

Réseaux de neurones à convolution

Introduction à l'apprentissage automatique – GIF-4101 / GIF-7005

Professeur : Christian Gagné

Semaine 10



10.1 Convolution et traitement des images

Convolution

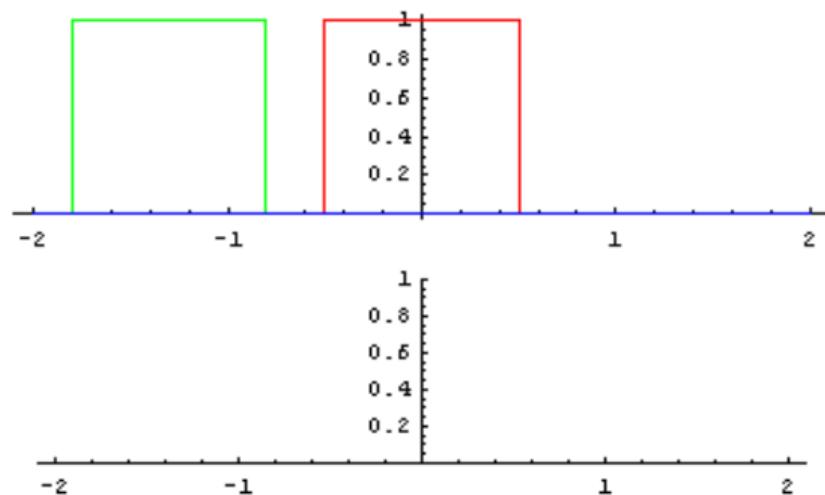
- Convolution : produit de deux fonctions sur un même domaine

$$f(x) * g(x) \equiv \int_{t=-\infty}^{\infty} f(x-t) g(x) dt$$

- Formulation discrète

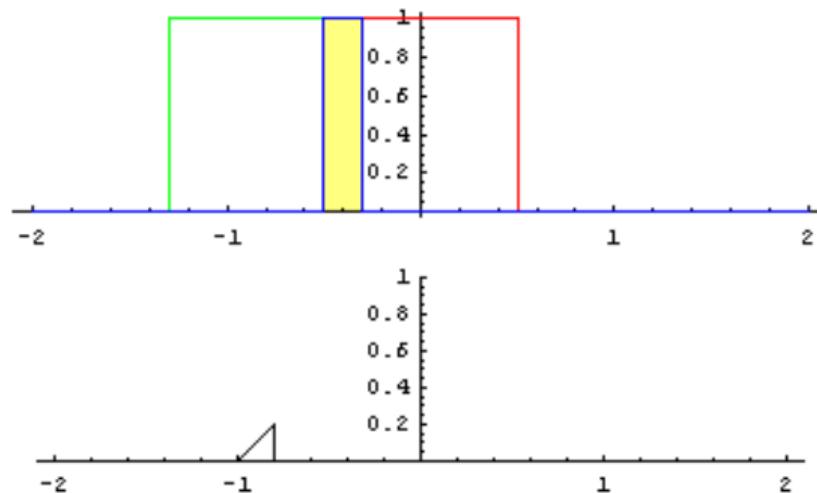
$$f(x) * g(x) \equiv \sum_{t=-\infty}^{\infty} f(x-t) g(x)$$

Exemple de convolution



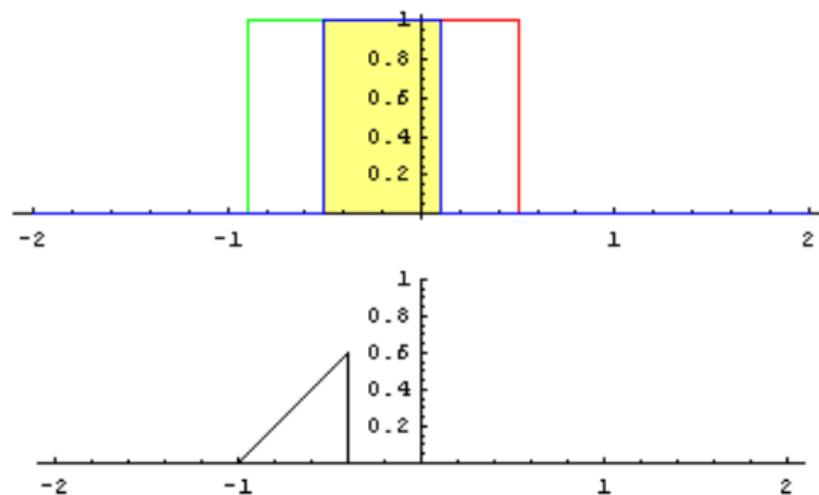
Par Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Exemple de convolution



