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 .”