

Le Deep Learning

L'intelligence artificielle recouvre l'ensemble des technologies susceptibles de comprendre leur environnement et de prendre des décisions en conséquence, sans supervision humaine. Aujourd'hui les ordinateurs sont capables de performances habituellement associées à l'intelligence humaine. Ils peuvent même dépasser nos champions dans différents domaines. Par exemple en 2015, le programme AlphaGo de Google DeepMind a battu le maître sud-coréen Lee Se-dol dans le jeu de société Go.

De nos jours, l'intelligence artificielle est sur toutes les lèvres. Demain elle sera non seulement l'organe vital de nos entreprises mais également les organes sensoriels de notre quotidien (nos yeux, nos oreilles, notre voix). L'intelligence artificielle nous accompagne déjà actuellement, à plus petite échelle. Son introduction dans notre environnement devrait croître de manière exponentielle à mesure que les performances des modèles seront vérifiées et que la puissance de nos processeurs évoluera. La course à la performance est lancée et le monde ne cessera d'évoluer en conséquence.

L'intelligence artificielle (ou l'IA *pour les intimes*) peut être divisée en deux grandes familles : le Machine learning et le Deep learning. Le site Web Foundation a réalisé un schéma qui synthétise de manière claire ces 3 notions :

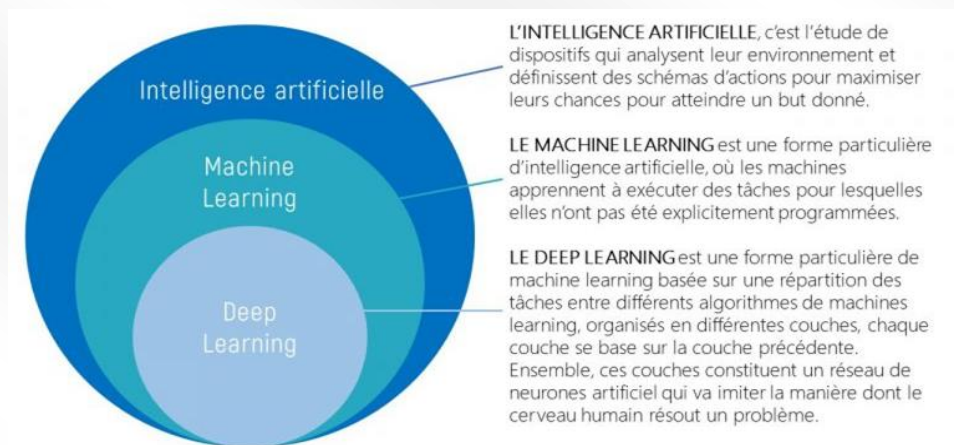


Figure 1 : Les grandes familles de l'Intelligence artificielle - © Copyright Webfoundation.org

Le Machine learning est une approche de l'IA et le Deep learning un sous-domaine du Machine learning.

1. L'avènement du Deep Learning

C'est en 1957 que le psychologue américain Frank Rosenblatt pose les premières pierres du Deep learning. Il a fallu patienter quelques années pour concrétiser ses travaux. La technologie de l'époque n'était pas assez évoluée pour modéliser des algorithmes complexes. En effet, pour qu'un réseau de neurones fonctionne il faut deux éléments essentiels :

- **Une quantité de données très importante**, donc une capacité de stockage en conséquence,
- **Une puissance de calcul évoluée**, donc des ordinateurs avec de très bons processeurs.

L'entreprise Backblaze a étudié l'évolution du prix de stockage au cours du temps. Les résultats du graphique ci-dessous démontrent que plus la puissance de stockage augmente, plus le prix de stockage diminue :

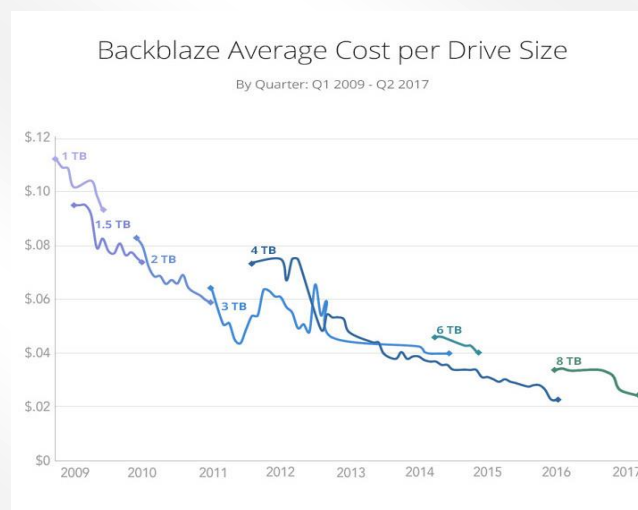


Figure 2 : Evolution du coût de stockage par gigaoctet- © Copyright Backblaze

Le coût de stockage diminue d'année en année tandis que la volumétrie des données explose.

