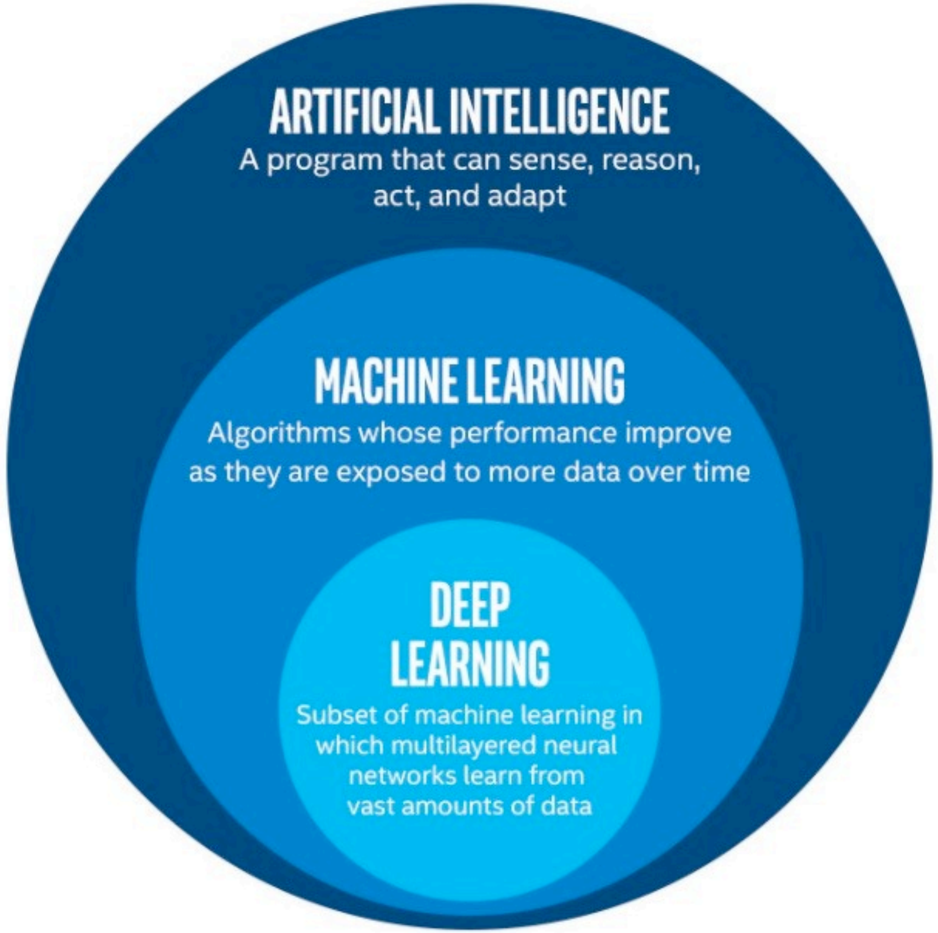
Machine learning

Autor: Cristopher McIntyre Garcia

Mini-Cours

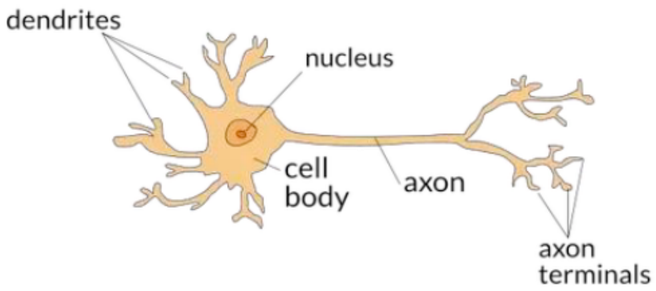
- ~~Jour 1: Introduction à l’image digital~~
- ~~Jour 2: Techniques de base de la vision artificiel et au traitement d’image~~
- ~~Jour 3: Filtre et technique traditionnel avancé~~
- Jour 4: Introduction a l’intelligence artificiel (ML et DL)
- Jour 5: Technique modern - Conversation de nos courent

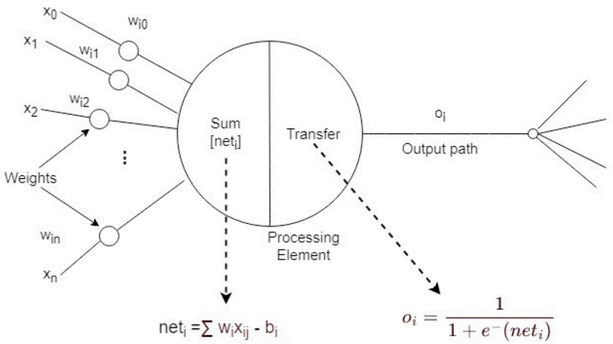
AI vs ML vs DL

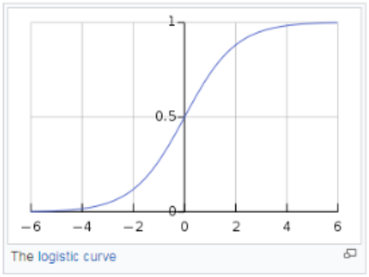

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

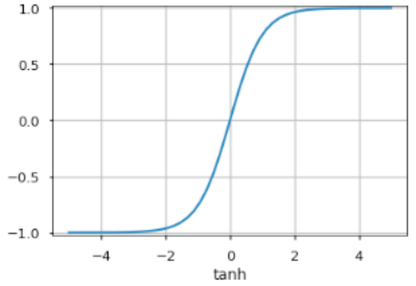
Neurone

- Entrée et sortie
- Poids calculé pendant la formation
- Sommation
- Fonction d’activation
  - Sigmoid, relu, tanh



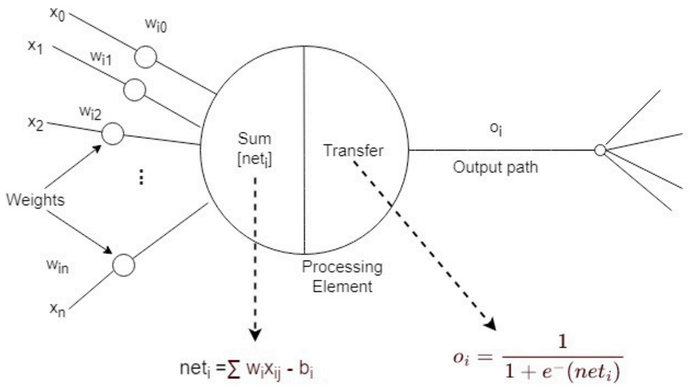


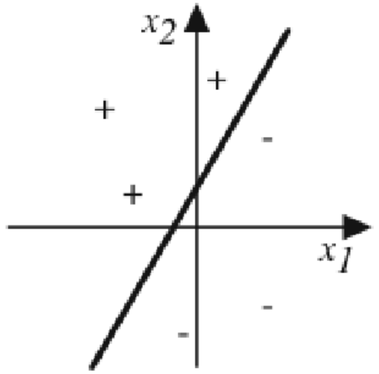


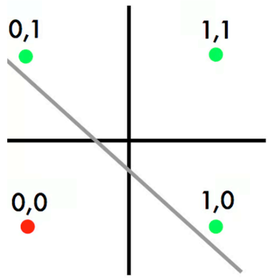


Perceptron

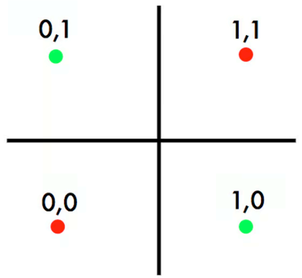
- Représente Limites linéaires
- Problème XOR
- Perceptron multicouche



