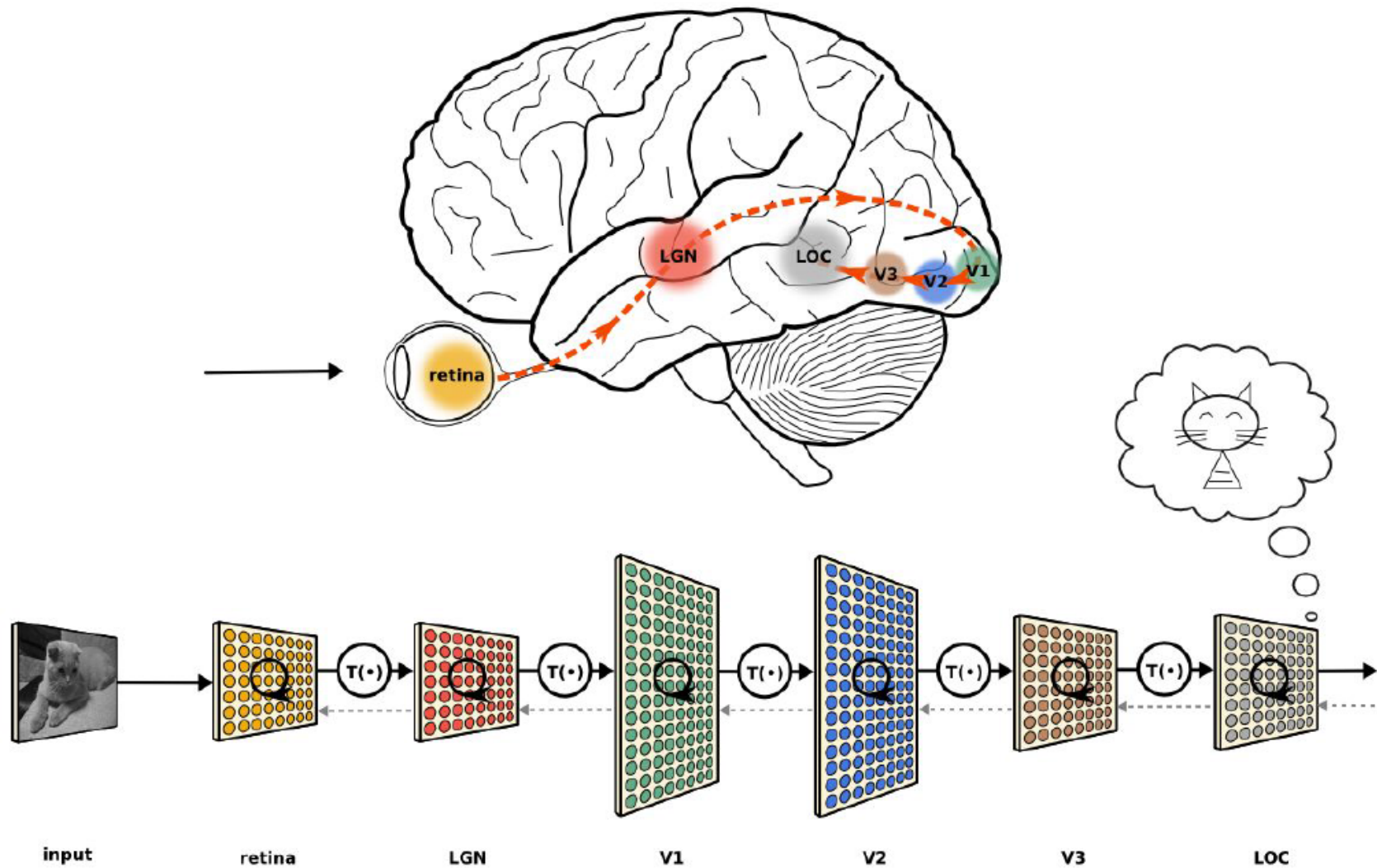




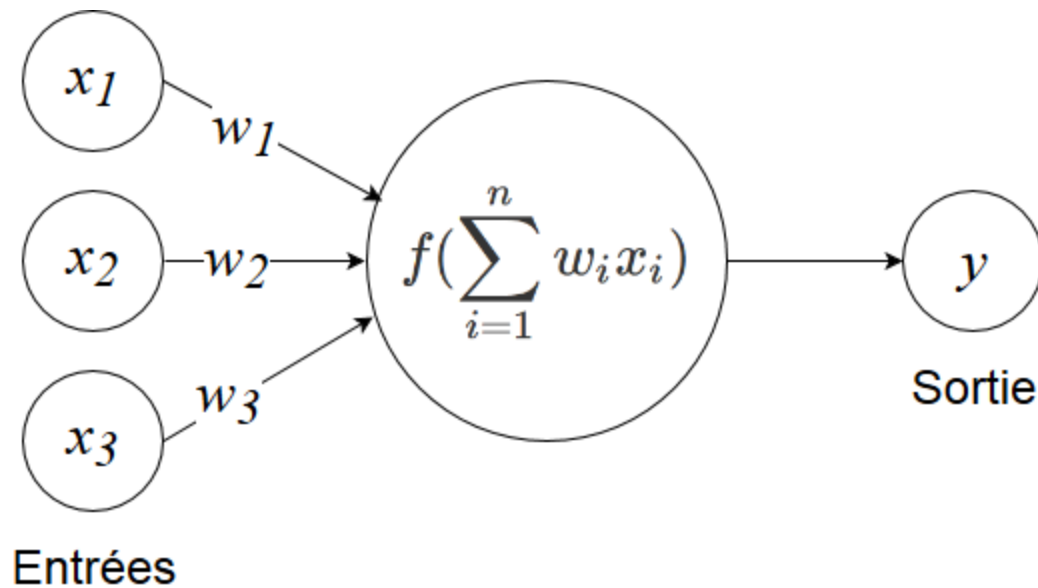
Introduction au deep Learning

Présentation partagée sous la licence Apache 2.0

Le Deep Learning

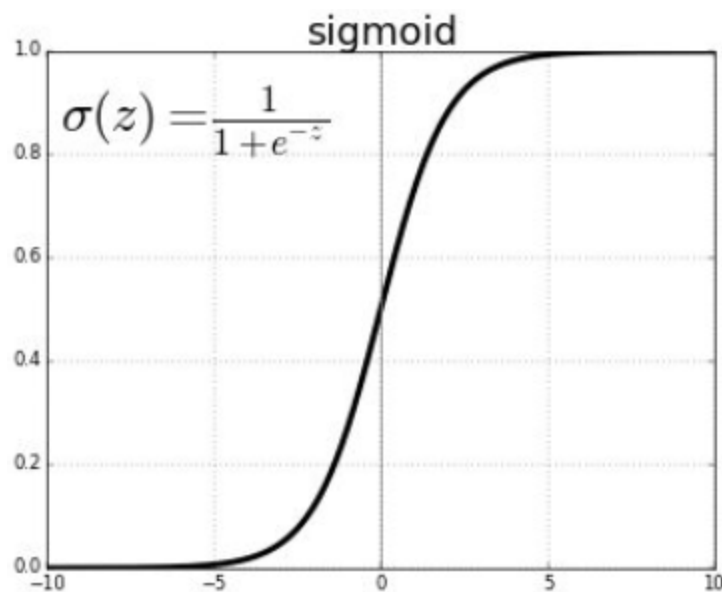


Un neurone

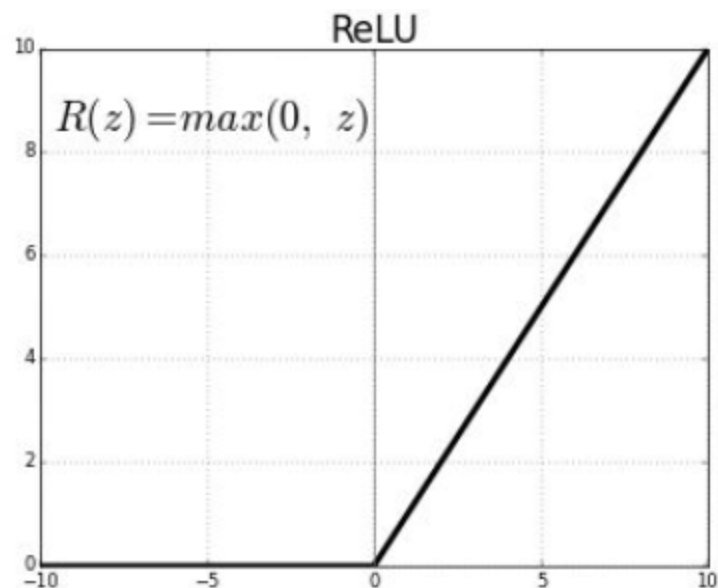


- f est la fonction d'activation
- Question : quelle fonction f choisir pour retrouver le modèle linéaire ?

