

RER – Deep Learning

Contexte :

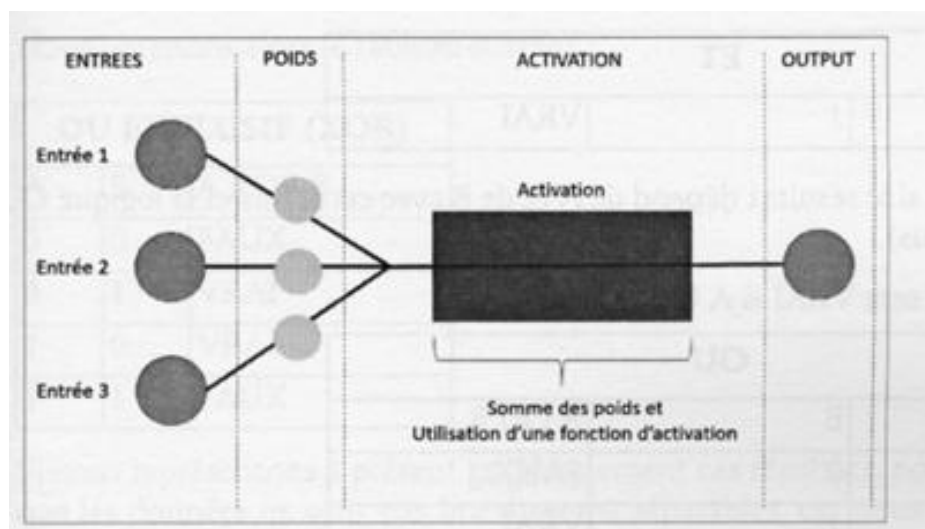
Se familiariser avec les différents concepts essentiels associés aux Réseaux Neuronaux et Deep Learning, dans un contexte d'apprentissage supervisé et comparer les performances d'algorithmes de Machine Learning (modèles non connexionnistes) et de Deep Learning.

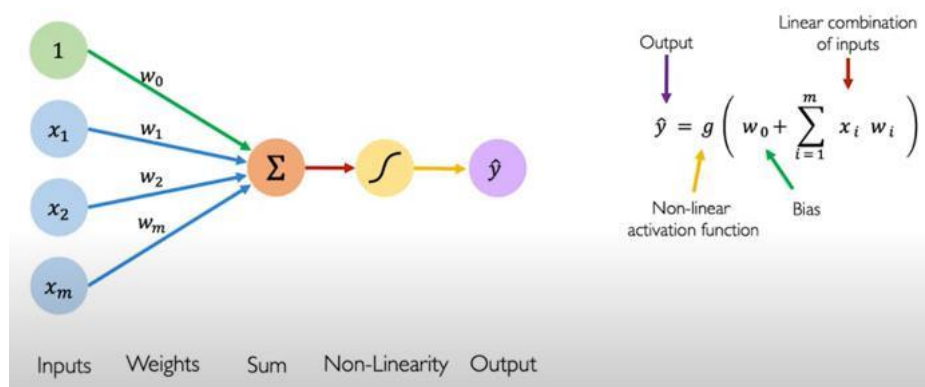
Problématique(s) :

Comment construire un réseau de neurones avec TensorFlow et Keras ? Comment utiliser ces bibliothèques ?

Mots clés :

