

Cours 7 - Les réseaux de neurones convolutifs (CNN)

Neila Mezghani

21 février 2022

Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

Introduction

Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

Pourquoi les réseaux de neurones convolutifs ? (1/3)

- Les PMC ont été utilisés auparavant pour la classification d'images
- Cependant, ils ont eu des difficultés à gérer des images de grande taille en raison de la croissance exponentielle du nombre de connexions avec la taille de l'image
- En effet, dans un PMC, chaque neurone est totalement connecté à chacun des neurones de la couche précédente et suivante.

Pourquoi les réseaux de neurones convolutifs ? (2/3)

- Par exemple, soit une image de taille $32 \times 32 \times 3$ (32 de large, 32 de haut, 3 canaux de couleur RGB). On a alors 3 072 pixels.
- Si on considère un seul neurone entièrement connecté dans la première couche cachée du PMC \implies on aurait 3 072 entrées ($32 \times 32 \times 3$).
- Pour une image $200 \times 200 \implies$ on aurait 120 000 entrées dans la première couche cachée d'un PMC par neurone
- Si on a plusieurs neurones, on obtient une énorme quantité de paramètres \implies Développement des réseaux de neurones convolutifs qui visent à limiter le nombre d'entrées

Pourquoi les réseaux de neurones convolutifs ? (3/3)

- Les réseaux de neurones convolutifs ou réseau de neurones à convolution (en anglais CNN ou ConvNet pour Convolutional Neural Networks) est une sous-catégorie de réseaux de neurones artificiels acycliques (aussi appelé à propagation avant ou feed-forward) = qui font passer directement les informations en entrée des noeuds de traitement vers les sorties
- Il s'agit d'un type de réseau de neurones non complètement connecté pour (principalement) alléger les modèles en nombre de paramètres.

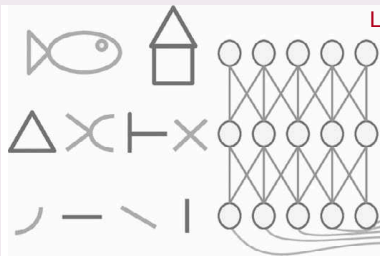
Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - **Historique/Inspirations**
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

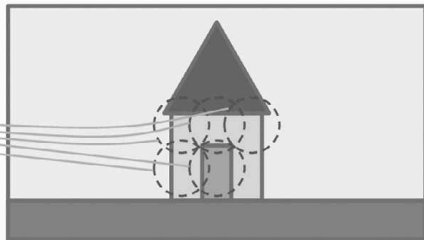
Historique/Inspirations (1/3)

- En 1958 et en 1959, les chercheurs Hubel et Wiesel ont mené une série d'expériences sur des chats et sur des singes et ils ont apporté des informations essentielles sur la structure du cortex visuel.
- Ils ont démontré que :
 - ◇ De nombreux neurones du cortex visuel ont un petit champ récepteur local. Ces neurones réagissent uniquement à un stimulus visuel qui se trouve dans une région limitée du champ visuel.
 - ◇ Certains neurones ont des champs récepteurs plus larges et réagissent à des motifs plus complexes, correspondant à des combinaisons de motifs de plus bas niveau.

Historique/Inspirations (2/3)



Les neurones biologiques du cortex visuel répondent à des motifs spécifiques dans de petites régions du champ visuel appelées champs récepteurs



Au fur et à mesure que le signal visuel traverse les modules cérébraux consécutifs, les neurones répondent à des motifs plus complexes dans des champs récepteurs plus larges

Extrait de : Deep Learning avec Keras et Tensorflow, Mise en oeuvre et cas concret. 2e édition, AurlienGéron, 2020

Historique/Inspirations (3/3)

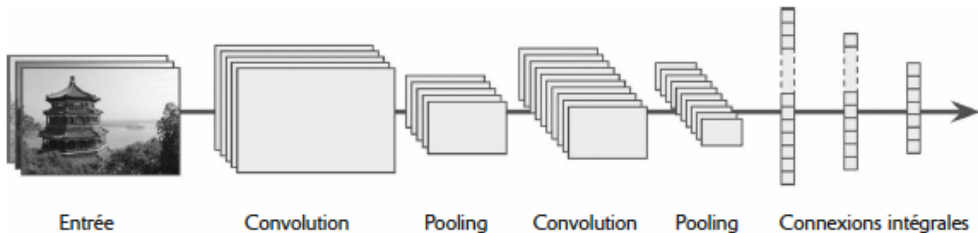
- Ces études du cortex visuel sont à l'origine du neocognitron, présenté en 1980, ancêtre des réseaux de neurones convolutifs.
- En 1998, Yann LeCun et al., a présenté la célèbre architecture [LeNet-5](#), largement employée par les banques pour reconnaître les numéros de chèques écrits à la main.

Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - **Architecture**
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

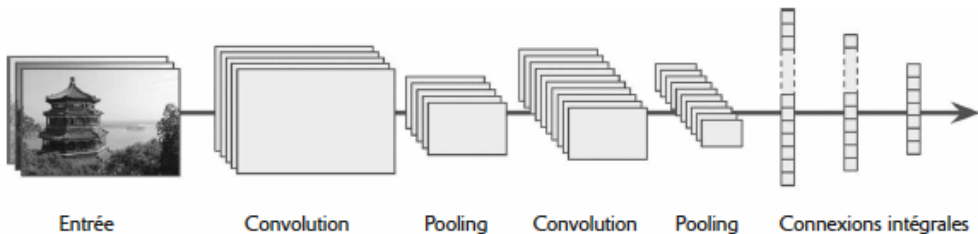
Architecture (1/2)

- Les architectures classiques de CNN empilent quelques couches de convolution qui sont suivies chacune (généralement) d'une couche ReLU → une couche de pooling → quelques autres couches de convolution → une autre couche de pooling... et ainsi de suite.



Architecture (2/2)

- À la fin de la pile de ces couches, on ajoute souvent un réseau de neurones constitué de quelques couches entièrement connectées (+ ReLU), et la couche finale qui produit la prédiction (par exemple, une couche softmax qui génère les probabilités de classe estimées).



Couche de convolution

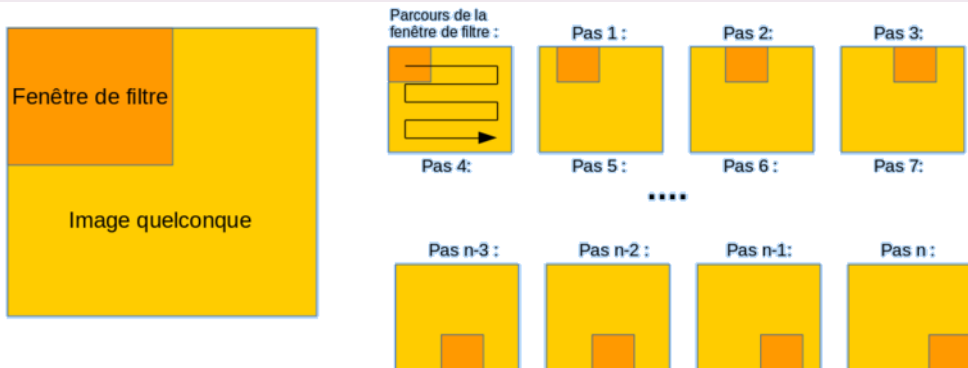
Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

Qu'est ce que la convolution ? (1/3)

- La convolution est un outil mathématique simple qui est très largement utilisé pour le traitement d'image.
- La convolution agit comme un filtrage.
- Consiste à définir une taille de fenêtre qui va se balader à travers toute l'image et d'effectuer une opération sur cette image

Qu'est ce que la convolution ? (2/3)



Adaptée de : <https://www.natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-rseaux-de-neurones-convolutifs>

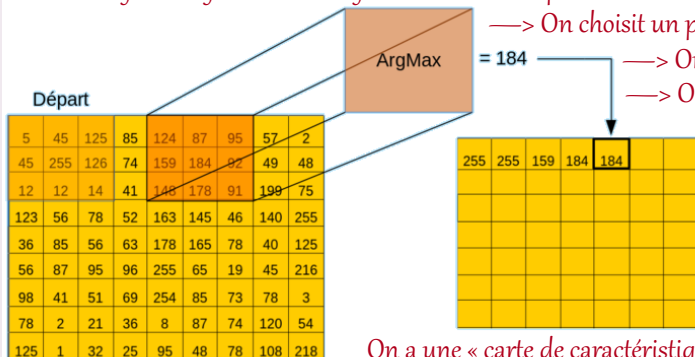
Qu'est ce que la convolution ? (3/3)

