
Utilisation de PyTorch avec un serveur GPU

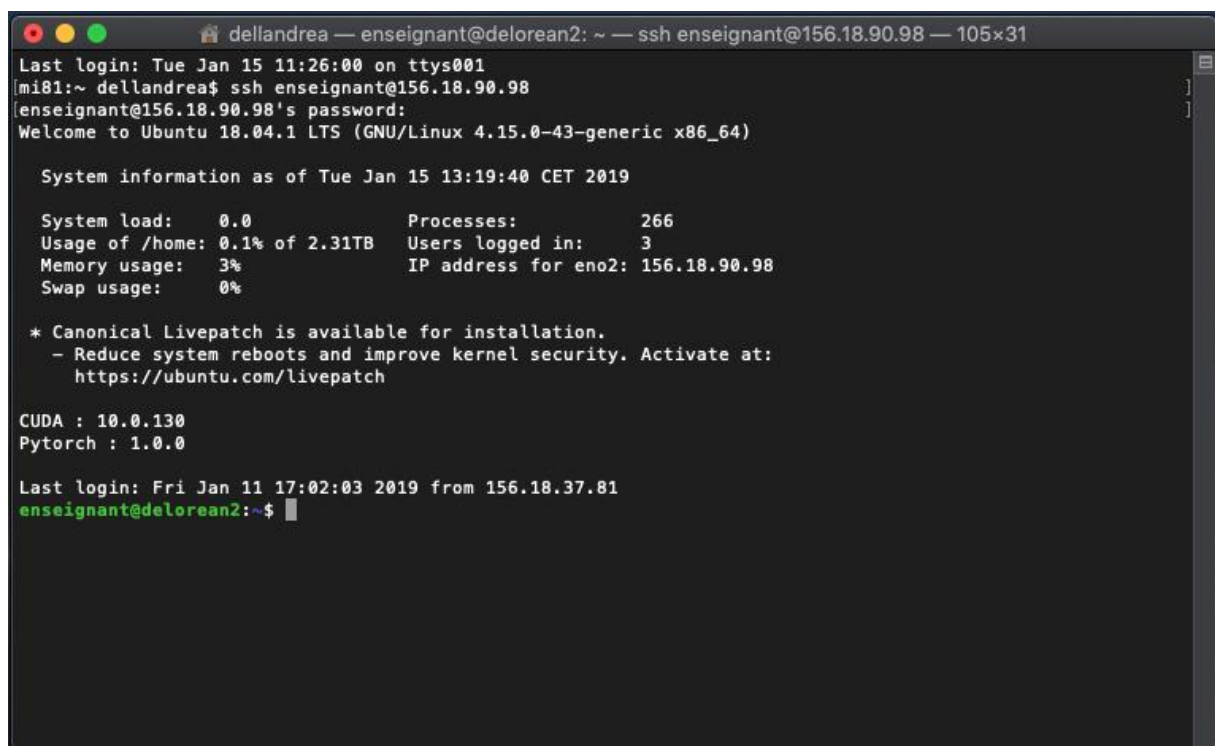
Deux serveurs GPU hébergés à la DSI sont disponibles pour réaliser des expérimentations. Chacun dispose de deux cartes graphiques. Votre chargé de TD vous indiquera l'adresse IP du serveur sur lequel vous pourrez vous connecter, et les identifiants/mots de passe à utiliser.

L'exécution des commandes se fera à partir d'un terminal, après s'être connecté en utilisant le protocole SSH. Sous MacOS et Linux, le programme (ssh) est préinstallé et utilisable directement dans le terminal. Sous Windows, il faudra utiliser un client SSH tel que Putty¹, Bitvise² ou MobaXterm³ ().

La copie des fichiers pourra se faire en utilisant un logiciel de transfert tel que le client FileZilla⁴.