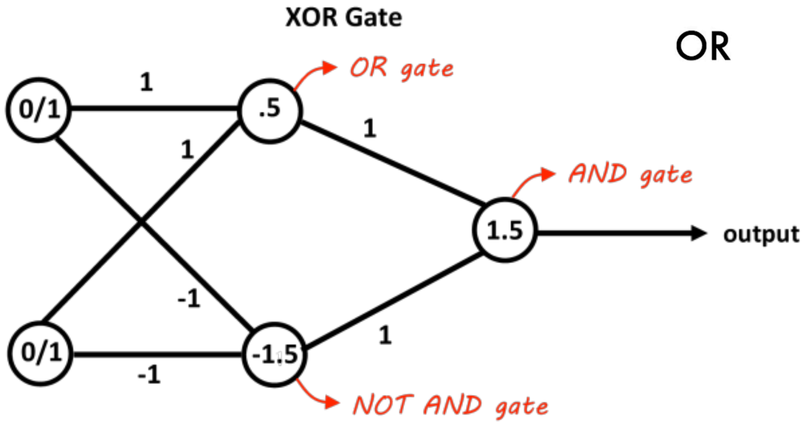




OR

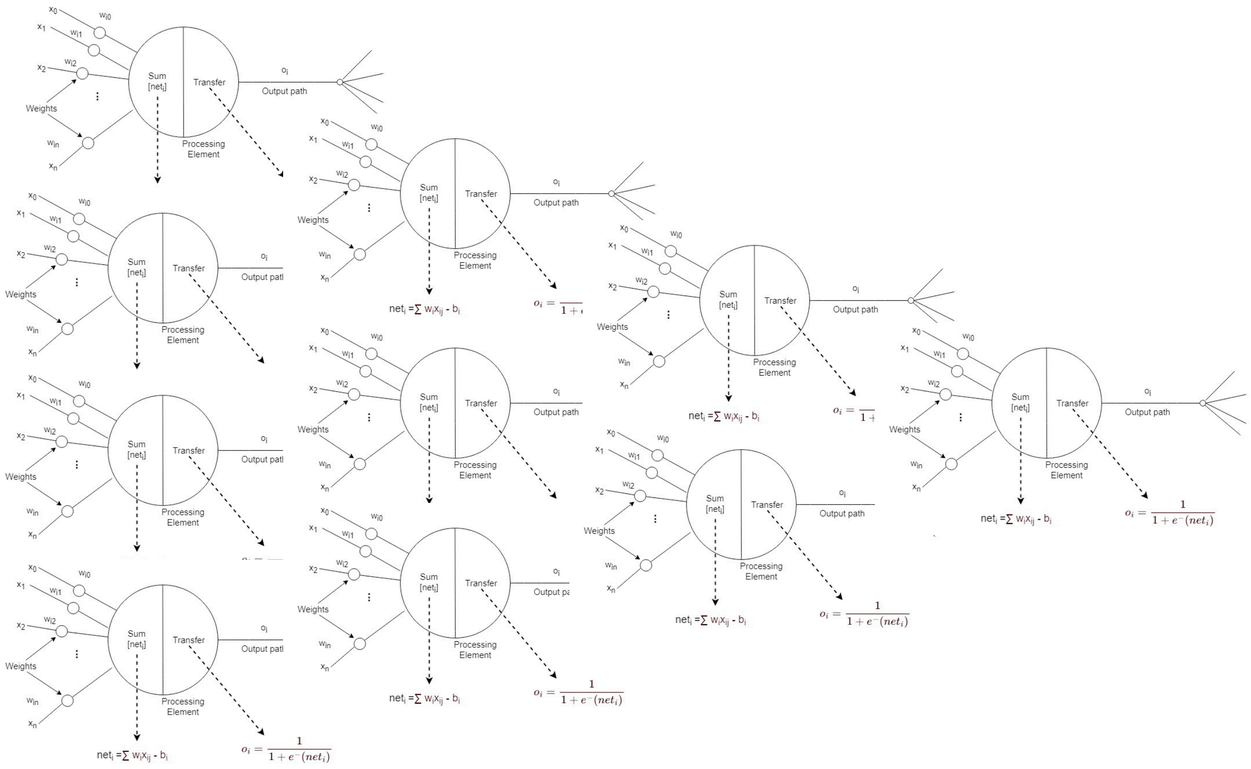


XOR



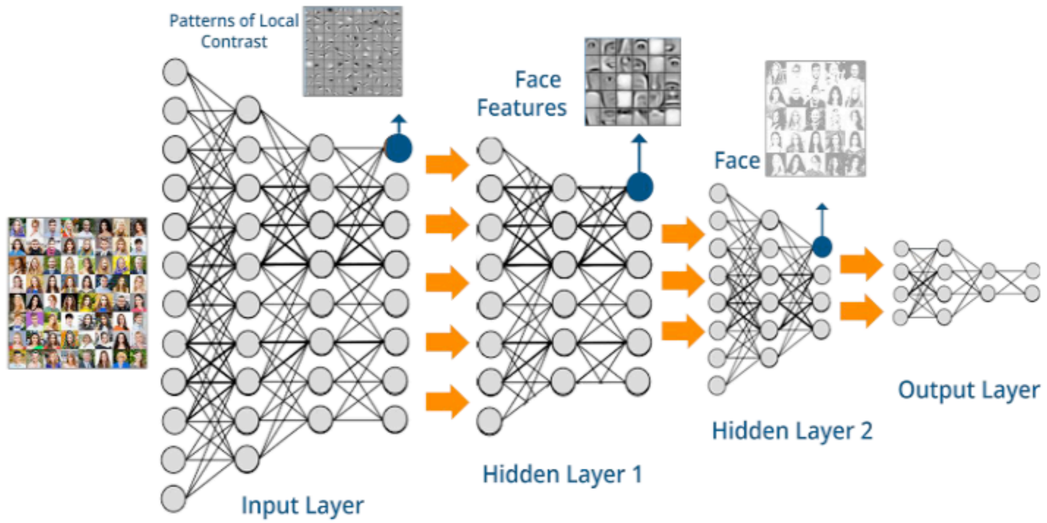
Multilayer perceptron

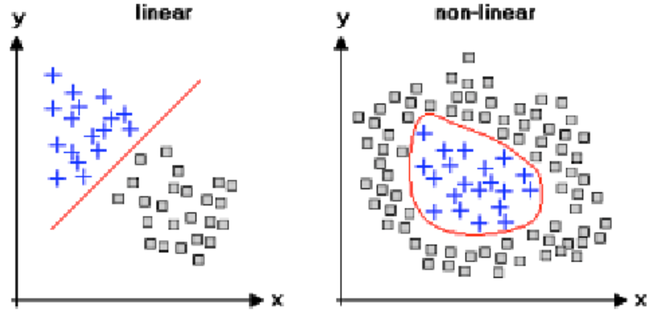
- Plusieurs neurone



Multilayer perceptron

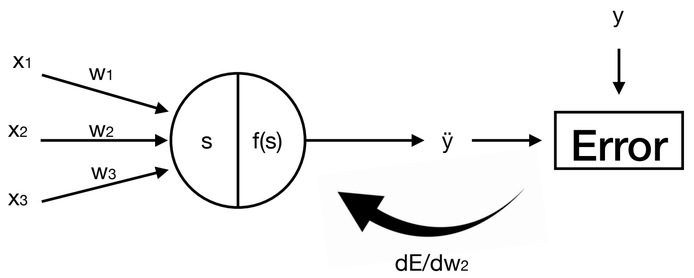
- Plusieurs neurone
- Caractéristiques
- Couches
- Modèle
- Formation pour plusieurs tâches

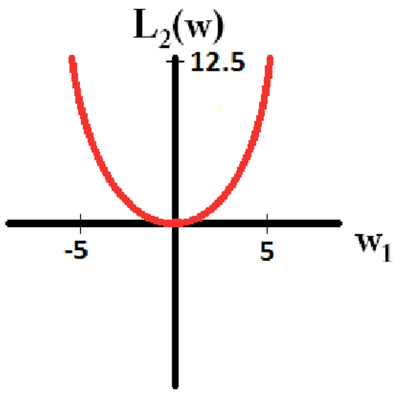


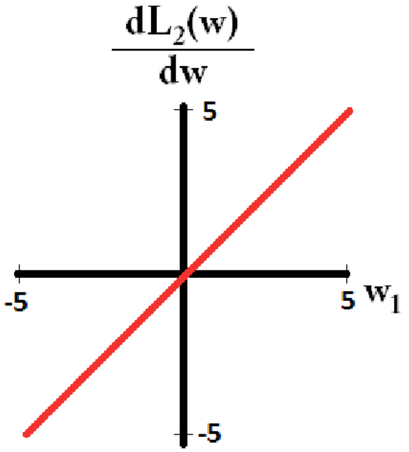


Formation

- Erreur entre vrais et prédit
  - Vrais: y
  - Prédit: ÿ
- Erreur L2
- Gradient (dérivé)
  - Impact du poids