—> On définit un filtre $3 \times 3 =$ fenêtre $3 \times 3 = 9$ pixels

—> On choisit un pas vertical = 1 et un pas horizontal = 1

—> On choisit un opérateur = arg max

—> On effectue une opération sur les 9 pixels



On a une « carte de caractéristiques » dont la dimension est plus petite que l'image en entrée

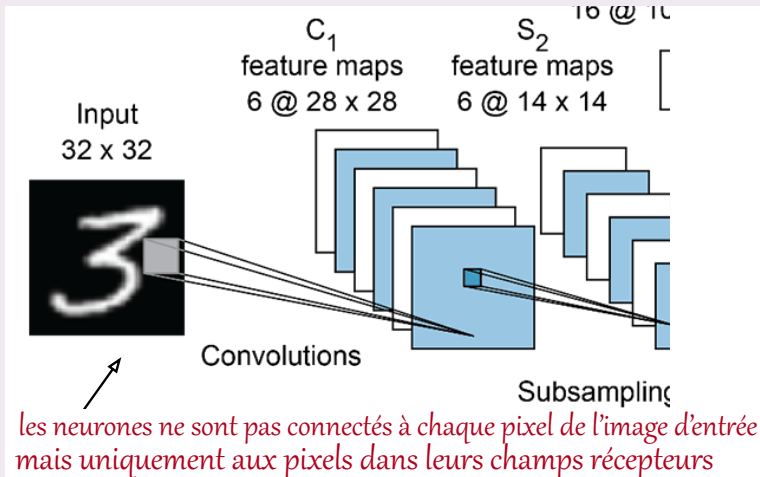
Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - **Description de la couche de convolution**
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

Description de la couche de convolution (1/2)

- La couche de convolution est le bloc le plus important d'un CNN.
- Constitue toujours au moins la première couche du réseau.
- Dans cette couche, les neurones ne sont pas connectés à chaque pixel de l'image d'entrée mais uniquement aux pixels dans leurs champs récepteurs (contrairement à ce qu'on a vu pour le PMC)

Description de la couche de convolution (2/2)



Rôle de la couche de convolution (1/3)

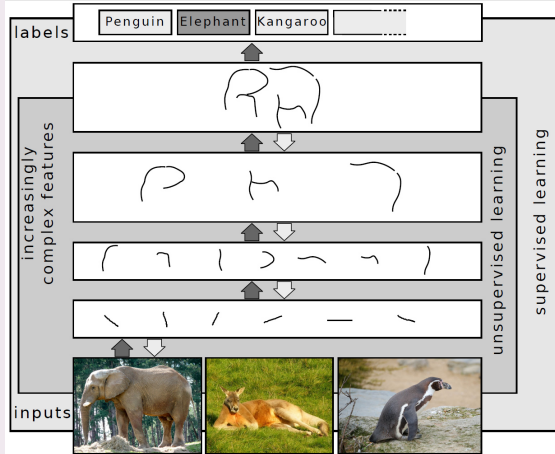
- La couche de convolution permet de repérer la présence d'un ensemble de features dans les images en entrée.
- Pour cela, un filtrage par convolution est réalisé : il consiste à balayer la fenêtre sur l'image et de calculer le produit de convolution à chaque partie de l'image balayée.
- Après la première étape nous avons donc autant de nouvelles images que de filtres aussi appelé noyaux de convolution.

Rôle de la couche de convolution (2/3)

- Une couche remplie de neurones qui utilisent le même filtre nous donne ainsi une carte de caractéristiques (*feature map*) qui fait ressortir les zones d'une image qui se rapprochent le plus du filtre.
- Un CNN contient de multiples filtres et ces filtres sont appliqués sur l'image d'origine.
- La phase de convolution peut aussi être vue comme des couches de neurones cachées où chaque neurone n'est connecté qu'à quelques neurones de la couche suivante.

