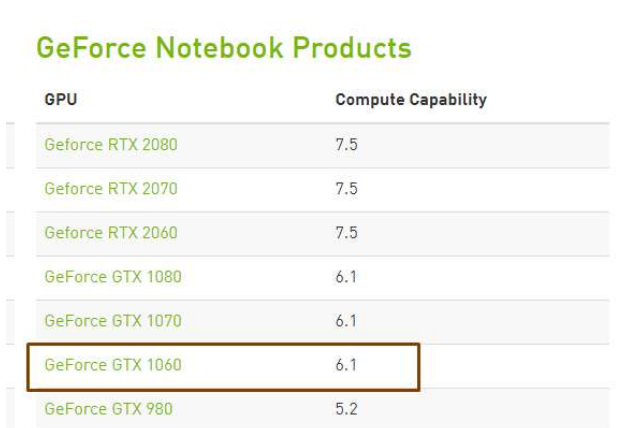


Ambiente Desarrollo: Windows 10 + NVIDIA GeForce GTX 1060 + TensorFlow GPU (Manual)

1. Verificar compatibilidad de tarjeta de video (NVIDIA GeForce GTX 1060):

- a. En el link <https://developer.nvidia.com/cuda-gpus> se debe buscar la referencia de la tarjeta de video para conocer si es compatible:



GPU	Compute Capability
GeForce RTX 2080	7.5
GeForce RTX 2070	7.5
GeForce RTX 2060	7.5
GeForce GTX 1080	6.1
GeForce GTX 1070	6.1
GeForce GTX 1060	6.1
GeForce GTX 980	5.2

- b. La tarjeta de video GTX 1060 cuenta con "Compute Capability" de 6.1, lo mínimo requerido es 3.5. Se puede revisar este link https://www.tensorflow.org/install/gpu#hardware_requirements

2. Verificar los requerimientos de software:

- a. En la documentación oficial de TensorFlow se pueden validar los requerimientos de software mínimos requeridos:
- https://www.tensorflow.org/install/gpu#software_requirements

Software requirements

The following NVIDIA® software must be installed on your system:

- [NVIDIA® GPU drivers](#) —CUDA 10.0 requires 410.x or higher.
- [CUDA® Toolkit](#) —TensorFlow supports CUDA 10.0 (TensorFlow >= 1.13.0)
- [CUPTI](#) ships with the CUDA Toolkit.
- [cuDNN SDK](#) (>= 7.4.1)
- (Optional) [TensorRT 5.0](#) to improve latency and throughput for inference on some models.

- b. En esta guía se quieren instalar las siguientes versiones:

- TensorFlow:** 1.13.1.
- CUDA Toolkit:** 10.0.130.
- cuDNN:** 7.4.1.

3. Instalar Visual Studio:

- a. "CUDA Toolkit 10.0.130" tiene como requisito tener instalado "Visual Studio". Si se revisa la documentación de CUDA <https://docs.nvidia.com/cuda/archive/10.0/cuda-installation-guide-microsoft-windows/index.html>, el Visual Studio requerido es 2017:

Table 1. Windows Operating System Support in CUDA 10.0

Operating System	Native x86_64	Cross (x86_32 on x86_64)
Windows 10	YES	YES
Windows 8.1	YES	YES
Windows 7	YES	YES
Windows Server 2016	YES	NO
Windows Server 2012 R2	YES	NO

