Chapitre 3
Programmation d'un perceptron multicouche

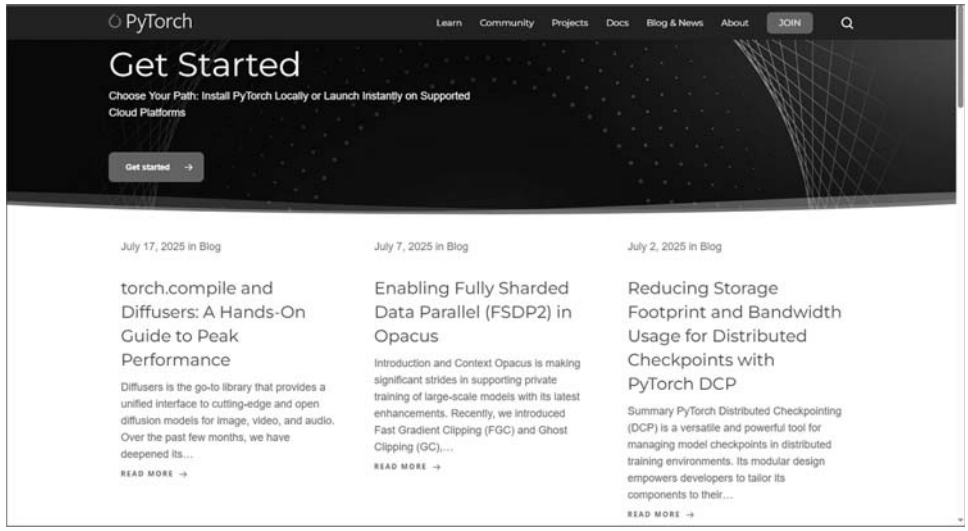
1. Frameworks pour le deep learning

1.1 Historique et panorama

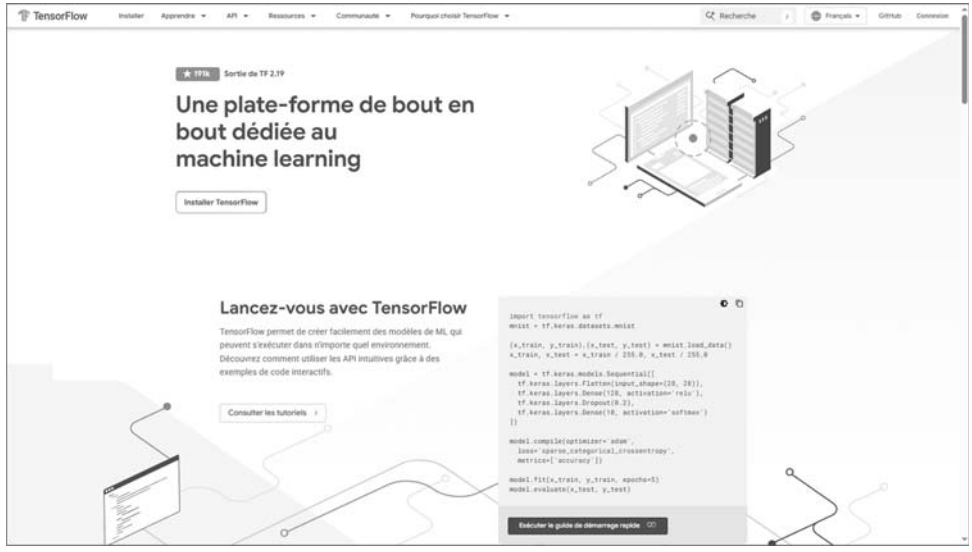
Avec l'essor du deep learning dans les années 2000, plusieurs outils spécialisés ont été développés par le monde académique. Parmi les plus connus on peut citer Theano (2007), développé au MILA (*Montreal Institute for Learning Algorithms*), PyBrain (2008) de l'université de Munich, ou encore Caffe (2013) créé par l'UC Berkeley en Californie.

Par la suite, plusieurs entreprises de la tech ont proposé leur propre logiciel. En 2015, Microsoft a introduit CNTK (*CogNitive ToolKit*). Amazon a soutenu activement MXNet, développé par des chercheurs des universités de Washington et de Carnegie Mellon.

