Explications toutes simples d’un réseau de neurone, et de son interprétation

<https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-non-lineaires/4730716-entrainez-un-reseau-de-neurones-simple>

<https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-non-lineaires/4732186-empilez-les-perceptrons>

Explications sur un CNN

<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn>

https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn

<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5088816-apprenez-a-construire-un-cnn-et-gagnez-du-temps-avec-le-transfer-learning>

<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5097666-tp-implementez-votre-premier-reseau-de-neurones-avec-keras>

Premier GAN, basé sur un réseau CNN un peu modifié

<https://towardsdatascience.com/gan-by-example-using-keras-on-tensorflow-backend-1a6d515a60d0>

Pour faire le lien entre les deux cours :

“Les *couches de convolution*, pooling et fully-connected correspondent à des instances des classes respectives [Conv2D](https://keras.io/layers/convolutional/), [MaxPooling2D](https://keras.io/layers/pooling/) et [Dense](https://keras.io/layers/core/) du module  keras.layers . Une couche ReLU peut être créée soit en instanciant la classe [Activation](https://keras.io/activations/), soit en ajoutant un argument au constructeur de la couche qui la précède.”

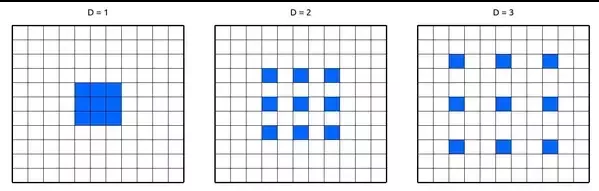
“Pour construire une couche de convolution, nous devons préciser le nombre de filtres utilisés, leur taille, le pas et le zero-padding. Ils correspondent respectivement aux arguments  filters ,  kernel\_size ,  strides  et  padding  du constructeur de la classe  Conv2D .”

Notion de batch\_size -> C’est le nombre d’image que l’on entraine en simultané. Plus on en fait, plus le calcul du gradien est bon, mais plus cela demande de l a mémoire

Cycle GAN : <https://machinelearningmastery.com/cyclegan-tutorial-with-keras/>

D'après le papier, pour apporter de la déformation, il faut utiliser de la convolution dilated, au lieu de prendre de la convolution classique

L'idée est juste que le filtre est éclaté, au lieu d'être sur des pixels adjacents comme d'habitude. D traduit le nombre de pixel en ecarté : donc D = 1 est de la convolution normale, et D> 1 est de la convolution dilated



Sur keras ; out = Conv2D(10, (3, 3), dilation\_rate=2)(input\_tensor)

Attention : specifying any dilation\_rate value != 1 is incompatible with specifying any stride value != 1.