Table 2. Windows Compiler Support in CUDA 10.0

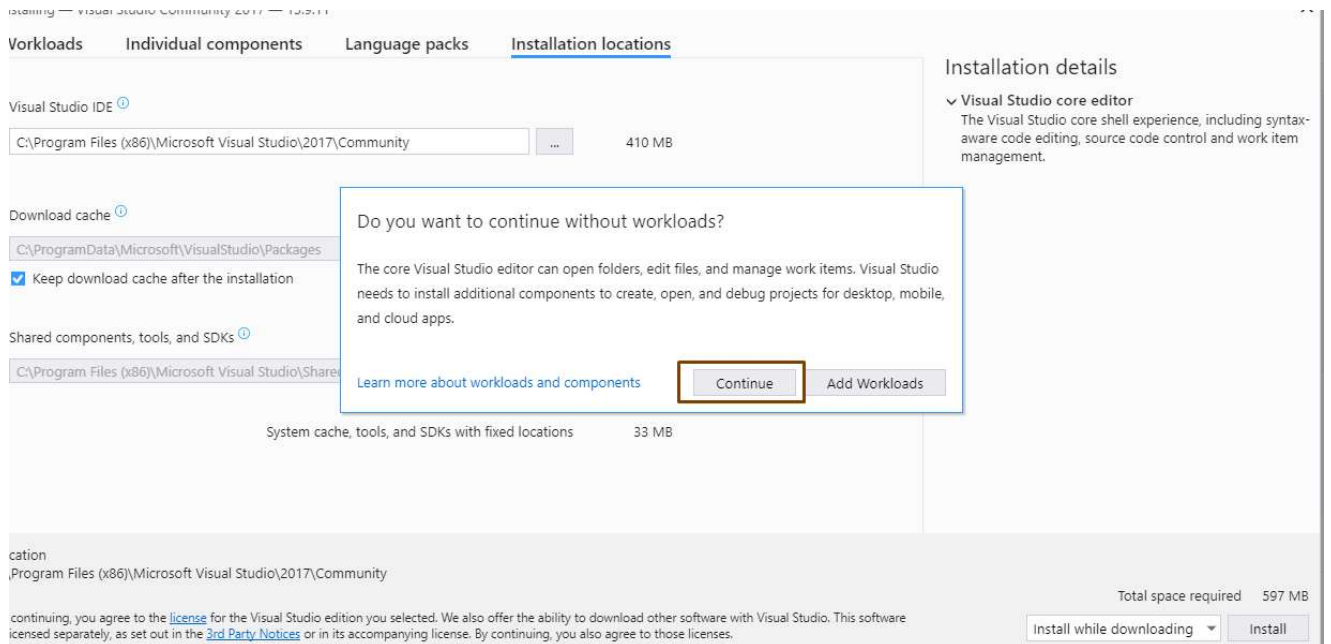
Compiler	IDE	Native x86_64	Cross (x86_32 on x86_64)
Visual C++ 15.x	Visual Studio 2017 (RTW and all updates)	YES	NO
Visual C++ 14.0	Visual Studio 2015	YES	NO
	Visual Studio Community 2015	YES	NO
Visual C++ 12.0	Visual Studio 2013	YES	YES
Visual C++ 11.0	Visual Studio 2012	YES	YES

x86_32 support is limited. See the [x86_32-bit Support](#) section for details.

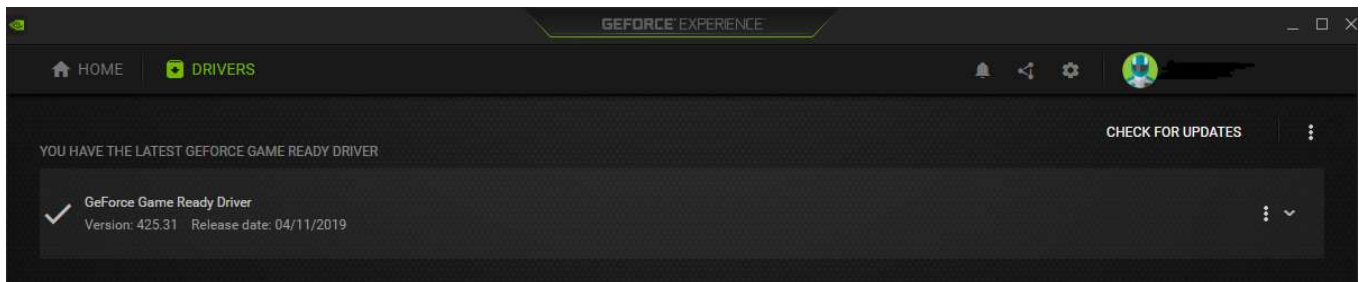
- b. Si se hace caso omiso (NO recomendado) al requisito del *paso b*, al momento de estar instalando CUDA se puede visualizar un mensaje como el siguiente:



- c. Para esta guía se descarga e instala "VS 2017 Community". No se necesitan tener instalados componentes de desarrollo, por lo cual se puede realizar instalación sin "Workloads", dejando por defecto lo seleccionado por el instalador:



4. Instalar Visual Studio Code:
 - a. "Visual Studio Code" lo uso como IDE por defecto para desarrollo de proyectos de I.A. y Python.
 - b. Link: <https://visualstudio.microsoft.com/>
5. Actualizar los drivers de la tarjeta de video:
 - a. Se realizó mediante programa "NVIDIA GeForceExperience":



- b. De igual manera se puede realizar por el sitio Web NVIDIA <https://www.nvidia.co.uk/Download/index.aspx?lang=en-us>:

NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products.

[Help](#)

Product Type:

NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products.

Help

Product Type: GeForce
 Product Series: GeForce 10 Series
 Product: GeForce GTX 1060
 Operating System: Windows 10 64-bit
 Windows Driver Type: Standard
 Download Type: Game Ready Driver (GRD)
 Language: English (US)

SEARCH

c. Verificar los drivers instalados:

```
C:\Program Files\NVIDIA Corporation\NVSMI
nvidia-smi
```

NVIDIA-SMI 425.31 Driver Version: 425.31 CUDA Version: 10.1									
GPU	Name	TCC/WDDM	Bus-Id	Memory-Usage	Disp.A	Volatile	Uncorr. ECC	GPU-Util	Compute M.
0	GeForce GTX 1060	WDDM	00000000:01:00:00	92MiB / 6144MiB	Off	0%	Default		

Processes:					GPU Memory Usage
GPU	PID	Type	Process name		
0	11492	C+G	...xperience\NVIDIA GeForce Experience.exe		N/A

d. Versión instalada del driver: 425.31.

6. Instalar Anaconda:

- Link: <https://www.anaconda.com/distribution/>
- Versión: Anaconda 2019.03 + Python 3.7

Anaconda 2019.03 for Windows Installer

Python 3.7 version

[Download](#)

[64-Bit Graphical Installer \(662 MB\)](#)
[32-Bit Graphical Installer \(546 MB\)](#)

Python 2.7 version

[Download](#)

[64-Bit Graphical Installer \(587 MB\)](#)
[32-Bit Graphical Installer \(493 MB\)](#)

c. Seleccionar las siguientes opciones:

Anaconda3 2019.03 (64-bit) Setup

Advanced Installation Options
 Customize how Anaconda integrates with Windows

Advanced Options

☐ Add Anaconda to my PATH environment variable
 Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☒ Register Anaconda as my default Python 3.7
 This will allow other programs, such as Python Tools for Visual Studio, PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

Anaconda, Inc.

< Back Install Cancel

7. Instalar CUDA Toolkit:

- Con base a los requisitos mencionados en el paso #2, la versión instalar de CUDA es la 1.0.130:

- b. Descargar CUDA v10.0.130 del sitio de NVIDIA <https://developer.nvidia.com/cuda-10.0-download-archive>:

CUDA Toolkit 10.0 Archive

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System: **Windows** **Linux** **Mac OSX**

Architecture: **x86_64**

Version: **10** **8.1** **7** **Server 2016** **Server 2012 R2**

Installer Type: **exe (network)** **exe (local)**

Download Installer for Windows 10 x86_64

The base installer is available for download below.