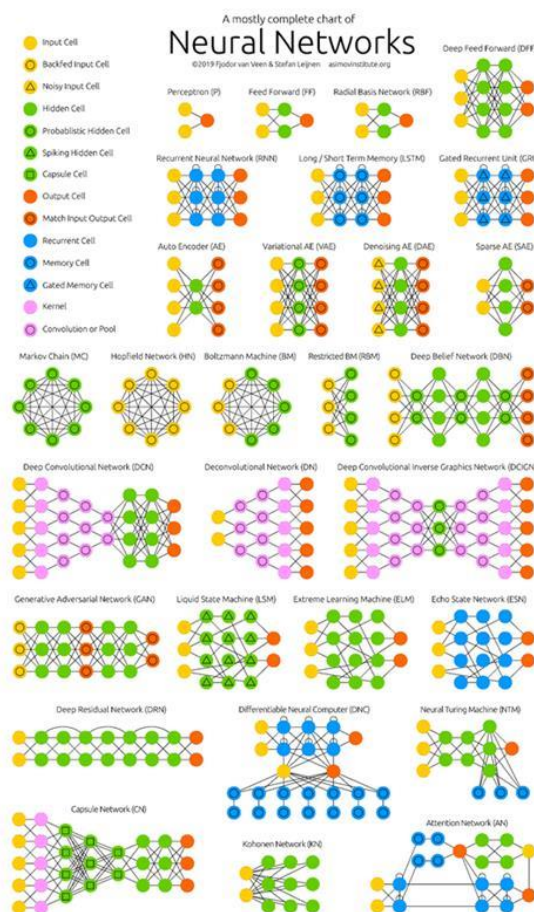
- Neurone artificiel : Structure mathématique ou informatique qui se comporte schématiquement comme un neurone biologique. Il reproduit le fonctionnement biologique. Il prend plusieurs entrées multipliées par des coefficients de pondération, réalise une sommation des entrées qui est ensuite transmise à une fonction d'activation. Si la somme excède un certain seuil le neurone artificiel s'active et produit une sortie.





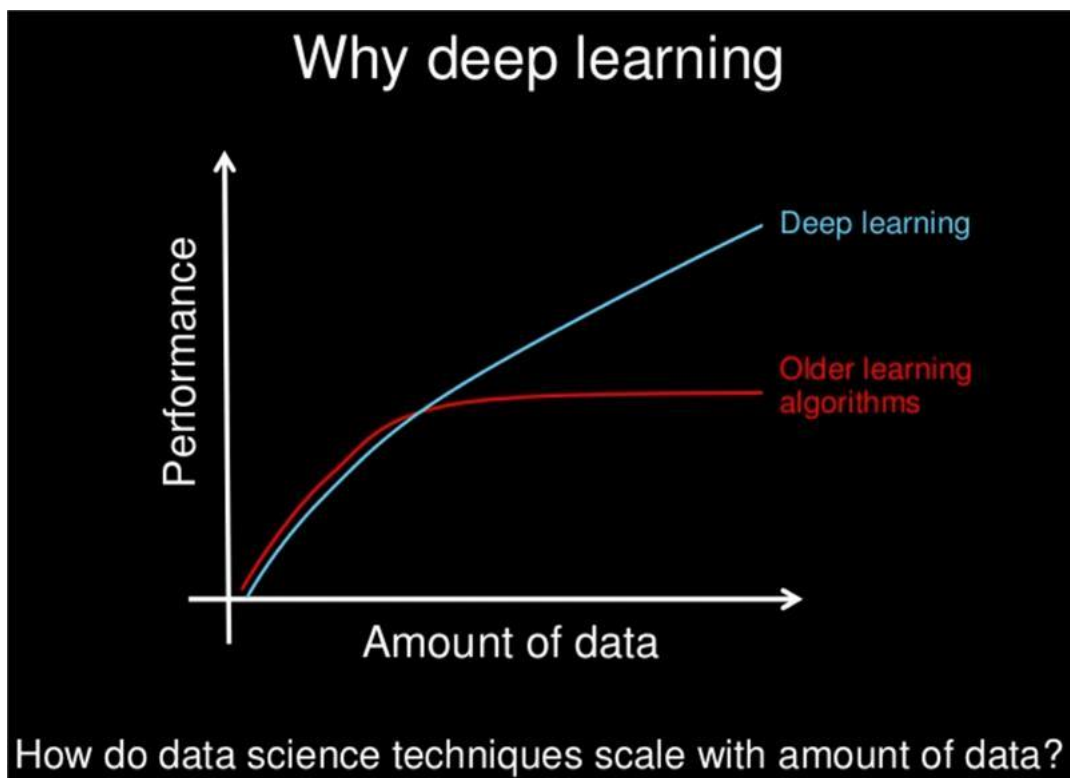
Neurone biologique	Neurone artificiel
Dendrites	Entrées (input)
Synapses	Poids
Axone	Sortie (output)
Activation	Fonction d'activation

- Réseau de neurones : Algorithme mathématique imaginé et conçu à partir des sciences cognitives. Combinaison de plusieurs neurones artificiels organisés en couches dans le but de multiplier leur capacité, utilisés pour faire du Machine Learning. Il existe plusieurs architectures du réseau de neurones détaillés ci-dessous.

