

IFT 780 : Devoir 2

Réseaux de neurones convolutifs

Dans ce devoir, vous devez implanter des architectures de **réseaux de neurones convolutifs**, le **dropout** et vous familiariser avec la **batch norm**.

Comme pour le devoir précédent, vous devez vous créer un environnement virtuel python suivant les recommandations sur la page web du cours. Vous devez ensuite exécuter les lignes de commande suivantes dans un terminal :

```
$ cd datasets/  
$ ./get_datasets.sh  
$ cd ../utils/cython/  
$ ./build_cython.sh
```

afin de télécharger la base de données CIFAR10 et de compiler le code cython. Vous devez également récupérer les fichiers .py que vous avez modifié dans votre tp1 dont :

```
layers/Dense.py  
model/LinearClassifier.py  
utils/model_loss.py  
model/Solver.py
```

Si d'aventure votre code du tp1 est incomplet, prenez contact avec le professeur qui vous donnera les fonctions manquantes.

Par la suite, les fichiers ipynb vous indiqueront les tâches à accomplir. Vous trouverez le code à rédiger dans les notebooks et dans les fichiers python (.py) aux endroits indiqués par des **TODO**. Vous trouverez également des questions à répondre dans certains fichiers ipynb. Les réponses aux questions doivent être dans les fichiers ipynb et en format *markdown* (voir <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet> pour plus de détails sur le standard *markdown*).

Les objectifs de ce devoir sont

- 1) Comprendre les opérations *dropout* et *batchnorm*.
- 2) Comprendre les opérations de convolution et de *max pooling*.
- 3) Comprendre le fonctionnement d'un réseau à convolution ainsi que les interactions entre ses couches.

À faire :

1. **[1 point]** Implanter et tester *dropout* (voir notebook tp2_dropout.ipynb).
2. **[0 points]** Comprendre la *batch-norm* (voir notebook tp2_batch_norm.ipynb).
3. **[3 points]** Comprendre l'opération de convolution (voir notebook tp2_convolution.ipynb).
4. **[2 points]** Implanter et tester un réseau convolutif 100% `numpy` (voir notebook tp2_cnn.ipynb).
5. **[1 point]** Réimplanter votre réseau convolutif, mais cette fois-ci avec **Pytorch** (voir notebook tp2_cnn_pytorch.ipynb).
6. **[3 points]** Implanter les blocs (dense, résiduel, goulot) du YourNet et obtenez de bonnes performances sur CIFAR-10 (voir notebook tp2_yournet.ipynb).

NOTE : le code peut s'exécuter sur la plupart des ordinateurs portables, même sans carte graphique. Si vous souhaitez ainsi tester votre code, vous devez installer pyTorch dans votre environnement virtuel suivant une commande comme celle-ci sur Ubuntu:

```
$ pip3 install torch torchvision torchaudio --extra-index-url  
https://download.pytorch.org/whl/cpu
```

Veuillez consulter le site suivant afin d'identifier la commande pip qui conviendra à votre ordinateur :

<https://pytorch.org/get-started/locally/>.

Si toutefois vous souhaitez utiliser un GPU et que votre ordinateur n'en a pas un, vous pouvez utiliser celui de **Google Colab** :

<https://colab.research.google.com/>

Un tutoriel simple à suivre est disponible sur la dite page de Colab. En résumé, vous devez

1. vous créer un compte gmail (si ce n'est pas déjà fait)
2. aller sur votre Google drive (<https://drive.google.com/>)
3. créer un répertoire pour le travail pratique (exemple "IFT780/TP2" dans le répertoire MyDrive/)
4. copier le contenu du code du TP2 dans le répertoire que vous venez de créer.
5. cliquez sur le fichier ipynb de votre choix.

Et bien entendu, n'oubliez pas de sélectionner un GPU (dans le menu : **Runtime** → **Change runtime type** → **GPU**).

NOTE IMPORTANT : veuillez prendre note que si votre ordinateur n'est pas équipé d'un bon GPU, le code « pytorch » fonctionnera très lentement sur le CPU de votre ordinateur. Pour tester si votre code utilise bel et bien un GPU, l'exécution des lignes suivantes devrait produire le « print » du mot « GPU »

```
import os  
import torch  
device = 'GPU' if torch.cuda.is_available() else 'CPU'  
print('Vous utilisez le {}'.format(device))
```