Fonctions d'activation couramment utilisées



Utilisation : à mettre en fin de réseau pour prédire une probabilité (entre 0 et 1)

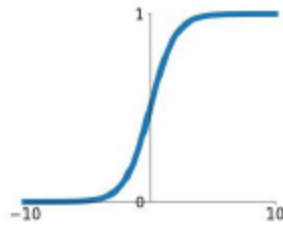


Utilisation : entre chaque couche pour dé-linéariser (à coût de calcul faible)

D'autres fonctions d'activation

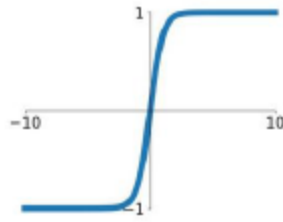
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



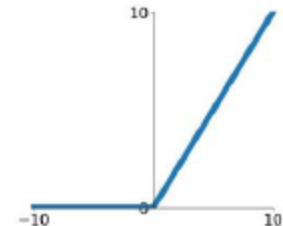
tanh

$$\tanh(x)$$



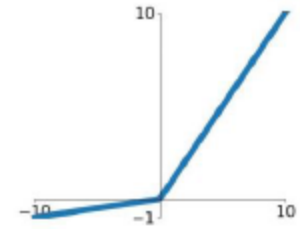
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

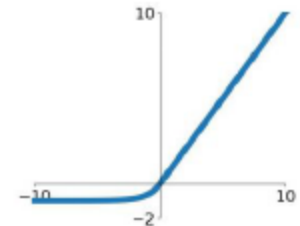


Maxout

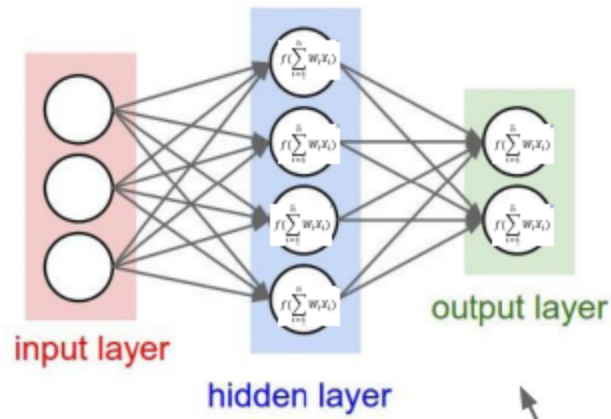
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

