



Ecole Thématique Internationale de l'Intelligence Artificielle au Cameroun (ETIIAC)

INTRODUCTION AU DEEP LEARNING

Dr Rose MOSKOLAÏ

26 novembre 2024

Objectifs de la présentation

- Comprendre les concepts de base de l'apprentissage profond (Deep Learning) et ses différences avec l'apprentissage automatique (Machine Learning).
- Explorer les principales architectures DL
- Aborder les méthodes d'évaluation et d'amélioration des modèles.



Plan de la présentation

- Généralités
- Structures et Fonctionnement
- Principales architectures
- Evaluation et amélioration des modèles
- Technologies utilisées

Historique de l'IA



1952 : premier logiciel qui joue aux échecs

1997 : **IBM Deep blue** bat le champion d'échecs

2016 : **AlphaGo** champion du monde du jeu de Go

2018 : invention des Transformers

2020 : lancement de GPT-3 par **OpenAI**

2021 : DALL-E permet de générer des images

2024 : lancement de GPT-4 par **OpenAI**

2024 : IA générative, LLM (Large Language Models)

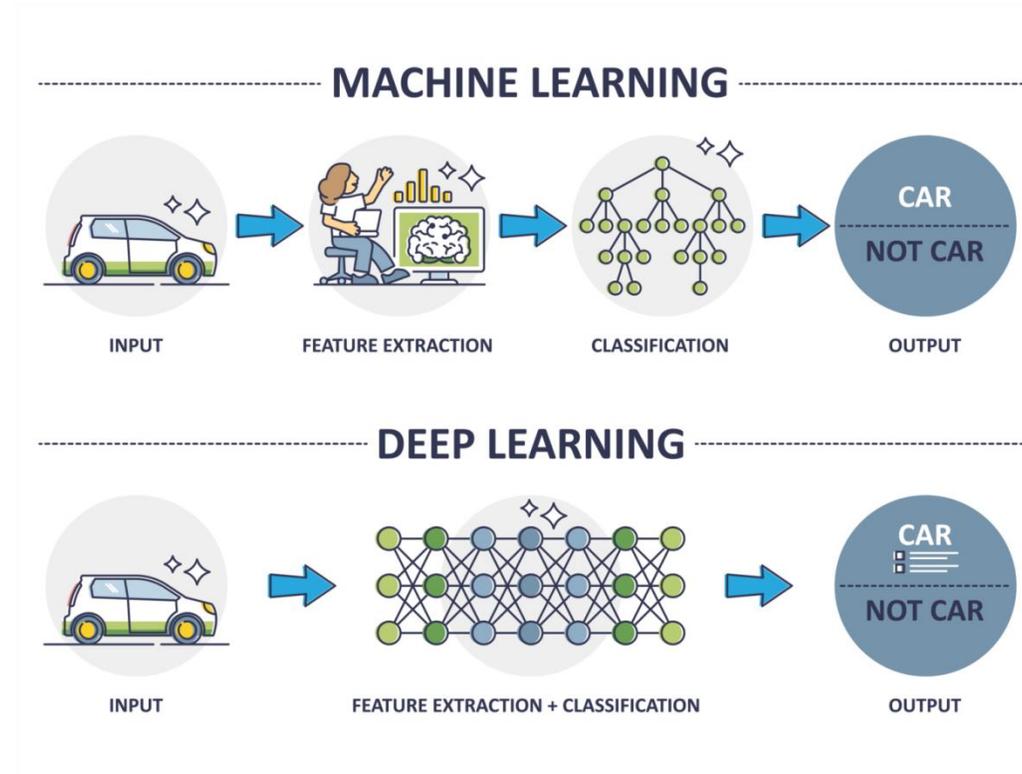
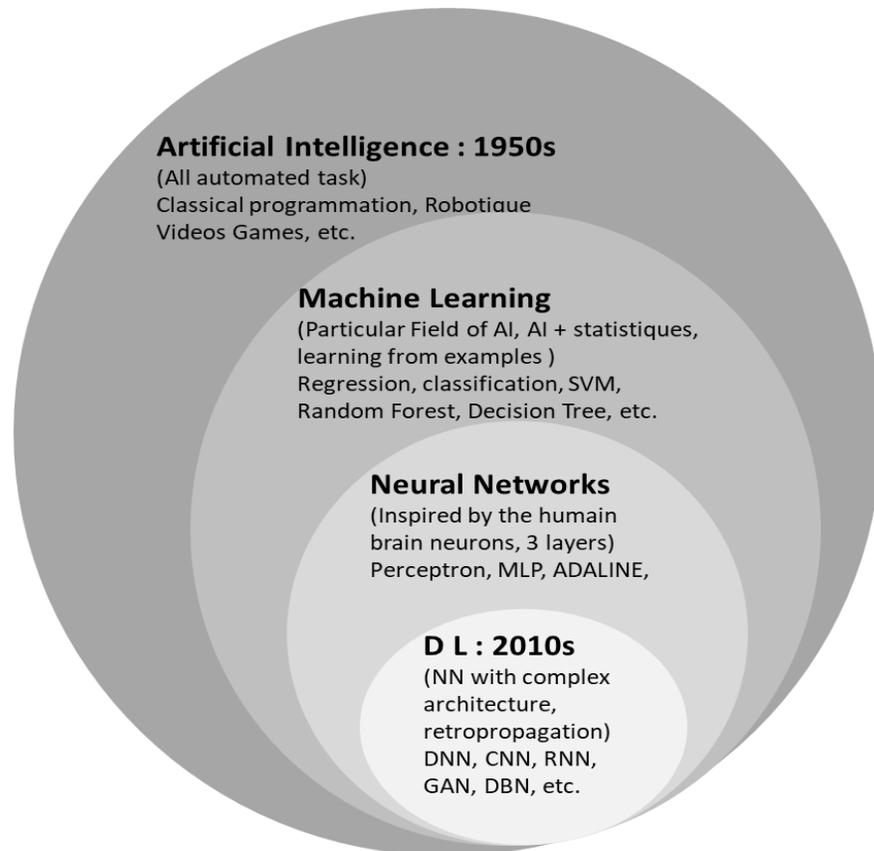
Définition

Deep Learning (DL) : partie de l'IA et de l'apprentissage automatique, qui permet à un système d'apprendre à résoudre un problème par lui-même à partir d'exemples.

IA : Champ général visant à simuler l'intelligence humaine.

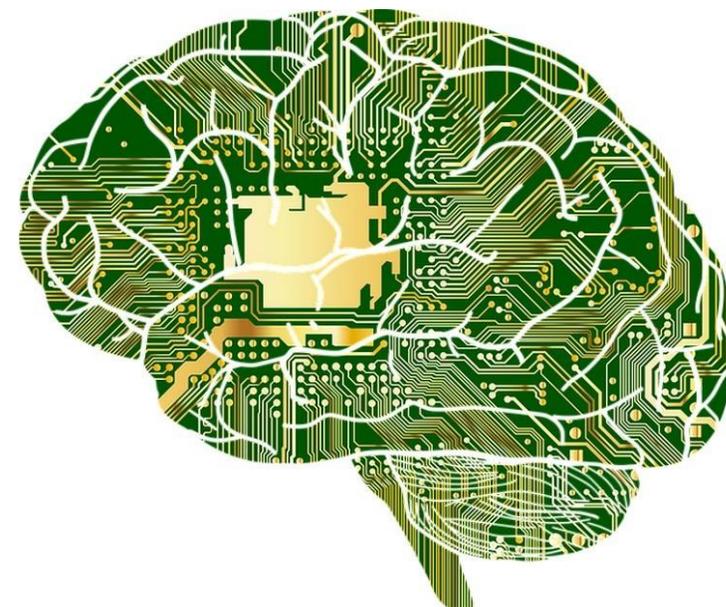
ML : Sous-domaine basé sur l'apprentissage à partir des données.

DL : Sous-domaine du ML utilisant des réseaux neuronaux profonds.



Caractéristiques du DL

- Apprentissage à partir de données brutes et complexes
- Modèles hiérarchiques (profondeur des réseaux)
- Extraction automatique des caractéristiques
- Utilisation intensive des données
- Capacité de généralisation
- Nécessite une puissance de calcul (GPU, TPU)
- Réutilisation de modèles (Apprentissage par transfert)
- Évolutivité...



Différence ML et DL

Caractéristique	ML	DL
Complexité des données	Requiert des données structurées.	Peut traiter des données brutes (images, sons, etc.).
Complexité des modèles	Régressions, arbres de décision, SVM.	CNN, RNN, transformeurs.
Requiert beaucoup de données ?	Oui, mais moins que le DL.	Oui, beaucoup de données et de puissance de calcul.
Exemple d'application	Classification des emails (spam vs non-spam).	Voitures autonomes, IA générative.
Apprentissage par transfert?	Non	Oui
Temps d'apprentissage	Généralement plus rapide à former que les modèles d'apprentissage profond.	Nécessite des temps de formation plus longs en raison d'architectures plus complexes.
Features extraction	Nécessite une intervention manuelle pour l'extraction et la sélection des caractéristiques.	Apprend les caractéristiques automatiquement, minimisant ainsi la nécessité d'une ingénierie manuelle des caractéristiques.

Pourquoi ce succès aujourd'hui ?

- Puissances de calcul insuffisantes dans les années 90
- Disponibilité de données labélisées (ImageNet)
- Capacités de stockage disponibles
- Disponibilité des puissances de calcul (GPU, TPU)
- Progression des algorithmes et des architectures

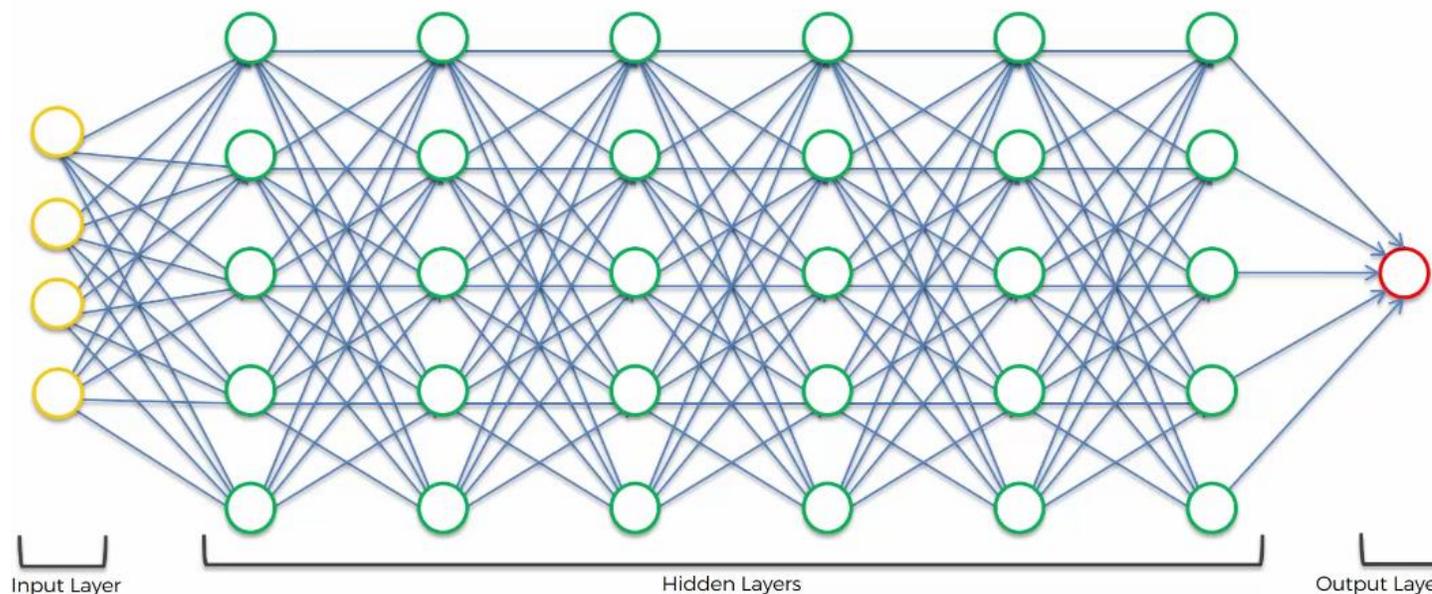


Plan de la présentation

- Généralités
- **Structures et Fonctionnement**
- Principales architectures
- Evaluation et amélioration des modèles
- Technologies utilisées

Réseau de Neurone Artificiel (RNA)

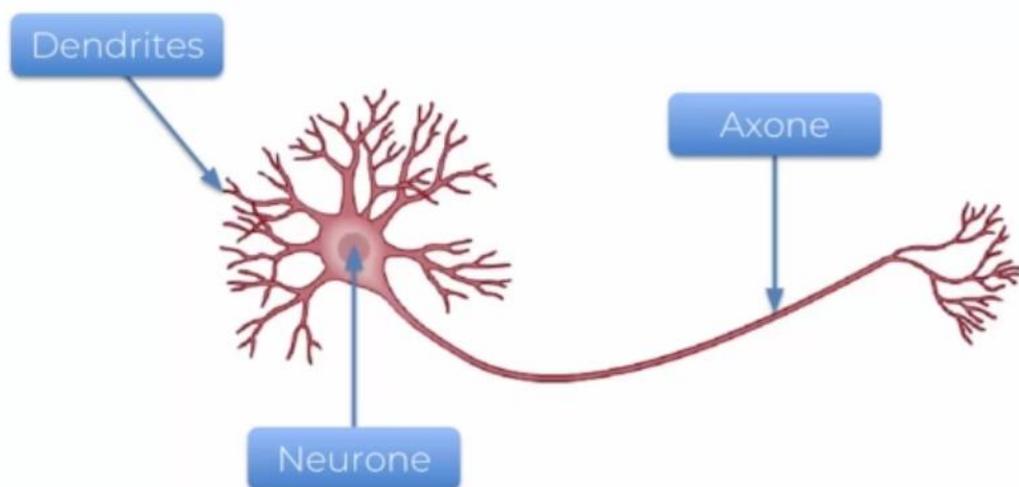
- Les architectures DL sont basées sur le concept des RNA.
- Les RNA désignent des structures composées de nœuds
- ML : Réseau avec une seule couche cachée
- DL : Plusieurs couches cachées et les connexions entre les neurones et les couches sont également plus complexes.