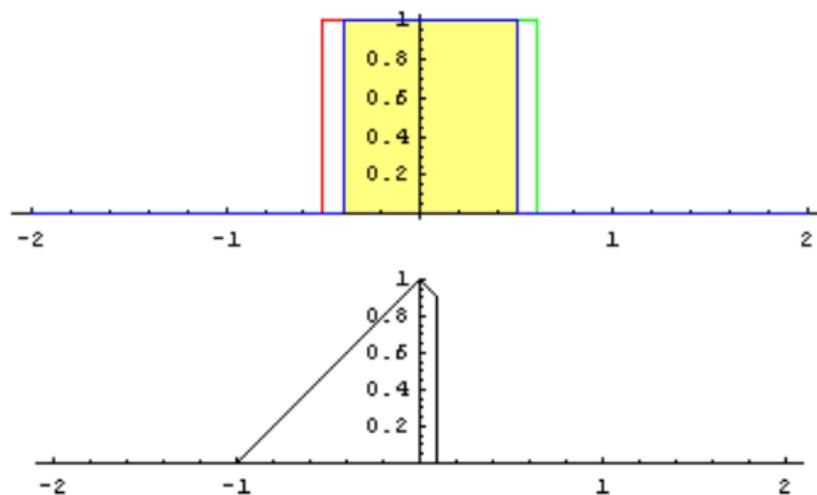
Par Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Exemple de convolution



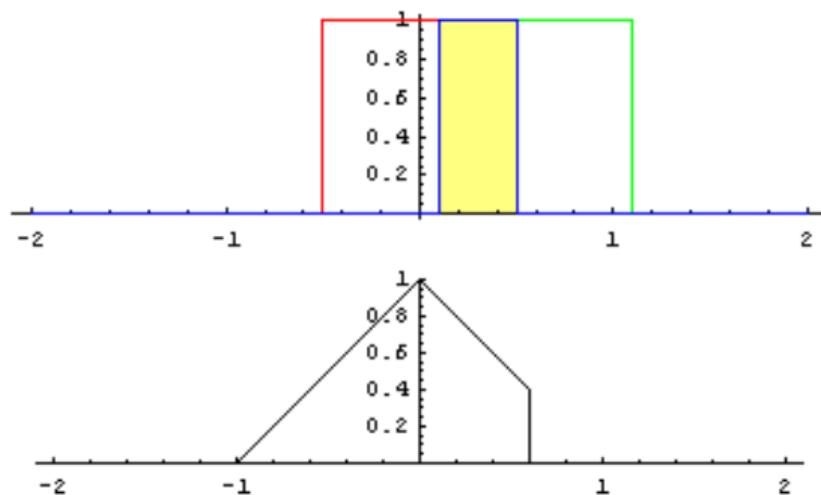
Par Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Exemple de convolution



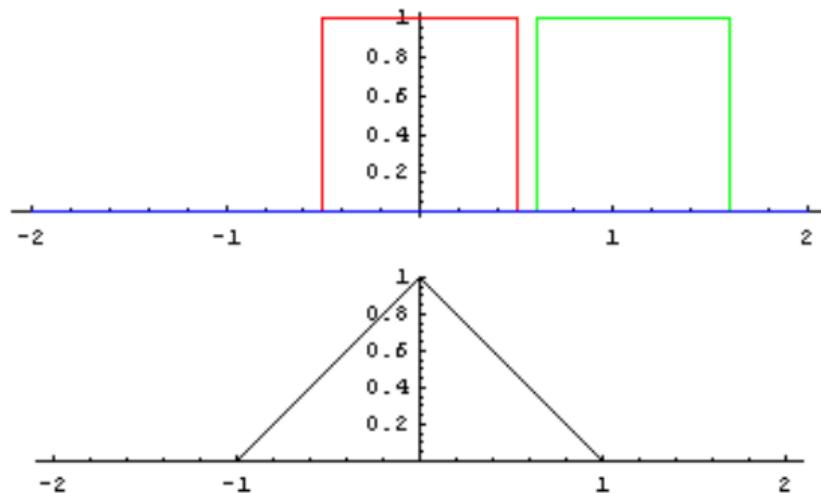
Par Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Exemple de convolution



Par Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Exemple de convolution



Par Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution et estimation de densité

- Distribution de Dirac décentrée

$$\delta(x - t) = \begin{cases} \infty & \text{si } x = t \\ 0 & \text{autrement} \end{cases}, \quad \int_{x=-\infty}^{\infty} \delta(x - t) dx = 1.$$