Une démarche possible pour réaliser et exécuter un programme sur le serveur GPU est donc:

1. Ouvrir un terminal et se connecter au serveur GPU par ssh :



```
dellandrea — enseignant@delorean2: ~ — ssh enseignant@156.18.90.98 — 105x31
Last login: Tue Jan 15 11:26:00 on ttys001
mi81:~ dellandrea$ ssh enseignant@156.18.90.98
enseignant@156.18.90.98's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-43-generic x86_64)

 System information as of Tue Jan 15 13:19:40 CET 2019

System load:  0.0          Processes:      266
Usage of /home: 0.1% of 2.31TB   Users logged in:  3
Memory usage:  3%           IP address for eno2: 156.18.90.98
Swap usage:    0%

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

CUDA : 10.0.130
Pytorch : 1.0.0

Last login: Fri Jan 11 17:02:03 2019 from 156.18.37.81
enseignant@delorean2:~$
```

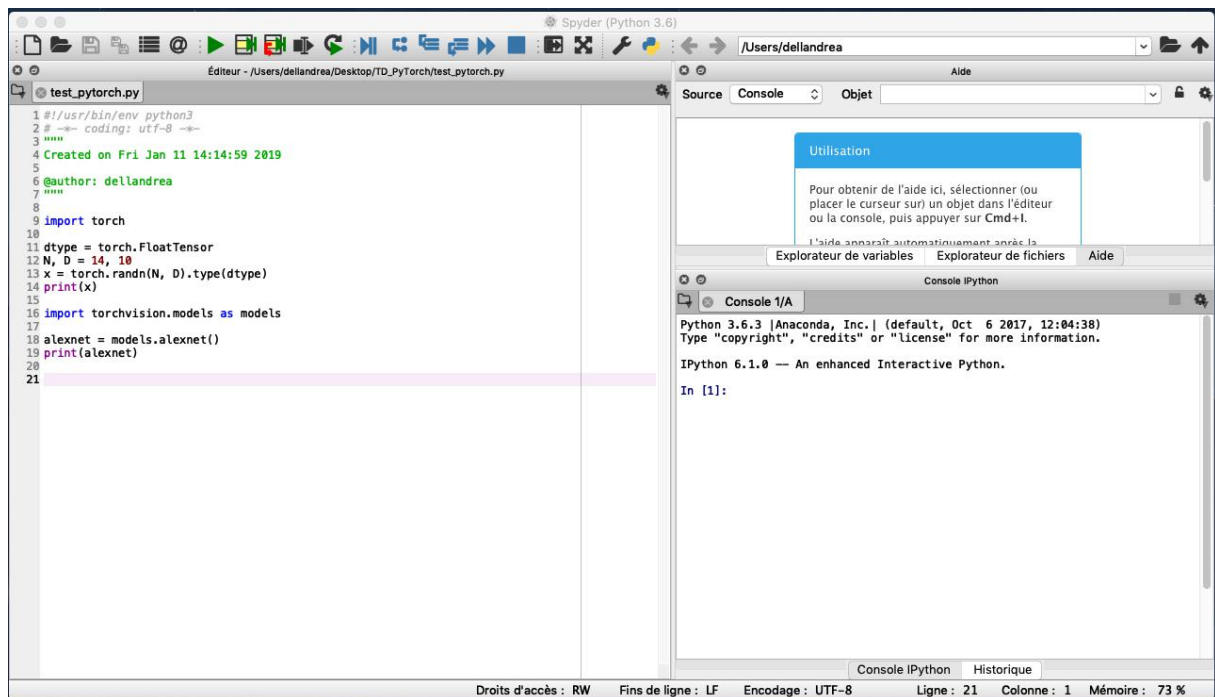
2. Ecrire le code python dans un fichier local, en utilisant par exemple Spyder :

