

# DEEP LEARNING EN PYTHON : COMMENT CHOISIR UN FRAMEWORK ?

BEL-LETOILE, Justine, [jbel-letaille@hellowork.com](mailto:jbel-letaille@hellowork.com), HelloWork, Pessac (Oratrice)

Thématique : Intelligence artificielle

Résumé : TensorFlow, PyTorch, Fastai... Quand on fait du Python, comment s'y retrouver parmi les nombreux outils pour le deep learning ? L'objectif de cette présentation n'est pas d'élire le meilleur framework, mais plutôt de partager les critères importants à prendre en compte pour sélectionner l'outil le plus adapté à votre projet. De quoi est fait l'écosystème DL aujourd'hui ? Quelles sont les différences notoires entre ces technos ? Et si on veut travailler sur GPU, faire du NLP ou déployer un modèle ?

Mots clés : python, framework, librairie, pytorch, tensorflow, jax

## 1. Introduction

Le choix de la technologie pour réaliser un projet de Deep Learning peut être difficile, voire cornélien. De multiples frameworks existent et on trouve sur internet de nombreux exemples qui utilisent une librairie python ou une autre. Quand on démarre un projet, comment s'y retrouver dans cette jungle d'informations ? L'objectif de cette présentation est de partager un ensemble d'éléments à considérer au moment de sélectionner un outil. En s'appuyant sur l'expérience engrangée au cours de plusieurs projets R&D, on verra quelles sont les questions pertinentes à se poser au moment de choisir.

## 2. Méthodologie

On commencera par un tour d'horizon rapide de différents frameworks de Deep Learning, en rappelant les concepts mathématiques sur lesquels ils reposent. En effet, qu'il s'agisse de TensorFlow, PyTorch, mais également MXNet ou JAX, leur point commun est de nous permettre de faire de la dérivation automatique et de mettre en application l'algorithme d'optimisation classique en apprentissage via réseau de neurones : passe forward / passe backward / calcul des gradients et mise à jour du modèle via un optimizer. On fera un zoom sur un exemple concret, en implémentant en TensorFlow et en PyTorch un réseau de neurones basique. L'idée est de mettre en avant les différences de syntaxe et de paradigme.

On distinguera aussi frameworks et librairies. Qu'apportent-elles de plus ? Surcouches sur un framework existant (Sonnet, Fastai, Flax,...) ou spécialisées sur un domaine (NLP avec transformers, Flair ou Fairseq, sequence models avec Trax, vision avec torchvision, etc.), bien souvent elles proposent des classes ou des recettes prêtes à l'emploi, pour mettre rapidement en place un script d'apprentissage, d'inférence, ou reproduire le code d'un papier.

Enfin, la dernière partie de la présentation sera consacrée au parcours de plusieurs critères qui vont guider notre choix. On s'appuiera sur les interrogations que l'on a traversées au sein de l'équipe Data de HelloWork lors du développement de modèles de Deep Learning (notamment pour du NLP), afin d'illustrer comment telle ou telle solution peut répondre à ces questions :

- Environnement technique : travaillez-vous sur CPU, GPU, ou les deux de façon interchangeable ? Dans un environnement de production, y a-t-il des contraintes de taille de modèle, déploiement sur un serveur, rapidité de l'entraînement ou performance de l'inférence ?
- Objectif du projet : S'agit-il d'une première expérience en Deep Learning et vous souhaitez comprendre les rouages ? Souhaitez-vous pouvoir profiter de modèles pré-entraînés ou bien comptez-vous repartir de zéro ? Est-ce plutôt dans la vision, le NLP ?
- Velléité d'être en charge des détails d'implémentation : êtes-vous dans le monde de la recherche, avec l'envie de créer des architectures très spécialisées et coder des modèles complètement custom, ou bien voulez-vous utiliser facilement un réseau connu ? Avez-vous besoin de monitorer finement l'entraînement (debug, étude des hyperparamètres, etc.) ?

### 3. Originalité / perspective

Cette présentation ne sera pas une simple confrontation TensorFlow vs. PyTorch, et surtout il ne s'agira pas de prouver qu'un certain framework de Deep Learning est meilleur que les autres. Au contraire, on souhaite montrer en quoi la solution la plus adaptée dépend du cas d'usage. L'idée est que grâce à ces pistes de réflexion, à l'issue de la présentation, chacun soit capable de résoudre son propre dilemme.

Un des aspects qui n'a pas été exploré dans cette vue d'ensemble est la question de la performance (en termes de temps d'entraînement, d'inférence et consommation mémoire). Pour comparer rigoureusement les différents frameworks, il s'agirait de mettre en place un protocole de comparaison en choisissant soigneusement les settings techniques (différentes machines CPU / GPU, plusieurs datasets et tâches d'entraînement, mesures comparées, etc.).

### Références

Beaucoup de ressources disponibles sur le web et parcourues au fur et à mesure des explorations, dont notamment :

- [A Brief History of Deep Learning Frameworks](#), Towards Data Science (2020)
- [PyTorch vs TensorFlow in 2022](#), AssemblyAI (2021)
- [TensorFlow, PyTorch, and JAX: Choosing a deep learning framework](#), InfoWorld (2022)
- [Why You Should \(or Shouldn't\) be Using Google's JAX in 2022](#), AssemblyAI (2022)
- [Using JAX to accelerate our research](#), DeepMind (2020)
- [Building the Future of TensorFlow](#), TensorFlow (2022)

Papiers présentant les frameworks :

- Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C. & Zhang, Z. (2015). MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems (cite arxiv:1512.01274Comment: In Neural Information Processing Systems, Workshop on Machine Learning Systems, 2016)
- Frostig, R.; Johnson, M.J.; Leary, C. Compiling machine learning programs via high-level tracing. In Proceedings of the Systems for Machine Learning, Montreal, QC, Canada, 4 December 2018.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library.. CoRR, abs/1912.01703.
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P. A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. & Zhang, X. (2016). TensorFlow: A system for large-scale machine learning.. CoRR, abs/1605.08695.