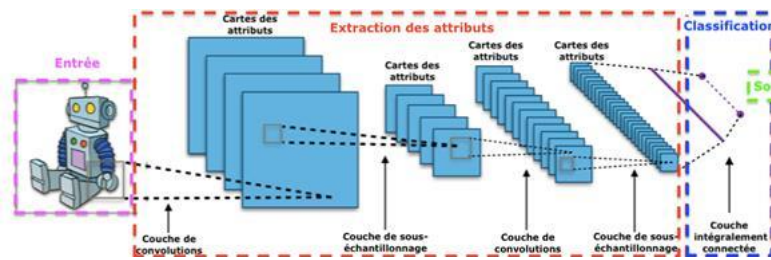


<https://www.asimovinstitute.org/neural-network-zoo/>

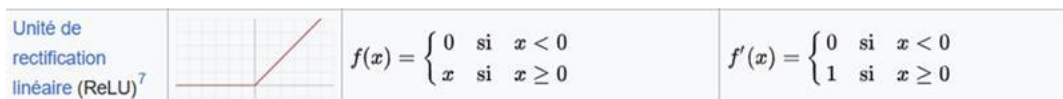
- **Perceptron** : Réseau de neurones à une couche qui permet de séparer des données linéairement séparables. Il peut être constitué d'un seul ou de plusieurs neurones. Un perceptron à une couche ne peut apprendre uniquement des fonctions linéaires.
- **Multilayer Perceptron (MLP)** : Assemblage d'éléments perceptrons en couches distinctes, connectées les unes aux autres de façon à posséder à minima 1 couche cachée (donc 2 pondérations). Les informations circulent de façon unidirectionnelle uniquement, de la couche d'entrée à la couche de sortie (feed forward).
- **Back Propagation** : (rétropropagation) algorithme d'apprentissage automatique supervisée utilisée pour mettre à jour le poids des neurones en calculant leurs sorties du réseau et en utilisant la méthode de descente du gradient pour minimiser cette erreur (rétropropagation).
- **Feed Forward** : Architecture de réseau de neurones qui n'autorise le transfert d'informations que dans le sens orienté de l'entrée vers la sortie. Les neurones d'une couche sont connectés à l'ensemble des neurones de la couche en aval.
- **Deep Learning** : (apprentissage profond). Sous domaine du Machine Learning qui exploite les réseaux de neurones artificiels. Plus le nombre de couche est élevé et plus le réseau est profond et puissant. Il doit contenir au moins 2 couches cachées.



- **CNN Réseau de neurones convolutif (Convolutional neural network)** :
Convolution = Opération mathématique (multiplication de matrices) qui prend 2 signaux en entrée et qui renvoie un nouveau signal.
Type de réseaux de neurones les plus performant notamment pour la classification d'images. Il consiste en une suite d'étapes de transformation de l'image d'entrée (convolution + *pooling*) suivi d'une étape de classification avec un classifieur plus classique.