- Convolution sur Dirac décentrés

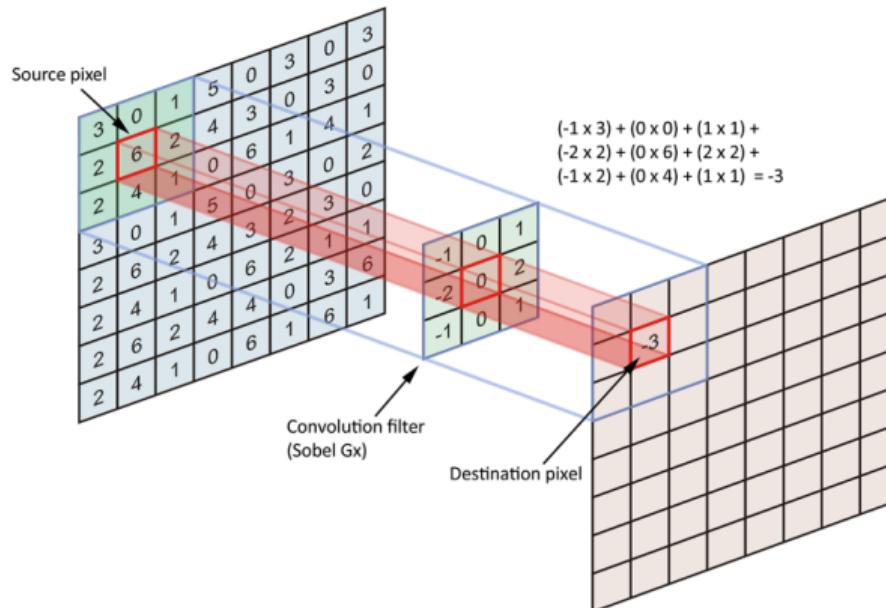
$$f(x) * \delta(x - u) = f(x - u)$$

- Estimation de densité à noyau : convolution du noyau avec plusieurs Diracs centrés sur les données

$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^N K\left(\frac{x - x^t}{h}\right) = \frac{1}{Nh} \sum_{t=1}^N K\left(\frac{x}{h}\right) * \delta(x - x^t)$$

Traitement d'images

- Convolution 2D est un élément de base du traitement d'images



Source : <https://thigiacmaytinh.com/wp-content/uploads/2018/05/kernel.png>, accédé le 13 novembre 2018.

Exemples de filtres

Noyau identité (3×3) :

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



Flou de Gauss :

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



Détection de contours :

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Amélioration de la netteté :

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



Par Michael Plotke, CC-BY-SA 3.0, <https://commons.wikimedia.org/wiki/File:Vd-Orig.png>,

<https://commons.wikimedia.org/wiki/File:Vd-Blur1.png>,

<https://commons.wikimedia.org/wiki/File:Vd-Edge3.png>, <https://commons.wikimedia.org/wiki/File:Vd-Sharp.png>.

Opérateur de Sobel

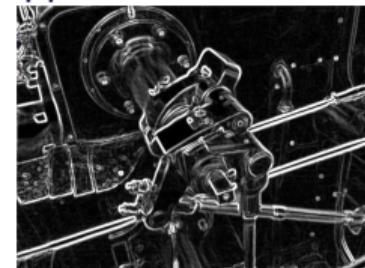
- Filtre classique pour la détection d'arêtes
 - Calcul les gradients locaux de l'intensité de l'image
 - S'appuie sur deux convolutions pour obtenir le gradient vertical \mathbf{G}_x et le gradient horizontal \mathbf{G}_y d'une image \mathbf{A} , le résultat est une image $\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}, \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Image d'origine :



Application de Sobel :



Par Simpsons contributor, CC-BY-SA 3.0, [https://commons.wikimedia.org/wiki/File:Valve_original_\(1\).PNG](https://commons.wikimedia.org/wiki/File:Valve_original_(1).PNG)

Par Simpsons contributor, CC-BY-SA 3.0, [https://commons.wikimedia.org/wiki/File:Valve_sobel_\(3\).PNG](https://commons.wikimedia.org/wiki/File:Valve_sobel_(3).PNG)