Rôle de la couche de convolution (3/3)

- La première couche de convolution apprend des features simples, qui représentent des éléments de structure rudimentaires de l'image (contours, coins...)
- Plus les couches de convolution sont hautes, c'est-à-dire loin de l'entrée du réseau, plus les features apprises sont complexes : celles-ci se composent des features plus simples des couches précédentes.
- Évidemment, nous n'avons pas à définir des filtres manuellement \implies Au cours de l'entraînement, la couche de convolution apprend automatiquement les filtres qui seront les plus utiles à sa tâche, et les couches supérieures apprennent à les combiner dans des motifs plus complexes.



← Les couches cachées supérieures et la couche de sortie associent ces structures intermédiaires pour modéliser des structures de haut niveau (par exemple, des visages).

← Les couches cachées intermédiaires combinent ces structures de bas niveau pour modéliser des structures de niveau intermédiaire (par exemple, des carrés et des cercles).

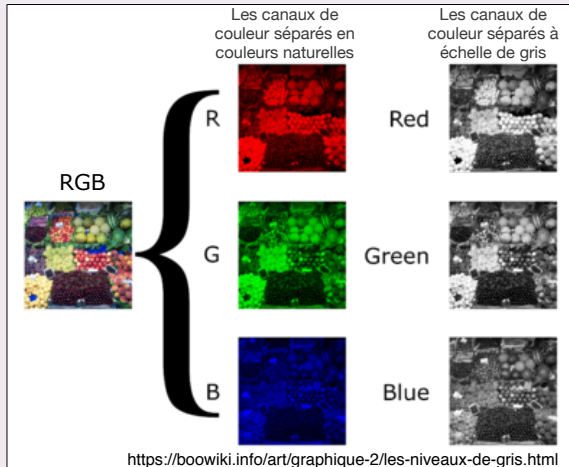
← Les couches cachées inférieures modélisent des structures de bas niveau (par exemple, des traits aux formes et aux orientations variées).

Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

Images d'entrée (1/2)

- Les images d'entrée sont constituées de multiples sous-couches, une par canal de couleur
- Pour les images en couleurs, on a trois canaux : rouge, vert et bleu (RVB, ou RGB).
- Pour les images en niveaux de gris, on a un seul canal.

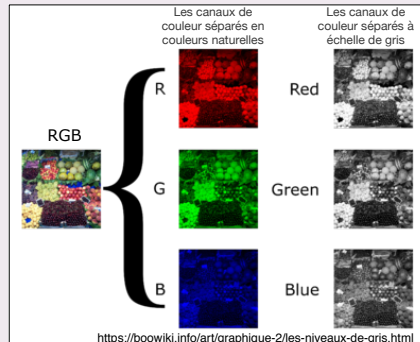


Images d'entrée (2/2)

Chaque image d'entrée est donc représentée par un tenseur à trois dimensions de forme ($width \times height \times channels$) :

- *width* : La largeur.
- *height* : La hauteur.
- *channels* : nombre de canaux.

Un mini-lot d'images est un tenseur de quatre dimensions (ajout de *batch_size*)



Hyperparamètres (1/3)

La couche de convolution possède quatre hyperparamètres :

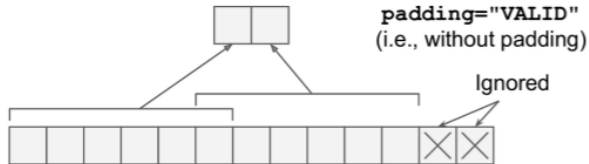
- ❶ Le nombre de filtres K
- ❷ La taille F des filtres : chaque filtre est de dimensions $F \times F \times channels$.
- ❸ Le pas contrôle le chevauchement des champs récepteurs (strides). Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand. Par exemple, un pas de 1 signifie qu'on déplace la fenêtre d'un pixel à la fois.

