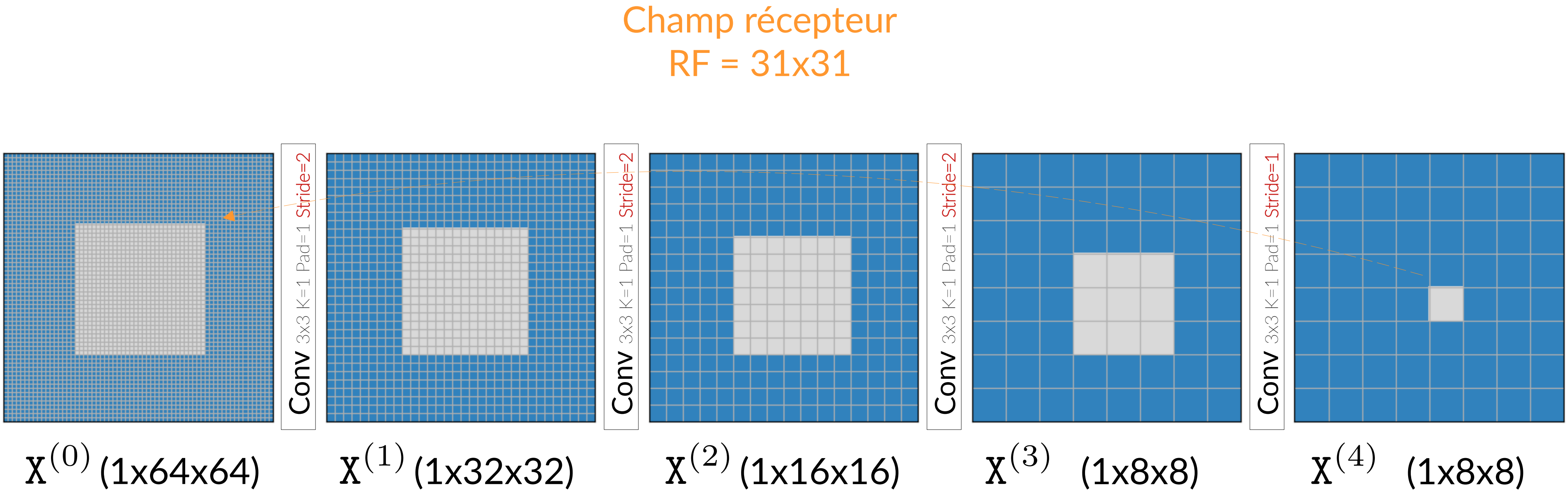
II)

Champ récepteur de deux couches de convolution
Stride = 2



II)

