

Instalación de TensorFlow

En el presente documento se enumeran los pasos a seguir para instalar TensorFlow de dos maneras:

1. CPU. Los pasos se llevaron a cabo en una versión de Linux y sin emplear un entorno virtual.
2. GPU. Además de instalar TensorFlow, se instaló CUDA y cuDNN. Asimismo, se empleó un entorno virtual.

Los entornos virtuales son recomendados para tener varias instalaciones aisladas dentro de un sólo equipo. Además, en caso de tener varios proyectos asociados en entornos virtuales, se puede garantizar un entorno seguro para ejecutarlos, sin ser afectados por los requerimientos del resto de proyectos.

SUGERENCIA. Si tienes proyectos o material de otras materias en donde no uses TensorFlow, te sugiero crear un entorno virtual para la materia e instalar allí las librerías de interés para el curso (TensorFlow y Matplotlib)

Instalación en equipos con CPU y Linux

La presente guía son los pasos que lleve a cabo para instalar Tensorflow 2.7.0 en Python 3.7.9 (sin usar conda o un ambiente virtual). Las características del equipo son:

Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz
RAM: 8 GB

Requisitos

- Python 3.6 a 3.9
- pip 19.0 o una versión posterior

Para verificar si se tienen las versiones usar:

```
python3 --version  
pip3 --version
```

En caso de no contar con la versión de Python, en la siguiente liga se encuentra un manual de instalación para Python 3.7.9

<https://tecadmin.net/install-python-3-7-on-ubuntu-linuxmint/>

Si se tiene la versión de Python, pero no la versión de pip3, actualizar con el siguiente comando:

```
python3 -m pip install --upgrade pip
```

```
emiranda@emiranda-Lenovo-Z40-70 ~ $ python3 -m pip install --upgrade pip
Defaulting to user installation because normal site-packages is not writeable
Collecting pip
  Using cached pip-21.3.1-py3-none-any.whl (1.7 MB)
Installing collected packages: pip
Successfully installed pip-21.3.1
```

Para instalar TensorFlow ejecutar el comando:

`pip install --upgrade tensorflow`

```
emiranda@emiranda-Lenovo-Z40-70 ~ $ pip install --upgrade tensorflow
Defaulting to user installation because normal site-packages is not writeable
Collecting tensorflow
  Downloading tensorflow-2.7.0-cp37-cp37m-manylinux2010_x86_64.whl (489.6 MB)
    | 489.6 MB 11 kB/s
Requirement already satisfied: typing-extensions>=3.6.6 in ./local/lib/python3.7/site-packages (from tensorflow) (4.0.1)
Collecting tensorflow-estimator<2.8,~>2.7.0rc0
  Downloading tensorflow-estimator-2.7.0-py2.py3-none-any.whl (463 kB)
    | 463 kB 3.9 MB/s
Collecting grpcio<2.0,>=1.24.3
  Downloading grpcio-1.42.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.0 MB)
    | 4.0 MB 2.9 MB/s
Collecting astunparse>=1.6.0
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting google-pasta>=0.1.1
  Downloading google-pasta-0.2.0-py3-none-any.whl (57 kB)
    | 57 kB 859 kB/s
Collecting keras-preprocessing>=1.1.1
  Downloading Keras Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
    | 42 kB 194 kB/s
Collecting termcolor>=1.1.0
  Downloading termcolor-1.1.0.tar.gz (3.9 kB)
  Preparing metadata (setup.py) ... done
Collecting opt-einsum>=2.3.2
  Downloading opt-einsum-3.3.0-py3-none-any.whl (65 kB)
    | 65 kB 585 kB/s
Requirement already satisfied: numpy>=1.14.5 in ./local/lib/python3.7/site-packages (from tensorflow) (1.21.4)
```

La instalación se probó con el siguiente código:

```

import tensorflow as tf

mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train, x_test = x_train / 255.0, x_test / 255.0
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)

model.evaluate(x_test, y_test, verbose=2)

```

FUENTE: <https://www.tensorflow.org/tutorials/quickstart/beginner>

Cabe mencionar que marca un *warning*, pero se ignora porque la instalación está en un equipo que no tiene GPU.

```

emiranda@emiranda-Lenovo-Z40-70 ~ $ python3 test_tf.py
2021-12-05 23:28:44.044138: W tensorflow/stream_executor/platform/default/dso_loader.cc:
64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cann
ot open shared object file: No such file or directory
2021-12-05 23:28:44.044173: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.

