IFT 603-712 : Devoir 4

Travail par équipe de 3

Pour ce devoir, **aucun code n'est fourni** : c'est à vous de proposer une solution complète au problème énoncé. Vous devez **entraîner, tester et comparer différents réseaux de neurones** dédiés à la classification d'images. Pour cela, vous utiliserez les bases de données **CIFAR-10** et **CIFAR-100**, toutes deux facilement accessibles via la bibliothèque *torchvision*. Pour ce faire, vous devez implémenter les trois modèles suivants :

1. **Un réseau de neurones pleinement connecté linéaire** (semblable à celui du TP3).
2. **Un réseau de neurones multicouches (MLP)**, toujours de type pleinement connecté, mais plus profond que celui du TP3 (à vous de déterminer le nombre de couches et de neurones).
3. **Un réseau à convolution (CNN)** multicouches, que vous concevrez vous-même.

Pour le réseau à convolution, soyez **originaux et créatifs**! Par exemple, définissez un nombre X de couches, chacune contenant Y opérations de convolution et Z filtres. C'est à vous de choisir la structure globale. Vous pouvez également :

- utiliser différentes fonctions d'activation ;
- combiner des filtres *same* et *valid* ;
- tester différents *strides* ;
- explorer plusieurs méthodes de *pooling* ;
- adopter une approche entièrement convolutive ou inclure des couches linéaires ;
- expérimenter plusieurs fonctions de perte (entropie croisée, erreur quadratique moyenne, etc.) voire des fonctions plus avancées comme la *focal loss*.

Exigences pour le code

Votre code doit respecter les contraintes suivantes :

- Le code doit **obligatoirement** utiliser la bibliothèque **PyTorch**.
- Les implémentations de vos réseaux doivent se trouver dans des fichiers séparés, par exemple *MLP.py* et *CNN.py* (ou tout autre nom approprié). Tous vos modèles doivent hériter de la classe *nn.Module.*
- Les fonctions liées à la gestion des données (téléchargement, *DataLoader*, transformations, etc.) doivent être regroupées dans une classe, par exemple *DataManager*.
- Les fonctions d'entraînement et de test doivent être regroupées dans une classe, par exemple *TrainTestManager*.
- L'essentiel de vos expériences doit être présent dans un **notebook Jupyter**, dont les cellules doivent pouvoir être exécutées facilement par le correcteur (évitez donc les chemins de fichiers *hardcodés*). Le notebook doit :
    - charger les données,
    - entraîner vos réseaux,
    - afficher les résultats et les courbes d'évolution.
- Les performances minimales attendues sont :
    - **75 % d'accuracy sur CIFAR-10** avec votre CNN,
    - **30 % d'accuracy sur CIFAR-100** avec votre CNN.
- Si vous ne disposez pas d'un ordinateur avec GPU compatible, il est fortement recommandé d'utiliser les ressources gratuites de **Google Colab** (https://colab.google/).

Rapport technique à remettre

En plus du code, vous devez rédiger un **court rapport technique** contenant :

- **Une courte introduction**.
- **La description et une illustration graphique** (schémas) des trois réseaux de neurones.
- **Les détails d'implémentation et d'entraînement**, incluant :
    - hyperparamètres,
    - fonctions de perte,
    - optimisateurs,
    - choix architecturaux.
- **Les temps d'entraînement et de test**, incluant une comparaison du temps nécessaire pour un *epoch* sur **GPU** versus **CPU**, pour chacun des réseaux.
- **Les courbes d'entraînement et de validation** (*loss* et *accuracy*) pour les deux bases de données, permettant d'observer l'évolution et la convergence des modèles.
- **Une section *Discussion***, dans laquelle vous expliquerez :
    - ce qui a bien fonctionné,
    - ce qui a moins bien fonctionné,
    - ce que vous recommanderez pour améliorer les performances.
- **Une bonne recherche d'hyperparamètres**, même sommaire, accompagnée d'un résumé des expériences effectuées.
- **Une section sur l'IA générative**, dans laquelle vous indiquerez clairement :
    - ce que vous avez fait vous-même,
    - ce que l'IA vous a aidé à produire.

Ressources recommandées

Avant de commencer, il est fortement recommandé de suivre le tutoriel Google Colab suivant (≈23 minutes)

https://www.youtube.com/watch?v=PytsxNfR8GI

Vous pouvez aussi consulter l'excellent tutoriel PyTorch officiel (≈ 60 minutes) :

https://docs.pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

Si vous préférez apprendre via des vidéos structurées, voici une série complète sur PyTorch :

https://www.youtube.com/playlist?list=PL_lsbAsL_o2CTlGHgMxNrKhzP97BaG9ZN