

# Cours: Apprentissage Profond – Fondements et Architectures (Deep Learning I)

Dieu-Donné FANGNON

ISMADE: Master 1-2 (SE), 2025-2026

E-mail: [dieu-donne.fangnon@inria.fr](mailto:dieu-donne.fangnon@inria.fr)

Total hours: 30H

Office: Online - ISMADE

Web: [linkedin.com/dieu-donnef](https://www.linkedin.com/dieu-donnef)

Class Hours: Saturday 19-22:00pm

Class Room: Online - ISMADE

Lab Room: Online - ISMADE

Lab Hours: 20:30-22:00pm

---

## Description du cours

Ce cours (dont tout le matériel sera en anglais) introduit les fondements théoriques et pratiques de l'apprentissage profond (Deep Learning I). Il vise à fournir une compréhension progressive des réseaux de neurones, depuis le neurone artificiel jusqu'aux architectures modernes utilisées en vision par ordinateur, en mettant l'accent sur l'intuition mathématique, la modélisation et la mise en œuvre pratique avec des frameworks actuels (Keras et PyTorch).

Le cours adopte une approche pédagogique progressive, reliant systématiquement les concepts théoriques aux applications concrètes.

## Objectifs pédagogiques

À l'issue de ce cours, l'étudiant sera capable de :

- Comprendre le fonctionnement interne des réseaux de neurones artificiels
- Concevoir et entraîner des réseaux de neurones multi-couches (MLP)
- Appliquer les MLP à des tâches de régression et de classification
- Comprendre les limites des MLP pour les données visuelles
- Expliquer et utiliser les réseaux de neurones convolutifs (CNN)
- Comprendre les principes fondamentaux des modèles de détection d'objets
- Utiliser les autoencodeurs pour la réduction de dimension et la représentation latente

- Comprendre les enjeux de l'explicabilité des modèles profonds (XAI)
- Implémenter des modèles en Keras et PyTorch et comparer leurs paradigmes

—

## Prérequis

- Bases en machine learning supervisé
- Algèbre linéaire (vecteurs, matrices)
- Calcul différentiel (notions de dérivées)
- Programmation en Python

## Installations

Pour la partie pratique de cet enseignement, il est possible de travailler sur des machine distantes ou bien en local sur votre ordinateur. Si vous travaillez sur votre ordinateur, il est nécessaire d'avoir installé de quoi programmer en python version 3 ainsi que les librairies:

- numpy
- matplotlib
- sklearn
- pytorch
- keras version 3 avec pytorch en backend

Il est conseillé d'utiliser un environnement virtuel pour éviter des éventuels conflits avec des contraintes d'autres cours et d'autres versions. Pour gérer différents environnements en python dans les notebooks, vous pouvez suivre ce document: <https://towardsdatascience.com/get-your-conda-environment-to-show-in-jupyter-notebooks-the-easy-way-17010b76e874>

Aide pour l'installation sur un système Ubuntu possédant déjà une installation python par anaconda:

- `conda create -n mlkeras3` (vous pouvez bien-sûr choisir un autre nom)
- `conda create -n mlkeras3` (vous pouvez bien-sûr choisir un autre nom)
- `conda activate mlkeras3`
- `conda install scikit-learn, matplotlib`
- installer pytorch en suivant les instructions du site <https://pytorch.org> (choisissez conda et python, les autres options dépendent de votre ordinateur)
- `pip install --upgrade keras`

Il faut ensuite éditer le fichier `./keras/keras.json` et modifier le backend en "torch"

Pour vérifier l'installation, lancer python puis:

- `import keras`
- `print(keras.__version__)`

Les sujets de TP sont généralement sous la forme d'un notebook jupyter. Si vous lancez les TPs localement sur votre machine, il est intéressant d'utiliser certaines extensions des notebooks comme par exemple ceux cités sur cette page [jupyter-notebook-extensions](#).

Même sans extension, il peut être intéressant de comprendre la syntaxe <https://markdownguide.offshoot.io/basic-syntax/> des cellules 'Markdown'.