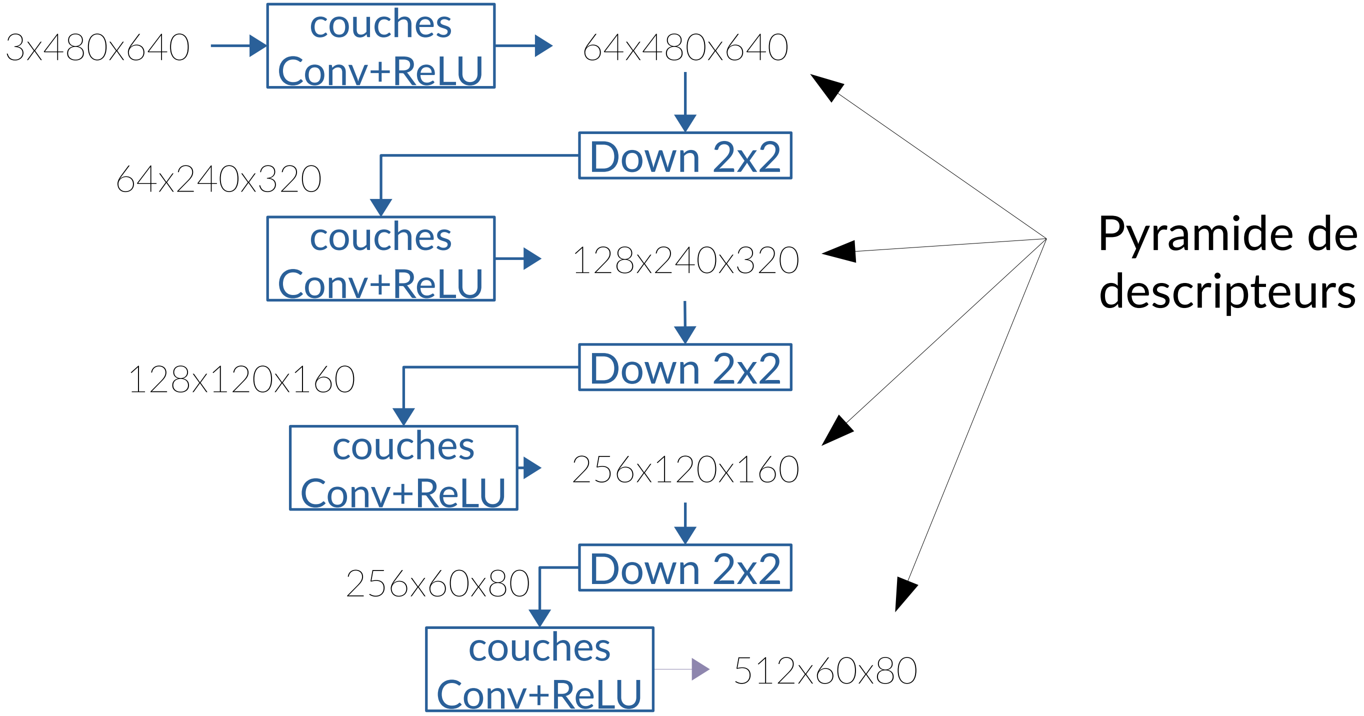
Champ récepteur de quatre couches de convolution
Stride = 2



→ Croissance rapide du RF … mais baisse de la résolution
→ Architecture U-Net

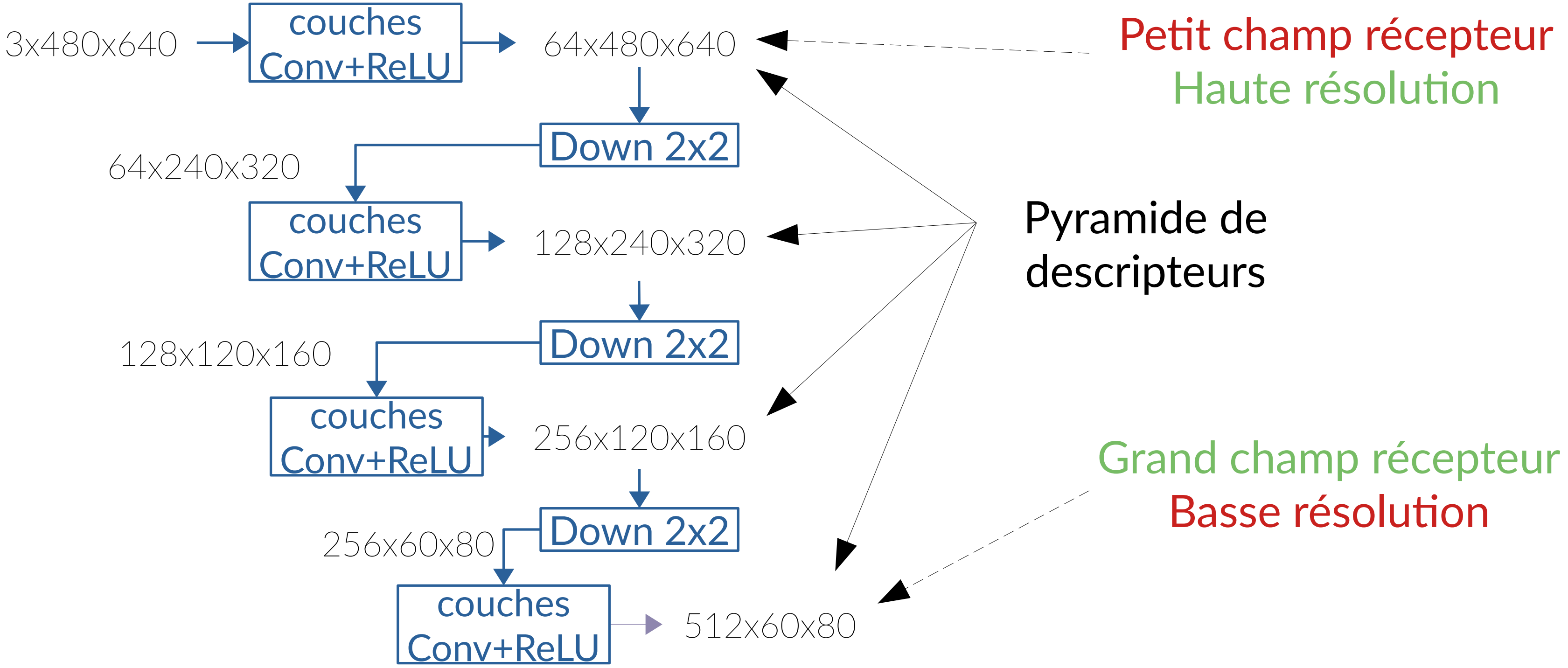
II)