Hyperparamètres (2/3)

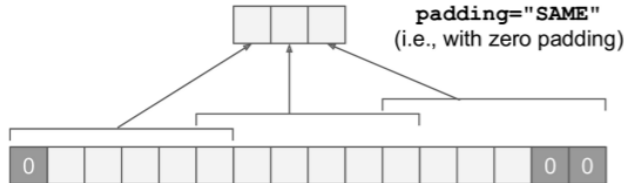
La couche de convolution possède quatre hyperparamètres :

- ④ La marge (à 0) ou zero padding (padding) : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

Hyperparamètres (3/3)



Si padding = « VALID », la couche de convolution n'utilise pas de marge de zéros et peut ignorer certaines lignes et colonnes dans les parties inférieure et droite de l'image d'entrée



Si padding = « SAME », la couche de convolution ajoute une marge de zéros si nécessaire.

Exemple : Couche de convolution (1/2)

L'objectif est de réaliser un convolution (`tf.nn.conv2d`) sur l'image `china.jpg` et d'afficher les carte de caractéristiques suite à la convolution

```
# Creation des filtres (7*7)
```

```
filters = np.zeros(shape=(7, 7, channels, 2), dtype=np.float32)
```

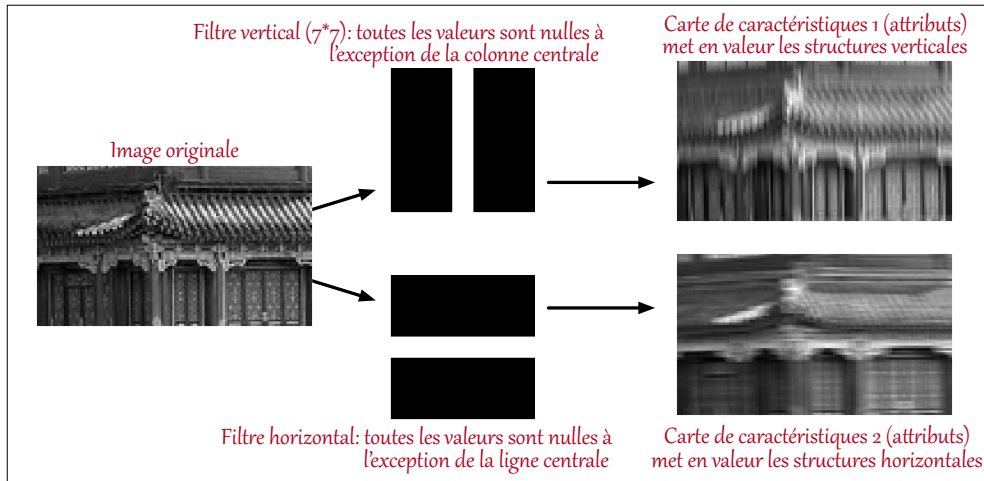
```
filters[:, 3, :, 0] = 1 # Ligne verticale
```

```
filters[3, :, :, 1] = 1 # Ligne horizontale
```

```
# Convolution
```

```
outputs = tf.nn.conv2d(images, filters, strides=1,  
                        padding="SAME")
```

Exemple : Couche de convolution (2/2)



La couche de pooling

Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - **La couche de pooling**
 - Paramétrage de la couche de pooling
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

La couche de pooling (1/4)

- La couche de *pooling* (mis en commun) est souvent placée entre deux couches de convolution : elle reçoit en entrée plusieurs *feature maps*, et applique à chacune d'entre elles l'opération de pooling.
- L'opération de pooling (ou sub-sampling) consiste à réduire la taille des images tout en préservant leurs caractéristiques importantes.

La couche de pooling (2/4)

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

Dimension (4*4)

Subsampling



max pooling
Filtre (2*2)
Pas (2,2)

9	7
8	6

Dimension (2*2)

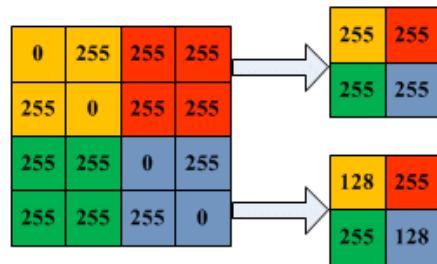
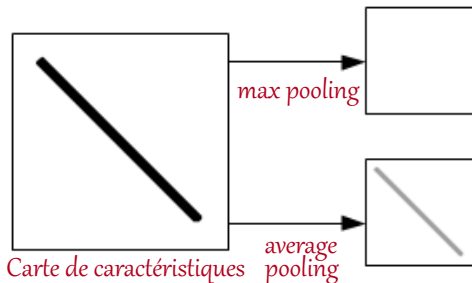
La couche de pooling (3/4)

- Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau.
- Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture de réseau de neurones convolutifs pour réduire le sur-apprentissage.
- L'opération de pooling crée aussi une forme d'invariance par translation puisque tous les neurones d'un même noyau (filtre) sont identiques, le motif détecté par ce noyau est indépendant de localisation spatiale dans l'image

La couche de pooling (4/4)

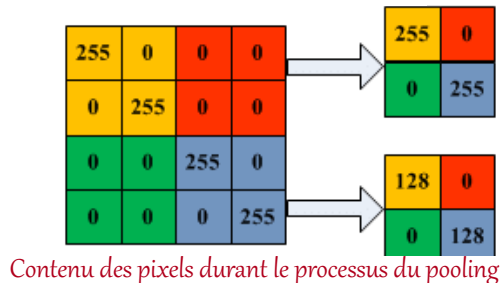
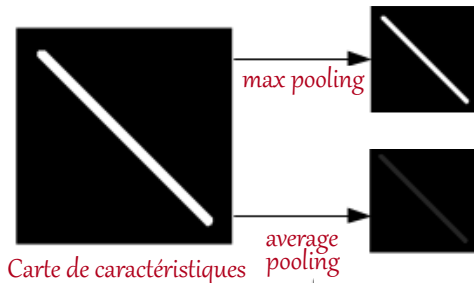
- La forme la plus courante est une couche de pooling avec des tuiles de taille 2×2
- La valeur de sortie peut être :
 - ◇ La valeur maximale en entrée \implies max-pooling
 - ◇ La valeur moyenne en entrée \implies average-pooling
 - ◇ La valeur moyenne en entrée \implies L2-norm pooling

Illustration des avantages/ inconvénients du max et de l'average pooling



Contenu des pixels durant le processus du pooling

Illustration des avantages/ inconvénients du max et de l'average pooling



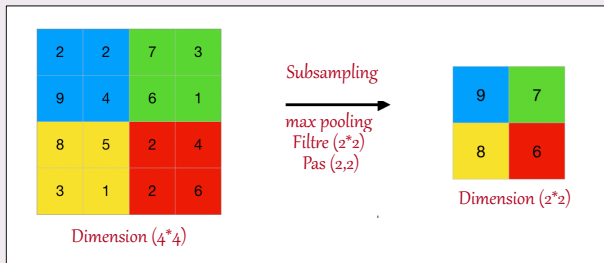
Plan du cours

- 1 Introduction
 - Pourquoi les réseaux de neurones convolutifs ?
 - Historique/Inspirations
 - Architecture
- 2 Couche de convolution
 - Qu'est ce que la convolution ?
 - Description de la couche de convolution
 - Paramétrage de la couche de convolution
- 3 La couche de pooling
 - La couche de pooling
 - **Paramétrage de la couche de pooling**
- 4 La couche de correction (ReLU) et la couche Fully connected
- 5 Exemples d'architectures de CNN

Hyperparamètres

La couche de pooling présente seulement deux hyperparamètres :

- La taille F des cellules : l'image est découpée en cellules carrées de taille $F \times F$ pixels
- Le pas S : les cellules sont séparées les unes des autres de S pixels



Exemple : Couche de pooling (1/2)

L' average pooling (`tf.nn.AveragePooling2D`) et le max pooling (`tf.nn.MaxPooling2D`) peut être obtenu par :

```
max_pool = keras.layers.MaxPool2D(pool_size=2)
avg_pool = keras.layers.AveragePooling2D(pool_size=2)
```

Exemple : Couche de pooling (2/2)