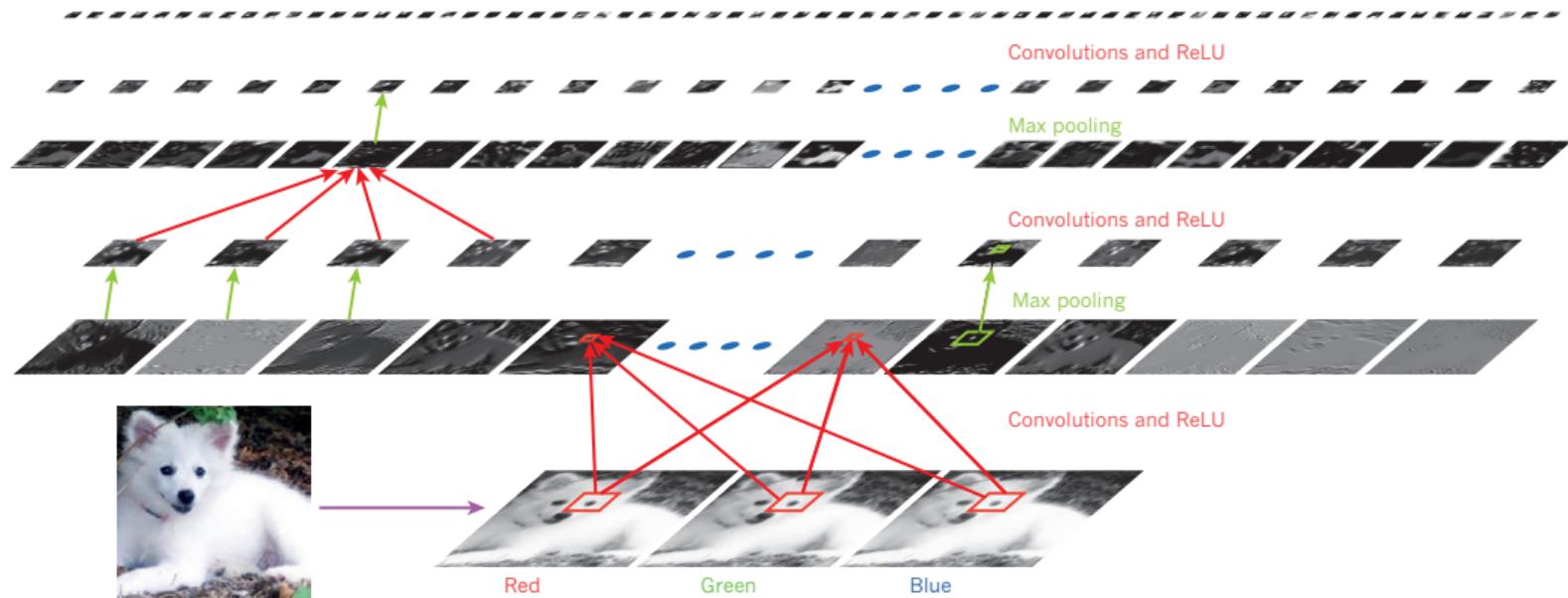
10.2 Réseaux de neurones à convolution

Réseaux de neurones à convolution

- Idée : définir des réseaux de neurones comportant des opérations de convolution
 - Apprendre les valeurs numériques des filtres convolués
 - Définir un réseau exploitant des éléments de la structure des données
 - Son ou parole : données temporelles (convolutions 1D)
 - Image : données spatiales (convolutions 2D)
 - Vidéo : données spatio-temporelles (convolutions 3D)
 - Enchaînement d'étages de convolutions, filtrant sortie de couche précédente
 - Permet une modélisation plus compacte que réseaux pleinement connectés et invariante en translation
- Quelques composants d'un réseaux à convolution
 - Couche de filtres convolués sur les différents canaux
 - Pooling : valeur maximale (max pool) ou moyenne (avg pool) dans une certaine fenêtre convoluée
 - Fonctions de transfert : ReLU, etc.
 - Près de la sortie, couches pleinement connectées (comme avec perceptron multicouche)