Cependant, toutes ces solutions n'ont pas survécu à la concurrence des deux frameworks majeurs actuels : TensorFlow, dévoilé en novembre 2015 par Google, et PyTorch, présenté en 2016 par Facebook (devenu maintenant Meta). Ces ateliers logiciels ont en commun de fonctionner avec le langage Python et d'être disponibles dans les environnements Windows, macOS et Linux. Ils sont disponibles gratuitement (open source), avec des mises à jour régulières qui prennent en compte les dernières découvertes sur l'apprentissage profond.



Page d'accueil de PyTorch (https://pytorch.org)



Page d'accueil de TensorFlow (https://www.tensorflow.org)

Ce tour d'horizon ne serait pas complet sans parler des solutions développées en Chine. Le produit le plus connu est PaddlePaddle présenté en septembre 2016 par la société Baidu (surnommée le « Google chinois »). Cette plateforme, concurrente des produits précédents, est largement utilisée en Chine, mais son usage se développe également à l'international.

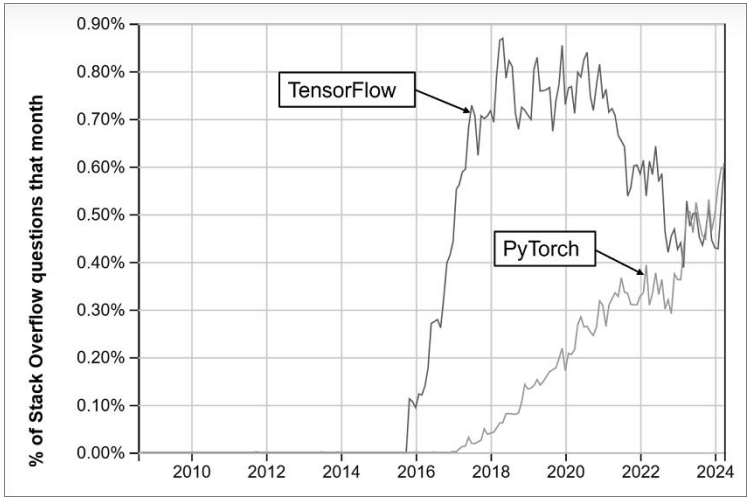
Page d'accueil chinoise de PaddlePaddle (https://www.paddlepaddle.org.cn)

Une autre bibliothèque très populaire pour le machine learning est Scikit-Learn (https://scikit-learn.org/). Ce logiciel est le résultat d'un projet collaboratif, initié en 2007, dans lequel l'INRIA (institut français de recherche) est très actif. Il est surtout spécialisé sur les algorithmes d'apprentissage automatiques « classiques » comme les régressions, les arbres de décision, les SVM (*Support Vector Machine*), le k-means, etc. Sa vocation n'est pas de concurrencer TensorFlow ou PyTorch dans le domaine de l'apprentissage profond (deep learning) par réseaux de neurones, bien qu'il puisse être utilisé en complément pour certaines tâches (nous verrons un exemple dans le chapitre suivant).

1.2 Quel framework choisir ?

Sur Internet de nombreux sites comparent régulièrement TensorFlow et PyTorch : il suffit de saisir « TensorFlow vs PyTorch » sur le moteur de recherche de Google pour obtenir des milliers de réponses. Ces dernières années, ces deux outils ont beaucoup évolué et se sont rapprochés en termes de fonctionnalités. Il est encore souvent considéré que PyTorch a la préférence du monde académique, tandis que TensorFlow reste plus largement adopté dans l'industrie, même si les choses peuvent évoluer très vite.

Stackoverflow.com est une plateforme communautaire très populaire où les développeurs peuvent poser des questions sur des problèmes de programmation et partager leurs connaissances. Chaque année, ce site réalise une enquête (https://survey.stackoverflow.co/) pour recueillir des informations sur les pratiques des programmeurs : technologies et outils utilisés, préférences, salaires, etc. Il y a encore 3 ans, TensorFlow était largement préféré à PyTorch. Dans l'étude de 2024, on constate une égalité des scores, tant chez les professionnels que chez les programmeurs débutants. Cette tendance est confirmée par les statistiques sur les mots-clés utilisés, au fil du temps, dans les requêtes sur Stack Overflow :



Évolution des requêtes entre TensoFlow et PyTorch

TensorFlow est un choix pertinent pour de nombreux projets, en raison de son écosystème riche et intégré. Son interconnexion avec des outils Google tels que Drive, Colab et Datasets facilite le développement, tandis que des ressources comme TFX et Cloud TPUs sont utiles pour le déploiement en production et l'accélération du calcul.