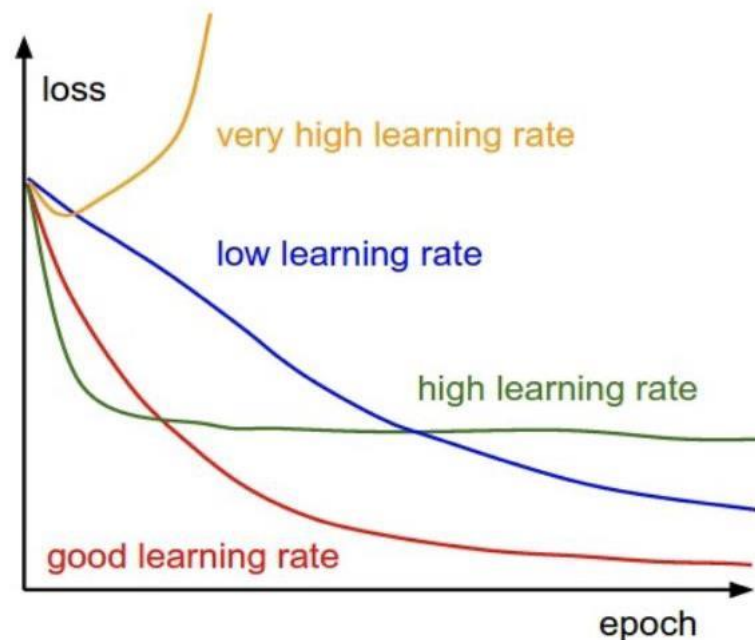


- *Recurrent neural network* : Type de réseau de neurones qui traite des données séquentielles comme des séries de mots ou des données dans une série temporelle. La force du réseau de neurone récurrent est leur capacité à prendre en compte des informations contextuelles suite à la récurrence du traitement de la même information. Dans un réseau récurrent, il existe au moins un neurone qui envoie des informations dans sa propre couche ou dans une couche en amont.
- *TensorFlow* : Plateforme open source de Deep Learning spécialisée dans le calcul tensoriel (calcul matriciel général). Elle est développée par Google. Elle intègre aujourd'hui Keras via une API pour l'entraînement des modèles.
- Keras : interface de programmation (API) python open source développée sur Theano, TensorFlow et CNTK. Elle a pour vocation de faciliter le développement de modèles de deep learning. Elle présente des fonctions d'export des modèles variées, et possède un large écosystème de modules.
- Epoch : Hyperparamètre représentant le nombre de cycles à travers l'ensemble complet des données d'entraînement. Un epoch représente une itération complète à travers tout le jeu de données d'entraînement pendant la phase d'entraînement d'un modèle. Plus epoch est grand et plus la précision est meilleure mais plus le temps de traitement est long.
- batch_size : Hyperparamètre indiquant le nombre d'échantillons utilisés lors d'une itération. Appelé batch size ou taille du lot.
- Itération : représente une étape de mise à jour des poids (= apprentissage) en fonction de epoch et de batch_size
- Fonction d'activation : Fonction mathématique qui est appliquée à la sortie d'un neurone. Elle autorise le passage d'information ou non selon la valeur. Il en existe plusieurs : Step (seuil), sigmoïde (logistique), Soft Max, ReLU, tanh,
- ReLU : (Rectified Linear Unit). Fonction d'activation utilisée en Deep Learning. Elle retourne la valeur de la fonction somme si elle est positive ou zéro si elle est négative. Elle est utilisée pour accélérer l'apprentissage (plus rapide que les fonctions sigmoïde et tanh)



- *Learning Rate* (step) / Taux d'apprentissage : Indique la vitesse à laquelle les coefficients sont mis à jour. C'est un hyperparamètre important en Deep Learning qui contrôle la magnitude de la mise à jour des poids du réseau neuronal pendant l'apprentissage. Le learning rate détermine l'amplitude de la modification des poids à chaque itération de l'algorithme d'optimisation. Si le learning rate est trop petit, l'entraînement peut être très

lent ou stagner, tandis que s'il est trop grand, les poids peuvent osciller et l'entraînement peut ne jamais converger.