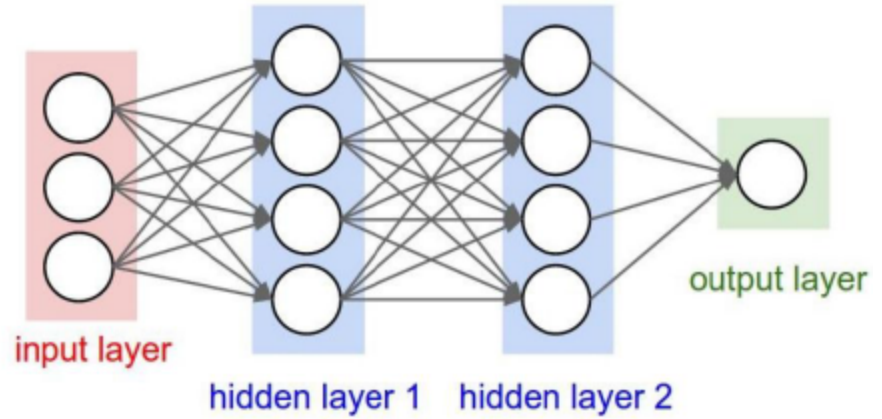
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Les couches / layers



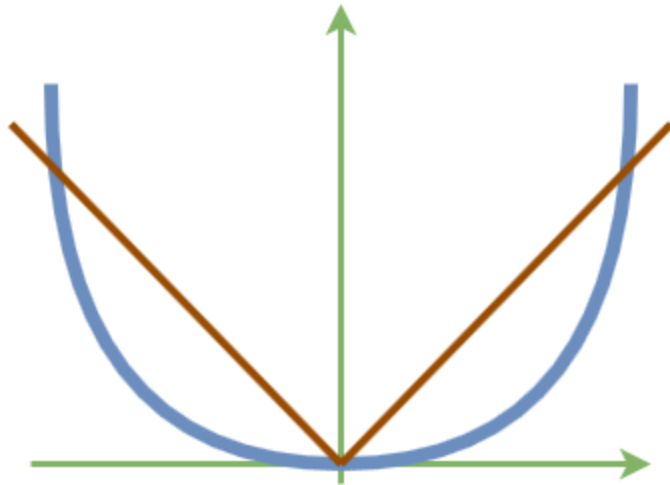
profondeur = 1



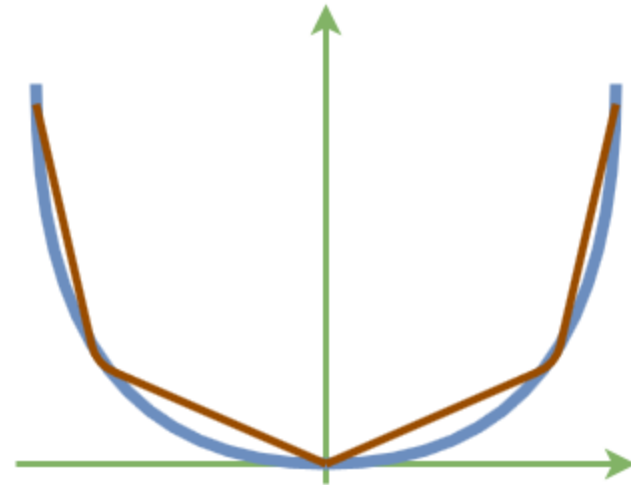
profondeur = 2

Intuition

Un réseau de neurones peut approcher n'importe quelle fonction.