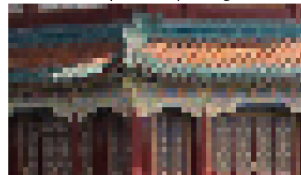
Input



Output average pooling



Output max pooling



La couche de correction (ReLU) et la couche Fully connected

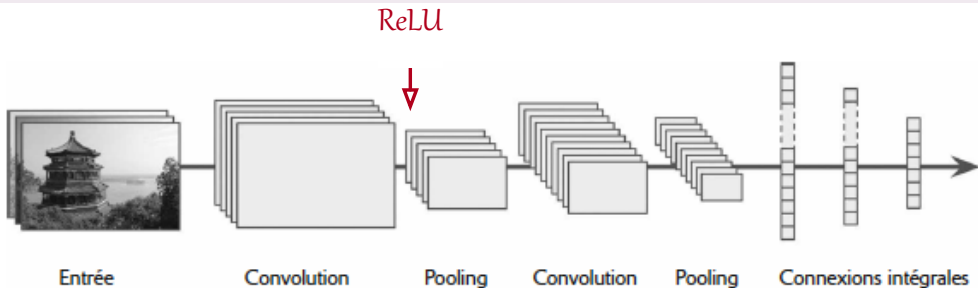
La couches de correction (ReLU) (1/2)

- Souvent, il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.
- On utilise souvent la correction ReLU (abréviation de Unité Linéaire Rectifiée) :

$$f(x) = \max(0, x)$$

aussi « fonction d'activation non saturante » qui augmente les propriétés non linéaires de la fonction de décision et de l'ensemble du réseau sans affecter les champs récepteurs de la couche de convolution.

La couches de correction (ReLU) (2/2)

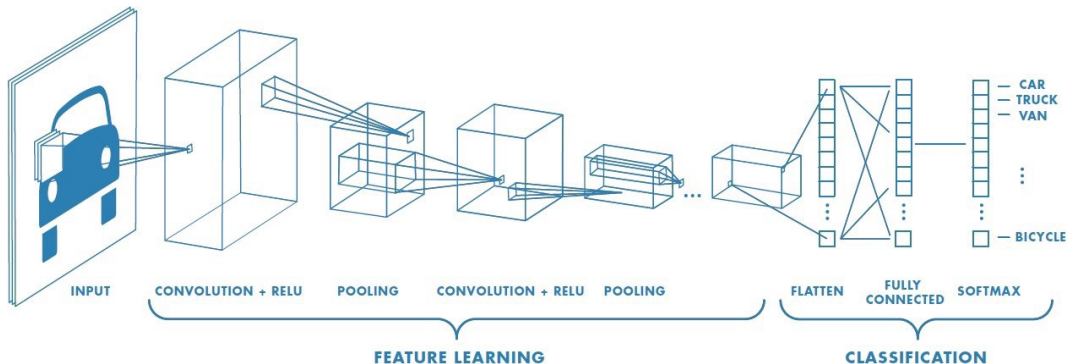


Deep Learning avec Keras et Tensorflow, Mise en oeuvre et cas concret. 2e édition, AurlienGéron, 2020

La couche Fully connected

- Cette couche est à la fin du réseau.
- Elle permet la classification de l'image à partir des caractéristiques extraites par la succession de bloc de traitement.
- Elle est entièrement connectée, car toutes les entrées de la couche sont connectées aux neurones de sorties de celle-ci.
- Chaque neurone attribue à l'image une valeur de probabilité d'appartenance à la classe i parmi les C classes possibles.

Résumé



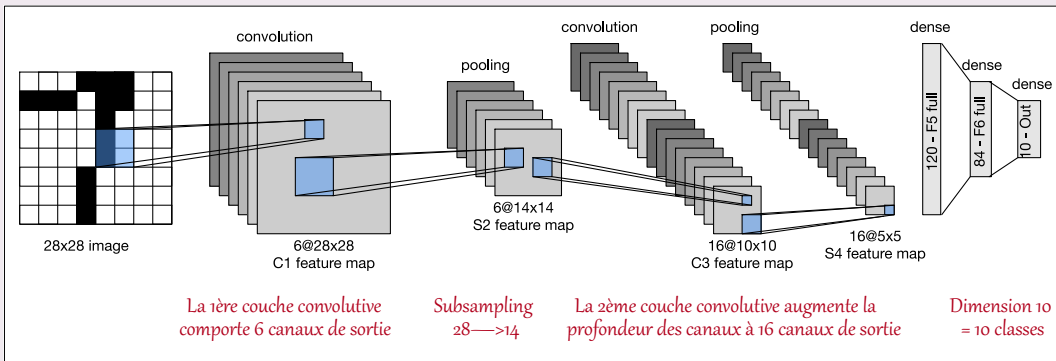
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Exemples d'architectures de CNN

LeNet-5 (1/3)

- Les études du cortex visuel sont à l'origine du neocognitron, présenté en 1980, ancêtre des réseaux de neurones convolutifs.
- En 1998, Yann LeCun et al., a présenté la célèbre architecture [LeNet-5](#), largement employée par les banques pour reconnaître les numéros de chèques écrits à la main.
- Une des premières applications a été la reconnaissance automatique des codes postaux aux US.