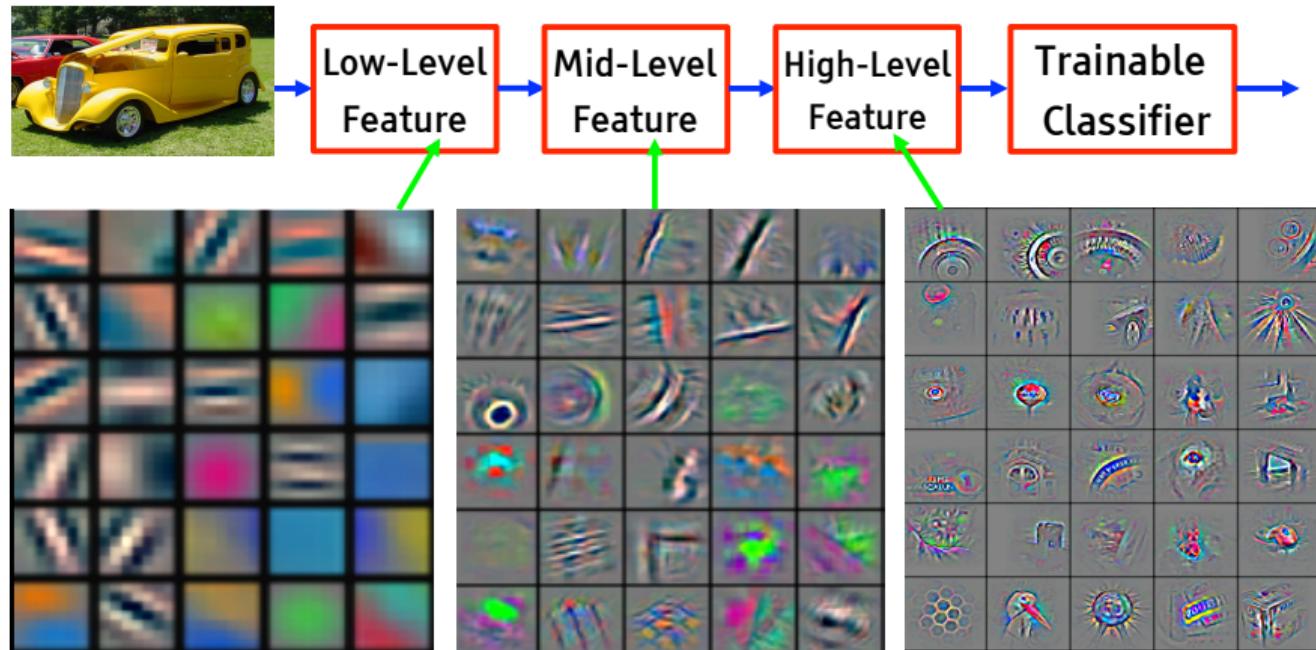
Réseau à convolution

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



Tiré de Y. LeCun, Y. Bengio et G. Hinton, Deep Learning, Nature, vol. 521, 28 mai 2015. Accédé en ligne le 6 novembre 2020 au <https://www.nature.com/articles/nature14539>.

Composition de filtres

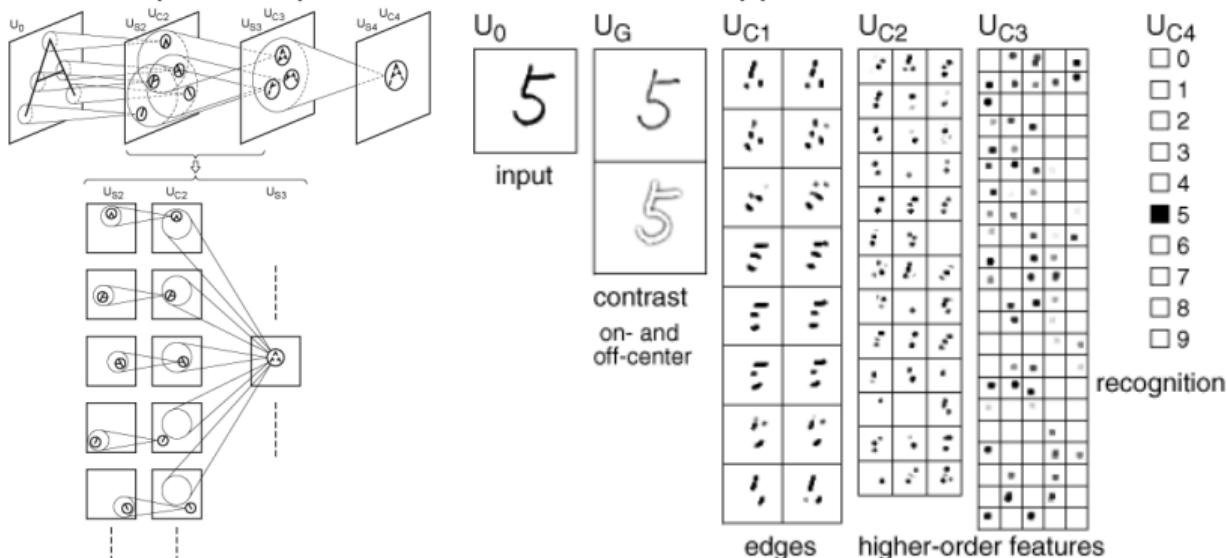


Tiré de G. Hinton, Y. Bengio et Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015. Accédé en ligne le 6 novembre 2020 au <https://nips.cc/Conferences/2015/Schedule?showEvent=4891>.

10.3 Exemples de réseaux à convolution

Neocognitron

- Neocognitron : l'ancêtre des réseaux à convolution
 - Proposé par Kunihiko Fukushima dans les années 1980
 - Source d'inspiration pour LeCun dans le développement des réseaux à convolution