Base Installer

Download [2.1 GB]

Installation Instructions:

1. Double click cuda_10.0.130_411.31_win10.exe
2. Follow on-screen prompts

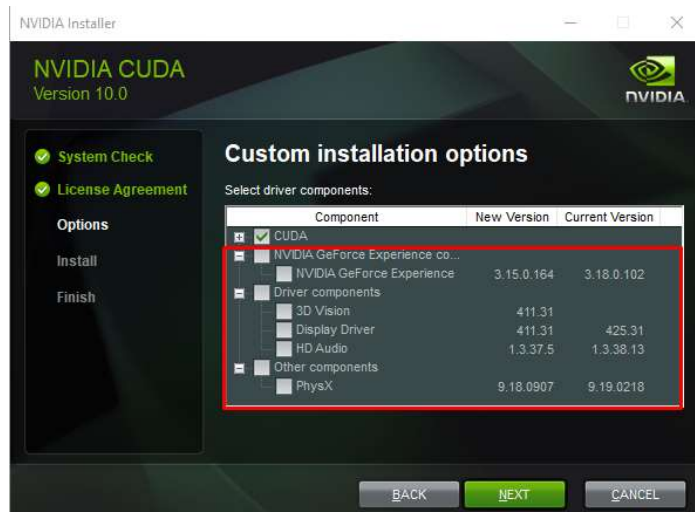
The checksums for the installer and patches can be found in [Installer Checksums](#).
For further information, see the [Installation Guide for Microsoft Windows](#) and the [CUDA Quick Start Guide](#).

- c. Instalar la base de CUDA Toolkit:

- Elegir la opción de instalación Custom:



- Solo instalar los componentes de CUDA. NO instalar los componentes "NVIDIA GeForce Experience", "Driver Components" y "Other Components" siempre y cuando las versiones instaladas en el sistema sean mayores que las que se sugieren instalar:



- Para mayor información de este paso, se puede revisar el siguiente link <https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html> para validar que CUDA 10.0.130 soporta versiones >= 411.31 del driver de la tarjeta de video en Windows:

Table 1. CUDA Toolkit and Compatible Driver Versions

CUDA Toolkit	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 10.1.105	>= 418.39	>= 418.96
CUDA 10.0.130	>= 410.48	>= 411.31
CUDA 9.2 (9.2.148 Update 1)	>= 396.37	>= 398.26
CUDA 9.2 (9.2.88)	>= 396.26	>= 397.44
CUDA 9.1 (9.1.85)	>= 390.46	>= 391.29
CUDA 9.0 (9.0.76)	>= 384.81	>= 385.54
CUDA 8.0 (8.0.61 GA2)	>= 375.26	>= 376.51
CUDA 8.0 (8.0.44)	>= 367.48	>= 369.30
CUDA 7.5 (7.5.16)	>= 352.31	>= 353.66
CUDA 7.0 (7.0.28)	>= 346.46	>= 347.62

- Si todo ha salido correctamente con la instalación, se debe visualizar la siguiente ventana:



- Ir a la ruta donde quedó instalado CUDA (C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0). Esta ruta es importante en pasos posteriores, ya que se debe adicionar a las variables de entorno de Windows.

- d. Para verificar la instalación de CUDA se puede ejecutar el siguiente comando "nvcc --version":

```
D:\Software\cmdr v1.3.11
A nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Sat Aug 25 21:08:04 Central Daylight Time 2018
Cuda compilation tools, release 10.0, V10.0.130
```

8. Instalar cuDNN:

- Con base a los requisitos mencionados en el paso #2, la versión instalar de cuDNN a instalar es la 7.4.1.
- Descargar cuDNN desde <https://developer.nvidia.com/rdp/cudnn-archive>. Requiere Inicio de sesión o Registro (Gratis) para poder obtener los instaladores.
- Se descarga el paquete "cuDNN Library for Windows 10" como se ve en la siguiente imagen:

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

Download cuDNN v7.4.2 [Dec 14, 2018], for CUDA 10.0

Download cuDNN v7.4.2 [Dec 14, 2018], for CUDA 9.2

Download cuDNN v7.4.2 [Dec 14, 2018], for CUDA 9.0

Download cuDNN v7.4.1 [Nov 8, 2018], for CUDA 10.0

Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos(x86_64 architecture)

cuDNN Library for Windows 7

cuDNN Library for Windows 10

cuDNN Library for Linux

cuDNN Library for OSX

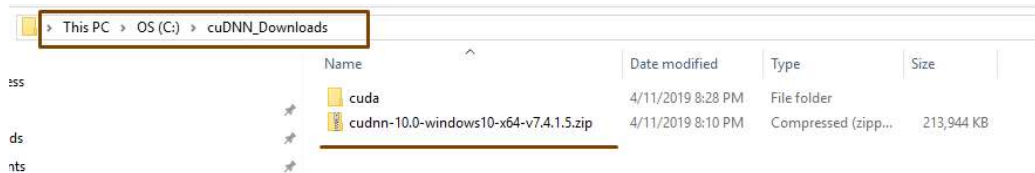
cuDNN Runtime Library for Ubuntu18.04 [Deb]

cuDNN Developer Library for Ubuntu18.04 [Deb]

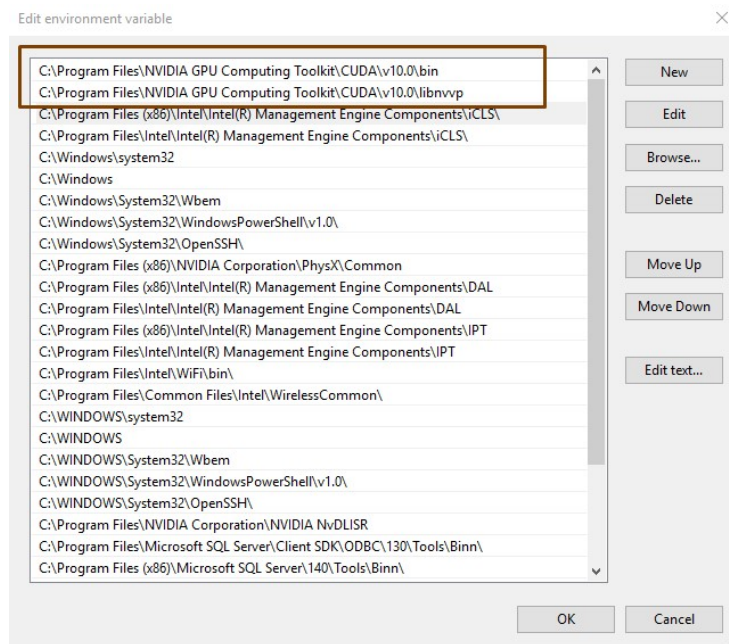
cuDNN Code Samples and User Guide for Ubuntu18.04 [Deb]

- d. En el paso anterior se descarga un archivo "cudnn-10.0-windows10-x64-v7.4.1.5.zip". Personalmente creé una carpeta "cuDNN_Downloads" en el disco C y se procede a

descomprimir el archivo:



- e. Siguiendo el "paso 3" de la documentación oficial de NVIDIA en el link <https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#installwindows>, se realiza lo siguiente:
- Se copia el archivo "C:\cuDNN_Downloads\cuda\bin\cudnn64_7.dll" en la ruta "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin".
 - Se copia el archivo "C:\cuDNN_Downloads\cuda\include\cudnn.h" en la ruta "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\include".
 - Se copia el archivo "C:\cuDNN_Downloads\cuda\lib\x64\cudnn.lib" en la ruta "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\lib\x64".
- f. Adicionar/Verificar en la variable de entorno PATH las carpetas "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin" y "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\libnvvp":



9. Crear "conda environment":

- a. Para crear el "environment" donde se va a usar TensorFlow se debe tener en cuenta que la versión de Python máxima soportada es la 3.5-3.6, tal y como se ve en la siguiente tabla:

GPU

Version	Python version
tensorflow_gpu-1.13.0	3.5-3.6
tensorflow_gpu-1.12.0	3.5-3.6

- b. Abrir "Anaconda Prompt" y crear un nuevo ambiente indicando la versión de Python 3.6. Para efectos de esta guía se crea un "environment" llamado *tensorflow*:

```
Anaconda Prompt
(base) C:\Users\dloperab>conda create --name tensorflow python=3.6
WARNING: The conda.compat module is deprecated and will be removed in a future release.
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\dloperab\Anaconda3\envs\tensorflow

added / updated specs:
- python=3.6
```

- c. Verificar los "environments" creados en Anaconda:

```
(base) C:\Users\dloperab>conda env list
WARNING: The conda.compat module is deprecated and will be removed in a future release.
WARNING: The conda.compat module is deprecated and will be removed in a future release.
# conda environments:
#
base                  *  C:\Users\dloperab\Anaconda3
tensorflow             C:\Users\dloperab\Anaconda3\envs\tensorflow
```

- d. Activar el "environment":

```
Anaconda Prompt - conda create -name tensorflow

(base) C:\Users\dloperab>activate tensorflow

(tensorflow) C:\Users\dloperab>
```

10. Instalar TensorFlow con soporte GPU:

- a. Escribir el comando "pip install tensorflow-gpu" en "Anaconda Prompt":

```
Anaconda Prompt - pip install tensorflow-gpu

(tensorflow) C:\Users\dloperab>pip install tensorflow-gpu
Collecting tensorflow-gpu
Using cached https://files.pythonhosted.org/packages/6d/c4/d5b81c2d40be9219a23852f630df9e1ed88a22f88da774e1bf69261f9b32/tensorflow_gpu-1.13.1-cp36-cp36m-win_amd64.whl
```

- b. Al terminar el punto anterior, se procede a probar la instalación de TensorFlow por medio del siguiente código en la consola:

```
Anaconda Prompt - conda install tensorflow-gpu - python

(tf_conda) C:\Users\dloperab>python
Python 3.6.8 [Anaconda, Inc.] (default, Feb 21 2019, 18:30:04) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import tensorflow as tf
>>> sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
2019-04-11 19:20:17.497910: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
2019-04-11 19:20:18.505442: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties:
  name: GeForce GTX 1060 major: 6 minor: 1 memoryClockRate(GHz): 1.733
  pciBusID: 0000:01:00.0
totalMemory: 6.00GiB freeMemory: 4.97GiB
2019-04-11 19:20:18.515288: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0
2019-04-11 19:20:19.057796: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-04-11 19:20:19.063518: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0: 0
2019-04-11 19:20:19.067379: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2019-04-11 19:20:19.070931: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 4716 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1060, pci bus id: 0000:01:00.0, compute capability: 6.1)
Device mapping:
/job:localhost/replica:0/task:0/device:GPU:0 -> device: 0, name: GeForce GTX 1060, pci bus id: 0000:01:00.0, compute capability: 6.1
2019-04-11 19:20:19.085673: I tensorflow/core/common_runtime/direct_session.cc:317] Device mapping:
/job:localhost/replica:0/task:0/device:GPU:0 -> device: 0, name: GeForce GTX 1060, pci bus id: 0000:01:00.0, compute capability: 6.1
>>>
```

Referencias:

- Changing your hardware to achieve faster Deep Learning on your PC: <https://towardsdatascience.com/pimp-up-your-pc-for-machine-learning-d11f68c2815>
- Install TensorFlow with GPU Support on Windows 10 (without a full CUDA install): <https://www.pugetsystems.com/labs/hpc/Install-TensorFlow-with-GPU-Support-on-Windows-10-without-a-full-CUDA-install-1172/>
- Ubuntu 18.04: Install TensorFlow and Keras for Deep Learning: <https://www.pyimagesearch.com/2019/01/30/ubuntu-18-04-install-tensorflow-and-keras-for-deep-learning/>