Si vous disposez d'une bonne connexion Internet, il est également possible de faire les TPs sur google colab <https://colab.research.google.com>.

## Plan du cours:

Les supports de cours, les TP.. seront dispo dans ce dossier Drive -> [Cours\\_Deep-Learning-I\\_ISMADE](#)

### 1 – Prise de mains en Python

- Installation de Python via [Anaconda](#)
- Quelques exercices de programmation en [python](#)

### 2 – Introduction à l'Apprentissage Profond

- Machine Learning vs Deep Learning
- Neurone artificiel et perceptron
- Fonctions d'activation
- Fonctions de perte
- Descente de gradient
- Sur-apprentissage et généralisation

### 3 – Réseaux de Neurones Multi-Couches (MLP)

- Architecture d'un MLP
- Propagation avant et rétropropagation
- Initialisation des poids
- Régularisation (L1, L2, Dropout)
- Choix des hyperparamètres

#### **4 – MLP pour la Régression**

- Régression non linéaire
- Fonctions de perte (MSE, MAE)
- Évaluation des performances
- Implémentation avec Keras

#### **5 – MLP pour la Classification d'Images**

- Représentation vectorielle des images
- Classification binaire et multi-classe
- Softmax et entropie croisée
- Limites des MLP pour la vision

#### **6 – Réseaux de Neurones Convolutifs (CNN) "Sur deux ou trois séances"**

- Convolution et filtres
- Cartes de caractéristiques
- Pooling
- Architectures CNN classiques
- Introduction au transfert d'apprentissage

#### **7 – Modèles de Détection d'Objets**

- Classification, localisation et détection
- Bounding boxes
- Principes des modèles R-CNN et YOLO
- Applications en vision par ordinateur

#### **8 – Autoencodeurs et Réduction de Dimension**

- Autoencodeurs simples
- Représentation latente
- Réduction de dimension non linéaire
- Lien avec la PCA
- Applications : compression, débruitage

## 9 – Explicabilité et Interprétabilité (XAI)

- Pourquoi expliquer un modèle ?
- Interprétabilité locale et globale
- Cartes de saillance
- Introduction à Grad-CAM
- Limites et enjeux éthiques

## 10 – Du Keras vers PyTorch

- Différences de paradigme
- Construction d'un modèle PyTorch
- Boucle d'entraînement
- Comparaison Keras / PyTorch

—

## Évaluation

- Test de compréhension du cours à chaque séance (5-10 min)
- Examen écrit en fin de cours
- Rendre un projet de groupe final (rapport de 10 pages maxi "en LaTeX" + Notebooks): présentations orales des projets.

## Bibliographie et ressources

### Livres

- Introduction to Deep Learning: [Accéder au livre via Drive](#)
- I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016. <https://www.deeplearningbook.org>
- F. Chollet, *Deep Learning with Python*, Manning, 2021. <https://www.manning.com/books/deep-learning-with-python-second-edition>
- A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly, 2022.

## Articles et papiers clés

- LeCun et al., *Deep Learning*, Nature, 2015. <https://www.nature.com/articles/nature14539>
- Krizhevsky et al., *ImageNet Classification with Deep Convolutional Neural Networks*, 2012. <https://arxiv.org/abs/1207.0580>
- Selvaraju et al., *Grad-CAM: Visual Explanations from Deep Networks*, 2017. <https://arxiv.org/abs/1610.02391>

## Vidéos et tutoriels

- Apprendre Python, programmation complet débutant: <https://www.youtube.com/watch?v=oUJolR5bX6g>
- Stanford CS231n – Convolutional Neural Networks <https://cs231n.stanford.edu>
- DeepLearning.AI – Neural Networks and Deep Learning <https://www.coursera.org/learn/neural-networks-deep-learning>
- PyTorch Official Tutorials <https://pytorch.org/tutorials/>
- TensorFlow / Keras Tutorials <https://www.tensorflow.org/tutorials>