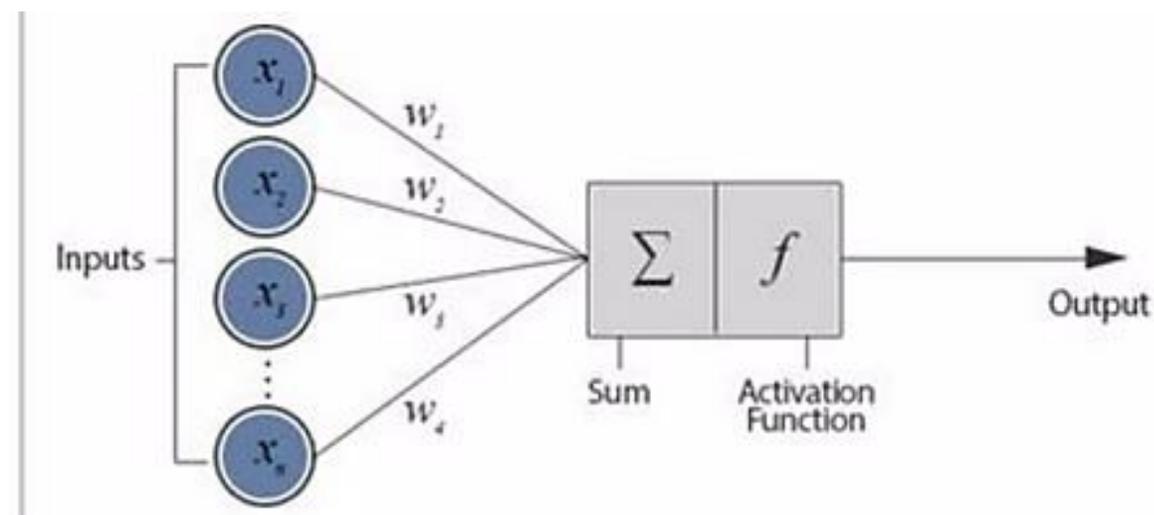
Le neurone

- Base fondamentale du réseau de neurone.
- Dendrite : partie qui reçoit les signaux.
- Axone : envoie les signaux
- La puissance se trouve dans le nombre de nombre de neurones (colonie de fourmie)

Neurone formel :



Modélisation mathématique :

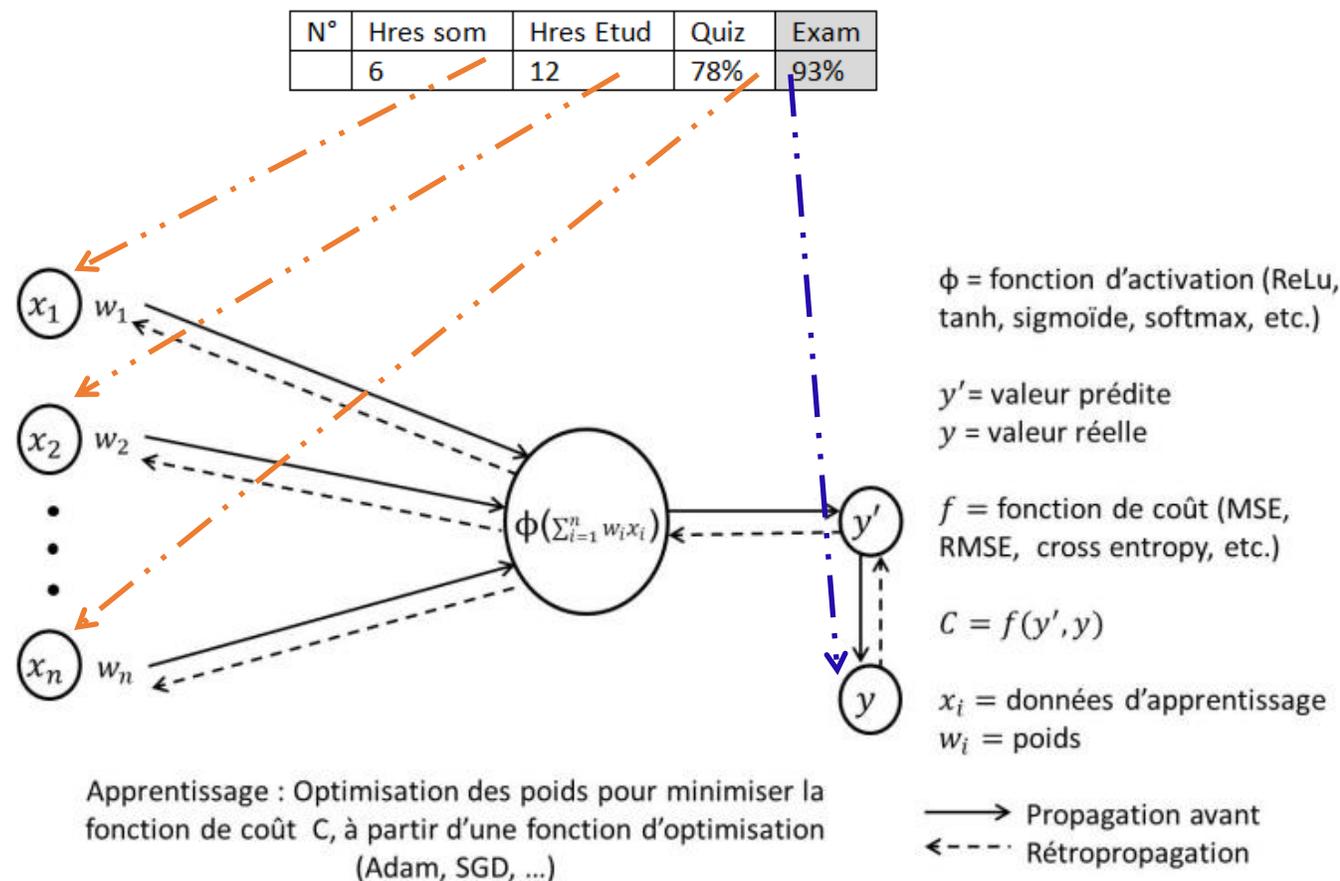


Apprentissage

Minimisation de la fonction de coût par la mise à jour des poids : rétro propagation

Les étapes :

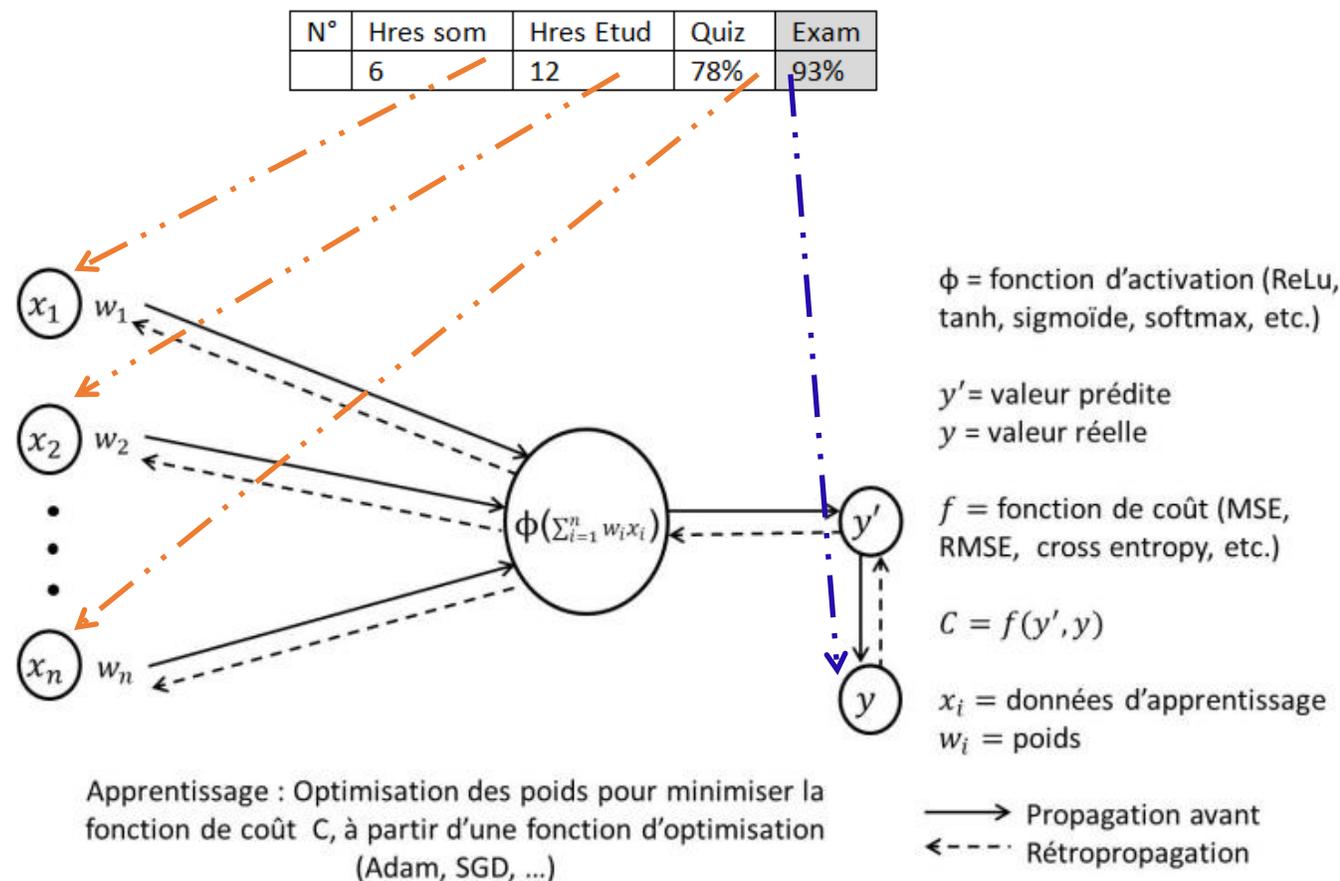
1. Poids initialisés avec des valeurs proches de 0.
2. Données envoyées à travers la couche d'entrée, avec une variable par neurone.
3. Propagation vers l'avant à travers les différentes couches jusqu'à la prédiction d'une valeur : les neurones sont activés en fonction des poids initiaux (application de la fonction d'activation).
4. Mesure de l'erreur avec la fonction de coût.



Apprentissage

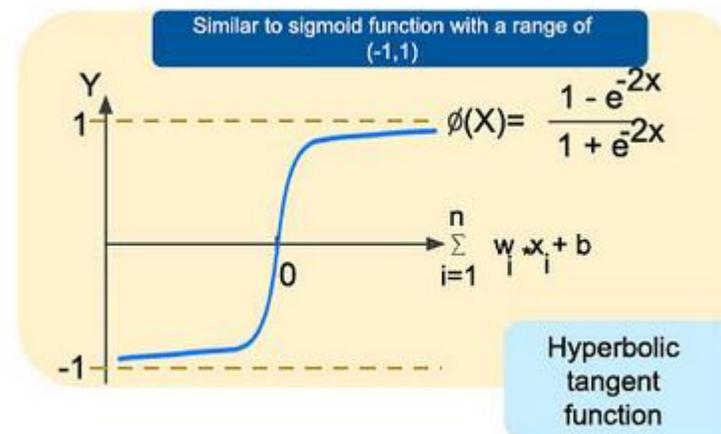
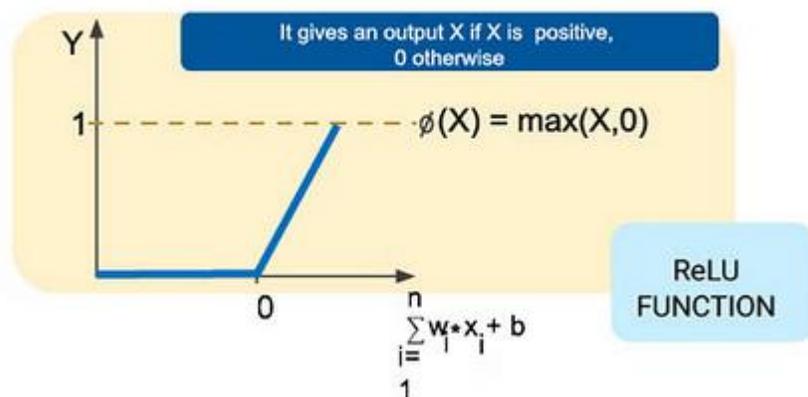
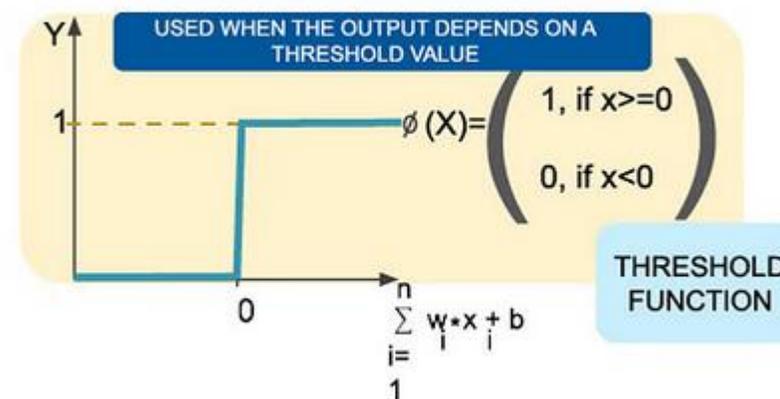
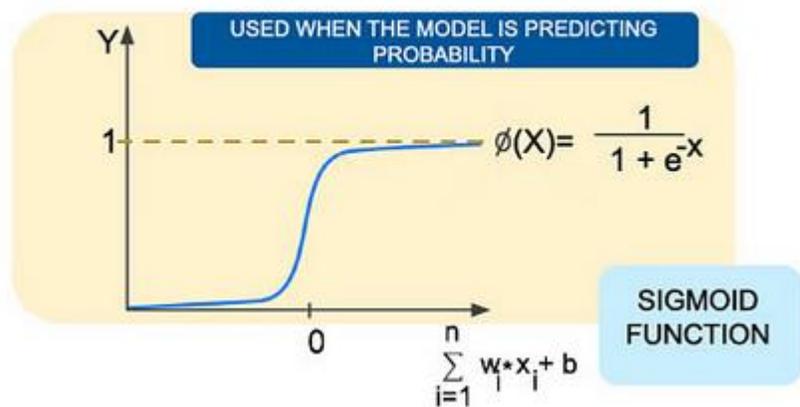
Minimisation de la fonction de coût par la mise à jour des poids : rétro propagation

- Rétropropagation de l'erreur à travers le réseau et mise à jour des poids grâce à une fonction d'optimisation.
- Répétition des étapes 1 à 5 après chaque observation (apprentissage renforcé) ou après un certain nombre d'observations (apprentissage par lots).
- Lorsque toutes les données sont passées dans le réseau, il s'agit d'une *epoch*. Plusieurs *epochs* sont exécutées autant que nécessaire.