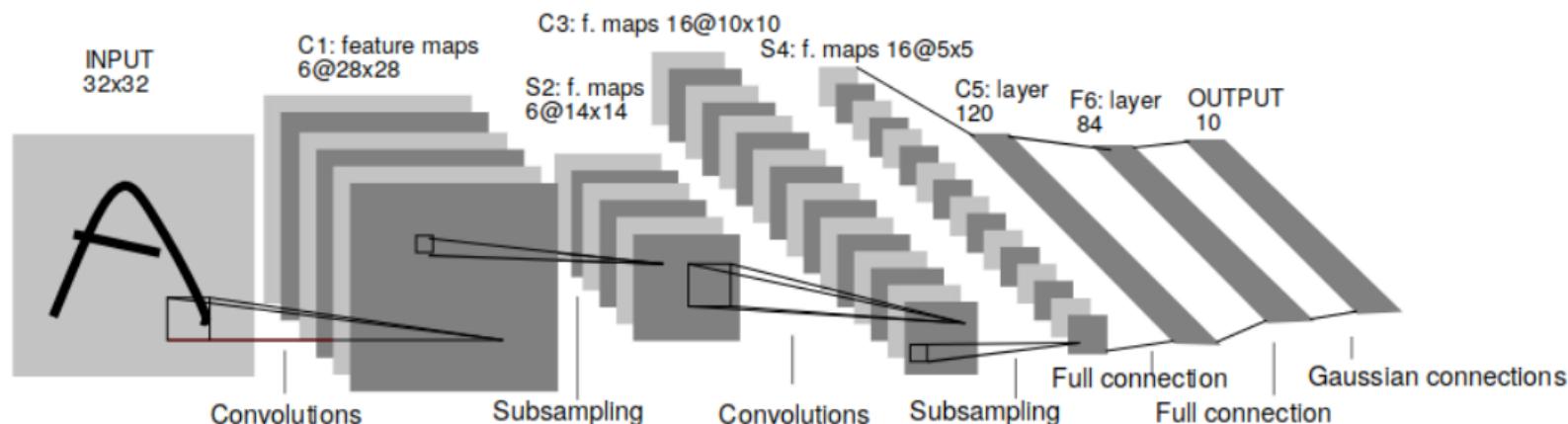


Source : <http://www.scholarpedia.org/article/File:ScholarFig2.gif>,
accédé le 13 novembre 2018.

Source :
<http://www.scholarpedia.org/article/File:ScholarFig4.gif>,
accédé le 13 novembre 2018.

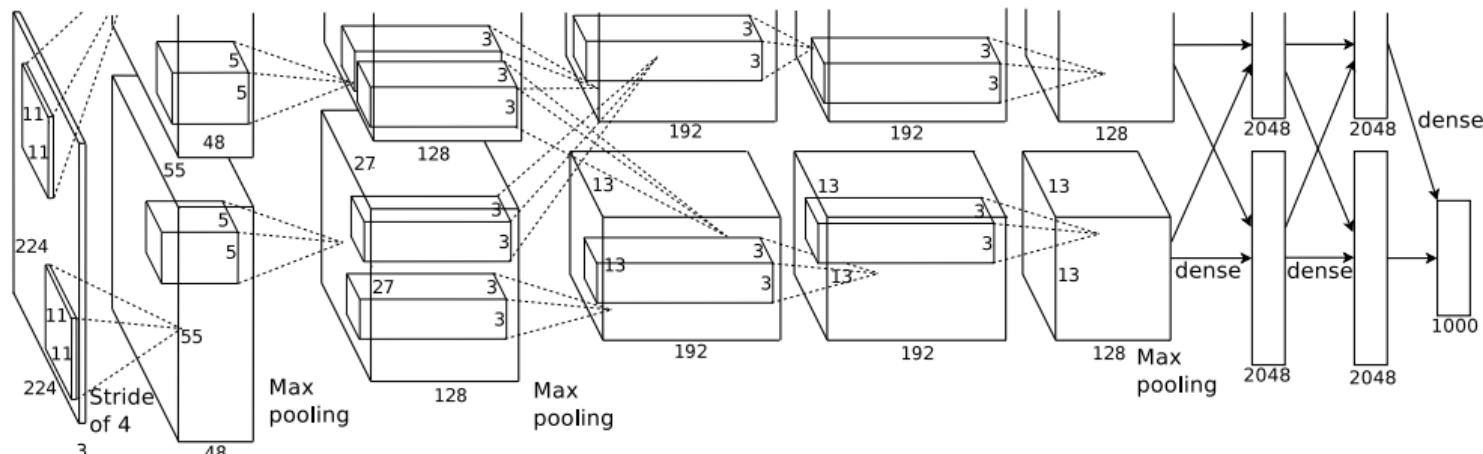
LeNet5

- LeNet5 : réseau à convolution classique, proposé dans les années 1990
 - 3 couches de convolutions, 2 couches de average pooling, 2 couches pleinement connectées
 - 60k paramètres (de 10M à 100M avec réseaux modernes)



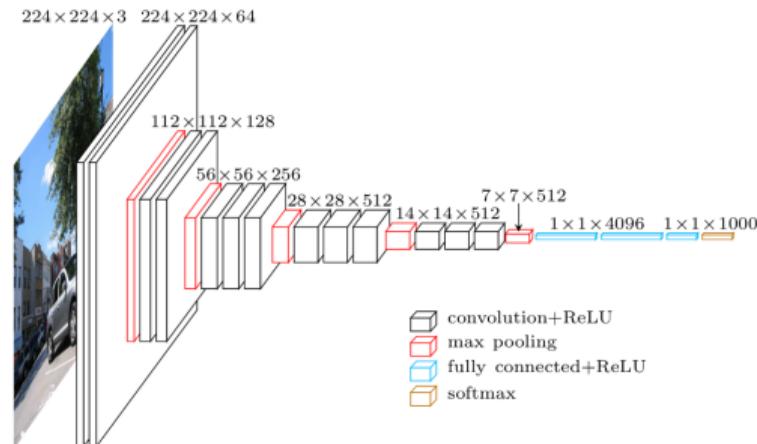
Tiré de Y. LeCun, L. Bottou, Y. Bengio et P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86(11), 1998. Accédé en ligne le 6 novembre 2020 au <http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf>.