A contrario, la puissance de nos ordinateurs évolue au cours du temps. Selon la loi de Moore, le nombre de transistors double tous les deux ans. Le graphique ci-dessous nous apprend qu'en 2010 les calculateurs surpassaient la capacité du cerveau des rats et qu'en 2045 la puissance des ordinateurs dépasserait l'entendement humain :

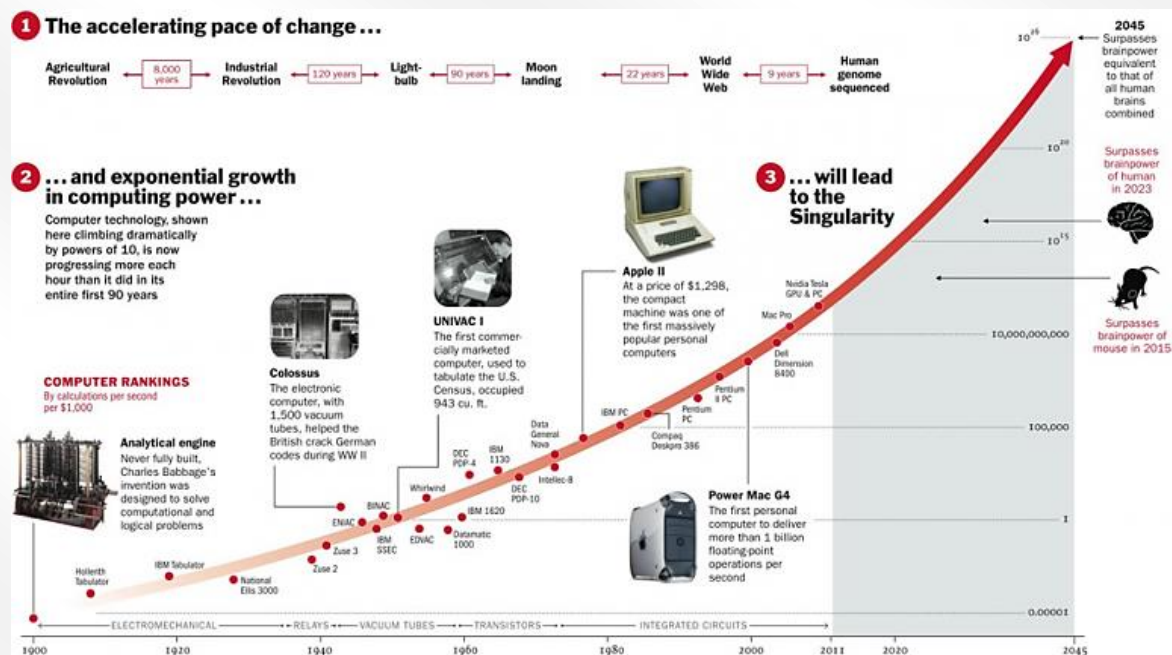


Figure 3 : Evolution de la puissance des ordinateurs - © Copyright Speechi

En 1980, Geoffrey Hinton devient le pionnier du Deep learning. Les ordinateurs sont alors assez puissants et la volumétrie des données assez importante pour tester ce nouveau type d'algorithme. Grâce à ses recherches, les ordinateurs sont capables d'apprendre et d'interpréter de manière autonome des corrélations entre différents phénomènes.