¹ <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

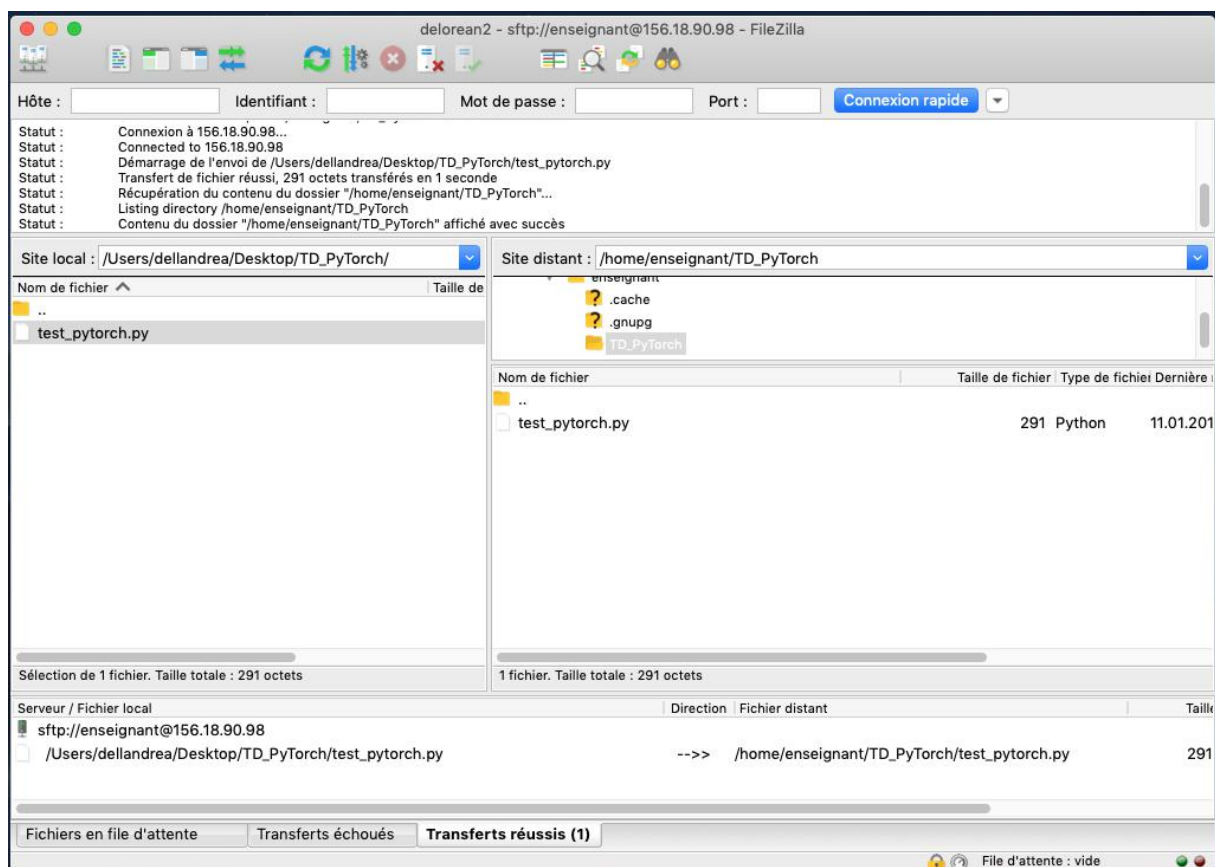
² <https://www.bitvise.com/download-area>