 études de cas avec l'utilisation de TensorFlow dans des entreprises comme Airbus, Coca-Cola ou encore PayPal :
https://www.tensorflow.org/about

Remarque

TensorFlow et PyTorch fonctionnent sur des principes similaires ; si besoin, il est donc assez facile de passer de l'un à l'autre.

2. Environnement de développement

2.1 Solution locale ou sur le cloud

Même si TensorFlow/Keras fait partie d'un écosystème, c'est avant tout une bibliothèque de modules écrits en Python. Il faut donc disposer d'un environnement de développement, pour écrire et mettre au point des programmes. Deux approches sont possibles : soit travailler en local sur son ordinateur, soit utiliser les ressources du cloud.

Dans le premier cas, de nombreux ateliers logiciels (IDE - *Integrated Development Environment*) sont disponibles, parmi lesquels :

- PyCharm (JetBrains) : https://www.jetbrains.com/fr-fr/pycharm/
- Visual Studio Code (Microsoft) : https://visualstudio.microsoft.com/fr/
- Spyder (inclus dans la distribution Anaconda) :
  https://www.spyder-ide.org/ et https://www.anaconda.com/

Tous ces outils sont disponibles sous Windows, macOS ou Linux. Certaines versions sont payantes, mais il existe également des versions gratuites pour les étudiants/enseignants ou pour évaluation avec des fonctionnalités réduites.

Plusieurs plateformes proposent des services pour travailler sur le cloud, comme :

– SageMaker AI (Amazon) : https://aws.amazon.com/fr/sagemaker-ai/
– Azure Machine Learning (Microsoft) : https://azure.microsoft.com/fr-fr/products/machine-learning/
– Vertex AI (Google) : https://cloud.google.com/vertex-ai

Ces solutions payantes (avec parfois des offres d'essai gratuites) sont destinées à gérer des modèles en production à grande échelle.

Dans son écosystème (cf. chapitre Découverte des réseaux de neurones - Ecosystème associé à TensorFlow), Google propose une plateforme en ligne appelée Colab (abréviation de Google Colaboratory). Elle est destinée aux petits projets et à la formation, pour expérimenter facilement des modèles de deep learning.

Nous avons choisi cet environnement (qui existe en version gratuite) pour développer les exemples traités dans cet ouvrage. Ce choix se justifie car il évite toutes les difficultés liées à la configuration de sa machine : version de Python, liste et versions des modules installés, taille mémoire, type de GPU, etc. Avec Colab, tous les utilisateurs ont accès à la même installation, qui est administrée et mise à jour par Google. De plus, les programmes sont des fichiers au format des notebooks Jupyter, particulièrement appréciés pour leur interactivité (exécution séparée des cellules de code, widgets…).

Remarque

Les exemples développés dans la suite de cet ouvrage fonctionnent également avec un environnement de développement local, sous réserve de la configuration installée.

2.2 Environnement Colab

2.2.1 Création d'un compte Google

Colab étant intégré à l'écosystème de Google, son utilisation impose de disposer d'un compte chez ce fournisseur. Si vous n'en possédez pas, il est nécessaire d'en créer un. Une manière de procéder consiste à ouvrir la liste des applications (1) dans la page d'accueil de Google (www.google.com), puis à sélectionner **Compte** (2). Une nouvelle page s'affiche alors, sur laquelle il faut cliquer sur le bouton **Créer un Compte** (3) :

Début de la procédure pour créer un compte Google

Vous aurez ensuite à renseigner différentes informations (nom, date de naissance, etc.) et surtout à choisir une adresse mail avec un mot de passe. Ils vous serviront de sésame pour bénéficier de tous les services de l'univers Google.