Fonctions d'activation

Sigmoid, ReLu, tanh, Seuil



Plan de la présentation

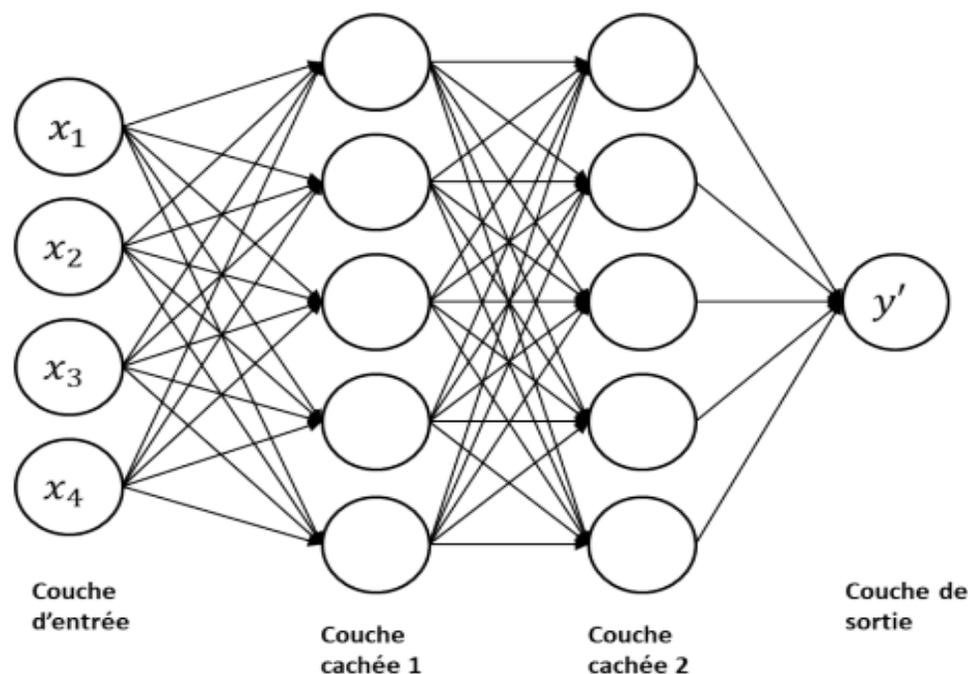
- Généralités
- Structures et Fonctionnement
- **Principales architectures**
- Evaluation et amélioration des modèles
- Technologies utilisées

Principales architectures

- 1. Multi-Layer Perceptron
- 2. Convolutional Neural Networks (CNNs)
- 3. Recurrent Neural Networks (RNNs)
- 4. Long Short-Term Memory Networks (LSTMs)
- 5. Generative Adversarial Networks (GANs)
- 6. Transformer Networks
- 7. Autoencoders
- 8. Graph Neural Networks (GNNs)
- ...

Multi-Layer Perceptron

Le *Multi-Layer Perceptron (MLP)* ou *feedforward network*, est considéré comme la première et la plus simple des architectures DL.

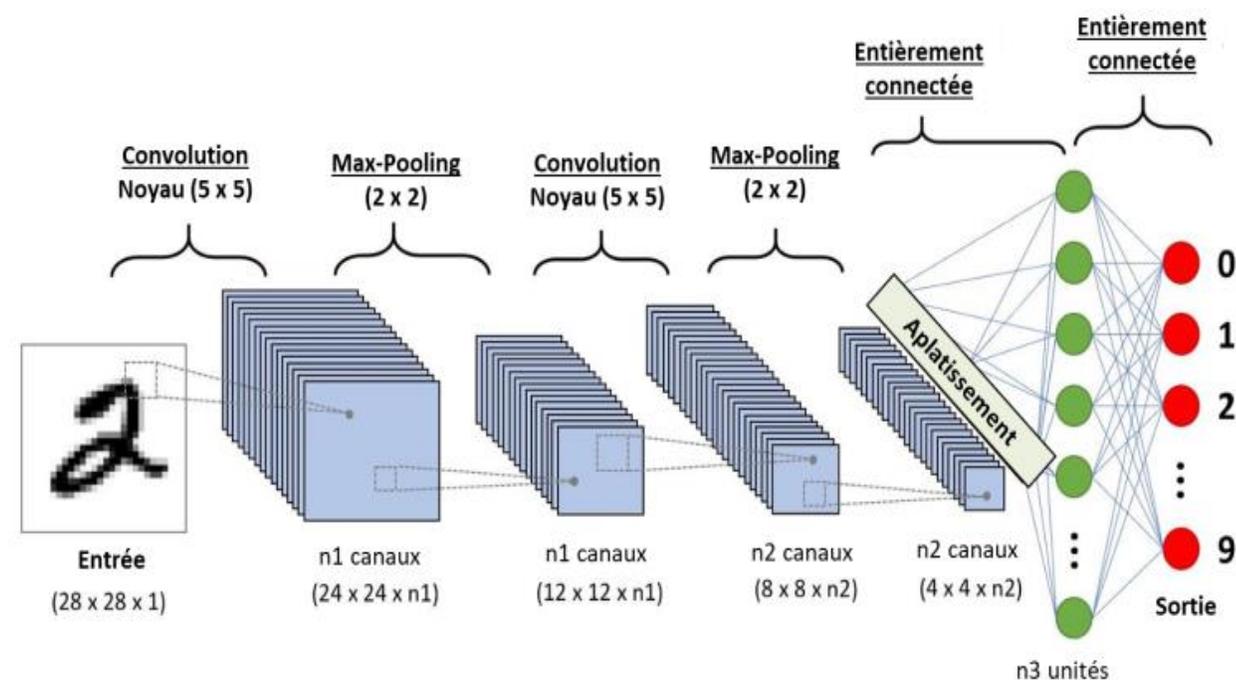


- Généralement utilisés pour les problèmes de régression et de classification
- Pas assez efficaces pour les tâches avancées de vision par ordinateur.
- Nombre total de paramètres peut devenir très élevé.
- Temps de traitement généralement inférieur au temps d'exécution pris par d'autres modèles DL.
- Les MLP nécessitent des données tabulaires ou de faible dimension en entrée

Convolutional Neural Network

Cette architecture est généralement utilisée dans le domaine de la vision par ordinateur et est appliquée dans plusieurs domaines tels que la détection d'objets, la reconnaissance de visages sur les médias sociaux, la détection d'anomalies, etc.

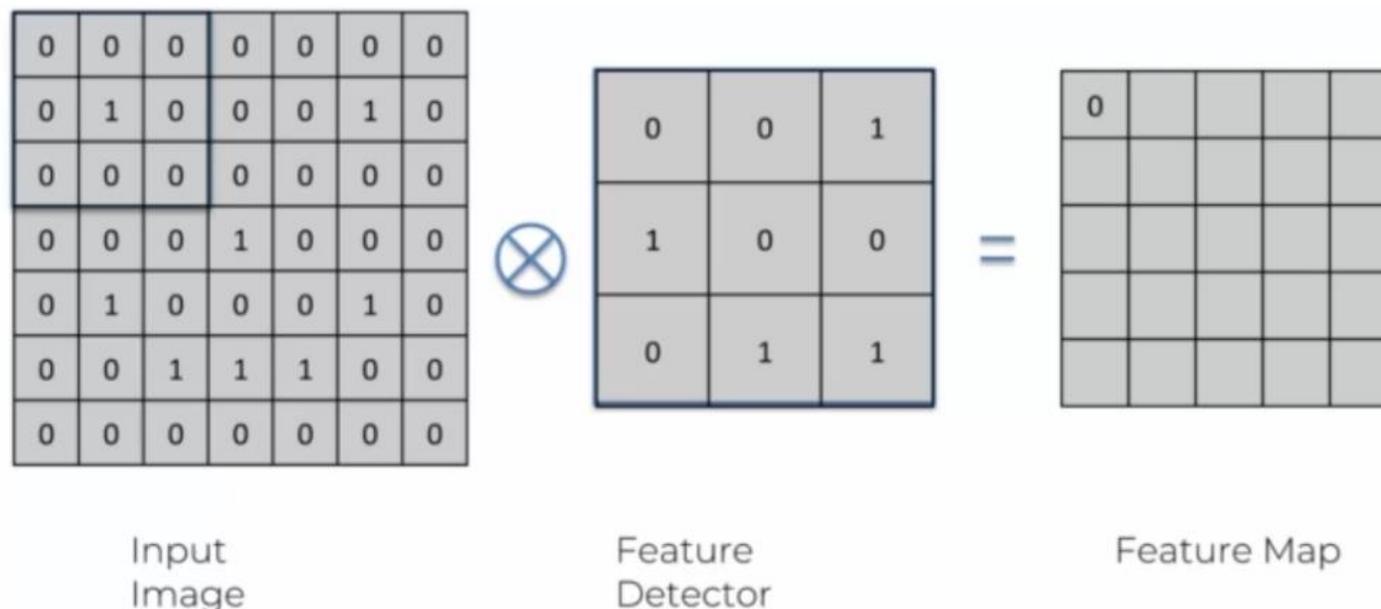
- Application de filtres aux données d'entrée
- Réduction de la taille des données
- Résultat disposé dans un vecteur soumis à un MLP
- Les couches non entièrement connectées
- Composé généralement de 4 couches de base.



Couche de convolution

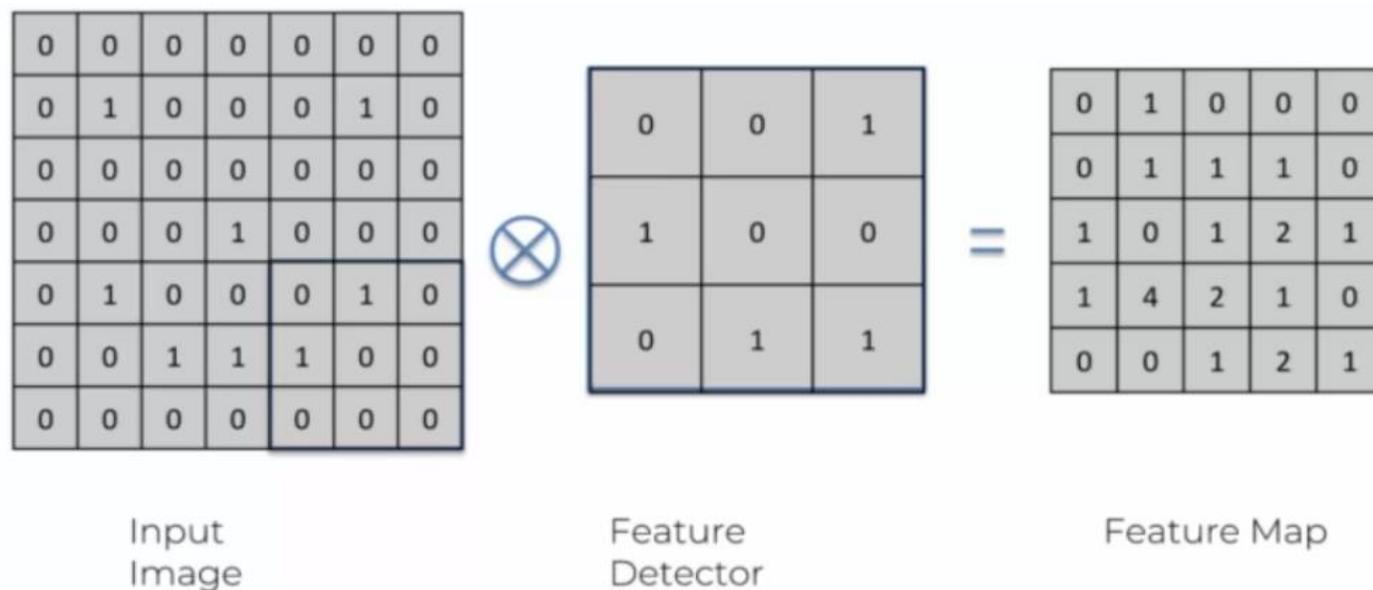
Les étapes de convolution permettent d'extraire les caractéristiques représentatives de l'image par des opérations mathématiques de convolution ou filtrage. Les images obtenues après les opérations de convolutions sont appelées *features map*.

On sélectionne un *feature* detector(encore appelé **Kernel** ou **filtre**) de dimension généralement 3X3, 5X5, 7X7 (3X3 le standard).