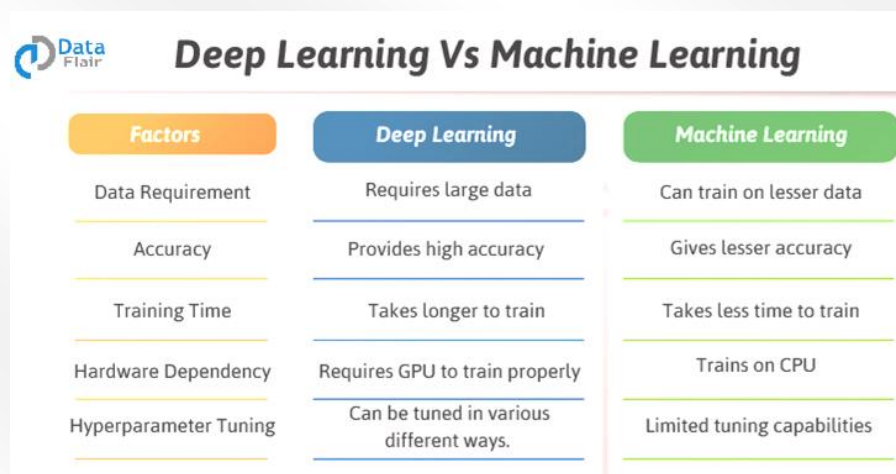
L'idée du Deep learning est de répliquer le fonctionnement du cerveau humain. Le cerveau est un des outils biologiques les plus complexes et performants pour apprendre, comprendre et interagir avec le monde environnant. Le nombre de neurones dans le cerveau humain se compte en milliard. Le cerveau est un réseau de neurones interconnecté. Il nous permet de réaliser inconsciemment de véritables prouesses telles que le déchiffrement presque instantané d'un chiffre manuscrit, la classification des objets qui nous entourent ou la compréhension de la perspective.

Les modèles de Deep Learning mettent en relation un nombre prédéfini de neurones artificiels au sein de plusieurs couches afin de détecter des informations de plus en plus spécifiques dans les données d'entrée. Ils produiront une prédiction ou une classification en sortie.

Tout comme le cerveau humain, plus nous fournissons des exemples d'apprentissage, plus le modèle se trompera, plus il apprendra et plus il sera performant.

2. La différence entre le Machine learning et le Deep learning

Le Machine learning et le Deep learning diffèrent de par leur complexité et le type de problèmes auxquels ils répondent. Le Machine learning est moins coûteux à mettre en place au sein des entreprises car il nécessite moins de ressources pour fonctionner. Cependant il ne pourra résoudre des problèmes complexes comme la vision par ordinateur, l'analyse en profondeur des textes ou l'analyse des sons.



Factors	Deep Learning	Machine Learning
Data Requirement	Requires large data	Can train on lesser data
Accuracy	Provides high accuracy	Gives lesser accuracy
Training Time	Takes longer to train	Takes less time to train
Hardware Dependency	Requires GPU to train properly	Trains on CPU
Hyperparameter Tuning	Can be tuned in various different ways.	Limited tuning capabilities

Figure 4 : Le Deep learning VS. Le Machine learning- © Copyright DataFlair

Le Machine learning désigne l'application de méthodes statistiques aux algorithmes de programmation. L'idée est de pouvoir généraliser les apprentissages à de nouvelles données. L'objectif est de construire des modèles qui ajustent le mieux les données observées à la courbe d'apprentissage. L'algorithme détermine la meilleure approximation des données sans que l'humain n'ait à lui imposer les coefficients du modèle. Cependant un expert des données doit au préalable nettoyer et trier les données pour améliorer au maximum les performances. Il doit choisir des caractéristiques ayant du sens vis à vis de la variable à prédire car le modèle établit des corrélations à partir de toutes les variables que l'expert lui fournit en entrée.

Avec le Deep learning, le tri préalable des variables n'est pas utile car c'est l'algorithme qui va trouver lui-même les corrélations dans les données. L'algorithme va adapter les liaisons entre ses neurones et mettre à jour les coefficients du modèle, sans supervision humaine.

Pour imaginer, si nous voulons cuisiner une tarte aux pommes façon Machine learning nous devrions présélectionner tous les ingrédients nécessaires avant de mettre le tout dans notre robot cuistot. Le robot se chargerait de définir les meilleures quantités de chaque ingrédient pour la recette parfaite. Si maintenant nous la cuisinons façon Deep learning nous n'aurions même pas besoin de présélectionner les ingrédients, le robot se chargerait non seulement de déterminer les meilleurs ingrédients pour réaliser la tarte aux pommes mais également leur quantité. Le travail en amont est donc moins important mais le robot nécessite plus de mémoire et de puissance pour traiter les informations.