```

La salida del entrenamiento es:

```

Epoch 1/5
 1/1875 [.....] - ETA: 12:36 - loss: 2.3059 - accuracy: 0.093
21/1875 [.....] - ETA: 4s - loss: 1.9159 - accuracy: 0.4122
1875/1875 [=====] - 5s 2ms/step - loss: 0.2946 - accuracy: 0.91
55
Epoch 2/5
1875/1875 [=====] - 4s 2ms/step - loss: 0.1437 - accuracy: 0.95
70
Epoch 3/5
1875/1875 [=====] - 4s 2ms/step - loss: 0.1064 - accuracy: 0.96
73
Epoch 4/5
1875/1875 [=====] - 5s 3ms/step - loss: 0.0889 - accuracy: 0.97
27
Epoch 5/5
1875/1875 [=====] - 4s 2ms/step - loss: 0.0748 - accuracy: 0.97
69
2021-12-05 23:29:11.081531: W tensorflow/core/framework/cpu_allocator_impl.cc:82] Alloca
tion of 31360000 exceeds 10% of free system memory.
313/313 - 1s - loss: 0.0750 - accuracy: 0.9769 - 586ms/epoch - 2ms/step
emiranda@emiranda-Lenovo-Z40-70 ~ $

```

Instalación en equipos con tarjeta gráfica

Antes de hacer la instalación, se sugiere tener la siguiente información:

1. Modelo de tarjeta gráfica Nvidia

1. Windows

<https://www.pcmag.com/how-to/what-graphics-card-do-i-have#:~:text=You%20can%20find%20out%20what,list%20your%20GPU%20right%20there.>

2. Ubuntu

`sudo lshw -C display`

`sudo ubuntu-drivers devices`

2. Driver adecuado para su tarjeta gráfica

1. Windows

<https://la.nvidia.com/Download/index.aspx?lang=la>

2. Ubuntu:

`sudo ubuntu-drivers devices`

<https://la.nvidia.com/Download/index.aspx?lang=la>

3. Versiones de CUDA y cuDNN

<https://docs.nvidia.com/deeplearning/cudnn/latest/reference/support-matrix.html>

4. Requerimientos para instalar Tensorflow

<https://www.tensorflow.org/install>

Pasos para instalar:

1. Instalar el driver de la tarjeta gráfica Nvidia
2. Instalar CUDA
3. Instalar cuDNN

En la liga <https://docs.nvidia.com/deeplearning/cudnn/latest/reference/support-matrix.html> se dan las instrucciones para llevar a cabo la instalación de CUDA y cuDNN para Windows o varias versiones de Linux.

4. Verificar la instalación

`nvidia-smi`

```
administrator@ubuntu:~$ nvidia-smi
Tue Jun 13 03:11:38 2023
```

NVIDIA-SMI 525.89.02 Driver Version: 525.89.02 CUDA Version: 12.0													
GPU		Name	Persistence-MI	Bus-Id	Disp.A	Volatile Uncorr. ECC							
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage		GPU-Util	Compute M.						
								MIG M.					
0		NVIDIA GeForce ...	Off	00000000:03:00.0	Off			N/A					
31%	34C	P8	18W / 200W	2900MiB / 8192MiB		0%	Default	N/A					
Processes:													
GPU	GI	CI	PID	Type	Process name	GPU Memory Usage							
		ID	ID										
0	N/A	N/A	22010	C	python3	2898MiB							

FUENTE: <https://www.gpu-mart.com/blog/monitor-gpu-utilization-with-nvidia-smi>

Para la instalación de la librería se recomienda tener instalado Anaconda (<https://www.anaconda.com/>) o miniconda (<https://docs.anaconda.com/free/miniconda/>), los cuales permiten trabajar con entornos virtuales, o en su defecto crear un entorno virtual con python (<https://docs.python.org/es/3/tutorial/venv.html>)

1. Crear o activar el entorno virtual en donde ejecutaran los cuadernos Jupyter de la materia
2. Instalar tensorflow con pip

❖ Recomendaciones de instalación

<https://www.lavivienpost.com/install-tensorflow-gpu-on-windows-complete-guide/>
<https://blog.stackademic.com/use-gpu-in-tensorflow-on-ubuntu-22-04-f033e59cf5cb>

❖ Manejo de ambientes con anaconda

<https://docs.anaconda.com/free/navigator/tutorials/manage-environments/>
<https://medium.com/@aminasaheed223/a-comprehensive-tutorial-on-miniconda-creating-virtual-environments-and-setting-up-with-vs-code-f98d22fac8e2>