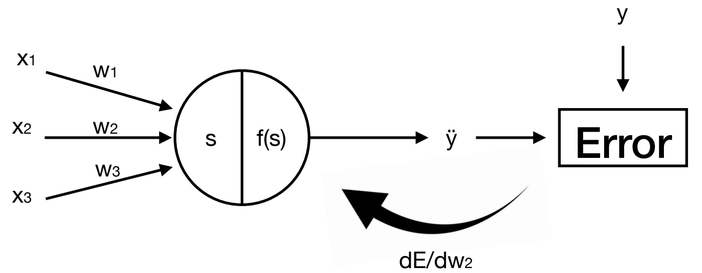


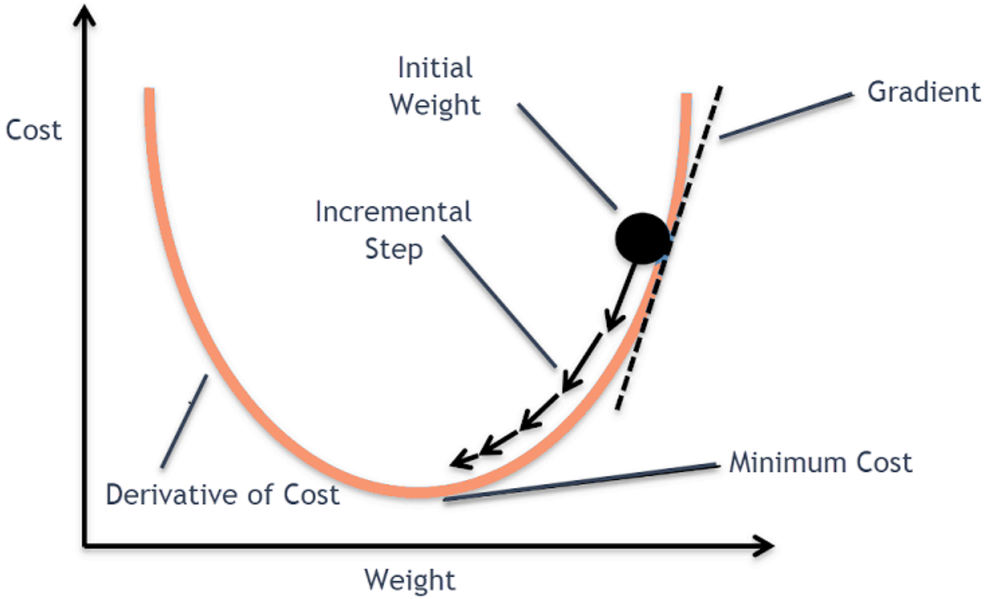




Gradient décent

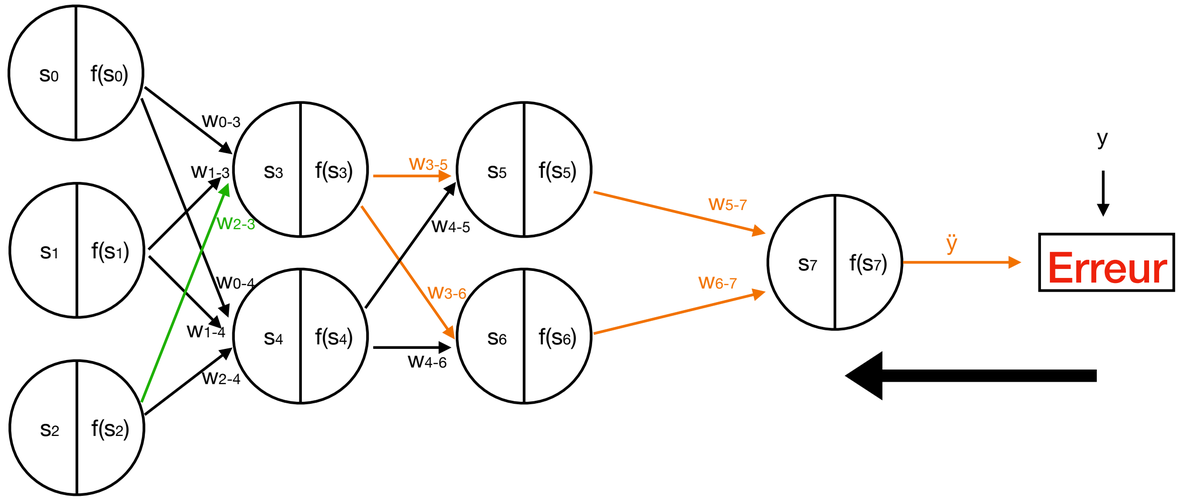
- Erreur minimal
- Étapes incremental
- Gradient dirigent
- Modifie le poids a chaque étape
- Chaine





Règle de la chaîne

- Dérivé partiel
- Fonction d’erreur
- Chaîne de l’erreur au poids



- Automatic avec Pytorch

$dE/dw_{2\text{-}3} =$

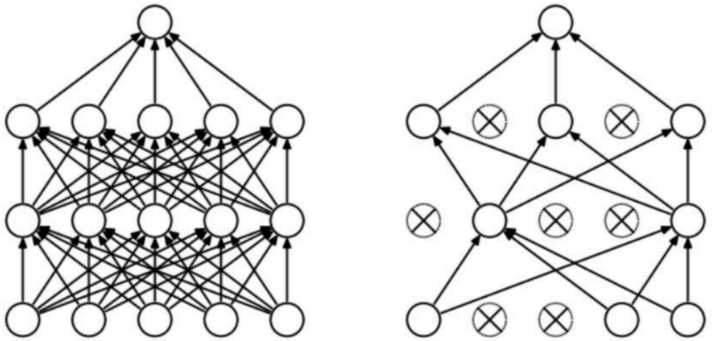
$dE/df(s_7) \times df(s_7)/ds_7 \times$

$(ds_7/df(s_5) \times df(s_5)/ds_5 \times ds_5/df(s_3) + ds_7/df(s_6) \times df(s_6)/ds_6 \times ds_6/df(s_3)) \times$

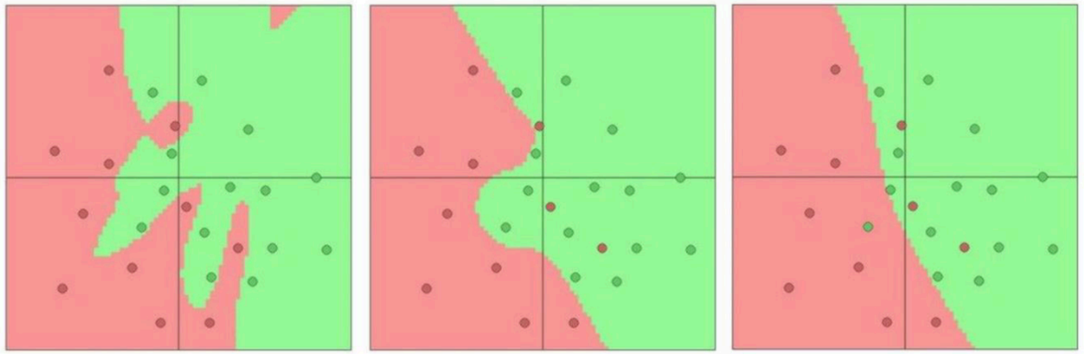
$df(s_3)/ds_3 \times ds_3/dw_{2\text{-}3}$

Régularisation Dropout

- Pas dépendre fortement d'un neuron particulier
- Sur-ajustement

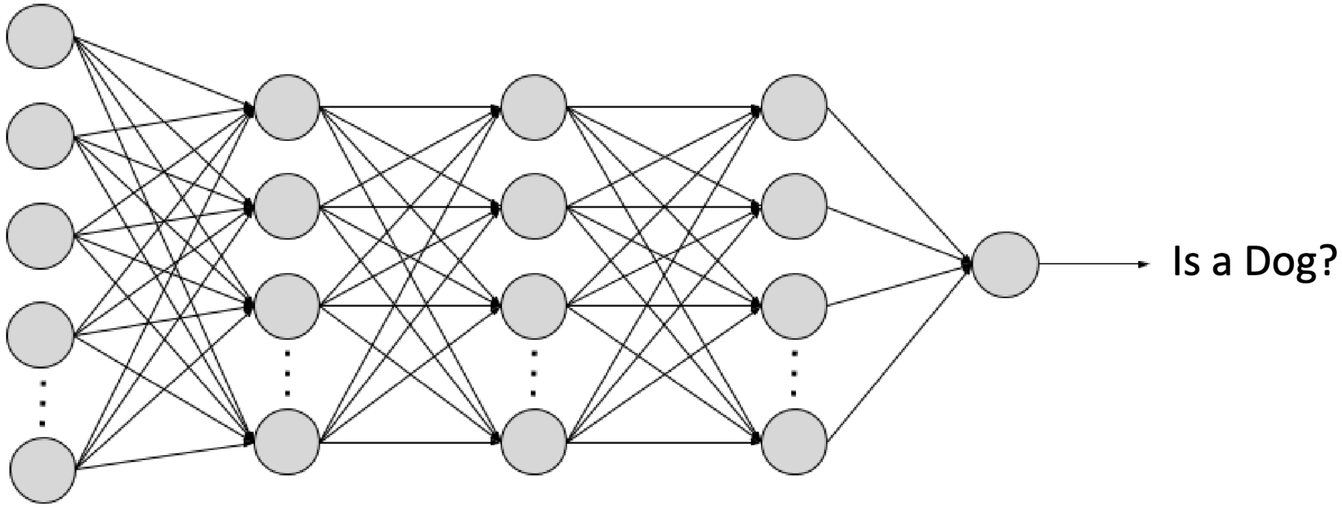


(a) Standard Neural Net (b) After applying dropout.



