

## 1. Instalar Anaconda (Opcional pero Recomendado)

Anaconda permite crear ambientes separados para versiones específicas de python sin afectar la versión instalada por defecto en el sistema. seguir los siguientes pasos:

1. Descargar el software Anaconda en su versión para Python 3.6 desde el sitio web oficial:  
<https://www.anaconda.com/download/#download>
2. Ejecutar el instalador siguiendo los pasos del asistente de instalación.
3. Una vez instalado Anaconda, abrir un terminal
4. Con el terminal abierto, ejecutar el siguiente comando para crear un ambiente con el nombre: tensorflow

```
$ conda create -n tensorflow pip python=3.6
```

5. Después de crear el ambiente, se debe activar con el siguiente comando:

```
$ source activate tensorflow
```

Con lo cual el prompt deberá cambiar observándose de la siguiente manera:

```
(tensorflow)$
```

Una vez realizado todo lo anterior se puede cerrar el terminal.

## 2. Requerimientos

### 2.1. Verificar GPU

Antes de realizar la instalación de Tensor Flow con soporte gpu, se debe verificar que se cuenta con una gpu que sea compatible con Nvidia CUDA GPU. Para esto se puede acceder al siguiente enlace: <https://developer.nvidia.com/cuda-gpus> y verificar.

**Nota: Para este caso se realizo en un portatil asus de referencia N46VB con una GPU Geforce 740M.**

### 2.2. Desabilitar Secure UEFI Boot

Antes de realizar cualquier paso, se debe deshabilitar el secure Boot accediendo a la bios.

### 2.3. Recomendaciones

- Hacer backup de los archivos
- Instalar sobre un ubuntu en blanco

### 3. Instalación de CUDA 8.0 y cuDNN v6.0

1. Añadir los repositorios para instalar los drivers de Nvidia. Abrir un terminal y ejecutar el siguiente comando:

```
$ sudo add-apt-repository ppa:graphics-drivers/ppa
```

2. Instalar el compilador GCC:

```
$ sudo apt-get update  
$ sudo apt-get install build-essential
```

3. Descargar e instalar el driver de Nvidia apropiado. Para este caso se utilizo la version 387.34, ya que a la fecha funciona . Ejecutar los siguientes comandos:

```
$ sudo apt-get install nvidia-387  
$ sudo apt-get install mesa-common-dev  
$ sudo apt-get install freeglut3-dev
```

una vez se complete la instalacion, reiniciar el computador. Si todo va bien, el ubuntu inicia normalmente.

4. En este paso se instalara Nvidia CUDA. se debe tener cuidado con las versiones de cuda y de tensorflow. A la fecha las que funcionan son:
  - TensorFlow 1.4
  - CUDA 8.0

Descargar CUDA 8.0 GA2 (Feb 2017) runfile de:

<https://developer.nvidia.com/cuda-80-ga2-download-archive>

Nota: el archivo se llama

```
cuda_8.0.61_375.26_linux.run
```

5. una vez descargado el archivo, ubicarse en la carpeta de descarga y ejecutar el siguiente comando.

```
$ sudo sh cuda_8.0.61_375.26_linux-run
```

Durante la instalación, el proceso preguntara varias veces si se quiere instalar cosas adicionales. lo importante en este punto es seleccionar que NO (n) cuando el proceso realice la siguiente pregunta:

```
Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 ...?  
(y)es/(n)o/(q)uit: n
```

Otras que aparecen son:

- want to install the CUDA 8.0 Toolkit (Respondemos:yes)
- the Toolkit Location (Respondemos:keep the default)
- a symbolic link (Respondemos:yes) and whether to install the Samples (yes - it helps for testing, put them in the default location asked)

Al finalizar este paso puede aparecer el siguiente warning, el cual se puede ignorar:

```
***WARNING: Incomplete installation! This installation did not install the  
CUDA Driver...
```

6. para que el ubuntu reconozca los comandos de CUDA se deben adicionar las variables al archivo .bashrc. Para esto se accede a este archivo con cualquier editor de texto y se agregan las siguientes líneas al final del archivo:

```
export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}  
export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

Después de guardar los cambios, cerrar todos los terminales, abrir uno nuevo y ejecutar el siguiente comando:

```
$ source ~/.bashrc
```

para validar que hasta este punto todo va bien, se ejecuta el siguiente comando:

```
$ nvidia-smi
```

con lo cual debemos obtener la siguiente respuesta:

```
+-----+  
| NVIDIA-SMI 384.98 Driver Version: 384.98 |  
+-----+  
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |  
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |  
+=====+  
| 0 GeForce GTX 780 Ti Off | 00000000:01:00.0 N/A | N/A |  
| 17% 36C P8 N/A / N/A | 486MiB / 3017MiB | N/A Default |  
+-----+
```

|                     |                          |              |                      |  |             |  |
|---------------------|--------------------------|--------------|----------------------|--|-------------|--|
|                     | 1 GeForce GTX 780 Ti Off |              | 00000000:02:00.0 N/A |  | N/A         |  |
|                     | 17% 32C P8 N/A / N/A     |              | 1MiB / 3020MiB       |  | N/A Default |  |
| +-----+-----+-----+ |                          |              |                      |  |             |  |
| +-----+-----+-----+ |                          |              |                      |  |             |  |
|                     | Processes: GPU Memory    |              |                      |  |             |  |
|                     | GPU PID Type             | Process name | Usage                |  |             |  |
| =====               |                          |              |                      |  |             |  |
|                     | 0 Not Supported          |              |                      |  |             |  |
|                     | 1 Not Supported          |              |                      |  |             |  |
| +-----+-----+-----+ |                          |              |                      |  |             |  |

si esto sucede el CUDA 8.0 quedo bien instalado y funcionando.

7. En este paso se va a instalar el cuDNN. Se debe descargar el cuDNN v6.0 (April 27, 2017) de: <https://developer.nvidia.com/rdp/form/cudnn-download-survey>. Se debe tener cuenta de Nvidia o registrarse. el archivo se llama:

```
cudnn-8.0-linux-x64-v6.0-tgz
```

Después de descargar, ubicarse en la carpeta de descarga y ejecutar los siguientes comandos para instalar cuDNN:

```
tar -zxvf cudnn-8.0-linux-x64-v6.0-ga.tgz
$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include/
$ sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64/
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h
$ sudo chmod a+r /usr/local/cuda/lib64/libcudnn*
```

**Hasta este punto CUDA 8.0 y cuDNN 6.0 se han instalado correctamente.**

## 4. Instalación de TensorFlow

Con el ambiente creado en la sección 1 (Instalar anaconda (opcional pero recomendado)) se procede con la instalación de tensorflow.

1. Se debe activar el ambiente creado en la seccion 1 y se procede a instalar tensor Flow con el siguiente comando:

```
(tensorflow)$ pip install --ignore-installed --upgrade https://storage.
googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-1.4.0-cp36-cp36m-
linux_x86_64.whl
```

2. Una vez instalado tensor Flow, se debe realizar la validación de la instalación. Para esto se debe cerrar el prompt y abrirlo nuevamente. Paso seguido se de activar el ambiente creado en la seccion 1.

```
$ source activate tensorflow\\
```

Una vez se tiene el ambiente activado se debe invocar python

```
(tensorflow)$ python
```

ingresar una por una cada una de las líneas de código mostradas a continuación:

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

Después de ingresar la última línea de código y haber pulsado enter, el software debe responder con el siguiente mensaje:

**Hello, TensorFlow!**

Hasta este punto solo se ha instalado la herramienta tensorflow, en los pasos siguientes se mostrará el proceso para instalar software adicional y librerías de Python necesarias para el desarrollo del curso de redes neuronales y aprendizaje profundo.

## 5. Instalación de Software Adicional y Librerías de Python

### 5.1. Instalación de la librería scikit-learn:

Para instalar esta librería nos debemos ubicar en el ambiente de tensorflow creado en los pasos 4 y 5. Una vez allí ejecutamos el comando:

```
(tensorflow)$ conda install -c anaconda scikit-learn
```

y seguimos los pasos de instalación aceptando todas las preguntas durante el proceso de instalación.

### 5.2. Instalación de la librería matplotlib

Nuevamente nos debemos ubicar en el ambiente de tensor Flow y ejecutar el siguiente comando:

```
(tensorflow)$ conda install -c conda-forge matplotlib
```

### 5.3. Instalacion del IDE Spyder

Para este caso de igual forma se sigue el mismo procedimiento que en los pasos anteriores. Ubicados en el ambiente de tensor Flow, ejecutar el siguiente comando:

```
(tensorflow)$ conda install spyder
```

**Nota:** Este ultimo paso de instalar spyder es opcional, ya que se puede acceder a este mediante el anaconda-navigator invocando el comando:

```
(tensorflow)$ anaconda-navigator
```

## Referencias

<https://www.quantstart.com/articles/installing-tensorflow-on-ubuntu-1604-with-an-nvidia-gpu>