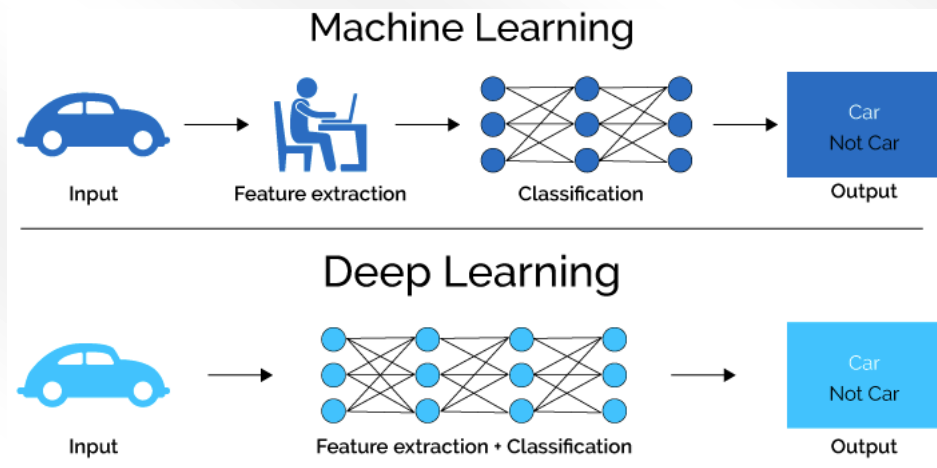


Figure 5 : Supervision humaine VS Apprentissage profond- © Copyright Xenonstack

3. Le Deep learning en entreprise

Le Deep learning fournit actuellement les meilleures solutions à de nombreux problèmes de reconnaissance d'images, reconnaissance vocale et de traitement du langage naturel. Ils sont déployés à grande échelle par des entreprises telles que Google, Microsoft et Facebook.

Une étude de McKinsey Global Institute (cf. Ressources [5]) montre comment les techniques d'apprentissage en profondeur peuvent accompagner des analyses plus traditionnelles déjà déployées au sein des entreprises, comme le Machine learning. Les réseaux de neurones profonds révolutionnent de nombreux secteurs d'activité car :

- Ils permettent de traiter de grands volumes de données tout en améliorant la performance des prédictions,
- Ils permettent d'exploiter des sources de données jusqu'alors inutilisées comme les images des capteurs pour la maintenance prédictive, des sons enregistrés lors d'échanges téléphoniques pour le suivi de la clientèle, des corpus de textes pour analyser les tendances de marché pour la finance et bien d'autres.

L'implémentation du Deep learning en entreprise prend tout son sens dans des secteurs où les sources de données sont variées et les données non structurées :

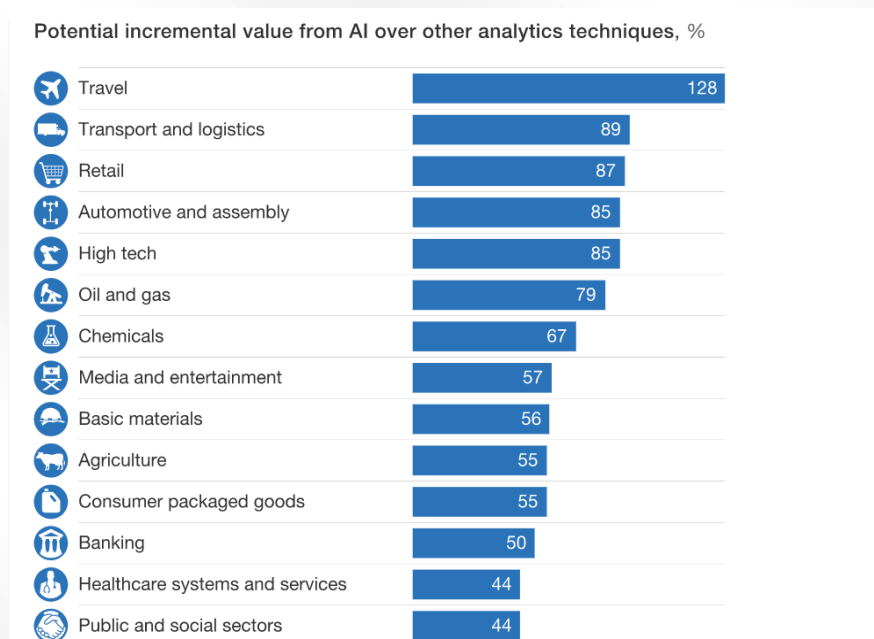


Figure 6 : Les secteurs d'activité où le Deep learning a le plus de valeur- © Copyright McKinsey Global Institute

4. Le réseau de neurones à propagation avant

Un réseau de neurones à propagation avant est un réseau acyclique (non bouclé), à la différence des réseaux de neurones récurrents. L'information ne se propage que dans le sens avant, de l'entrée vers la sortie.