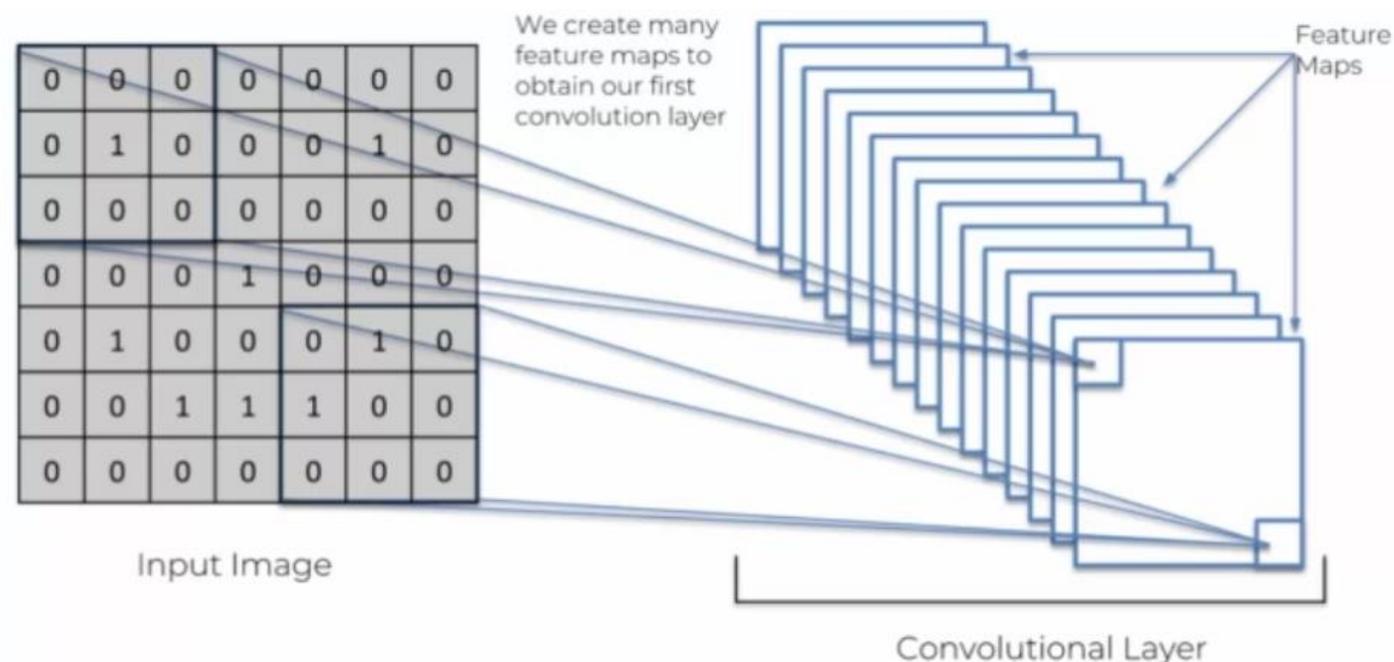
Couche de convolution

- On superpose le filtre puis on fait multiplication terme à terme et on additionne le tout pour avoir la nouvelle valeur. Ensuite on fait un pas vers la droite (stride de pas = 1)
- On continue jusqu'à la fin puis on descend d'un niveau parcourir tous les pixels.
- L'opération permet de détecter les *features* importants (valeurs de pixels élevés).



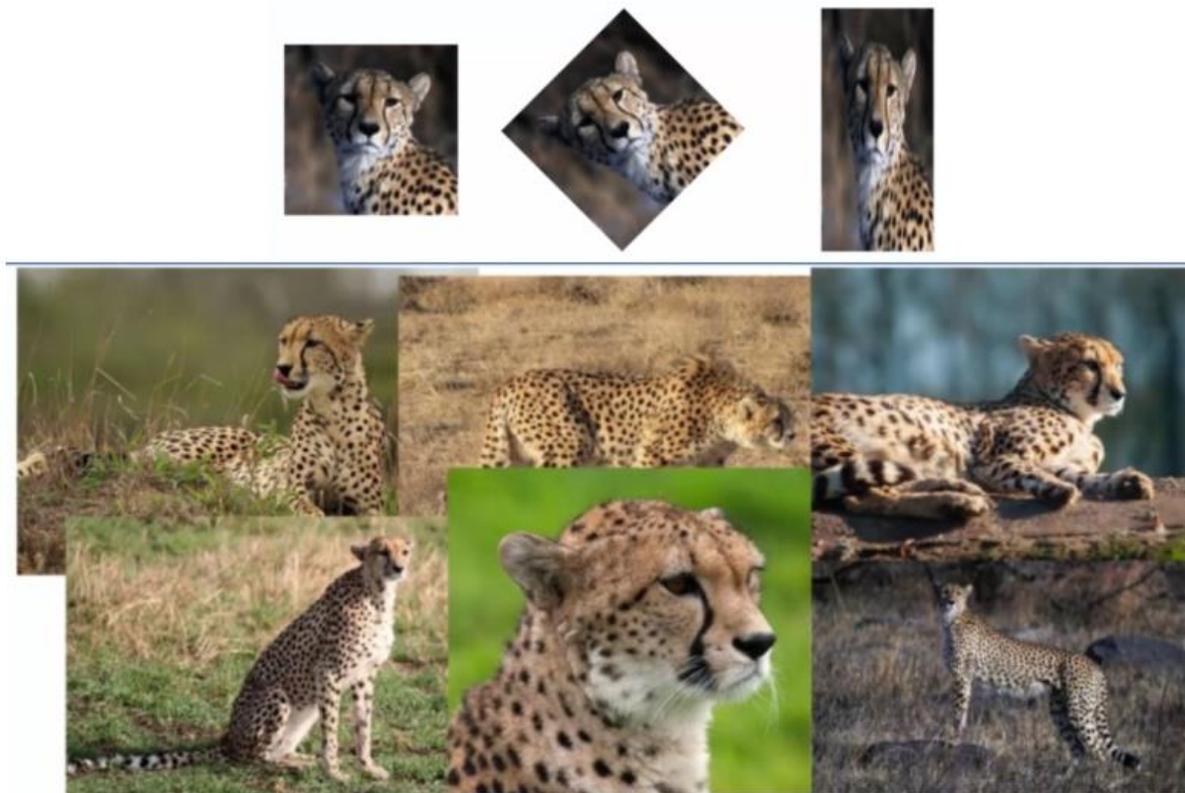
Couche de convolution

Après plusieurs itérations, le réseau crée plusieurs *features map* correspondants à une *feature* importante. Le but de la convolution est détecter les *features* dans une image avec un filtre et de les conserver dans une *features map* pour mémoriser les données importantes.



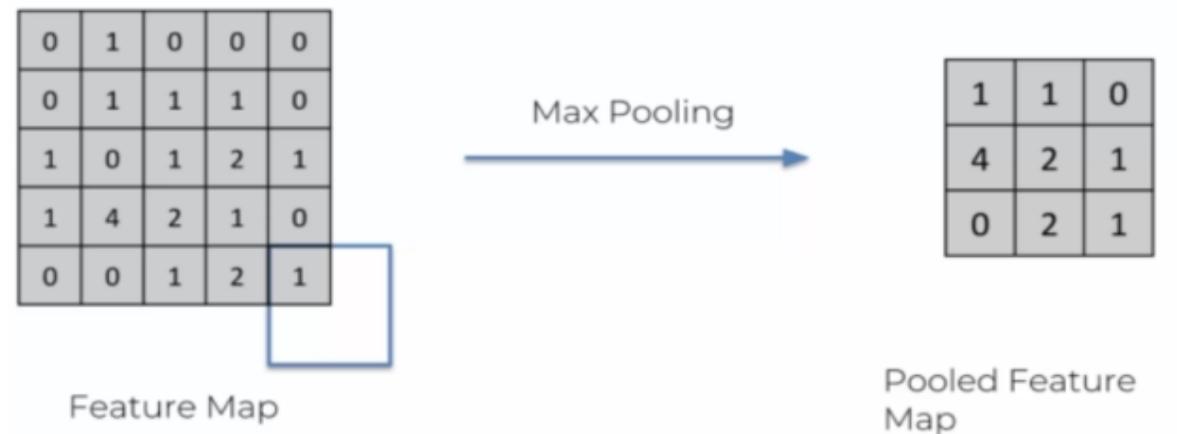
Couche de Pooling

- Parfois appelées *subsampling* ou *downsampling*, les étapes de *pooling* réduisent la taille de chaque donnée mais conservent les informations importantes.
- Elles confèrent au réseau la propriété d'invariance spatiale.



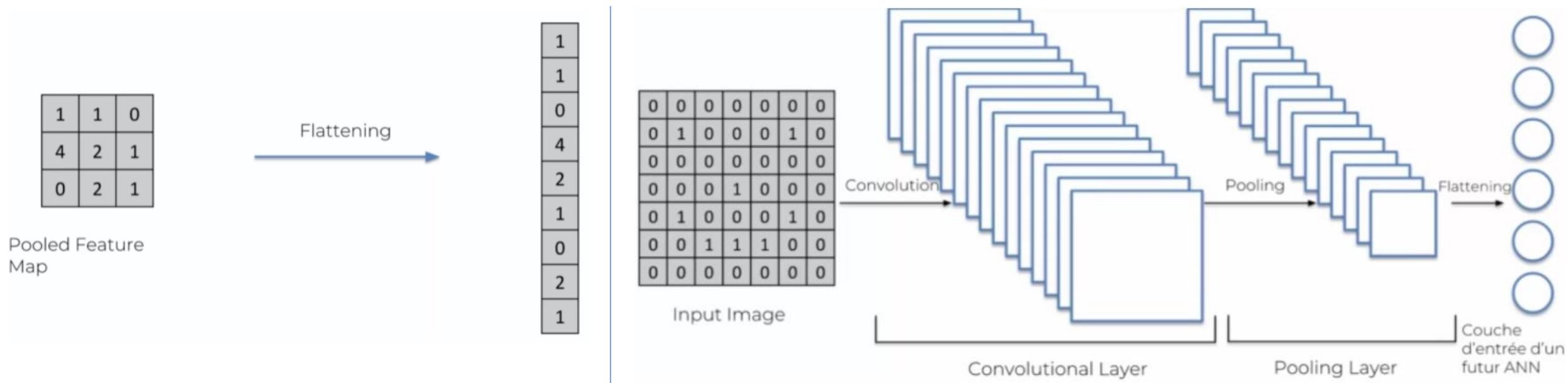
Max Pooling

- Le Max pooling, assez similaire à la convolution, mais on prend le maximum.
- Stride de 2 avec un carré de 2X2.
- On parcourt toute l'image jusqu'à la fin.
- Min Pooling, Average Pooling



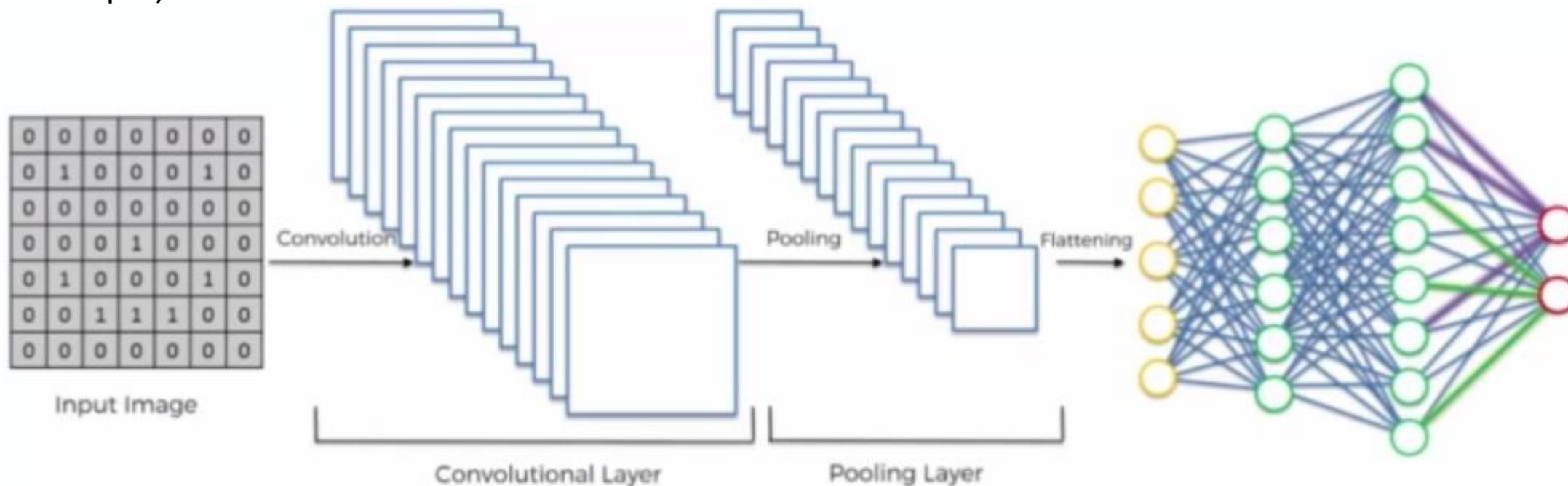
Couche de Flattening

Les valeurs des matrices sont disposées sous forme de colonne dans un vecteur. Cette disposition des valeurs est nécessaire pour faire passer chacune d'elles dans des neurones d'un RNA. Pour faire passer chacune des valeurs dans des neurones.