Couche cachée à 2 neurones



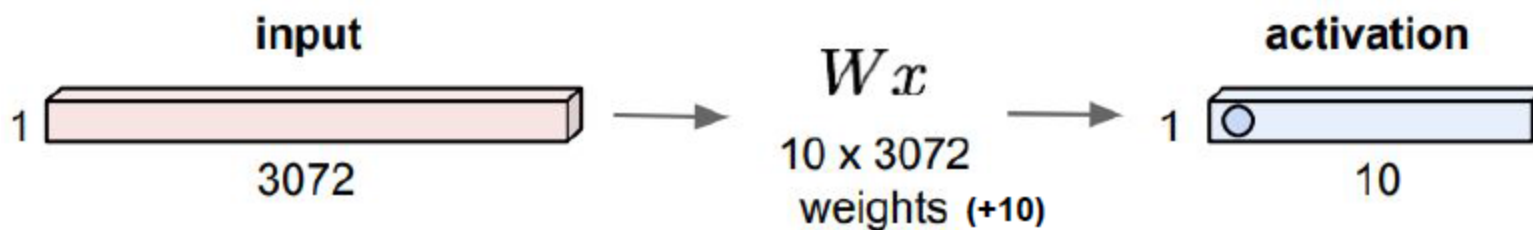
Couche cachée à 4 neurones

Classifier une image avec un réseau de neurones sans couche cachée

- Objectif : classifier une image 32x32 en 10 classes

32x32x3 image ->

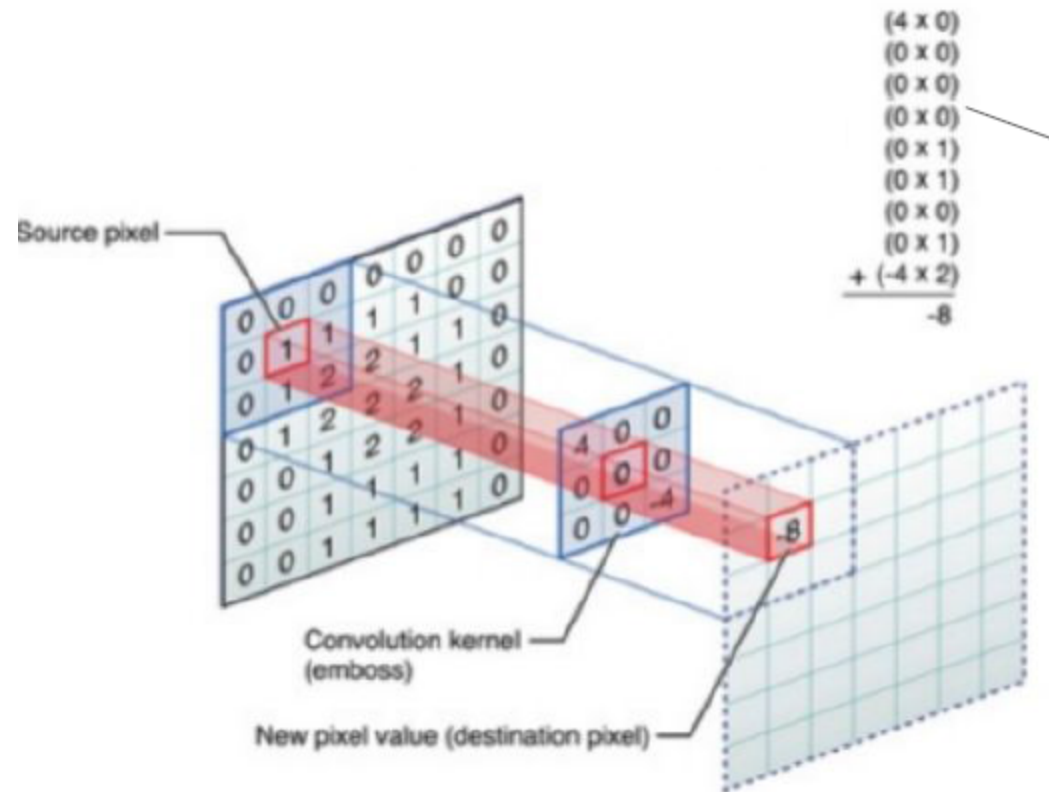
Aplatir en 1x3072



- Plus de 30 000 paramètres pour un petit réseau et une petite image
- Explose avec la résolution de l'image et la complexité du réseau