- AlexNet : réseau pour la reconnaissance d'objets
 - Gagnant du concours ImageNet 2012
 - Implémenté pour calculs sur GPU
 - Souvent utilisé comme modèle de base pour transfert de représentations
 - 8 couches de convolution, quelques couches de max pooling, 3 couches pleinement connectées



Tiré de A. Krizhevsky, I. Sutskever, et G. Hinton, *Imagenet classification with deep convolutional neural networks*. NIPS, 2012. Accédé en ligne le 6 novembre 2020 au <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

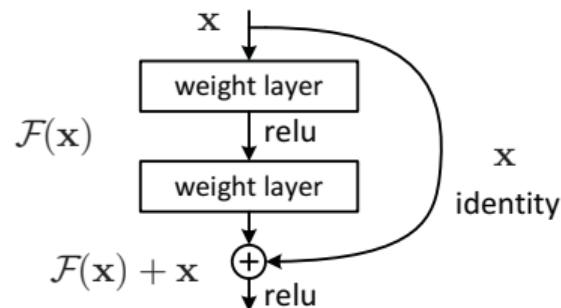
- VGGNet : plus grande profondeur avec topologie simplifiée
 - Gagnant du concours ImageNet 2013
 - Profondeur est un élément critique pour de bonnes performances
 - Similaire à AlexNet, mais avec seulement convolutions 3×3 , max pooling 2×2 , 3 couches pleinement connectées et 16 couches au total (VGG-16)



Source : <https://heuritech.files.wordpress.com/2016/02/vgg16.png>, accédé le 13 novembre 2018.

ResNet

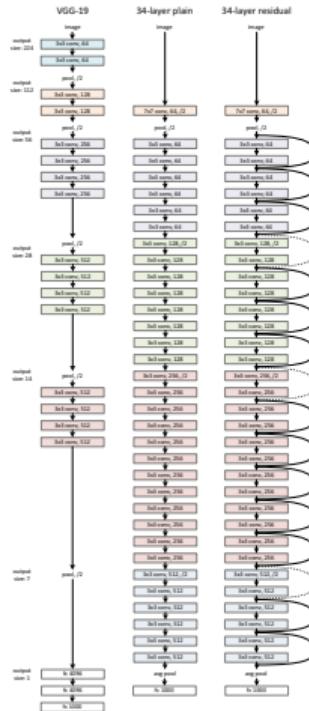
- Réseaux résiduels : permettre des connections directes entre couches non adjacentes (*skip links*)



Tiré de K. He, X. Zhang, S. Ren, et J. Sun, Deep residual learning for image recognition. CVPR, 2016. Accédé en ligne le 6 novembre 2020 au <https://arxiv.org/abs/1512.03385>.

- Permet des réseaux beaucoup plus profonds et performants
 - Gagnant de compétition ImageNet 2015 (3,57 % d'erreur top 5)
 - Facilite l'optimisation et la propagation du signal à travers le réseau
 - Bloc résiduel doit faire mieux qu'un traitement directement sur le bloc précédent

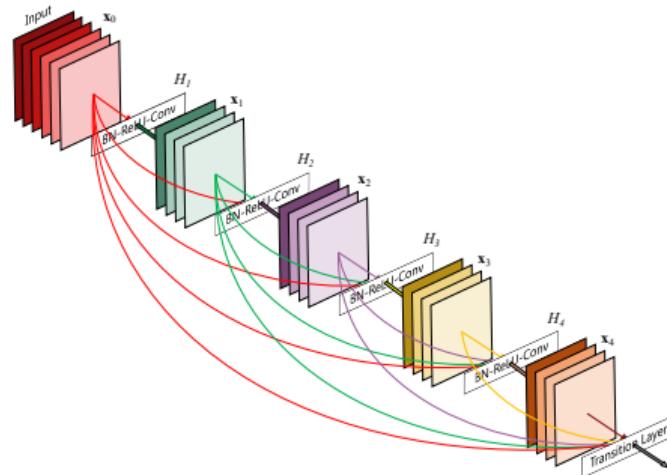
ResNet



Tiré de *K. He, X. Zhang, S. Ren, et J. Sun, Deep residual learning for image recognition. CVPR, 2016*. Accédé en ligne le 6 novembre 2020 au <https://arxiv.org/abs/1512.03385>.