Couche entièrement connectée

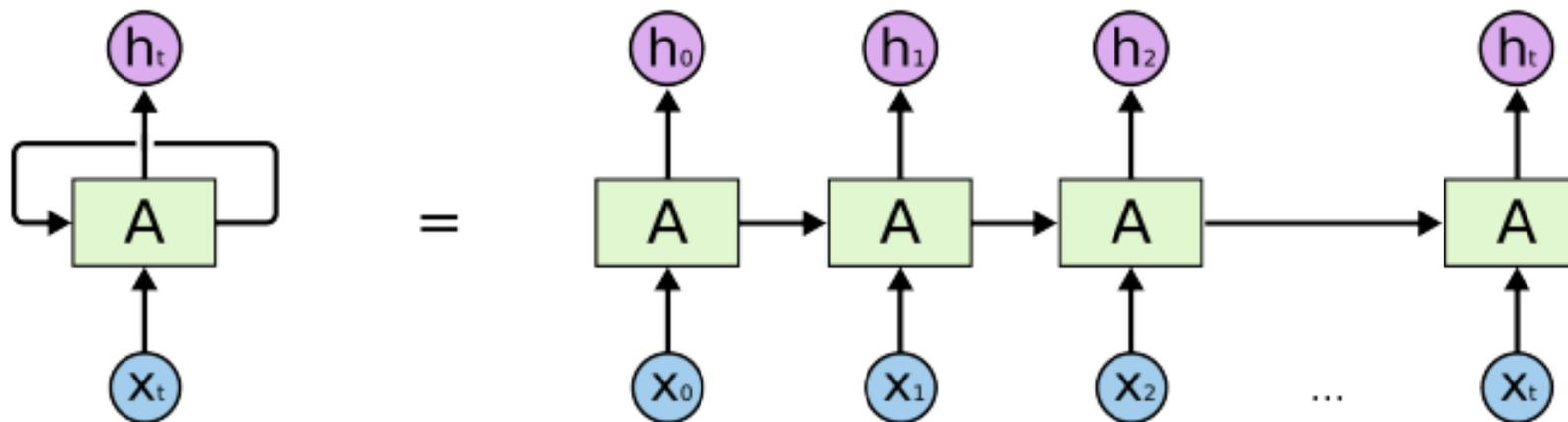
- Rajout d'un nouveau MLP qui va être complètement connecté aux données issues de la couche précédente.
- Fonction *softmax* ajoutée à la fin (somme de probabilités à 1 pour un problème de classification par exemple)



Réseaux de neurones récurrents

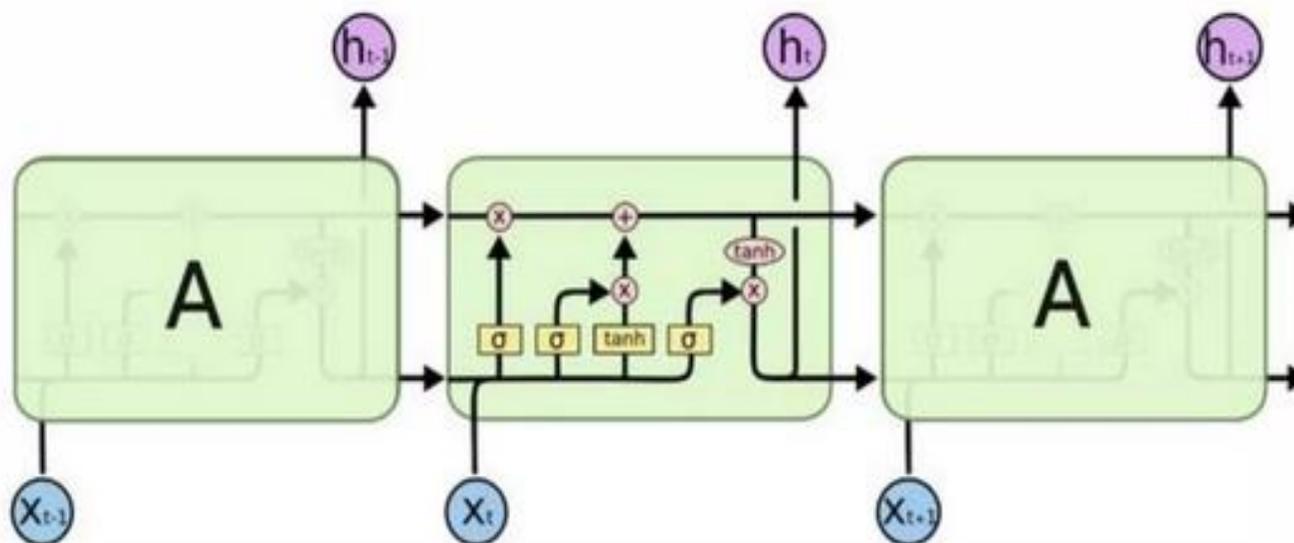
Les réseaux de neurones récurrents ou *Recurrent Neural Network* (RNN)

- Utilisés lorsque les données d'entrée sont séquentielles (séries chronologiques, le son, ou NLP).
- Disposent de capacités d'apprentissage exceptionnelles et adaptés à la conception de modèles prédictifs.



Long-Short Term Memory (LSTM)

- Les LSTM sont un type particulier de RNN avec des unités de mémoire appelées « cellules », capable d'apprendre les dépendances à long terme.
- Plus efficaces pour des tâches telles que la reconnaissance vocale et la prédiction de séries temporelles.
- Trois mécanismes qui contrôlent le flux d'informations dans le réseau : la porte d'entrée, la porte d'oubli et la porte de sortie.



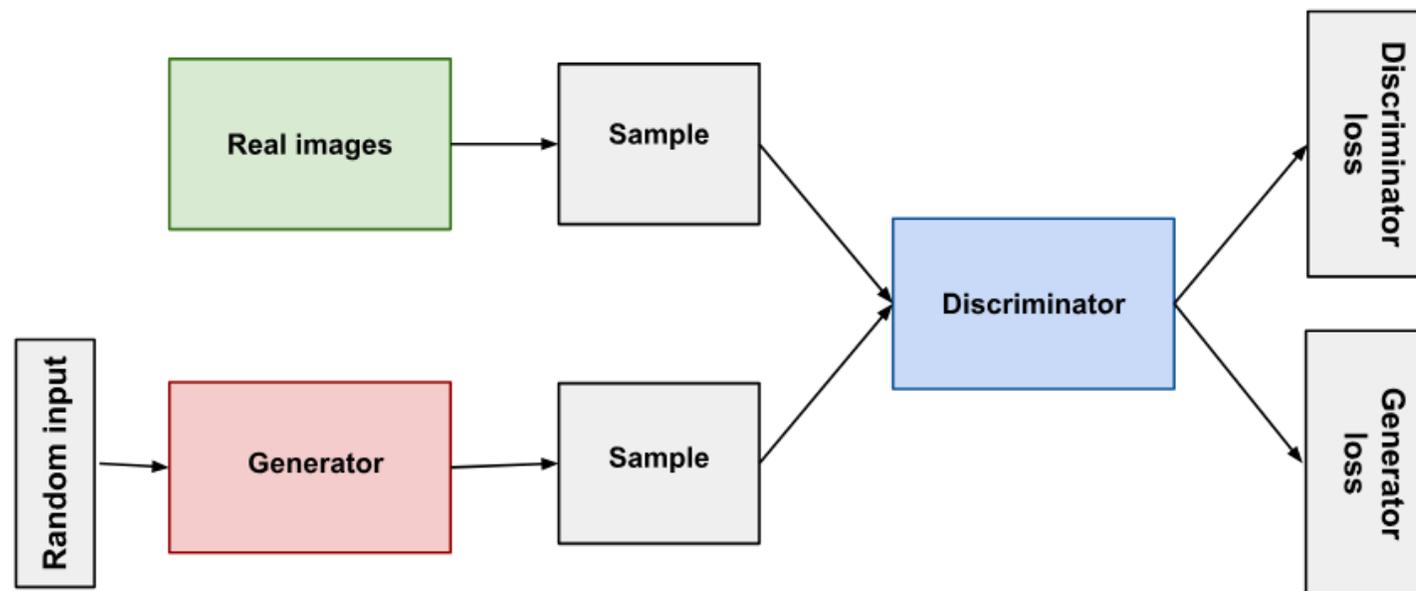
Gated Recurrent Units (GRU)

- Les GRU présentent des similitudes avec les réseaux à mémoire à long terme (LSTM), mais se caractérisent par une structure plus simple pour le processus d'apprentissage.
- Les GRU ont une architecture plus simple que les LSTM, avec moins de paramètres et de calculs. Cette simplicité augmente l'efficacité en termes de temps de formation et de ressources informatiques, ce qui rend les GRU avantageuses pour diverses applications.
- Il comporte deux portes : la porte de réinitialisation et la porte de mise à jour.

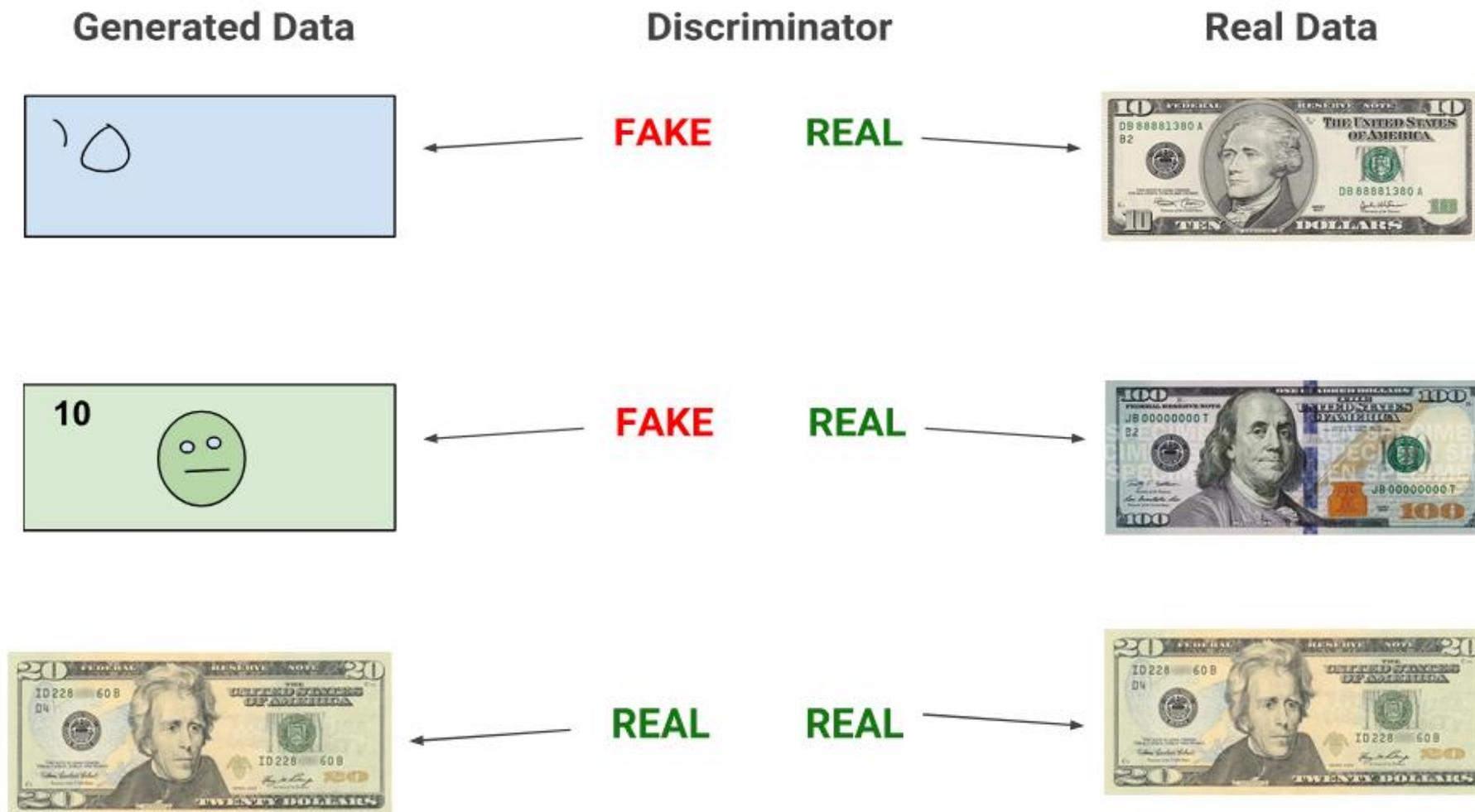
Generative Adversarial Networks

L'objectif principal des GANs est de générer des données qui sont indiscernables des vrais exemples. Ils ont été utilisés pour créer des images, des vidéos et des sons réalistes.

- **Générateur** : Produit des données synthétiques.
- **Discriminateur** : Tente de distinguer les données synthétiques des données réelles.



Generative Adversarial Networks



Applications des GAN

Exemples de GAN utilisés pour générer de nouveaux exemples plausibles pour des ensembles d'images.



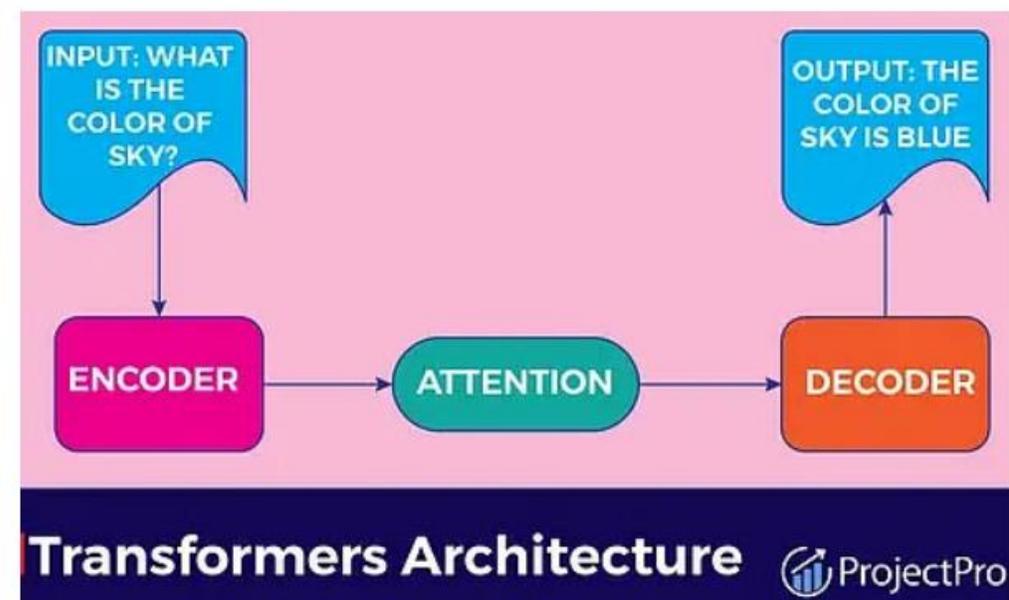
Applications des GAN

Exemples de visages photoréalistes générés par des GAN, tirés de Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017.