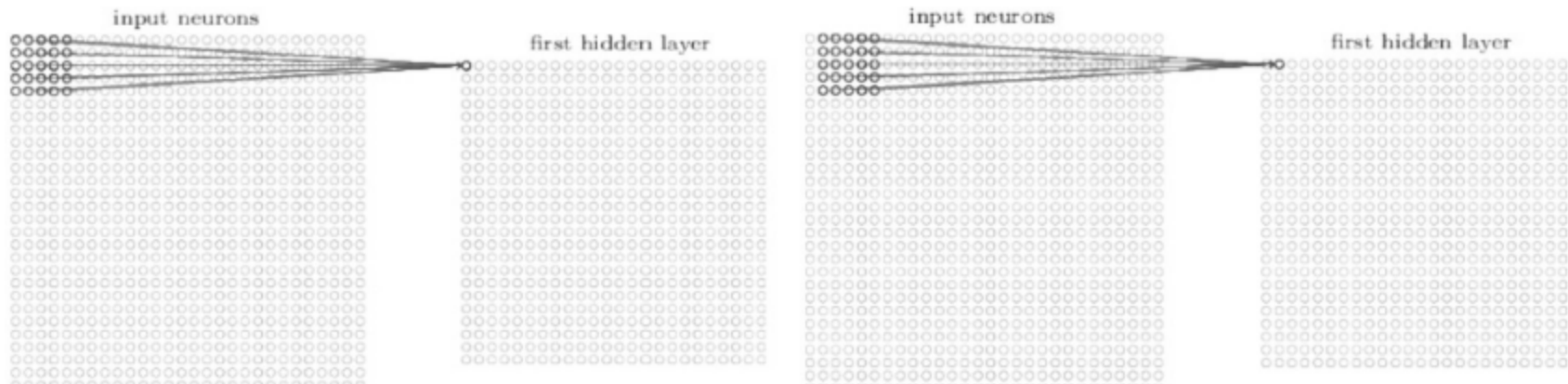
Réseaux de neurones convolutionnels

Convolution sur une image



- Multiplication pixel par pixel (produit scalaire)