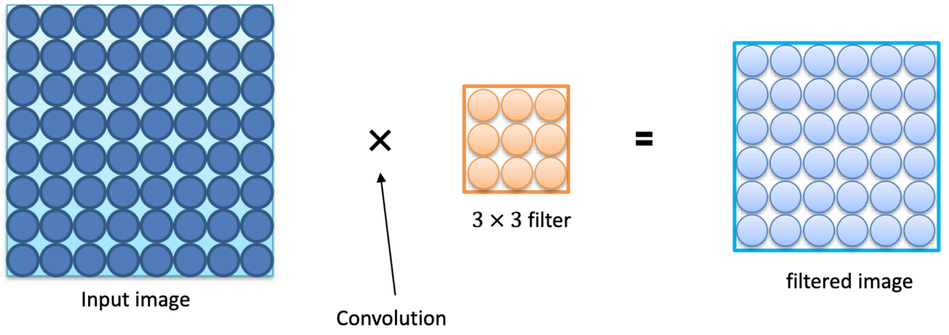
Réseau entièrement connecté sur les images

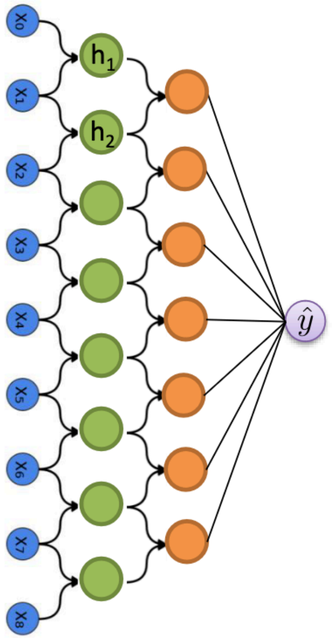
- Un neurone pour chaque pixel
- Trop de poids entre les couches
- 3 Milliard dans la première couche



Convolutional neural network

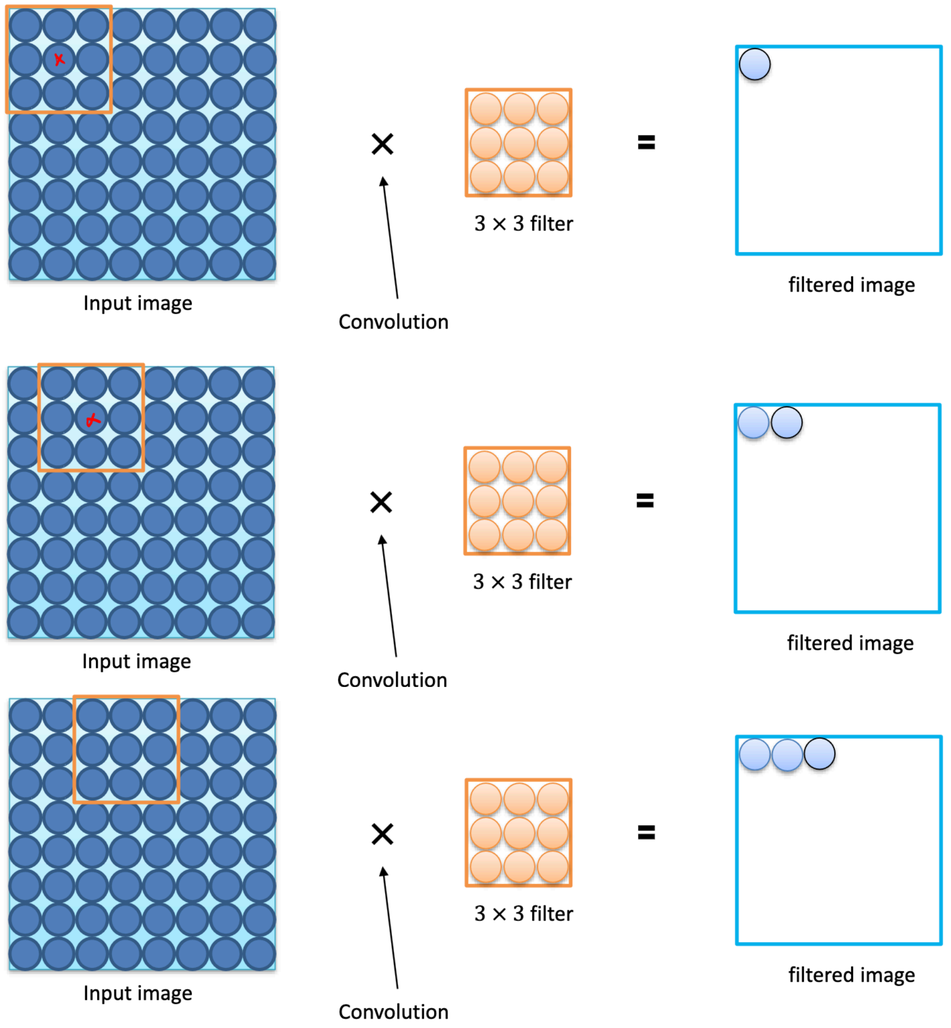
- Connecter chaque unité cachée à un patch de neurones de la couche précédente
- Partager la matrice des paramètres
- Entrées locales





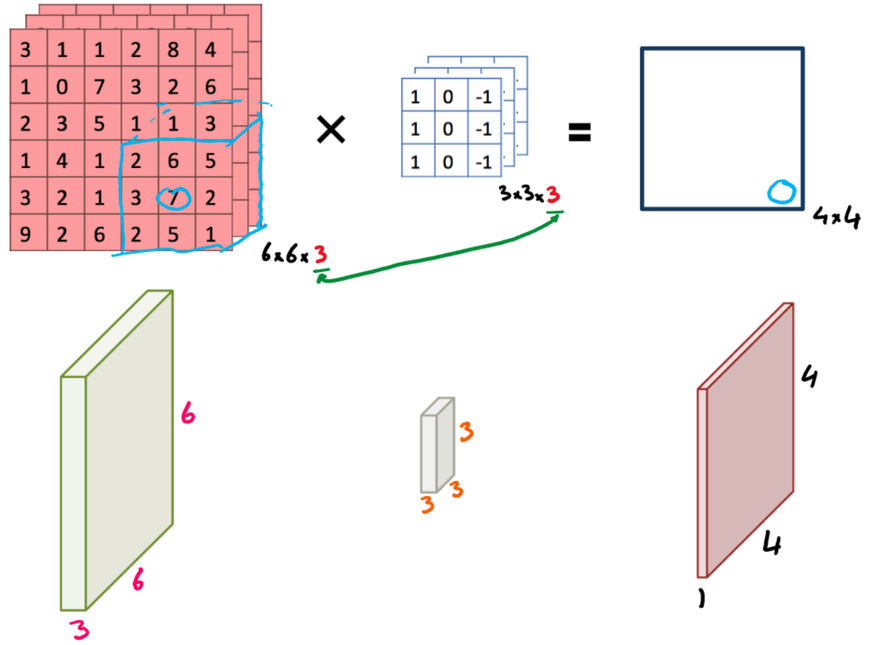
ConNet

- Faites glisser le filtre
- Extraction des paramètres



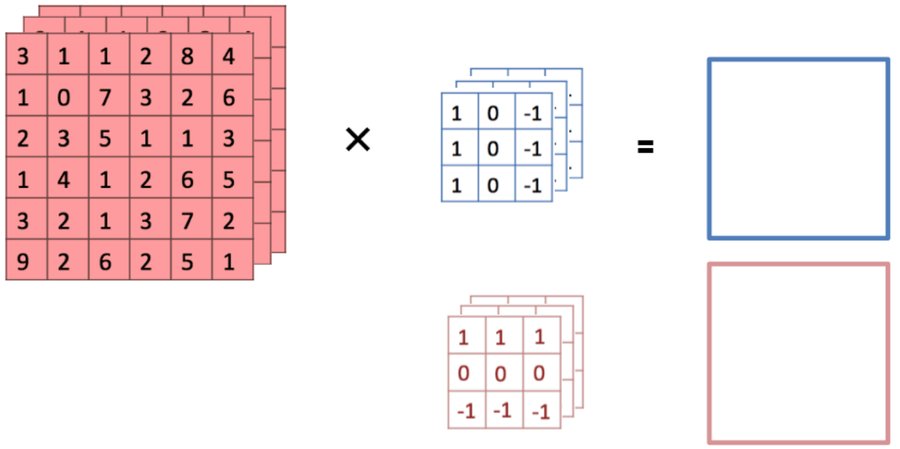
Convolution sur RGB

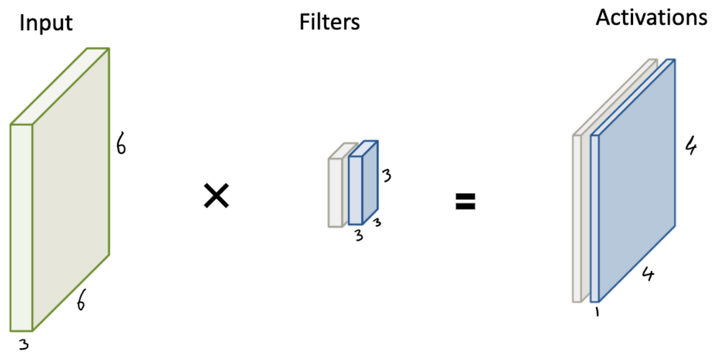
- Pas de glissement dans la direction du canal
- Résultat est une matrice 2D



