GUIDE TO PRACTICE WITH NEURAL NETWORKS FOR MASTER STUDENTS

Attention, document en franglais !

Emmanuel Viennet, Jan 2022

UNIVERSITÉ PARIS 13

1- Visionner attentivement le cours

3 vidéos, vous pouvez écouter la version française ou anglaise.

1. What is a Neural Network: introduction
2. Applications du Deep Learning
3. Backpropagation

(fichier vidéos sur Teams)

1- Prise en main de PyTorch

1. Si vous travaillez sur votre propre ordinateur, install PyTorch on your machine (*non necessaire si vous restez sur Collab*)
   https://pytorch.org/get-started/locally/
2. Follow the tutorial
   https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
   Watch the short video and study carefully
   https://pytorch.org/tutorials/beginner/blitz/tensor_tutorial.html#sphx-glr-beginner-blitz-tensor-tutorial-py

2- Notebook on gradient descent

- Open 01-example-linear-1d.ipynb
  https://drive.google.com/file/d/1Ks3ua_uZNl5t3iI-ycSoLDv74Wls8TPJ/view?usp=sharing
  This notebook shows you how to learn a very simple model by gradient descent. So simple *that no framework is needed*. Take time to understand all the code and answer the questions.

3- MLP Neural Networks for 2d classification

- Open 02-nn-2d.ipynb
  https://drive.google.com/file/d/1-sFvZGRuiAAh_wIpmC7sX4OL7CaVsO74/view?usp=sharing
  This notebook shows you how to use PyTorch to build simple neural networks working on 2-dimensional data.

Take time to understand all the code, play with it, answer the questions.

4- Fully connected MLP for Digits recognition

- Open 03-classif-MNIST-fullconnect.ipynb

This notebook shows you how to use Pytorch to build a fully connected MLP and apply it to MNIST digits recognition.

https://drive.google.com/file/d/1zh9B5QBMo8zsKgtguZAh8HGuGQrTlyee/view?usp=sharing

The file **mnist-original.mat** contains 70000 images of handwritten digits. Copy this file on your computer and adjust the path in the notebook.

Take time to understand all the code, play with it, answer the questions.

5- Exercise (not a neural network)

- Consider the model (1-dimension, 2 parameters)

$$\hat{y} = x^2.w_2 + x.w_1 + b$$

$$\text{loss} = (\hat{y} - y)^2$$

1. Compute (by hand): $\frac{\partial loss}{w_1} = ?$ $\frac{\partial loss}{w_2} = ?$
2. Code this computation in pure Python and optimize the model (like in the first notebook)
3. Code the same computation with **PyTorch**.

À la fin de la journée (mercredi 11/1)

- Envoyer un (un seul) mail à emmanuel.viennet+m3ir@gmail.com

avec les liens vers VOS trois notebooks, qui devront être completes et commentés le plus clairement possible.

Pour aller plus loin:

- Le cours de Yann LeCun à New York University
  - https://cds.nyu.edu/deep-learning/
  - Version française: https://atcold.github.io/NYU-DLSP21/fr/
- Tutoriel PyTorch: https://www.youtube.com/watch?v=XriwHXfLi6M