

# Instruction d'installation

Vous aurez préalablement besoin d'un accès à internet, d'une machine sous Ubuntu 18.04 et de la commande `wget`. Nous commenceront par installer le gestionnaire de paquets Anaconda. Pour l'installer, il vous suffit de copier coller les lignes suivantes :

```
1 $ wget https://repo.anaconda.com/archive/Anaconda3-2019.07-Linux-x86_64.sh
2 $ chmod +x Anaconda3-2019.07-Linux-x86_64.sh
3 $ ./Anaconda3-2019.07-Linux-x86_64.sh -b
```

Si l'installation c'est bien déroulé, vous pouvez configurer un environnement de travail avec conda. Il vous suffit de copier coller les lignes suivantes et de les exécuter :

```
1 #On désactive l'environnement de base
2 $ conda config --set auto_activate_base false
3
4 #On crée un nouvel environnement du nom d'euromillion
5 $ conda create --name euromillion python=3.7
6
7 #On active l'environnement créé
8 $ conda activate euromillion
9
10 #On installe les packages dont on aura besoin
11 $ conda install -y tensorflow-gpu
12 $ conda install -y keras-gpu
13
14 #On désactive l'environnement si on ne compte pas l'utiliser tout de suite
15 $ conda deactivate
```

Nous allons ici installer les bibliothèques nécessaires à la création d'un réseau de neurones. Pour cela il vous faudra une machine ayant python, une distribution Ubuntu et une connexion internet.

Nous allons utiliser les bibliothèques python, TensorFlow, Keras pour cela nous utiliserons anaconda qui nous permettra de les installer simplement.

On peut se rendre sur le site anaconda pour obtenir le fichier d'installation de anaconda, il suffit ensuite d'aller dans la section download, ensuite choisir la version à télécharger selon son système d'exploitation (Windows, Linux ou MacOS), nous choisirons Linux pour notre tutoriel.

Nous choisirons la version 3.7 car la version 2.7 de python ne sera plus maintenue en 2020 (source officielle).

Et nous prendrons la version 64-Bit (x86).

On peut aussi faire :

```
1 $ wget https://repo.anaconda.com/archive/Anaconda3-2019.07-Linux-x86_64.sh
```

Qui téléchargera directement le bon fichier dans le répertoire courant.

Une fois le fichier téléchargé il faut lui donner le droit d'exécution, en tapant la commande suivante dans votre terminal là où est situé votre fichier :

```
1 $ chmod +x Anaconda3-2019.07-Linux-x86_64.sh
```

Cela va permettre au fichier de s'exécuter et d'installer anaconda, on exécute donc le fichier avec -b pour gagner du temps:

```
1 $ ./Anaconda3-2019.07-Linux-x86_64.sh -b
```

Une fois anaconda installé, nous allons installer des packages pour anaconda pour pouvoir utiliser les réseaux de neurones. Comme nous allons utiliser des packages et que nous ne voulons pas forcément avoir toutes ces bibliothèques sur toutes nos applications, nous allons donc créer un environnement python avec anaconda pour pouvoir installer les packages que l'on veut sur un python isolé des autres. Et les packages à installer variant d'un projet à l'autre il peut être intéressant d'avoir des environnements bien séparés pour ne pas tout mélanger.

Pour cela nous allons utiliser anaconda qui nous permet de créer un environnement assez facilement. Nous allons créer notre environnement projeteuromillion: Nous allons indiquer la version de python que nous souhaitons utiliser nous prendrons la dernière disponible.

```
1 $ conda create --name projeteuromillion python=3.7
```

Maintenant nous allons nous utiliser cet environnement:

```
1 $ conda activate projeteuromillion
```

Maintenant les packages que nous téléchargeront seront installés seulement dans cet environnement python.

Nous allons maintenant installer le package TensorFlow pour gpu: l'option -y permet d'éviter d'avoir à répondre yes lorsque python nous le demande.

```
1 $ conda install -y tensorflow-gpu
```

Le package est installé, on installe aussi la surcouche Keras:

```
1 $ conda install -y keras-gpu
```

maintenant si on fait un list on peut voir les packages qui ont été installés :

```
1 $ conda list
```

Donc l'installation de la librairie tensorflow est faite. Pour quitter l'environnement dans lequel vous êtes faites:

```
1 $ conda deactivate
```

Vous serez maintenant de nouveau dans votre environnement de base.

```
1 $ conda config --set auto_activate_base false
```

Permet de désactiver le (base) sur le terminal.

```
(base) etudiant@d20j:~$
```

*l'environnement base*

Normalement rendue à ce stade là vous avez un tensorflow exécutable.