Convolution sur une image

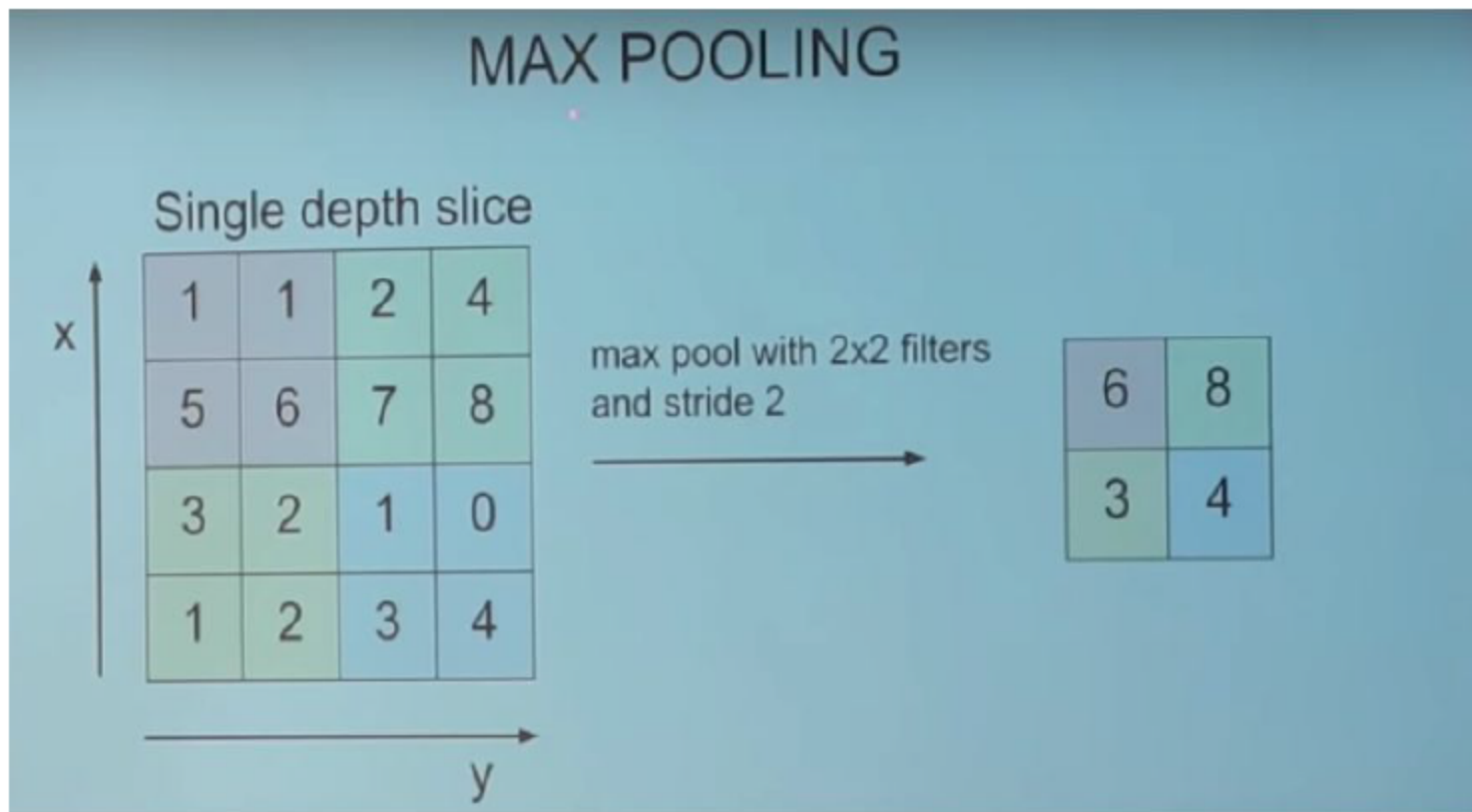


- 1 filtre 5x5
- Exemple en images : <http://setosa.io/ev/image-kernels/>

Autres types de couches

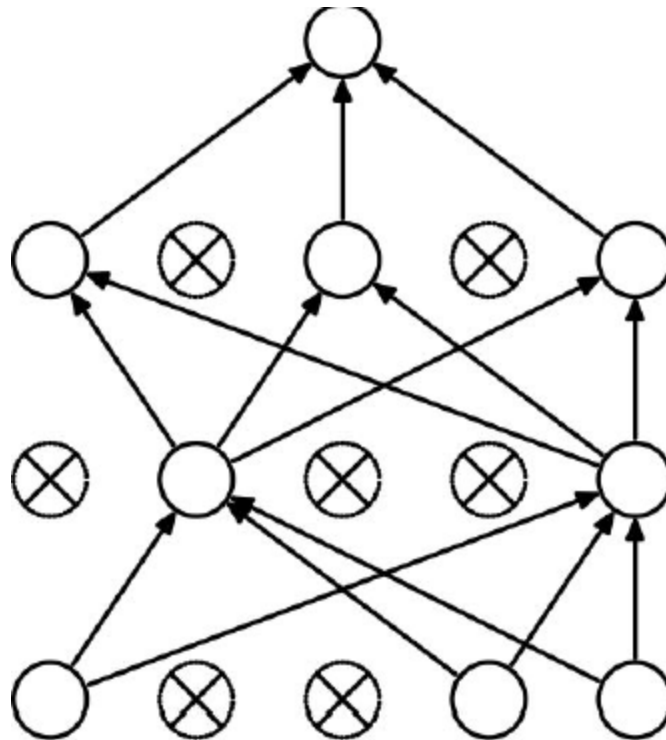
- MaxPooling
- DropOut

Max Pooling : Réduire la dimension



Dropout : supprimer aléatoirement des neurones

Méthode de régularisation



Exemple de réseau convolutionnel complet