4.1. Son architecture

Tout réseau de neurones comprend trois types de couches :

- La couche d'entrée
- La couche cachée
- La couche de sortie

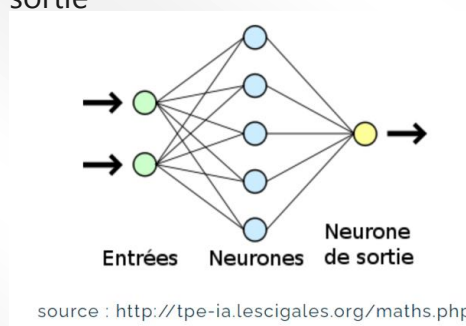


Figure 7 : Architecture d'un réseau de neurones à 1 couche cachée

Chacune de ces couches est constituée de plusieurs neurones. Les neurones contiennent des valeurs quantitatives continues.

Les neurones de la couche d'entrée représentent nos variables indépendantes d'entrée, standardisées, pour un seul enregistrement (ou un individu).

Les neurones des couches cachées_(n) prennent en entrée une somme pondérée des neurones de la couche précédente_(n-1). Chaque neurone de la couche précédente_(n-1) a un **poids** dans la prise de décision du neurone de la couche traitée_(n). Plus le poids du neurone est important plus il aura de l'importance dans le réseau. Le modèle procède alors à une multiplication matricielle entre la matrice des poids en entrée de la couche_(n) et les valeurs des neurones de la sortie précédente_(n-1).

Les neurones des couches cachées représentent les caractéristiques, les spécificités apprises par le modèle. Ce sont des liens imperceptibles pour l'homme que le modèle aura déterminé au cours de son apprentissage. En sortie de la couche cachée le modèle décide si oui ou non le neurone sera activé et pris en compte dans la décision de la prochaine couche. Pour ce faire une fonction d'activation est appliquée aux entrées des neurones. La fonction la plus utilisée au niveau des couches cachées est la **fonction redresseur ou RELU** : $\phi(x) = \max(0, x)$ car elle apporte de la non-linéarité au modèle sans pour autant modifier la sortie si celle-ci est positive.

Les neurones de la couche de sortie prennent en entrée les sorties des fonctions d'activation de la dernière couche cachée et appliquent généralement une **fonction d'activation sigmoïde** à la somme pondérée effectuée : $\phi(x) = \frac{1}{(1+e^{-x})}$ pour une classification binaire ou la **fonction d'activation softmax** : $\phi(x_j) = \frac{e^{x_j}}{(\sum_1^k (1+e^{x_k}))}$ pour une classification multiclassées. La fonction sigmoïde permet d'obtenir une probabilité (donc une quantité continue entre 0 et 1) et la fonction softmax permet d'obtenir des probabilités pour chaque label et de fixer leur somme à 1. Le label ayant la probabilité la plus grande remporte la partie.

A savoir qu'il existe également d'autres types de fonction d'activation comme la fonction seuil et la fonction tangente hyperbolique. Les fonctions d'activation permettent d'apporter de la non-linéarité dans le modèle ce qui permet de répondre à tout type de problèmes naturels (selon le théorème d'approximation universel).

Plus le nombre de couches cachées augmente, plus le réseau est profond et complexe. Le nombre de couche correspond au nombre de matrice de poids contenu dans le réseau. Les poids déterminent à quel point le neurone est important dans la prise de décision du modèle. L'apprentissage du modèle consiste à mettre à jour les poids du modèle pour généraliser au mieux les données d'entrée.

4.2. Son apprentissage

Pendant la phase d'apprentissage, des données labellisées sont fournies au modèle. L'objectif du modèle à propagation avant est de prédire une quantité ou une classe, le plus proche possible de la vraie valeur. La valeur de sortie prédite par le modèle \hat{y} est comparée à la vraie valeur de l'observation y . Pour vérifier si la valeur prédite est proche de la vraie valeur nous allons appliquer une fonction de coût. La fonction de coût mesure la qualité des prédictions effectuées par le modèle vis-à-vis de la réponse attendue. En d'autres termes la fonction de coût quantifie les erreurs de prédiction effectuées par le modèle.