Plusieurs filtre

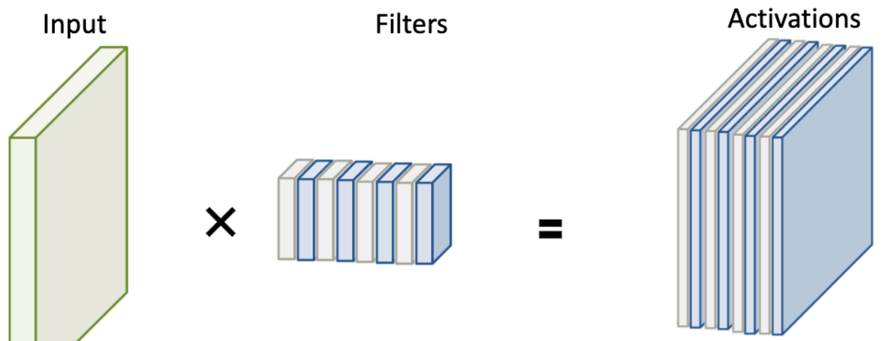
- Résultat augmenté
- Capture the propriété augment
- Filtre Prewitt

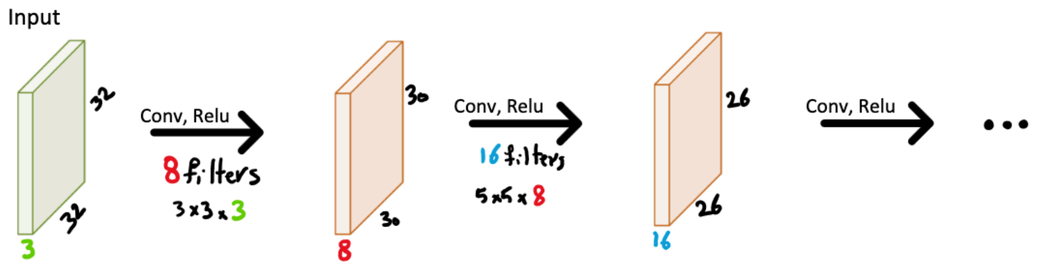




Plusieurs filtre

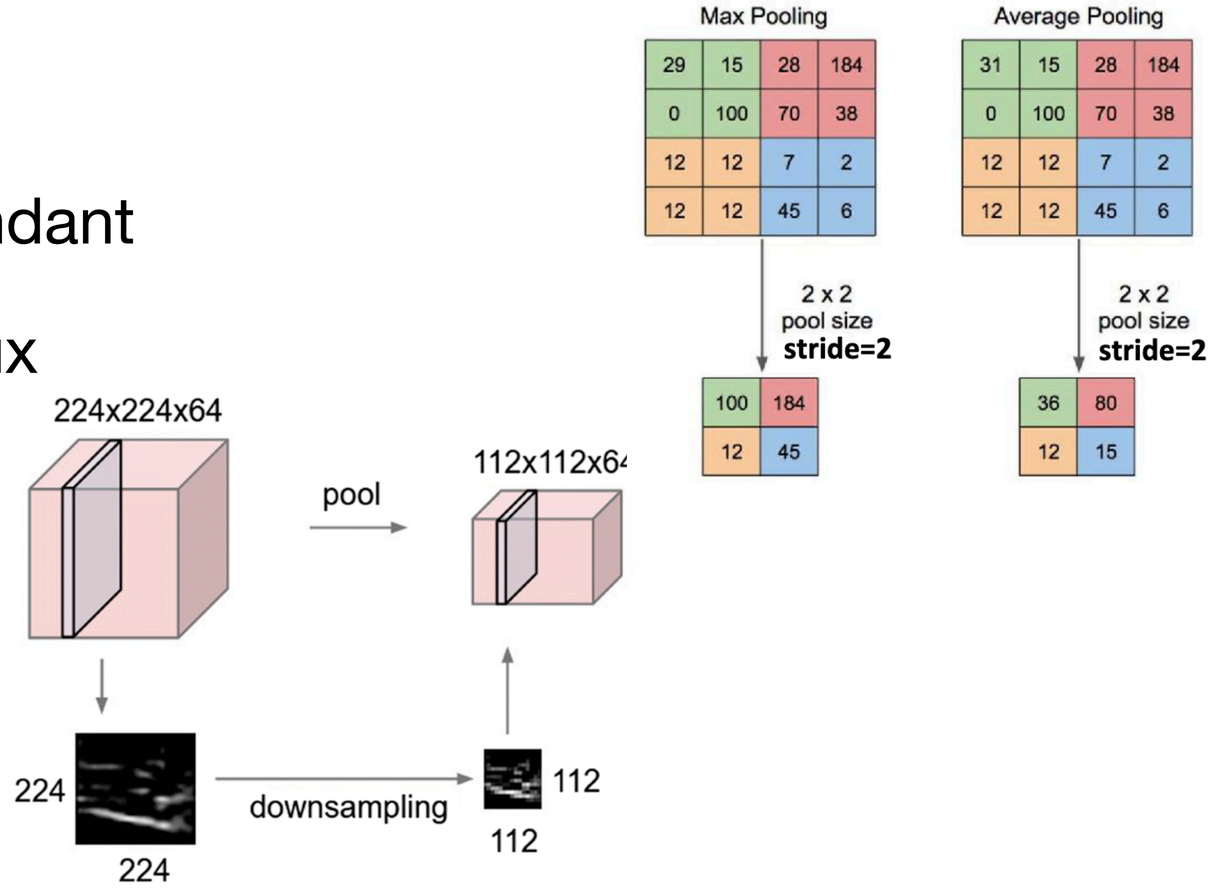
- Augmentation de propriété découvert



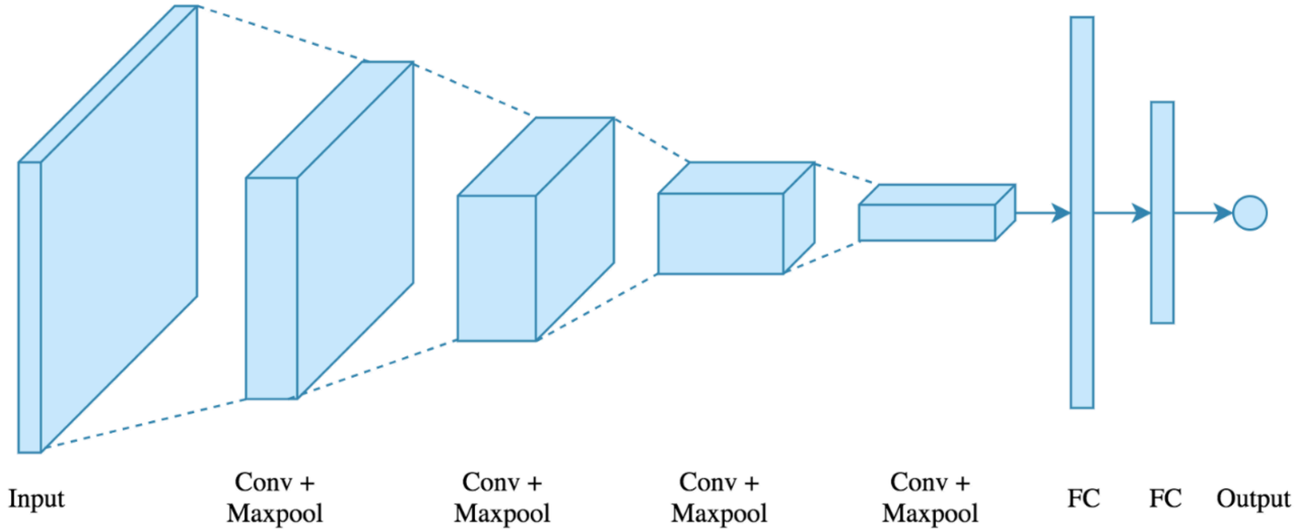


Pooling

- Une couche de pooling
- Effectue un échantillonnage descendant
- Ne modifie pas le nombre de canaux
- Pas de paramètres à apprendre