U-Net, également appelé « Feature Pyramid Network » (FPN)

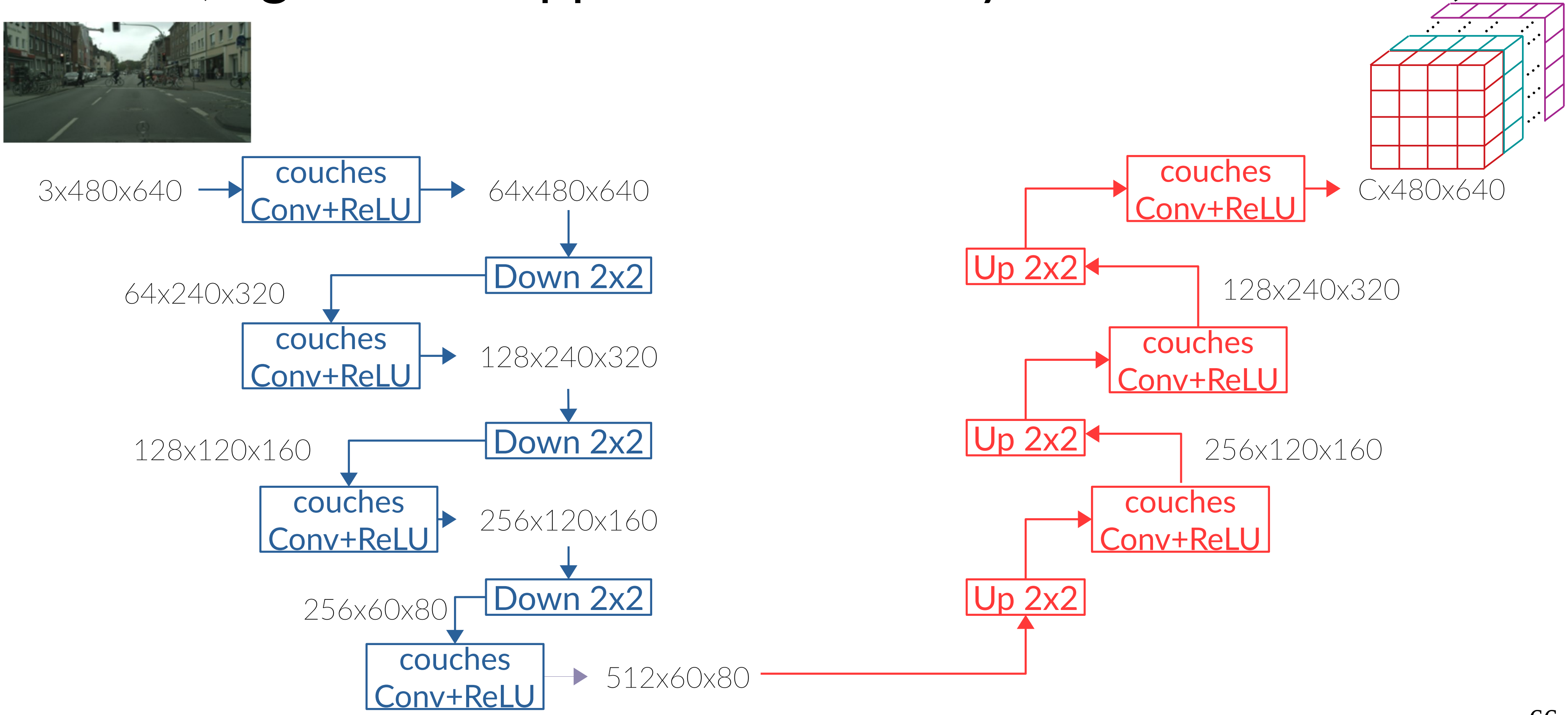


II)

U-Net, également appelé « Feature Pyramid Network » (FPN)

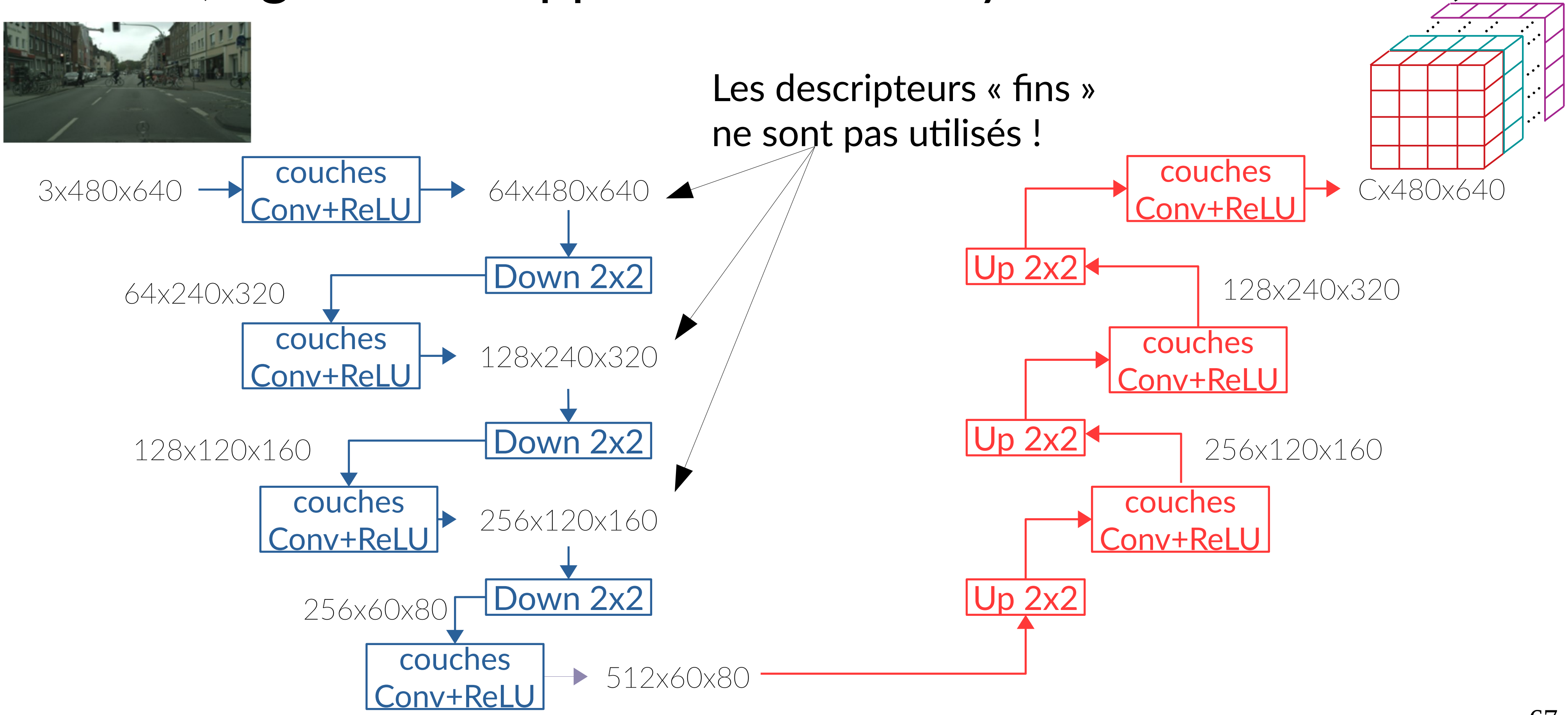


U-Net, également appelé « Feature Pyramid Network » (FPN)