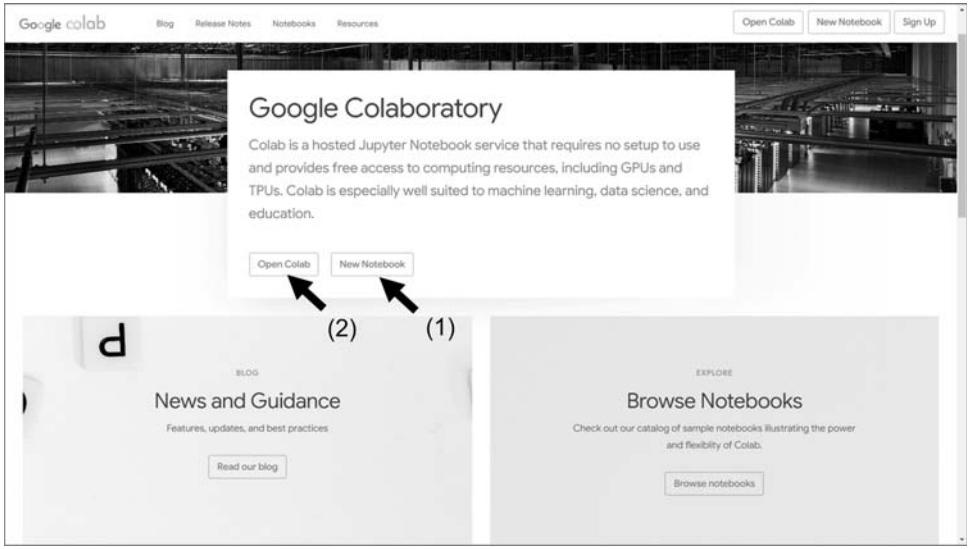
Remarque

Un compte Google n'est pas nécessaire pour utiliser TensorFlow/Keras dans un environnement de développement local.

Colab est compatible avec les principaux navigateurs tels que Firefox ou Microsoft Edge. Il est cependant conseillé d'utiliser Google Chrome, afin d'assurer une meilleure fluidité avec Google Drive, service du cloud où seront stockés les fichiers utilisés (notebooks, modèles, jeux de données). Si ce n'est pas déjà fait, nous vous suggérons donc d'installer ce navigateur sur votre ordinateur.

2.2.2 Lancement de l'application

Après avoir lancé Chrome, connectez-vous à l'adresse https://colab.google (sans l'extension « .com ») pour accéder à la page d'accueil de Colab :

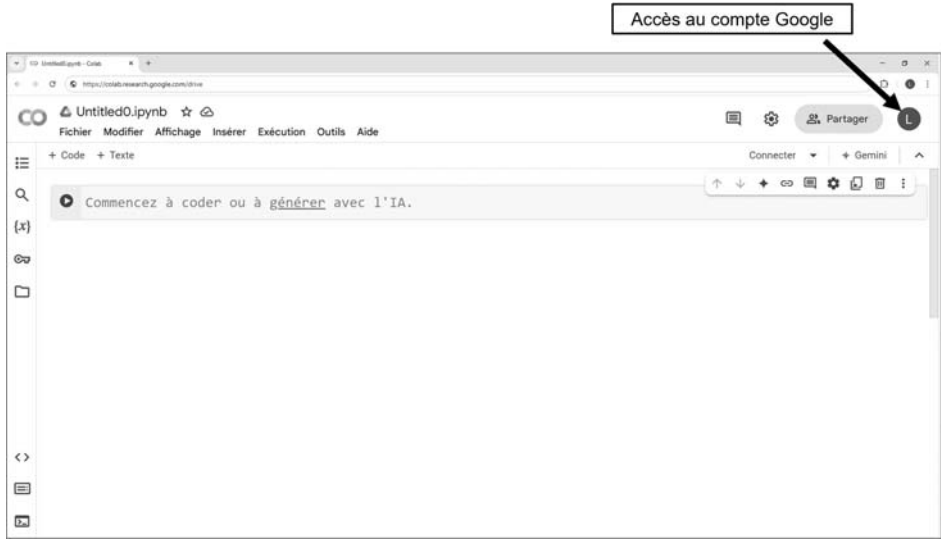


Page d'accueil de Colab (https://colab.google)

De nombreuses ressources sont disponibles, telles que des tutoriels, des exemples de code et de la documentation. Vous pouvez les consulter, mais tout n'est pas forcément accessible lorsqu'on débute.

Deux solutions sont possibles pour entrer dans l'environnement de travail : **New Notebook** (1) qui lance la création d'un nouveau programme, ou **Open Colab** (2) qui ouvre une fenêtre pour sélectionner le notebook à charger.

La sélection de **New Notebook** entraîne l'affichage de la page suivante :



Fenêtre principale de l'application Colab

Les menus apparaissent dans la langue du compte Google utilisé pour lancer Colab, dans cet exemple le français.

Remarque

Au lieu de passer par la page d'accueil, l'environnement Colab est aussi accessible via l'adresse https://colab.research.google.com/. Nous verrons dans la suite qu'il est également possible d'ouvrir un notebook directement depuis Drive.