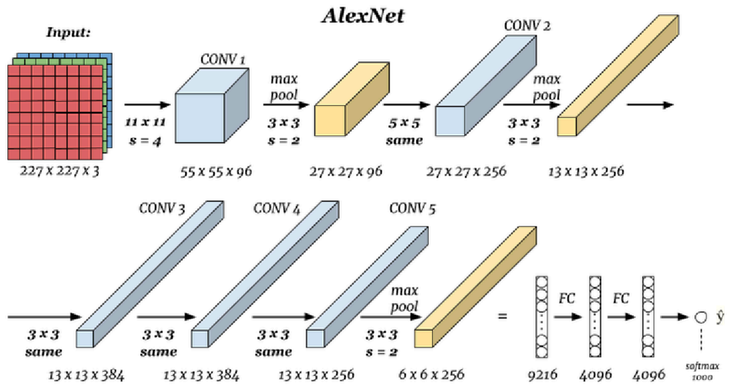


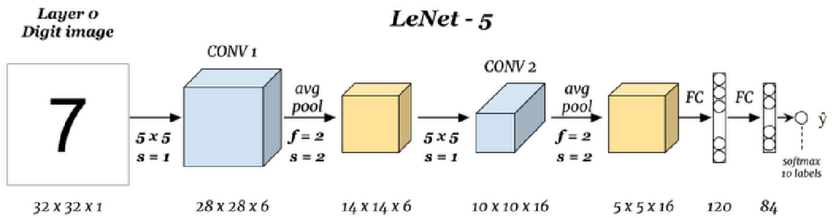
Example de ConvNet

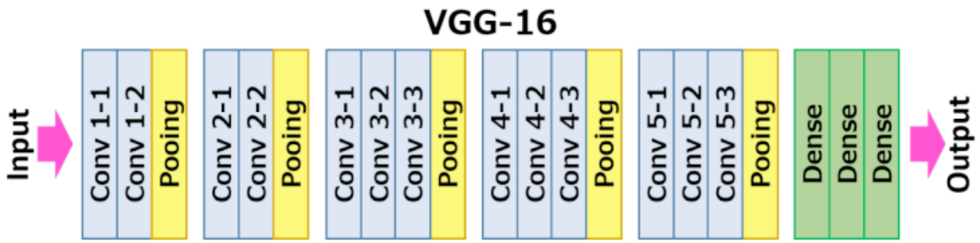


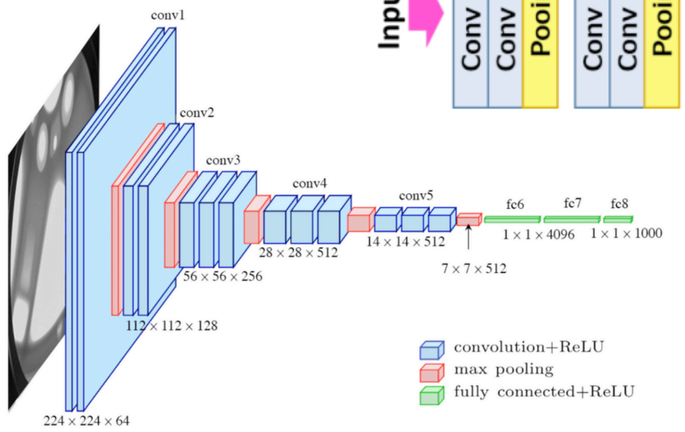
ConvNets

- LeNet
- AlexNet
- VGG



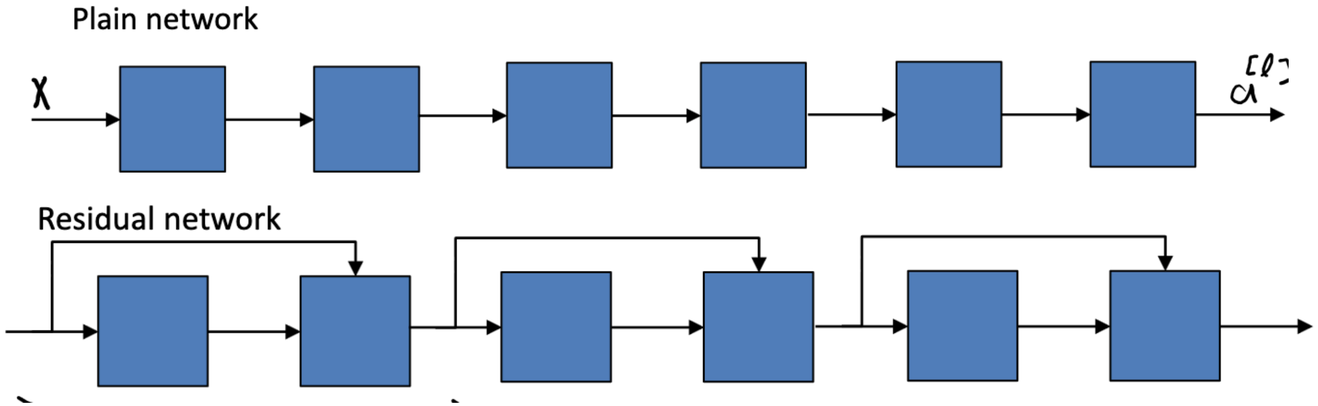


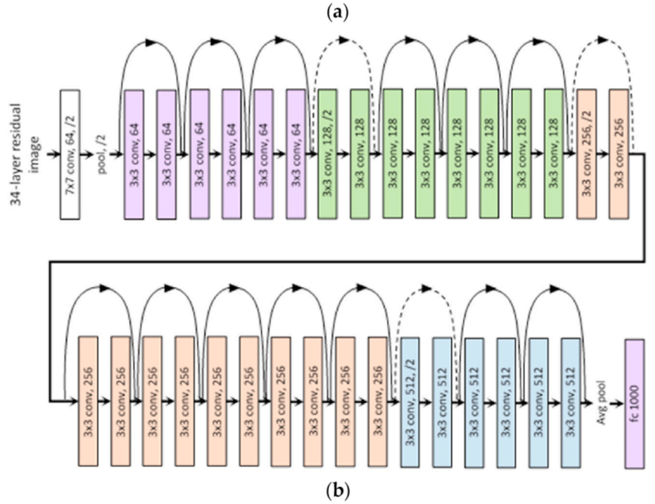




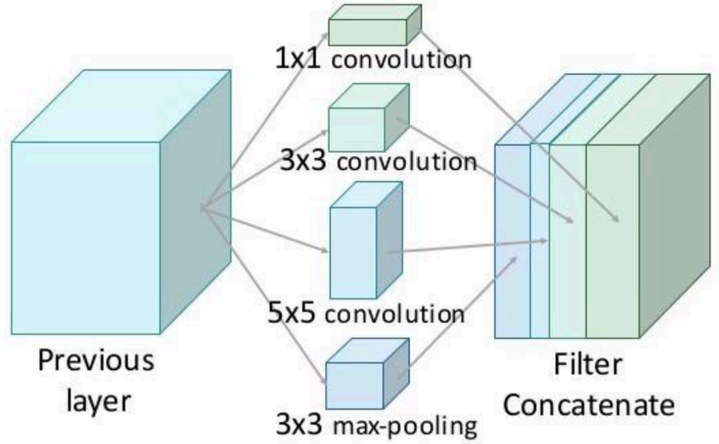
Blocs résiduels

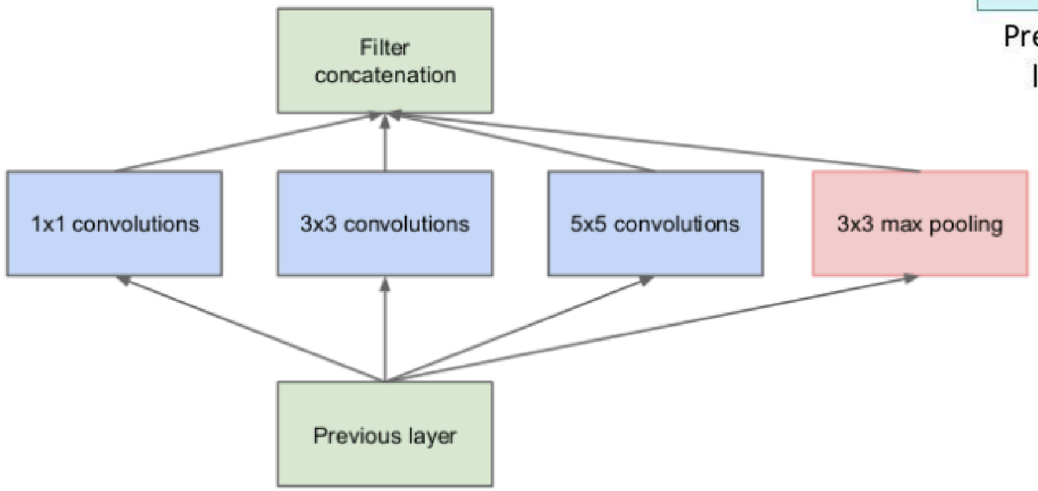
- Réutilisation des fonctionnalités
- Stabiliser l'apprentissage
- Débit de gradient renforcé