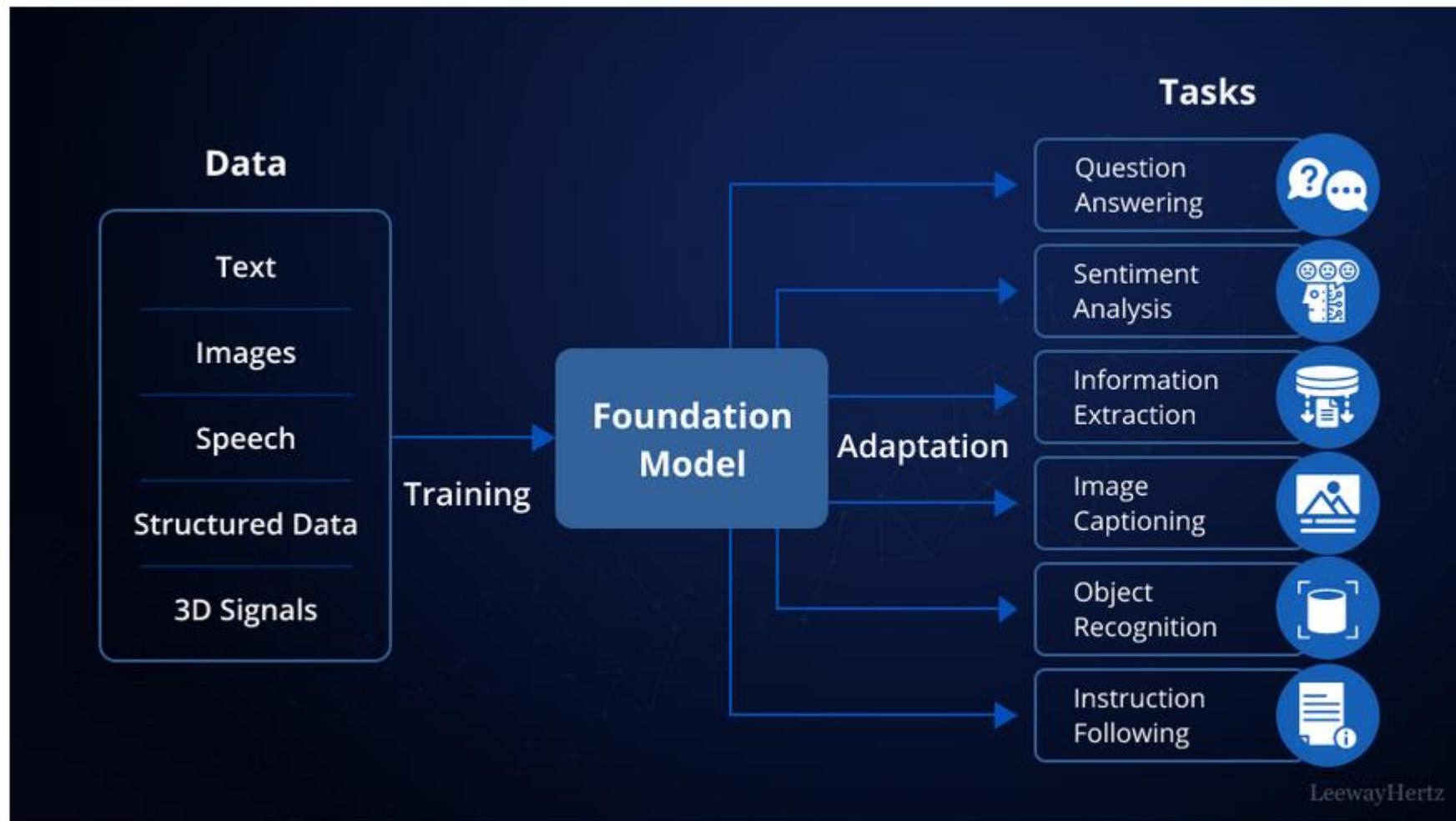


Transformers

- Architecture révolutionnaire en DL, qui ont transformé des domaines comme le **NLP**, la **CV**, et même les **ST**.
- Reposent principalement sur un mécanisme d'**attention** permettant de se concentrer sur les parties importantes des données d'entrée.
- Créés pour surmonter les limitations des **RNN (Réseaux récurrents)**
- Permettent le traitement parallèle des données, accélérant les calculs.
- Gèrent les dépendances à long terme dans les séquences.



Transformers- LLM



Applications des Transformers

■ Traitement du langage naturel (NLP)

Traduction automatique (Google Translate).

Résumé automatique de texte.

Génération de texte (ChatGPT, Bard).

Analyse des sentiments.

■ Vision par ordinateur

Classification et segmentation d'images.

Génération d'images réalistes.

■ Audio et vidéo

Reconnaissance vocale (speech-to-text).

Synthèse vocale (text-to-speech).

Analyse et génération de vidéo.

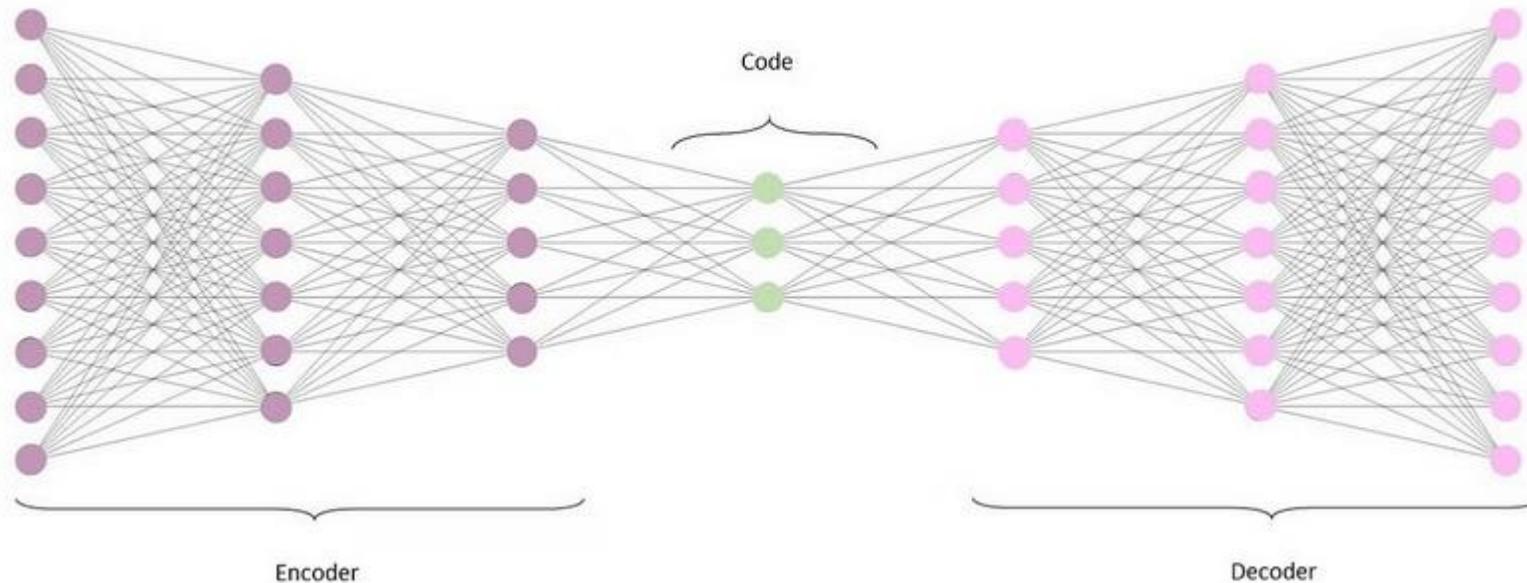
■ Sciences et recherche

Prédiction des structures protéiques.

Analyse de données médicales.

Autoencoder

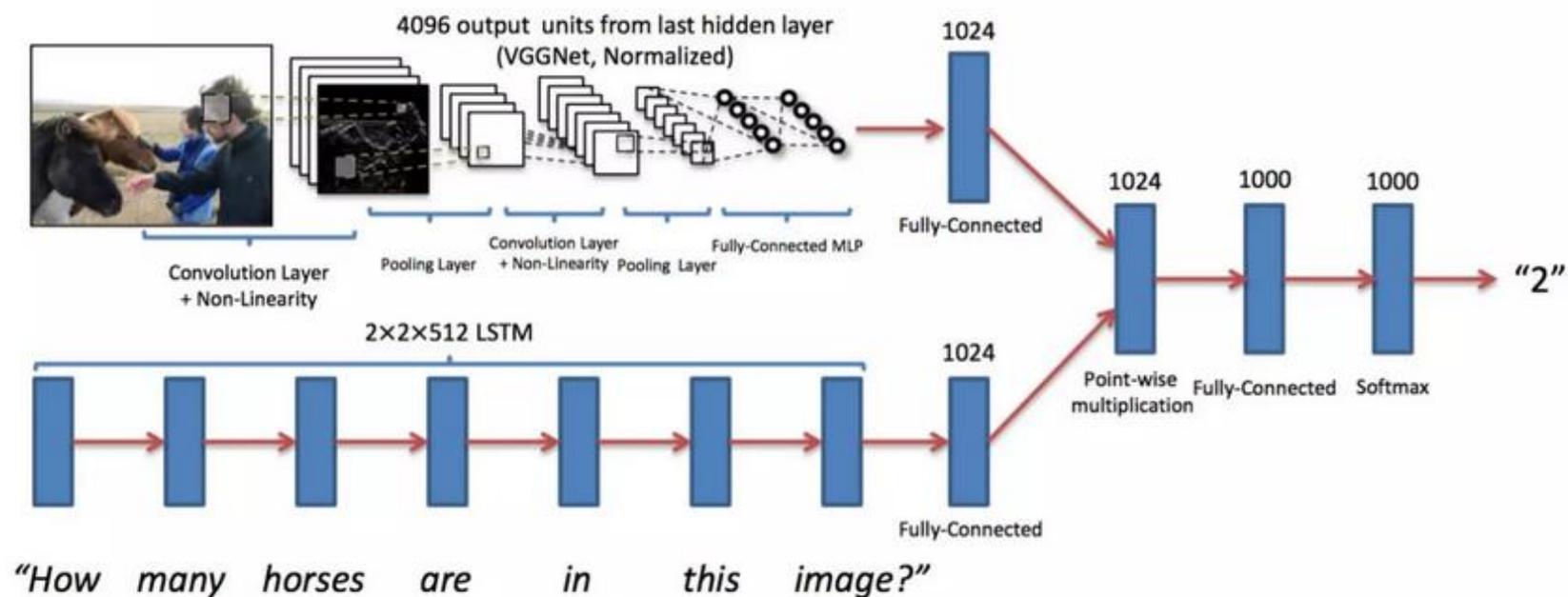
- Les autoencodeurs sont des modèles d'apprentissage non supervisés pour des tâches telles que la compression de données, le débruitage et l'apprentissage de caractéristiques. Ils apprennent à coder les données dans une représentation de dimension inférieure, puis à les décoder pour revenir aux données d'origine.



Réseaux hybrides

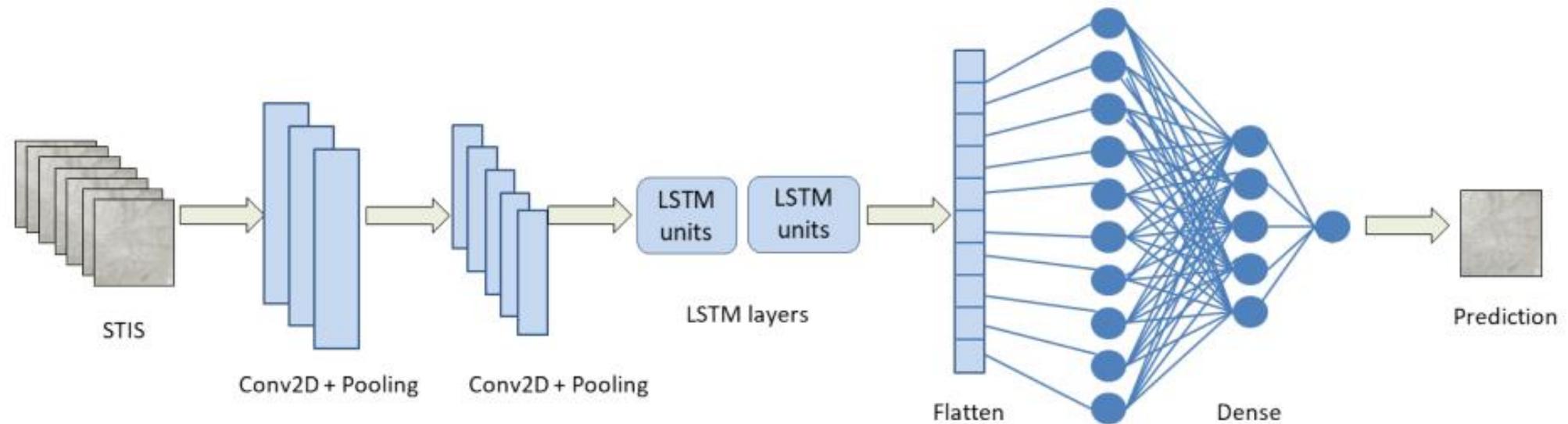
Un réseau de neurones hybride (HNN) est un modèle informatique avancé qui intègre plusieurs architectures de réseaux neuronaux pour exploiter leurs points forts tout en atténuant leurs faiblesses.