La fonction de coût la plus populaire pour un problème de régression est **la fonction des moindres carrés** $\mathcal{C} = \sqrt{\sum(\hat{y} - y)}$. Pour un problème de classification nous utilisons une fonction de coût logarithmique : **binary cross entropy** pour une classification binaire et la **categorical cross entropy** pour une classification multilabels. L'objectif est de minimiser la fonction de coût grâce à l'algorithme de la descente du gradient stochastique (**adam**). Ainsi le gradient est renvoyé dans le réseau afin de mettre à jour les poids. On appelle cette étape la rétropropagation.

Tout l'enjeu de l'apprentissage réside dans la mise à jour des poids. Ce sont les valeurs des poids qui influencent la sortie du modèle. La mise à jour des poids se fait après avoir calculé la fonction de coût pour chaque observation dans le cas d'un apprentissage renforcé ou après un nombre fixé d'observations (**batch size**) dans le cas d'un apprentissage par mini lot. Les valeurs de poids sont modifiées de manière simultanée en fonction de l'impact qu'ils ont sur l'erreur. **Le taux d'apprentissage** détermine la vitesse à laquelle nous effectuons la descente du gradient.

Lorsque tout le jeu de données d'apprentissage est rentré dans le neurone cela forme une **époque**. Nous pouvons faire plusieurs époques afin de former au mieux le modèle.

4.3. Son implémentation

Il existe différentes librairies de Deep learning en Python :

- **Theano** (module de calcul numérique open source basé sur Numpy) qui permet de faire des calculs rapides :

→ utilisation du CPU (calcul classique de l'ordinateur pour ouvrir un document par exemple)

→ utilisation du GPU de l'ordinateur (calculs en parallèle pour les jeux vidéo, ou le Deep learning). Le GPU est intéressant pour la propagation avant simultanée de l'activation des neurones et pour la rétropropagation (la mise à jour des poids).

- **TensorFlow** qui permet de recoder de nouveaux réseaux de neurones à partir des fonctionnalités de bases :

→ utilisation du CPU

→ utilisation du GPU

- **Keras** qui permet de ne pas réinventer la roue : Keras rassemble les deux modules précédents pour créer des réseaux de neurones profonds avec quelques lignes de codes.

Exemple d'implémentation d'un réseau artificiel de neurones à 2 couches cachées pour un problème de classification binaire, en utilisant la librairie Keras:

```
from keras.models import Sequential # module qui permet d'initialiser le réseau de neurones
from keras.layers import Dense # module qui permet de créer les couches du réseau

# Initialiser le réseau
clf = Sequential()

#Ajout des couches cachées
# La fonction Dense prend plusieurs arguments en entrée :

- units= nombre de neurones de la couche cachée que l'on souhaite créer. Il n'y a pas de règle préétablie mais l'astuce est de prendre la moyenne entre le nombre de variables d'entrée et le nombre de neurones en sortie.

- activation : fonction d'activation des neurones de la couche

- kernel_initialization : choix des poids pour les neurones. Lors de l'initialisation les poids sont choisis de manière aléatoire suivant différentes lois.

- input_dim = seulement pour la première couche cachée afin de spécifier le nombre de neurones en entrée du réseau.

clf.add(Dense(units = 128, kernel_initializer = 'uniform', activation = 'relu', input_dim = 13))
clf.add(Dense(units = 64, kernel_initializer = 'uniform', activation = 'relu'))

#Ajout de la couche de sortie
clf.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))

#Compilation du réseau pour définir les méthodes d'apprentissage
clf.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

#Entraînement du modèle
clf.fit(X_train, y_train, batch_size = 10, epochs = 100)

# Prédictions
pred= clf.predict(X_test)

#Évaluer le modèle
# Train_loss= score_train[0], accuracy_train=score_train[1]
score_train = clf.evaluate(X_train, y_train)
score_test = clf.evaluate(X_test, y_test)
```

Ressources

- [1] <https://experiences.microsoft.fr/business/intelligence-artificielle-ia-business/comprendre-utiliser-intelligence-artificielle/>
- [2] <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>
- [3] <https://penseeartificielle.fr/difference-intelligence-artificielle-machine-learning-deep-learning/>
- [4] <http://neuralnetworksanddeeplearning.com/chap2.html>
- [5] <https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning>
- [6] <https://stats.stackexchange.com/questions/154879/a-list-of-cost-functions-used-in-neural-networks-alongside-applications>