

# Travaux Pratiques : De la conception au déploiement de modèles de Deep Learning

Louis Fippo Fitime

Claude Tinku

Kerolle Sonfack

Département Génie Informatique, ENSPY  
Université de Yaoundé I

22 septembre 2025

## Résumé

Ce document de Travaux Pratiques est destiné aux étudiants en informatique souhaitant acquérir des compétences pratiques en ingénierie du Deep Learning. Il couvre l'ensemble du cycle de vie d'un modèle d'apprentissage profond, depuis la conception et l'entraînement jusqu'au déploiement et au suivi en production.

## 1 Partie 1 : Fondations du Deep Learning

### 1.1 Concepts théoriques

#### 1.1.1 Modèles linéaires et optimisation stochastique

Les modèles linéaires constituent la base de nombreux algorithmes d'apprentissage automatique. Ils visent à établir une relation entre des variables d'entrée et une sortie à l'aide d'une combinaison linéaire de paramètres. L'apprentissage consiste à ajuster ces paramètres en minimisant une fonction de perte.

La descente de gradient classique calcule le gradient de la fonction de perte sur l'ensemble du jeu de données avant chaque mise à jour des paramètres. Cette approche est stable mais coûteuse en temps et en mémoire lorsque les données sont volumineuses.

La descente de gradient stochastique (SGD), quant à elle, met à jour les paramètres en utilisant un échantillon ou un mini-lot de données. Elle est plus rapide, plus scalable et mieux adaptée aux grands jeux de données, ce qui explique son utilisation privilégiée en Deep Learning.

#### 1.1.2 Réseaux de neurones artificiels

Un réseau de neurones artificiels est composé de plusieurs couches :

- **La couche d'entrée** reçoit les données brutes.
- **Les couches cachées** apprennent des représentations intermédiaires à l'aide de transformations non linéaires.

- **La couche de sortie** produit la prédiction finale du modèle.

L'apprentissage repose sur la rétropropagation du gradient (*backpropagation*). Après le calcul de la sortie, l'erreur est mesurée à l'aide d'une fonction de perte puis propagée en sens inverse afin de mettre à jour les poids du réseau.

## 1.2 Exercice 1 : Réseau de neurones avec Keras

### Question 1 : Couches Dense, Dropout et Softmax

Les couches **Dense** permettent d'apprendre des combinaisons linéaires des caractéristiques d'entrée, suivies d'une fonction d'activation non linéaire. La couche **Dropout** est une technique de régularisation qui désactive aléatoirement certains neurones durant l'entraînement afin de limiter le surapprentissage.

La fonction d'activation **Softmax** est utilisée en sortie car il s'agit d'un problème de classification multiclasse. Elle transforme les sorties du réseau en probabilités dont la somme est égale à 1.

### Question 2 : Optimiseur Adam

Adam (Adaptive Moment Estimation) est un optimiseur qui améliore la SGD en adaptant automatiquement le taux d'apprentissage pour chaque paramètre. Il combine les avantages du momentum et des méthodes adaptatives, ce qui permet une convergence plus rapide et plus stable, notamment pour les réseaux profonds.

### Question 3 : Vectorisation et calculs par lots

La vectorisation est appliquée via l'utilisation de tableaux NumPy, permettant de traiter plusieurs données simultanément sans boucles explicites. Le calcul par lots est implémenté grâce au paramètre `batch_size`, ce qui permet d'entraîner le modèle sur des sous-ensembles de données, améliorant l'efficacité et la stabilité de l'apprentissage.

## 2 Partie 2 : Ingénierie du Deep Learning

### 2.1 CI/CD et déploiement

#### Question 1 : Pipeline CI/CD

Un pipeline CI/CD permet d'automatiser la construction, les tests et le déploiement de l'application. À chaque modification du code, des outils comme GitHub Actions peuvent automatiquement construire l'image Docker, exécuter des tests et déployer le service sur une plateforme cloud telle que Google Cloud Run ou Amazon ECS.

#### Question 2 : Monitoring en production

Après le déploiement du modèle, plusieurs indicateurs doivent être surveillés :

- Les performances du modèle (précision, dérive des données).
- Les métriques système (latence, utilisation CPU/mémoire).
- Les indicateurs métier (volume de requêtes, taux d'erreur).

Ces indicateurs permettent d'assurer la fiabilité et la qualité du service en production.

### **3 Conclusion**

Ce TP a permis de parcourir l'ensemble du cycle de vie d'un modèle de Deep Learning, depuis sa conception jusqu'à son déploiement. Il met en évidence l'importance des bonnes pratiques d'ingénierie logicielle pour assurer des modèles robustes, reproductibles et exploitables en production.