- CNN-RNN
- CNN-Transformer
- Autoencoder-GAN
- CNN-LSTM
- ConvLSTM
- Hybridation avec ML classique



CNN-LSTM

Un réseau de neurones CNN-LSTM utilisé sur une série temporelle d'images satellitaires.



Plan de la présentation

- Généralités
- Structures et Fonctionnement
- Principales architectures
- **Evaluation et amélioration des modèles**
- Technologies utilisées

Evaluation du modèle

Mesurer la qualité de l'apprentissage en DL est essentiel pour évaluer la performance d'un modèle, diagnostiquer ses problèmes, et s'assurer qu'il peut bien généraliser à de nouvelles données.

- Utiliser des métriques d'évaluation (varient selon le type de tâche).
- Évaluer la généralisation (à partir de nouvelles données)
- Diagnostic des erreurs
- Surveiller le sur-apprentissage (examiner les courbes de perte)
- Benchmarks et comparaisons
- Visualisation des résultats

Evaluation du modèle

Mesurer la qualité de l'apprentissage en DL est essentiel pour évaluer la performance d'un modèle, diagnostiquer ses problèmes, et s'assurer qu'il peut bien généraliser à de nouvelles données.

- Utiliser des métriques d'évaluation (varient selon le type de tâche).
- Évaluer la généralisation (à partir de nouvelles données)
- Diagnostic des erreurs
- Surveiller le sur-apprentissage (examiner les courbes de perte)
- Benchmarks et comparaisons
- Visualisation des résultats

Métriques d'évaluation

- **Mean Squared Error (MSE)** : mesure la différence quadratique moyenne entre les données prédites et les données réelles.
- **Peak signal to noise (PSNR)** : calcule le rapport signal/bruit maximal, en décibels.
- **Structural Similarity (SSIM)** : mesure la similarité structurelle entre une image prédite et l'image réelle.

$$MSE = \frac{1}{r \times c} \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} (x(i, j) - y(i, j))^2$$

$$PSNR = 10 \times \log_{10} \left(\frac{d^2}{MSE} \right)$$

$$SSIM(x, y) = \frac{(2\bar{x}\bar{y} + c_1)(2\sigma_{xy} + c_2)}{(\bar{x}^2 + \bar{y}^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Métriques d'évaluation

- **Pearson correlation coefficient (r)** : mesure la corrélation linéaire entre les données.
- **Prediction accuracy (PA)** : représente la précision de la prédiction pour chaque classe de couverture terrestre.
- **Number of pixels close to zero (NPCZ)** : représente le pourcentage de pixels ayant des valeurs proches de zéro dans l'image représentant la différence entre l'image prédite et l'image réelle.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

$$PA = \frac{1}{n} \sum_{i=1}^n (100 - |\frac{y_i - x_i}{x_i} \times 100|)$$

$$NPCZ = \frac{1}{r \times c} \sum_{i=1}^r \sum_{j=1}^c f(i, j)$$

$$f(i, j) = \begin{cases} 1, & \text{si } -\epsilon < d_{i,j} < \epsilon \\ 0, & \text{sinon.} \end{cases}$$

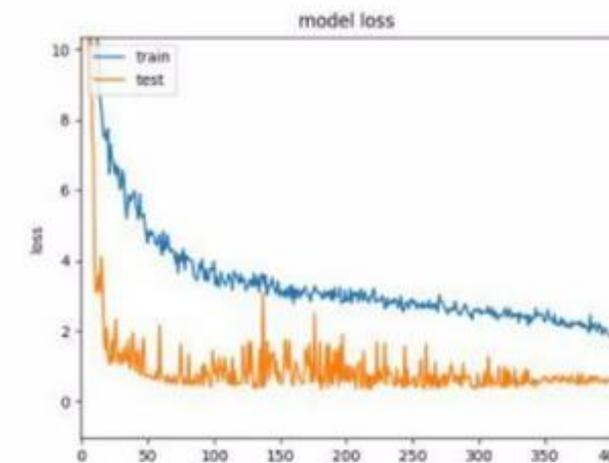
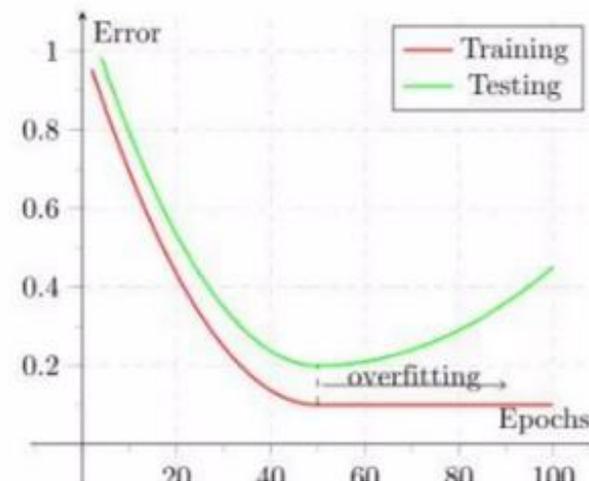
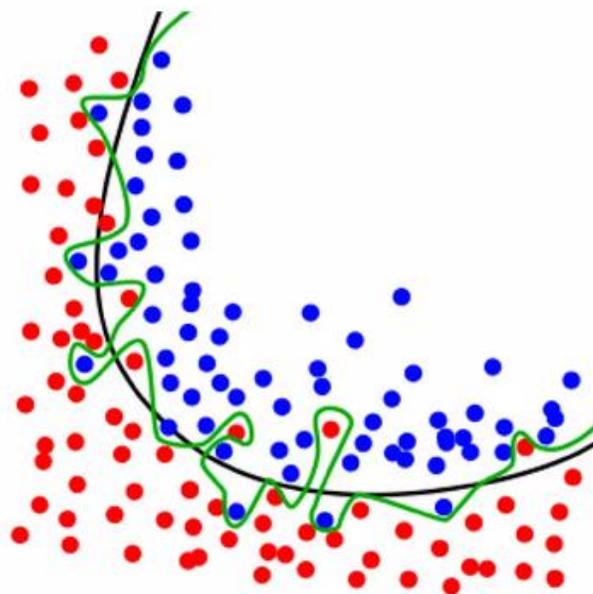
Le Sur-apprentissage

Le réseau n'est pas capable de généraliser

Architecture inadaptée

Pas assez de données

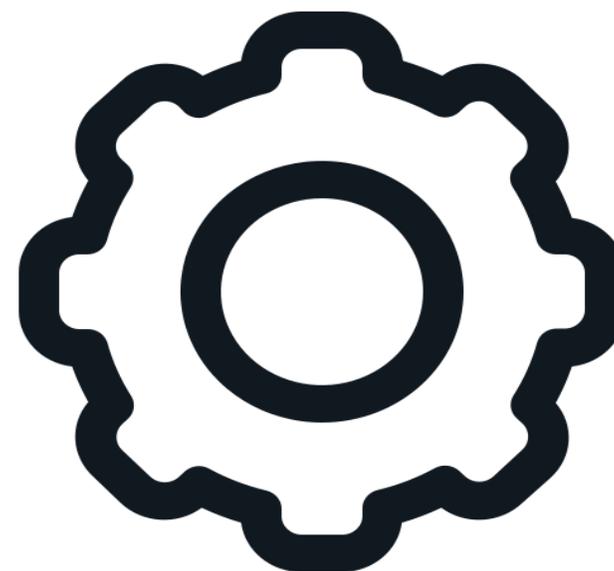
Trop d'apprentissage



Régularisation

La **régularisation** en DL est une série de techniques utilisées pour réduire le sur-apprentissage (**overfitting**) dans un modèle.

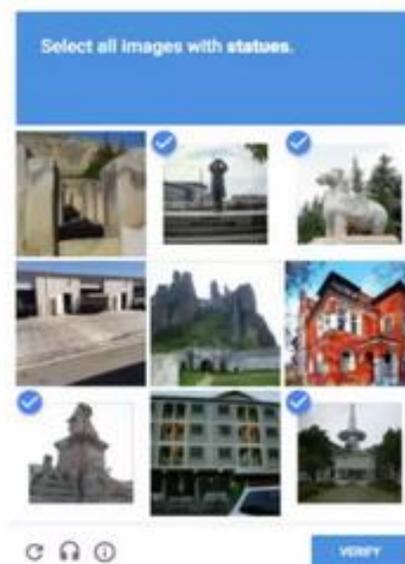
- Plus de données d'entraînement
- Augmentation de données
- Dropout
- Régularisation L1 & L2
- Early stopping
- Normalisation des données
- Batch normalization
- Réduction de la profondeur du réseau



Plus de données



- Datasets existants
- User Generated Content
(attention à assainir les data)
- Crowd sourcing



Augmentation de données



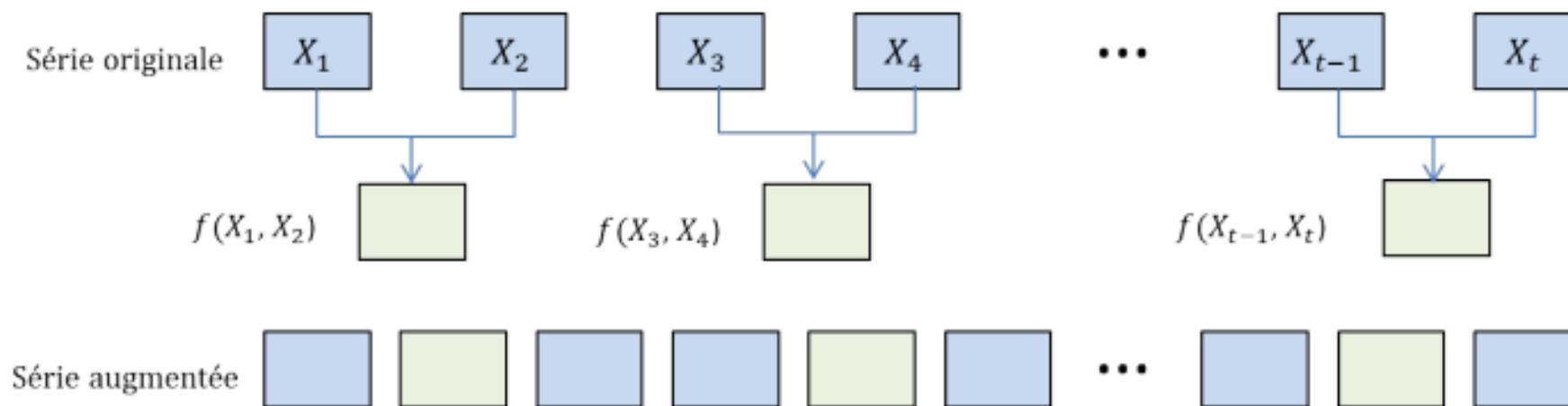
Enlarge your Dataset

- appliquer des transformations : rotation, déformation, homothétie, symétrie
- changer la luminosité, la colorimétrie, le focus, ...
- autres : ajouter du bruit, ajouter des éléments en foreground, découper, ...



Augmentation des ST

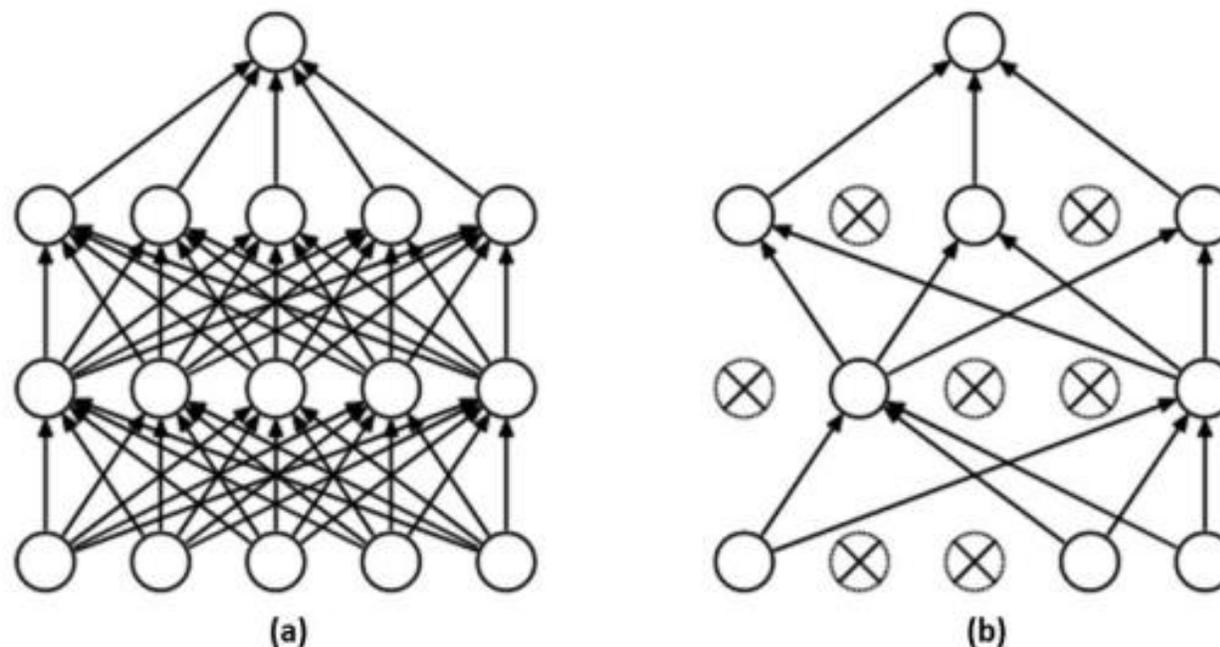
- Méthode d'augmentation qui considère chaque paire d'images de la série et à appliquer une opération mathématique sur les paires d'images.
- La taille de la série passe alors de N à $N + N/2$ éléments.