- Biais : Ordonnée à l'origine. Nœud supplémentaire dans un réseau de neurones qui n'a pas de connexion en entrée mais qui est connecté à tous les neurones de la couche suivante et qui présente une valeur fixe (souvent égale à 1).
- *Self Organised Map (SOM)* : carte auto-organisatrice. Type de réseau neuronal non supervisé qui permet la visualisation et l'analyse de données multidimensionnelle complexes. Il permet de projeter les données multidimensionnelles sur un plan 2D de façon organisée. C'est le même type de réseau de neurone que Kohonem, du nom du savant qui parle de réseau de neurones auto-organisé. Il utilise une méthode d'apprentissage « le vainqueur gagne » (les neurones les plus stimulées sont favorisées, ceux qui ne le sont pas sont dévalués)

Hypothèses :

- Un réseau de neurone peut être utilisé pour générer des films / vidéos par ex. à partir de texte donné par un utilisateur (**Axel**) **Vrai**, il existe des réseaux de neurones dédiés (CVGAN)
- Le *Deep Learning* est plus sensible à l'*overfitting* que les algorithmes non connexionnistes (**Étienne**) **Faux, c'est l'inverse**
- Dans le cadre des données Titanic, nous ne disposons pas de suffisamment de données pour faire un modèle performant avec le *Deep Learning* (**Briand**) **Vrai**, de façon empirique, l'*accuracy* sur un algorithme de *Deep Learning* est inférieur à celui obtenu avec un algorithme connexionniste
- Il existe des réseaux de neurones spécialisés (par ex. en *computer vision* ou en *NLP*). Càd des réseaux de neurones plus adaptés à certaines tâches (**Aude**) **Vrai**
- Chaque couche de réseau de neurones peut réaliser des fonctions différentes (**Loïc**), **Vrai** (par exemple, workshop XOR)

- Pour faire de la classification ou de la régression, 3 couches (1 couche d'entrée et deux couches cachées), càd 3 vecteurs de poids, suffisent (**Adrien**) **Vrai, mais ce n'est pas le plus optimal.**
- Le *Deep Learning* est plus utilisé que le *Machine Learning* (non connexionniste) (**Osman**), **Faux, cela dépend du nombre de données disponible pour l'entraînement.**
- Le Deep Learning est plus approprié pour la reconnaissance d'images ou photos (*computer vision*). (**Adeline**) **Vrai**
- Il faut un environnement de travail spécialement dédié à l'entraînement d'un modèle de *Deep Learning* (**Jean Paul**) **Partiellement vrai car le deep Learning utilise un nombre important de bibliothèques. C'est donc préférable.**
- MLP et CNN sont utilisés pour différents types de problèmes (**Tetyana**) **Vrai, MLP est plus adapté pour des données structurées et des problèmes de classification et de régression, tandis que les CNN sont plus adaptés pour des données non structurées telles que des images et des problèmes de vision par ordinateur**
- Le *batch_size* consiste en le nombre d'itérations réalisé par le réseau de neurone pour ajuster les poids (**Nicolas**), **Vrai, mais cela dépend du nombre d'Epoch Vrai peut Epoch = 1**
- *Epoch* consiste en un nombre d'itération réalisé par les réseaux de neurone pour ajuster les poids (**Seydou**) **Vrai mais cela dépend du nombre de batch_size. Vrai si batch_size = nb d'exemples.**

Plan d'action :

- Explorer les ressources
- Définir et comprendre les mots clefs
- Répondre à la problématique
- Vérifier les hypothèses
- Comparaison entre Keras et TensorFlow

Caractéristique	Keras	TensorFlow
Niveau d'abstraction	Élevé	Bas
Convivialité	Facile à utiliser et à comprendre	Plus difficile à prendre en main
Flexibilité	Limitée	Grande
Performance	Moins performant que TensorFlow	Très performant, en particulier pour les modèles à grande échelle
Modularité	Forte	Faible
Compatibilité avec d'autres frameworks	Peut être utilisé avec TensorFlow, Theano, CNTK	Est un framework utilisé dans plusieurs autres outils de machine learning
Types de modèles	Convolutionnels et récurrents	Convolutionnels, récurrents, et bien plus encore
Support communautaire	Fort	Très fort
Documentation	Complète et bien structurée	Complète et bien structurée

- Faire les Workshop (réaliser d'abord ScikitLearn, Keras puis TensorFlow, dans cet ordre)
- RER