

# Intelligence Artificielle 2

## Cours 4 - Intro. au Machine et Deep Learning

Steve Lévesque, Tous droits réservés © où applicables

# Table des matières

- 1 Introduction au concept de l'apprentissage automatique et profond (Machine Learning and Deep Learning)
  - Définition
  - Réseaux de neurones artificiel profond
  - Exemple - CNN pour MNIST

# Machine Learning et Deep Learning - Définition

**L'apprentissage profond (Deep Learning)** est un sous-ensemble de **l'Apprentissage Automatique (Machine Learning)** étant lui-même un sous-ensemble de **l'AI**.

Il peut être dit que les **sciences des données** à une partie prenante sur tout cela.

# Machine Learning et Deep Learning - Définition

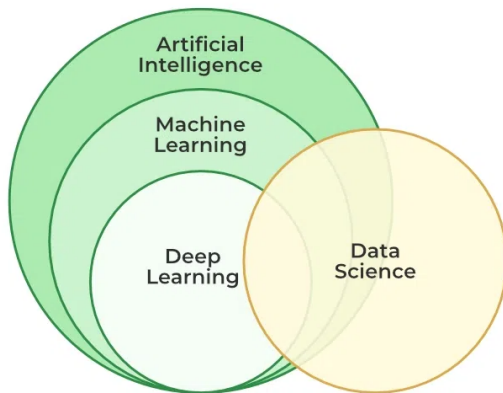


Figure: <https://www.geeksforgeeks.org/introduction-deep-learning/>

# Machine Learning et Deep Learning - Définition

Pourquoi est-ce que le Deep Learning est le sous-ensemble du Machine Learning ?

Puisque le Deep Learning utilise des données, entraîne avec le même principe en général, et est basé sur un aspect du Machine Learning étant les réseaux de neurones.

D'où la reconnaissance de mettre le Deep Learning comme un sous-ensemble spécialisé du Machine Learning (réseau de neurones profonds).

# Machine Learning et Deep Learning - Définition

Le Deep Learning est en lien avec les réseaux de neurones artificiels (le Machine Learning introduit le perceptron, le neurone simple), mais ceux-ci sont très profond dans le Deep Learning. Ceci permet de généraliser de manière très large avec des quantités de données phénoménales.

Avec les capacités des ordinateurs au niveau des processeurs et du stockage de nos jours, le Deep Learning prends tout son sens et est largement utilisé.

La littérature existe depuis un certain temps, avant que le sujet soit populaire, mais la concrétisation physique était peu viable, et donc celle-ci est restée discrète/cachée.

# Réseaux de neurones artificiel profond

Le neurone biologique traditionnel (neurosciences) à joué un rôle dans la conception du neurone d'aujourd'hui dans un réseaux de neurones en Machine/Deep Learning.

Il y a beaucoup d'aspects ressemblant au modèle biologique.

# Réseaux de neurones artificiel profond

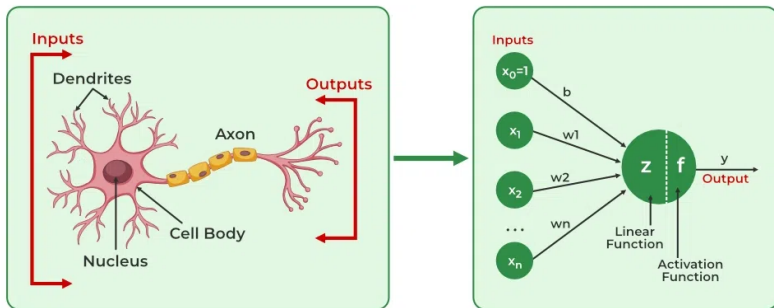


Figure: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>



# Réseaux de neurones artificiel profond - Différences ML et DL

Le **Deep Learning** est meilleur pour :

- Gérer des cas complexes et provenir des bons résultats tout de même vis-à-vis cette complexité
- Le traitement des images (extraction automatique des caractéristiques avec les neurones)

Par contre, moins efficace pour :

- Économiser des ressources (consomme beaucoup de GPU)
- Obtenir des résultats avec peu de données
- Le suivi/débuggage
  - Puisque celui-ci est un modèle de type “boîte noire”, on ne peut pas savoir exactement (à ce jour) les interprétations, il y a des aspects stochastiques (aléatoires)

# Réseaux de neurones artificiel profond - Différences ML et DL

Le **Machine Learning** est meilleur pour :

- L'interprétation théorique et lorsque l'utilisation des données est moindre
- Entraîner rapidement les modèles

Par contre, moins efficace pour :

- Ne généralise pas aussi bien lorsque beaucoup de données globales/généralisées sont disponibles pour l'entraînement
- Un modèle statistique n'est généralement plus suffisant si le domaine d'application utilise des données massives

# Exemple - CNN pour MNIST

Avec nos nouvelles connaissances, il est possible de revenir sur l'exemple d'introduction du tout début du cours.

On peut sauvegarder la topologie séquentielle dans un graphique pour analyser plus facilement les couches.

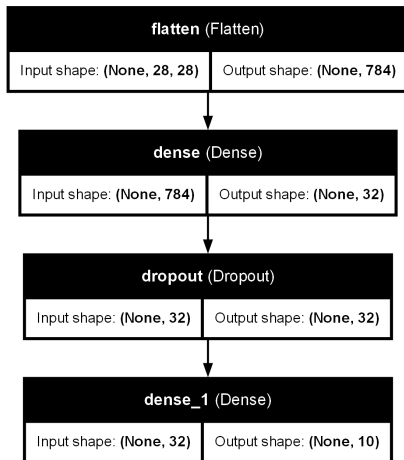
# Exemple - CNN pour MNIST

```
1 # Copyright 2020 Google LLC. https://www.apache.org/licenses/LICENSE-2.0
2
3 # All models in this course are sequential.
4 model = tf.keras.models.Sequential()
5
6 # The features are stored in a two-dimensional 28X28 array. Flatten that
7 # two-dimensional array into a one-dimensional 784-element array.
8 model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))
9
10 # Define the first hidden layer.
11 model.add(tf.keras.layers.Dense(units=32, activation='relu'))
12
13 # Define a dropout regularization layer.
14 model.add(tf.keras.layers.Dropout(rate=0.2))
15
16 # Define the output layer. The units parameter is set to 10 because
17 # the model must choose among 10 possible output values (representing
18 # the digits from 0 to 9, inclusive).
19 # Don't change this layer.
20 model.add(tf.keras.layers.Dense(units=10, activation='softmax'))
```

# Exemple - CNN pour MNIST

```
1  # Copyright 2020 Google LLC. https://www.apache.org/licenses/LICENSE-2.0
2  import tensorflow as tf
3  from tensorflow.keras.utils import plot_model
4
5  model = tf.keras.models.Sequential()
6  model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))
7  model.add(tf.keras.layers.Dense(units=32, activation='relu'))
8  model.add(tf.keras.layers.Dropout(rate=0.2))
9  model.add(tf.keras.layers.Dense(units=10, activation='softmax'))
10
11  plot_model(model, to_file='2.1-mnist_nn_model_layers_graphic.png',
12             show_shapes=True, show_layer_names=True)
```

# Exemple - CNN pour MNIST



# Bibliographie

- <https://www.geeksforgeeks.org/introduction-deep-learning/>
- <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>
- <https://www.datacamp.com/tutorial/machine-deep-learning>