Dropout, Early stopping

Dropout

Technique d'ensemble implicite où des neurones (et leurs connexions) sont aléatoirement désactivés pendant l'entraînement. Il réduit la dépendance entre les neurones en forçant le réseau à apprendre des représentations plus robustes.

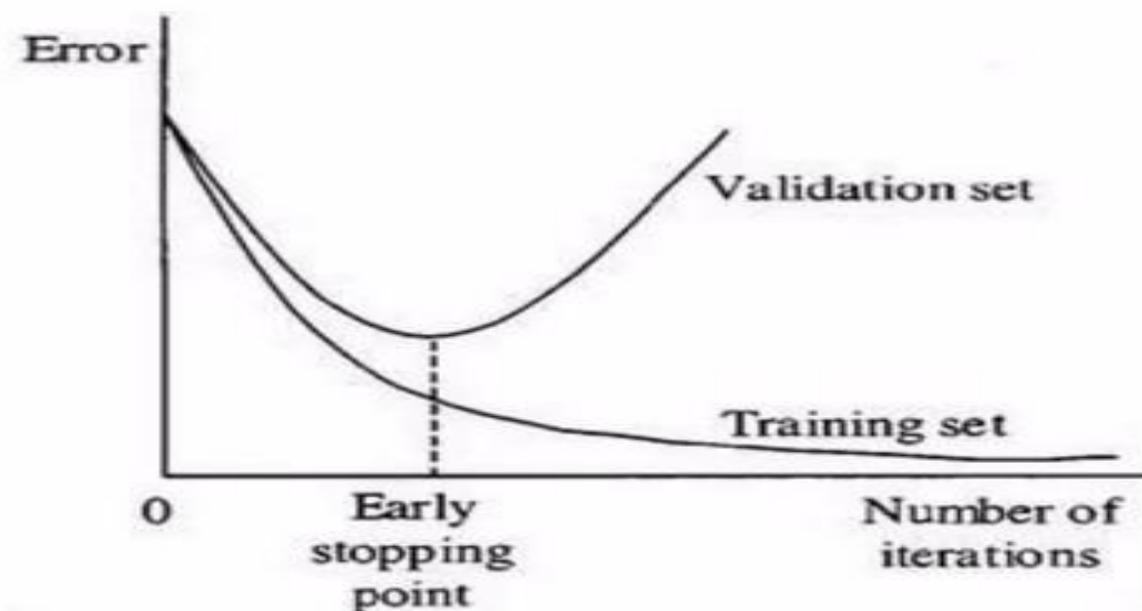


Dropout, Early stopping

Early Stopping

Permet d'arrêter l'entraînement lorsque la performance sur un ensemble de validation commence à se détériorer.

Nécessite de suivre la perte ou une métrique sur les données de validation.



Early Stopping

Choix des hyperparamètres

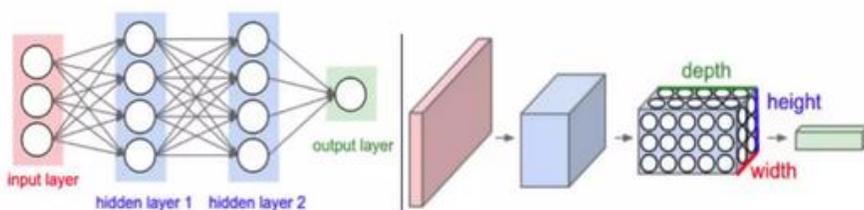
Hyperparamètres : valeurs définies par l'utilisateur avant le début du processus d'apprentissage.

Trois grands types :

- Hyperparamètres de couches (taille et nombre des noyaux, nombre de couches, méthodes d'activation des couches, etc.)
- Hyperparamètres de compilation du modèle (fonction d'optimisation, fonction de coût, taux d'apprentissage, etc.)
- Hyperparamètres d'exécution du modèle (taille du lot, nombre d'epochs, nombre d'étapes par epoch, etc.).

Choix des hyperparamètres

Recherche & Développement



Recherche et Développement :

- Tester plusieurs combinaisons
- Comparer les résultats

Automatisé (Network Architecture Search)

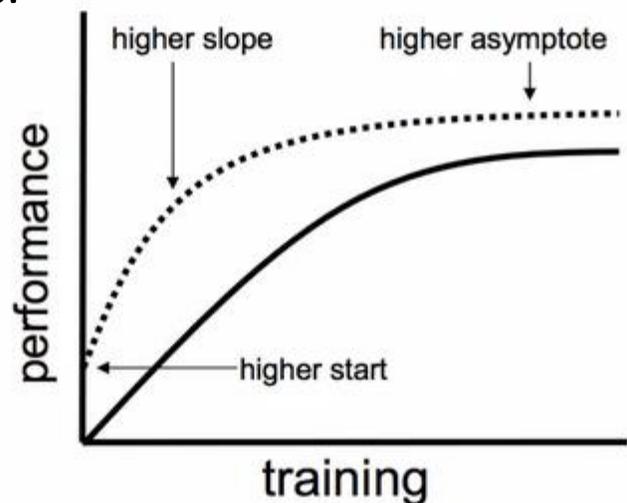
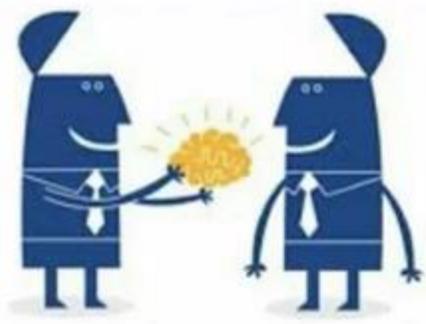


- GridSearch
- Random Search
- Optimisation hyperband
- AutoML (Google AutoML, AutoKERAS...)

Accélérer l'apprentissage

Le Transfer Learning ou apprentissage par transfert

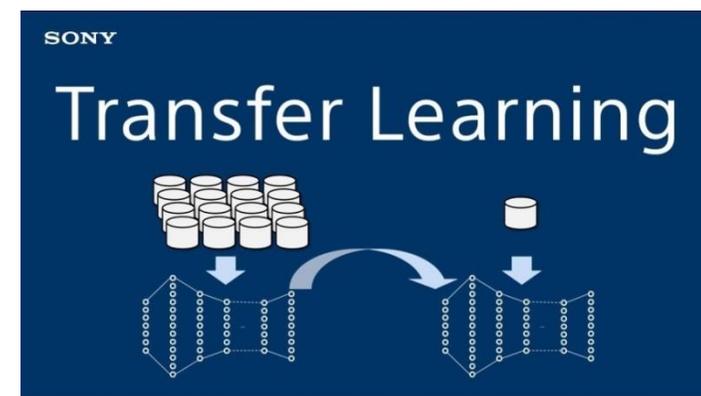
- Complète l'apprentissage d'un modèle préalablement entraîné à résoudre une tâche donnée
- Basé sur le fonctionnement du cerveau humain
- Réutilisation des neurones déjà entraînés pour des tâches similaires plus précises.



..... with transfer
 — without transfer

Transfer learning

Similitudes



Plan de la présentation

- Généralités
- Structures et Fonctionnement
- Principales architectures
- Evaluation et amélioration des modèles
- **Technologies utilisées**

Frameworks

TensorFlow est une bibliothèque d'apprentissage profond open-sourcedéveloppée par Google. Elle est développée en C++ et son implémentation se fait en Python.

PyTorch, qui est principalement développé par le laboratoire de recherche en IA de Facebook (FAIR).

Keras est un framework Python développé à l'origine par Francois Chollet . Son atout majeur est la réutilisation du code pour le CPU et le GPU.

Rasterio, GDAL/OGR, GeoPandas, ...



Notebooks

Bienvenue dans Colaboratory

Fichier Modifier Affichage Insérer Exécution Outils Aide

Sommaire

Premiers pas

Science des données

Machine learning

Autres ressources

Exemples

+ Section

+ Code + Texte Copier sur Drive

Connecter

Bienvenue dans Colab !

(Nouveau) Essayez l'API Gemini

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

Si vous connaissez déjà Colab, regardez cette vidéo pour découvrir les tables interactives, l'affichage de l'historique du code exécuté et la palette de commandes.

3 Cool Google Colab Features

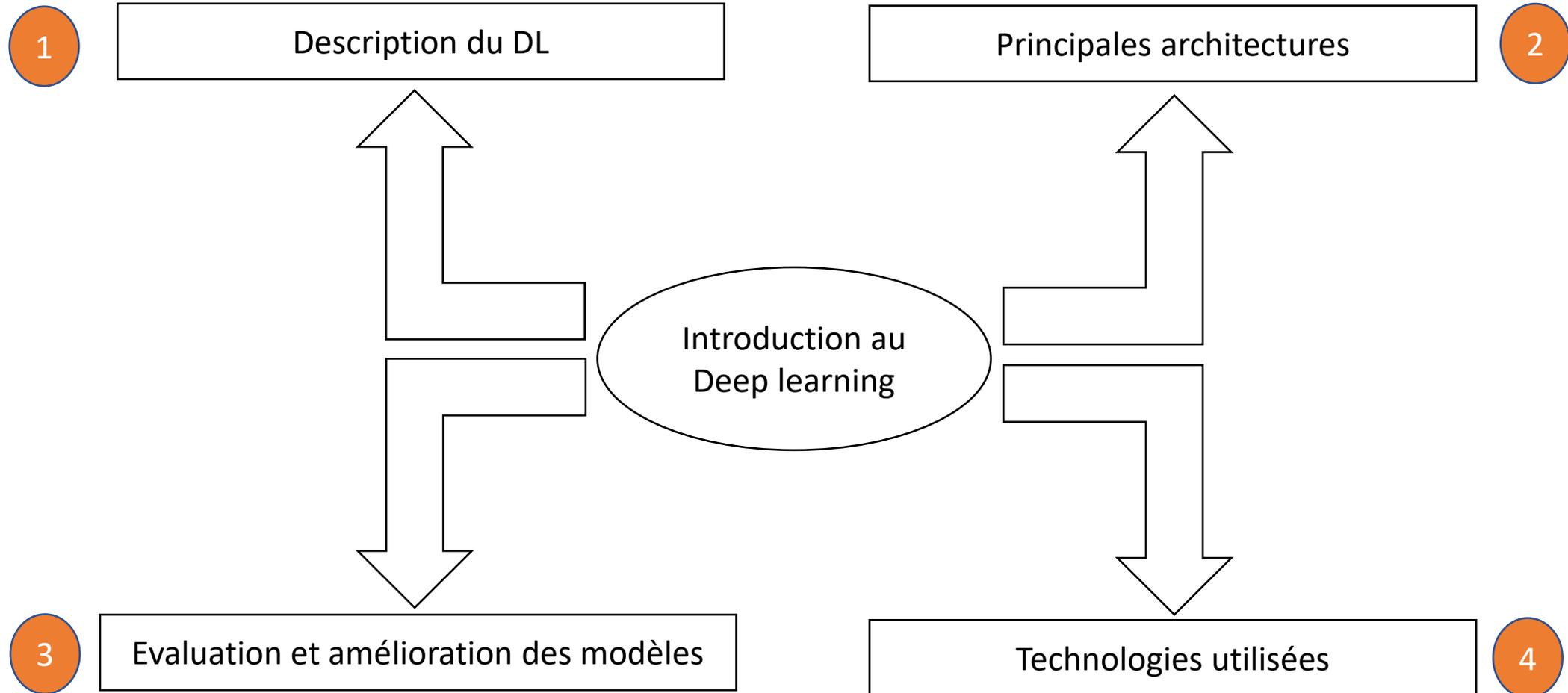
Matériel



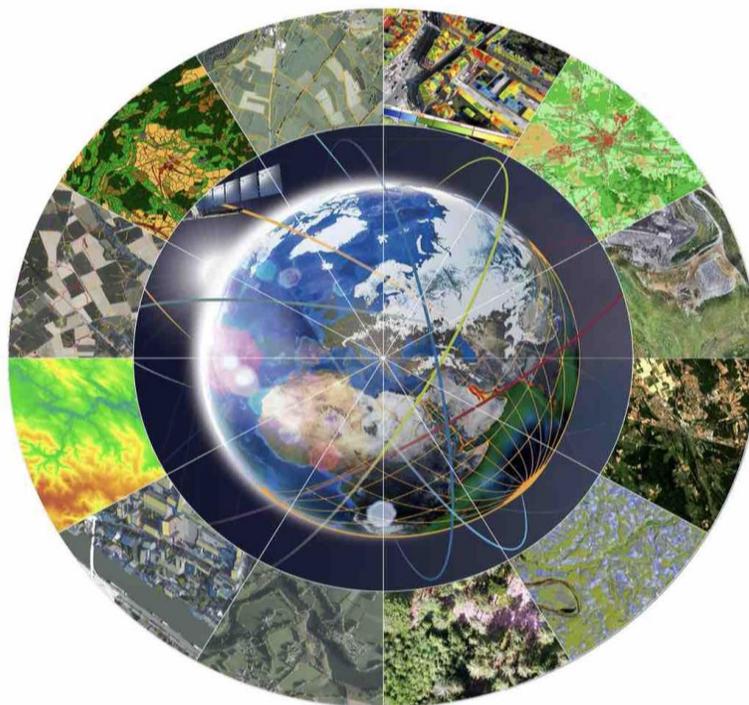
Cloud TPU



Bilan



MERCI POUR VOTRE ATTENTION



Dr Rose MOSKOLAÏ
rose.moskolai@gmail.com

Liens utiles

<https://www.simplilearn.com/tutorials/deep-learning-tutorial/>

<https://developers.google.com/machine-learning/>

<https://colab.research.google.com/>

[Application of Deep Learning Architectures for Satellite Image Time Series Prediction: A Review](#)

[Qu'est-ce que le deep learning ? | IBM](#)