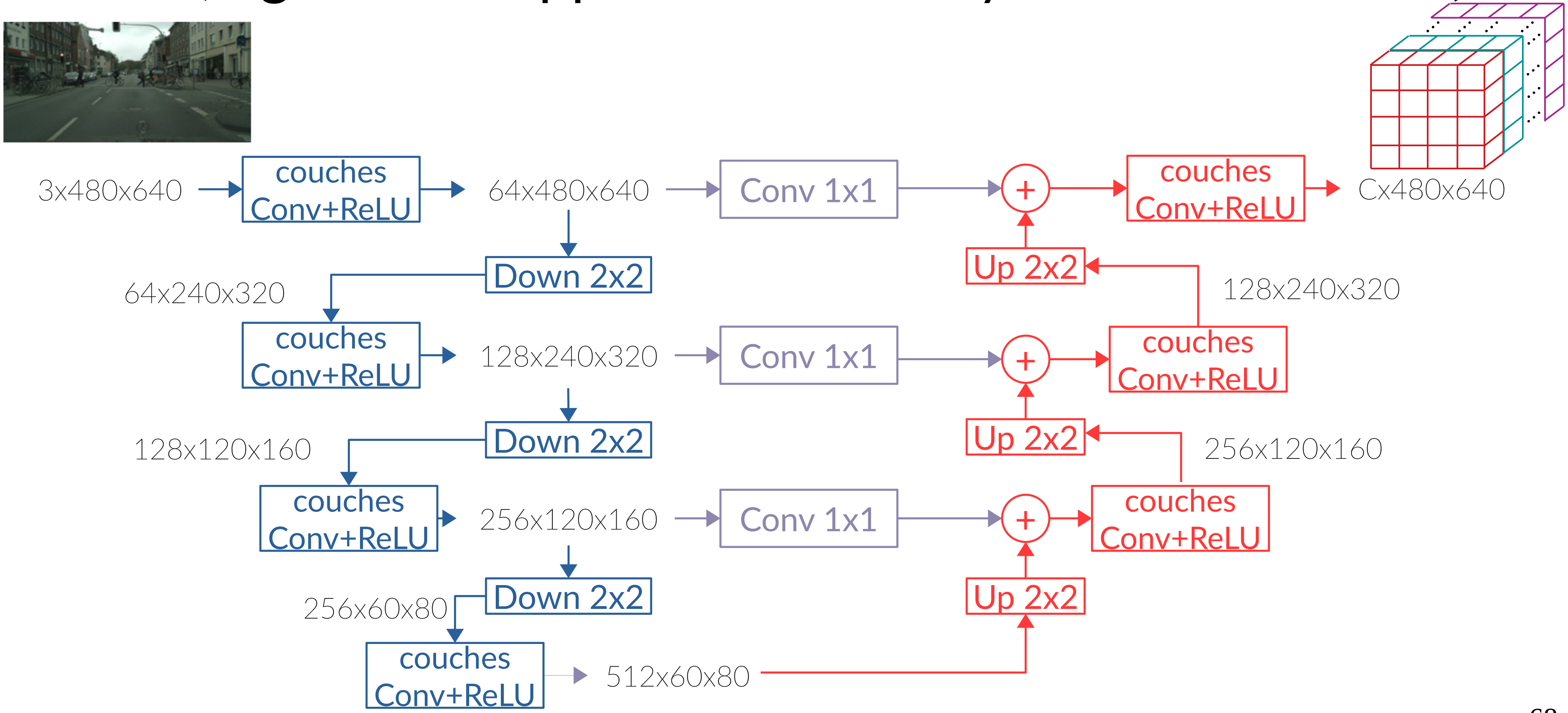


II)

U-Net, également appelé « Feature Pyramid Network » (FPN)



U-Net, également appelé « Feature Pyramid Network » (FPN)



II)

U-Net, également appelé « Feature Pyramid Network » (FPN)