LeNet-5 (2/3)



LeNet-5 (3/3)



a

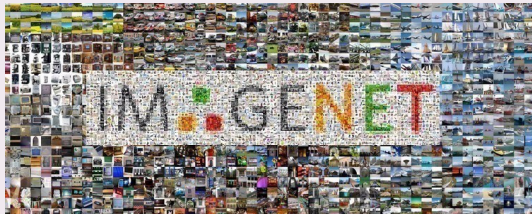
a. Illustration extraite de <http://yann.lecun.com/exdb/lenet/>

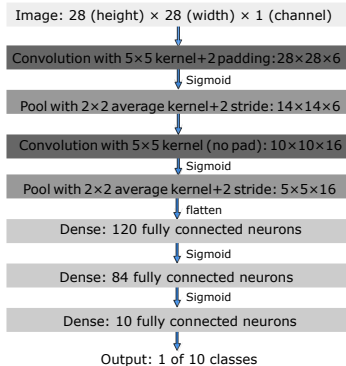
AlexNet (1/2)

- Alexnet est le nom de l'architecture d'un réseau profond de neurones à convolutions développé par une équipe de l'Université de Toronto
- L'équipe est dirigée par Geoffrey Hinton, dont faisaient partie Alex Krizhevsky, architecte principal qui lui a donné son nom AlexNet.
- Alexnet a remporté l'épreuve ImageNet en 2012.

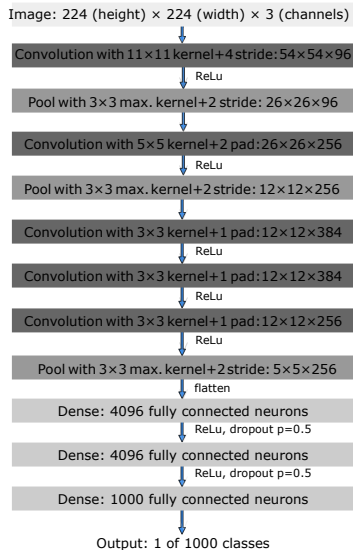
AlexNet (2/2)

- ImageNet est un ensemble de données de plus de 15 millions d'images haute résolution étiquetées appartenant à environ 22 000 catégories.
- Dans le défi de reconnaissance visuelle à grande échelle ImageNet (ILSVRC) auquel AlexNet a participé, un sous-ensemble d'ImageNet est utilisé, qui comprend environ 1000 images dans chacune des 1000 catégories.



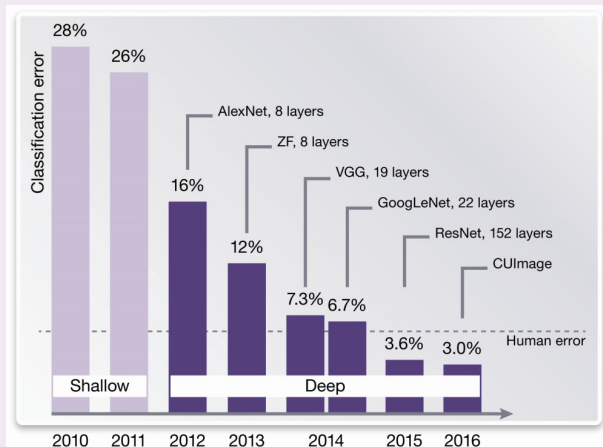


LeNet



AlexNet

Comparaisons des performances des différents CNN



Références intéressantes

Voici quelques liens intéressants à consultés (certains ont été utilisés pour les illustrations) :

- ❶ [Pense-bête de réseaux de neurones convolutionnels](#)
- ❷ [Les réseaux de neurones de convolution pour les néophytes](#)
- ❸ [The amazing world of TensorFlow](#)