DenseNet

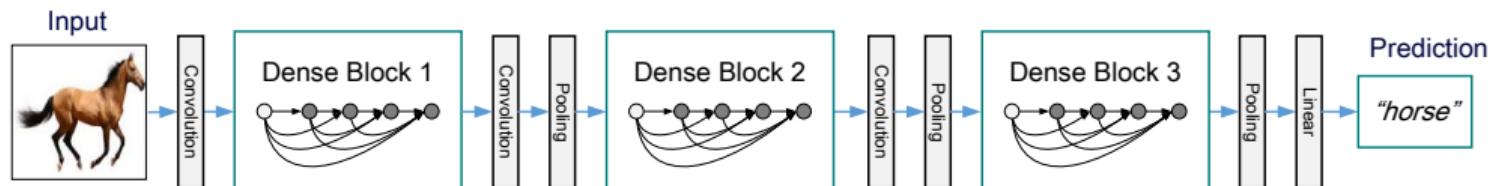
- Observation : réseaux à convolution peuvent être plus profonds et obtenir de meilleures performances avec connections proches dans tout le réseau à son entrée
- DenseNet : connecter chaque couche à toutes les couches qui précèdent
 - Réseau à L couches aura $L(L + 1)/2$ connections directes entre les couches



Tiré de G. Huang, Z. Liu, L. Van Der Maaten et K.Q. Weinberger, *Densely Connected Convolutional Networks*. CVPR, 2017. Accédé en ligne le 6 novembre 2020 au <https://arxiv.org/abs/1608.06993>.

DenseNet

- En pratique, on crée des blocs denses séparés par des couches de convolutions et de pooling

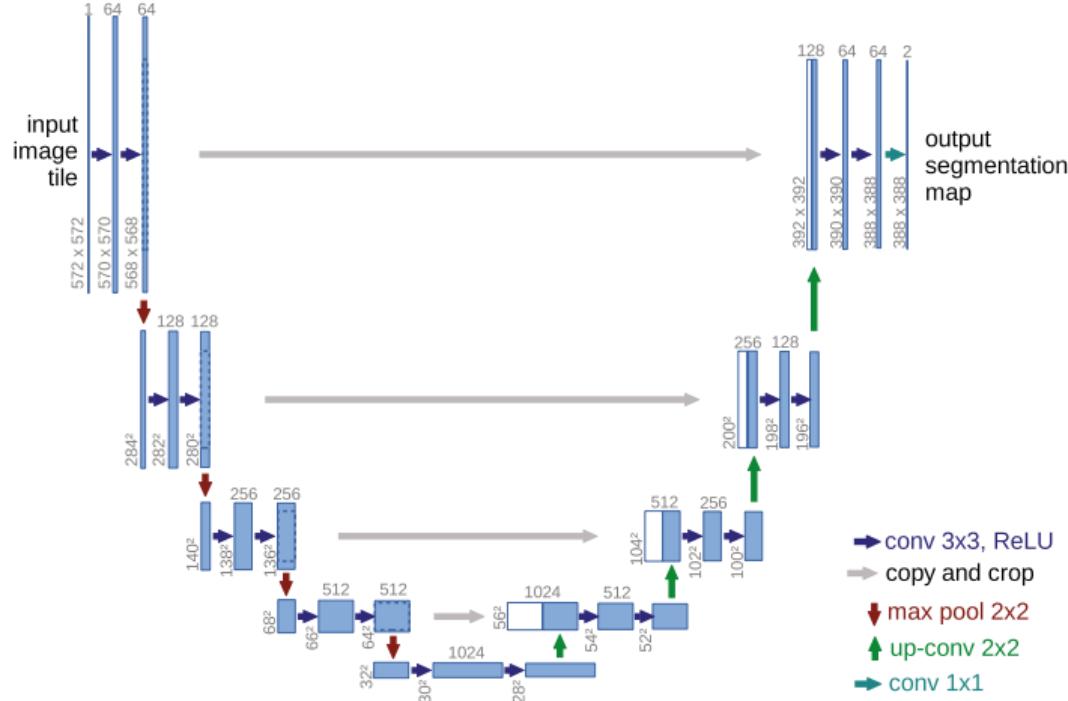


Tiré de G. Huang, Z. Liu, L. Van Der Maaten et K.Q. Weinberger, *Densely Connected Convolutional Networks*. CVPR, 2017. Accédé en ligne le 6 novembre 2020 au <https://arxiv.org/abs/1608.06993>.

- Chaque couche dans un bloc dense peut être relativement « étroite », c'est-à-dire peut comprendre peu de neurones

- Réseaux présentés jusqu'à présent d'abord proposés et testés pour reconnaissance d'objets (classement)
 - D'autres tâches possibles en vision : détection, suivi, etc.
- Segmentation : identifier régions cohérentes de l'image
 - Séparer les différentes régions
 - Donner un étiquette à chaque région
- U-Net : réseau proposé en imagerie biomédicale
 - Réseau pleinement convolutionnel, donne une image en sortie
 - Compression de l'information en milieu de réseau, similaire à un auto-encodeur
 - Skip links permettent de conserver structure spatiale

U-Net



Tiré de O. Ronneberger, P. Fischer, et T. Brox, U-net : Convolutional networks for biomedical image segmentation. MICCAI, 2015. Accédé en ligne le 6 novembre 2020 au <https://arxiv.org/abs/1505.04597>.