³ <https://mobaxterm.mobatek.net>

⁴ <https://filezilla-project.org/>



3. Transférer ce fichier sur le serveur avec le client FileZilla :



4. Exécuter le fichier python sur le serveur GPU :

```
dellandrea — enseignant@delorean2: ~/TD_PyTorch — ssh enseignant@156.18.90.98 — 121x78
Last login: Tue Jan 15 11:26:00 on ttys001
mi81:~ dellandrea$ ssh enseignant@156.18.90.98
enseignant@156.18.90.98's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-43-generic x86_64)

System information as of Tue Jan 15 13:19:40 CET 2019

System load:  0.0      Processes:    266
Usage of /home: 0.1% of 2.31TB   Users logged in: 3
Memory usage:  3%      IP address for eno2: 156.18.90.98
Swap usage:    0%

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

CUDA : 10.0.130
Pytorch : 1.0.0

Last login: Fri Jan 11 17:02:03 2019 from 156.18.37.81
enseignant@delorean2:~$ cd TD_PyTorch/
enseignant@delorean2:~/TD_PyTorch$ python test_pytorch.py
tensor([[ -1.8741e-01,  3.3130e-01, -4.4322e-01,  9.1984e-02,  6.1356e-01,
          1.5805e+00, -2.7530e-02,  1.5462e-01,  5.7853e-01,  2.5527e-01],
        [-2.0440e-01,  1.9694e-01,  1.7613e+00, -8.3053e-01, -1.2506e+00,
          4.7315e-01,  2.3847e-01, -5.9934e-01, -1.4521e+00, -1.4770e+00],
        [-1.2479e+00,  8.5246e-01, -2.1504e-01, -5.4391e-01,  1.0847e+00,
          8.2655e-01,  4.7420e-01,  1.2505e+00, -1.3467e+00,  8.5746e-01],
        [ 2.3580e-01,  1.1419e+00,  1.4725e+00, -9.2527e-01, -1.2751e+00,
        -2.1811e+00, -1.3010e-01, -3.7280e-01, -6.2903e-01, -1.0954e+00],
        [-2.7632e-01, -8.7158e-01, -1.4518e-01,  1.2119e+00,  5.1791e-01,
        -2.2531e-02, -7.9661e-01, -5.4442e-01, -1.4239e+00, -1.1506e+00],
        [ 6.8942e-01, -2.7273e-01,  4.4887e-01,  3.2061e-01,  8.9406e-01,
        -3.1503e-01, -6.1502e-01,  8.8183e-01,  6.6246e-01, -6.6656e-01],
        [-2.0457e+00, -1.1659e+00, -9.2297e-01,  5.9862e-01, -2.1354e-01,
        -3.8730e-01,  1.8303e-01, -9.7596e-01, -6.0013e-01, -8.0279e-01],
        [-4.0330e-01, -4.7294e-01, -1.1919e+00, -1.0336e+00, -1.6456e+00,
        -2.5234e-01,  3.8558e-01, -4.4679e-01,  1.3629e+00, -1.1733e+00],
        [ 5.4975e-01, -6.2449e-01,  1.1453e+00,  3.1061e-01,  1.2307e-02,
        1.3877e+00,  1.0378e+00, -5.5775e-01,  9.9396e-01, -8.0244e-01],
        [ 3.4683e-01, -8.6192e-01, -4.0258e-01,  1.5832e+00,  3.3182e-01,
        -1.4552e+00, -1.2283e+00,  1.0676e-01, -5.9510e-01,  1.3563e+00],
        [-1.6473e+00,  2.9967e+00,  1.3902e-01, -2.4636e-01, -2.1820e-01,
        -6.9258e-01,  9.6116e-01, -1.3109e+00,  1.9687e-01, -2.4093e-01],
        [-1.8461e+00, -7.1747e-01,  1.4644e-01, -8.7021e-01,  7.2109e-01,
        6.3802e-02, -1.4777e+00,  2.4418e-01, -5.9156e-01,  8.9379e-01],
        [-1.2569e+00, -1.8024e-03, -5.1996e-01, -5.7305e-01,  3.9382e-02,
        1.3762e+00,  4.7349e-01, -6.3693e-01,  1.4120e+00,  8.2492e-02],
        [ 1.3428e+00,  1.1223e+00,  9.0809e-01, -5.6057e-01, -1.6854e-01,
        -3.8964e-01,  9.7677e-01,  6.1724e-01, -1.3984e+00,  7.0738e-01]])

AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Dropout(p=0.5)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace)
    (3): Dropout(p=0.5)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
  )
)
enseignant@delorean2:~/TD_PyTorch$
```

Remarque : après chaque modification du fichier local dans Spyder, le fichier devra être à nouveau transféré sur le serveur avant d'être exécuté.

Les commandes Unix de base utiles lors de l'utilisation d'un terminal sont présentées ici : https://doc.ubuntu-fr.org/tutoriel/console_commandes_de_base