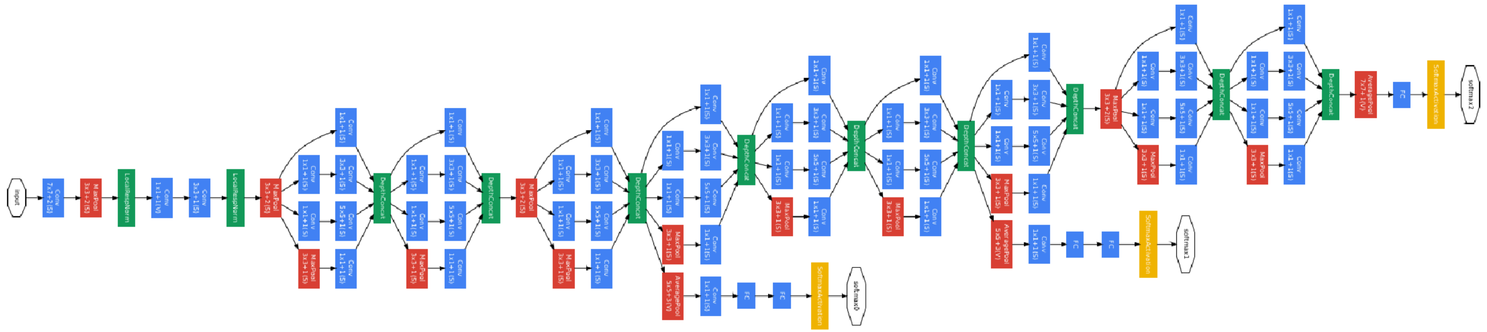




Réseau Inception

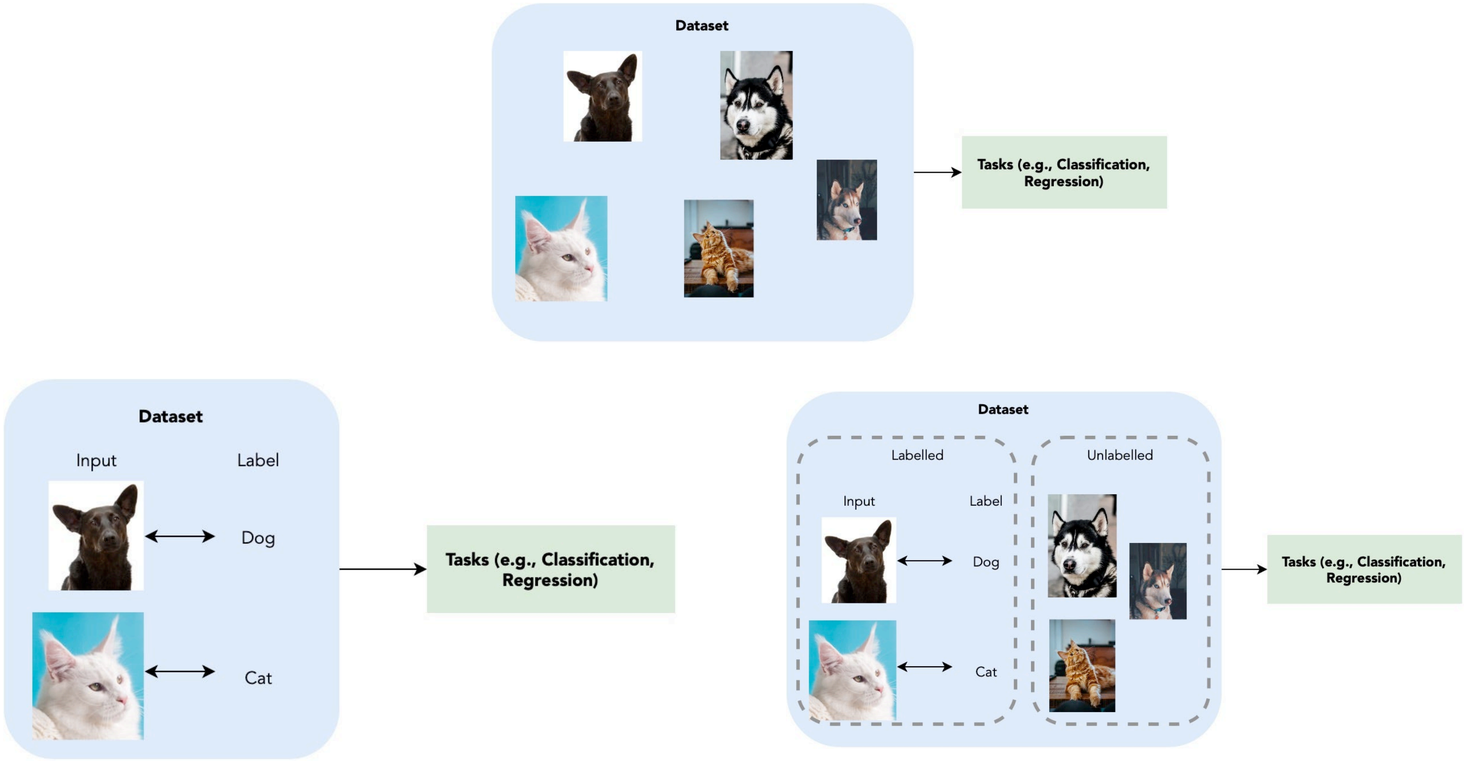






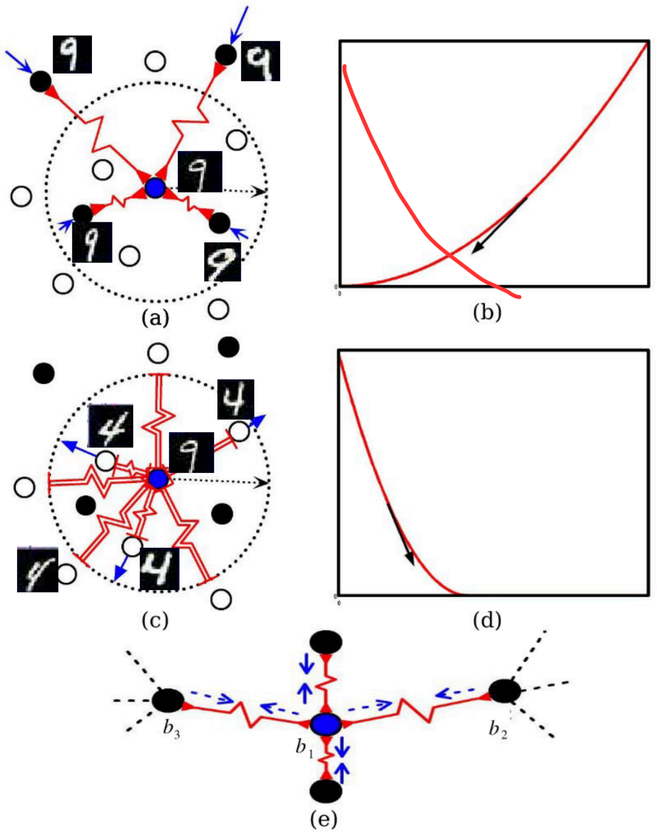
Les différents types d'apprentissage

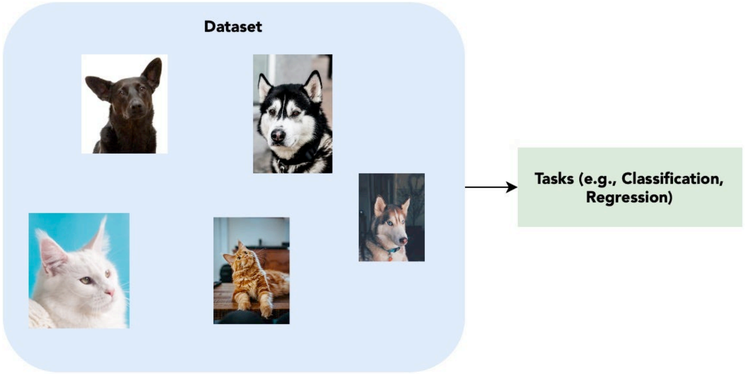
- Supervisé
- Semi-supervisé
- Non supervisé

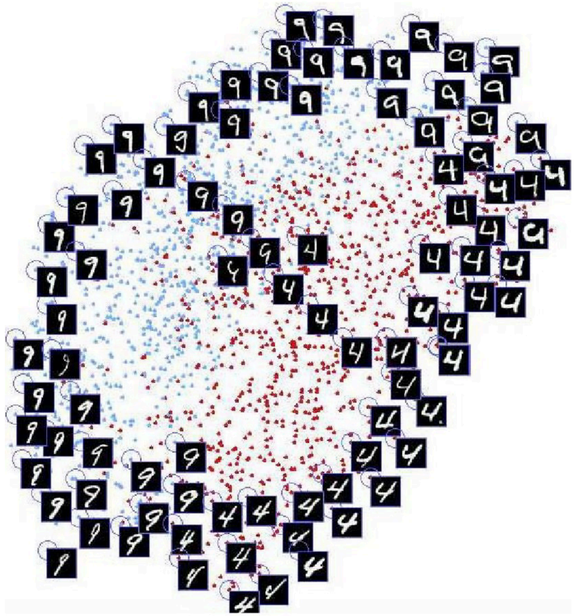


Apprentissage non supervisé

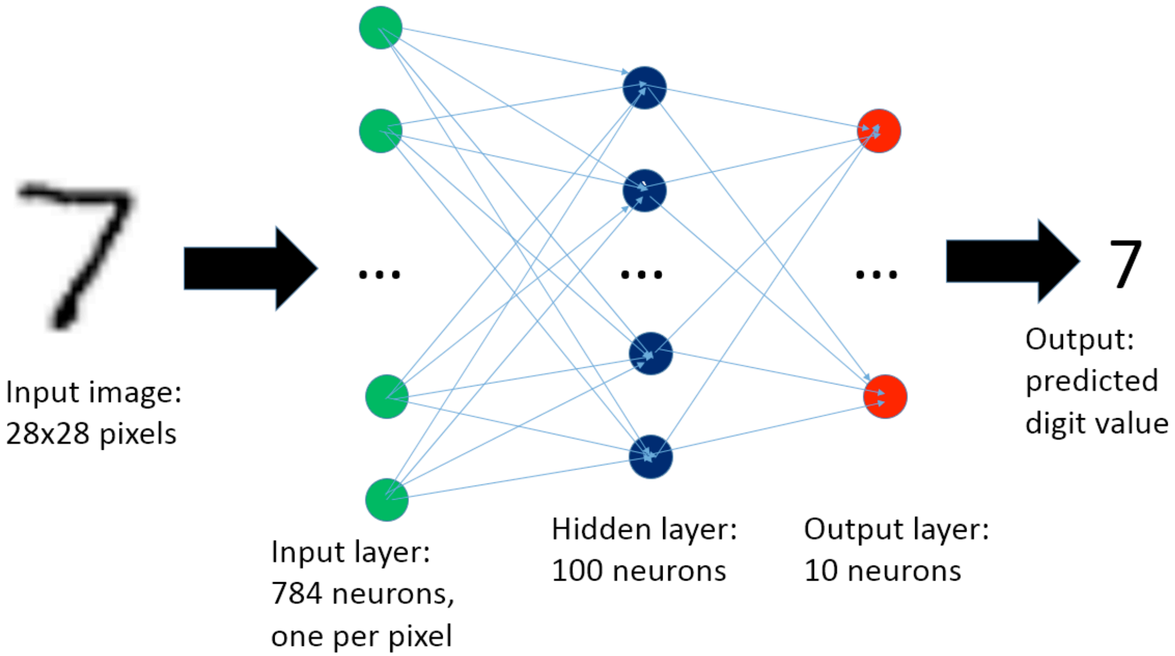
- Espace latent
- Attirance et repoussement

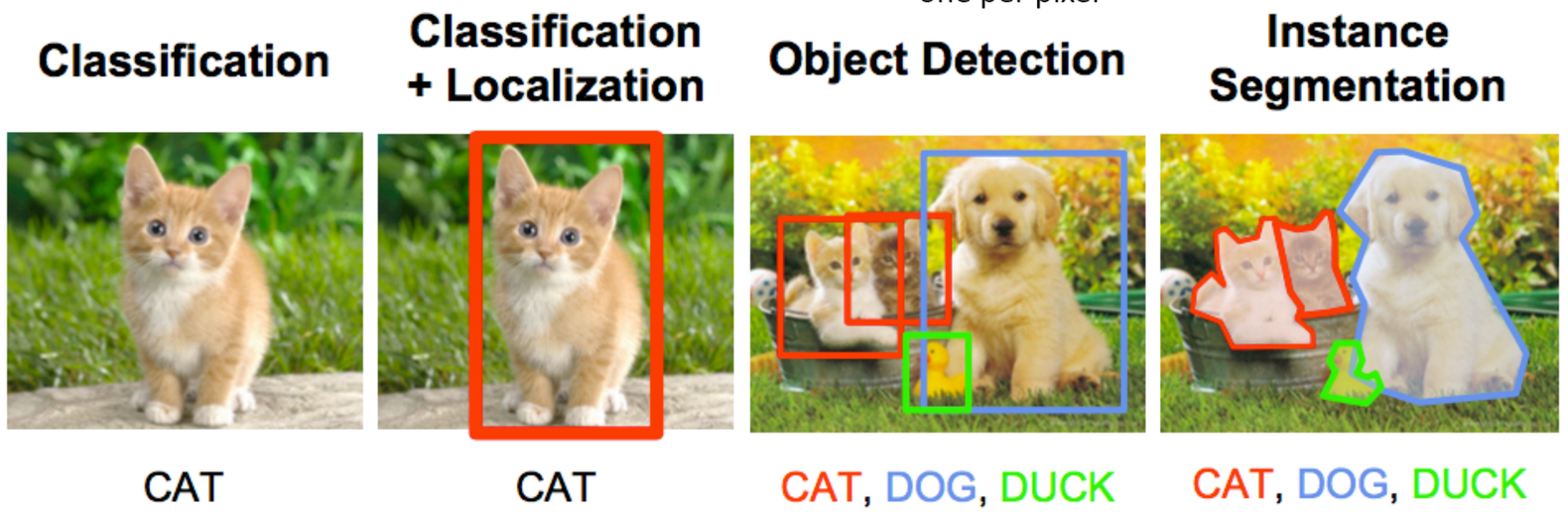






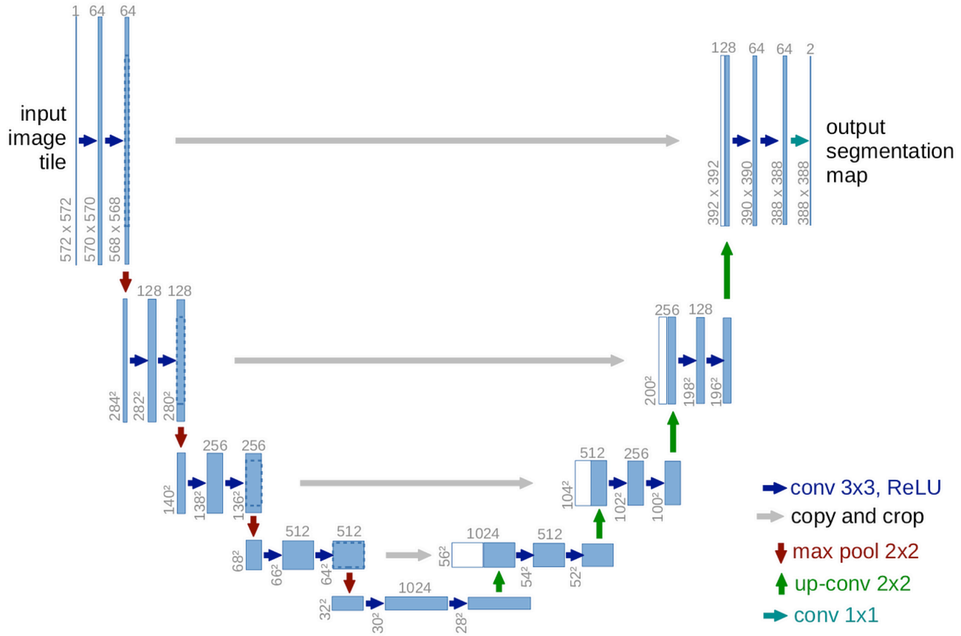
Applications

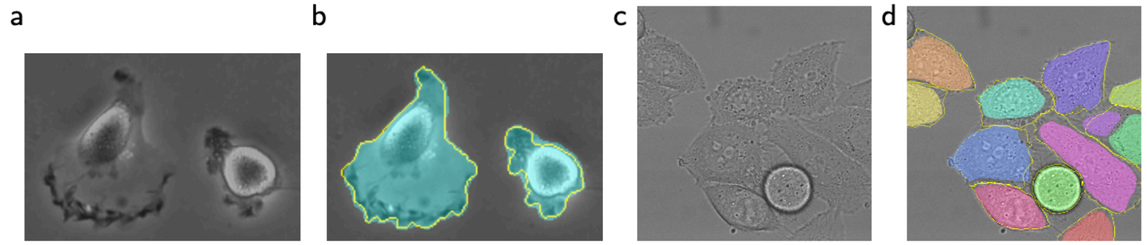




Segmentation sémantique

- UNet
- Image médical





Python + OpenCV

- GitHub
  - https://github.com/cmcin019/mini-cours
- Google Colab

Fin

- Merci