Ambiente Desarrollo: Windows 10 + NVIDIA GeForce GTX 1060 + TensorFlow GPU (Auto)

1. Verificar compatibilidad de tarjeta de video (NVIDIA GeForce GTX 1060):

- En el link <https://developer.nvidia.com/cuda-gpus> se debe buscar la referencia de la tarjeta de video para conocer si es compatible:

GeForce Notebook Products	
GPU	Compute Capability
GeForce RTX 2080	7.5
GeForce RTX 2070	7.5
GeForce RTX 2060	7.5
GeForce GTX 1080	6.1
GeForce GTX 1070	6.1
GeForce GTX 1060	6.1
GeForce GTX 980	5.2

- La tarjeta de video GTX 1060 cuenta con "Compute Capability" de 6.1, lo mínimo requerido es 3.5. Se puede revisar este link https://www.tensorflow.org/install/gpu#hardware_requirements

2. Instalar Visual Studio:

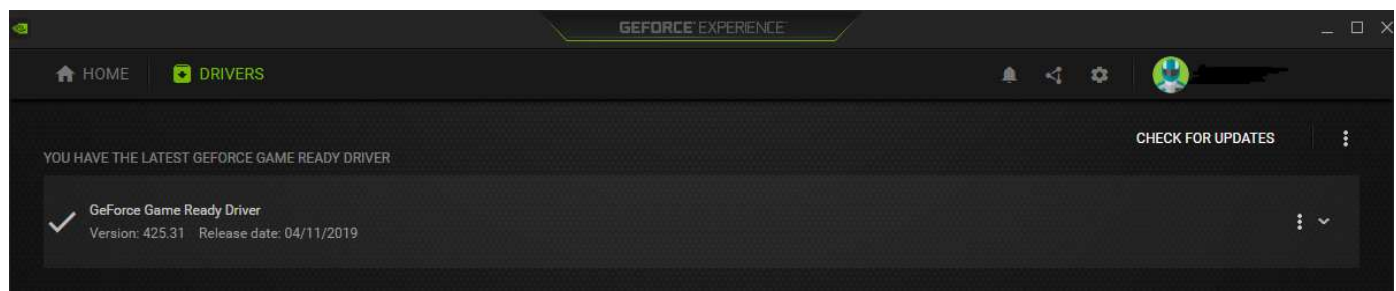
- Al momento de esta guía tengo instalado *Visual Studio 2019 Enterprise* para otros tipos de desarrollo que realizo. NO uso *Visual Studio* como herramienta de desarrollo para Python, para este propósito uso *Visual Studio Code*.

3. Instalar Visual Studio Code:

- Visual Studio Code* lo uso como IDE por defecto para desarrollo de proyectos de I.A. y Python.
- Link: <https://visualstudio.microsoft.com/>

4. Actualizar los drivers de la tarjeta de video:

- Se realizó mediante programa "NVIDIA GeForce Experience":



- De igual manera se puede realizar por el sitio Web NVIDIA <https://www.nvidia.co.uk/Download/index.aspx?lang=en-us>:

NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products.

Help

Product Type:	GeForce
Product Series:	GeForce 10 Series
Product:	GeForce GTX 1060
Operating System:	Windows 10 64-bit
Windows Driver Type:	Standard
Download Type:	Game Ready Driver (GRD)
Language:	English (US)

SEARCH

- Verificar los drivers instalados:


```

C:\Program Files\NVIDIA Corporation\NVSMI
nvidia-smi
Thu Apr 11 10:47:50 2019
+-----+
| NVIDIA-SMI 425.31      Driver Version: 425.31      CUDA Version: 10.1     |
+-----+-----+
| GPU Name               TCC/WDDM | Bus-Id | Disp.A | Volatile Uncorr. ECC |
+-----+-----+-----+-----+
| Fan  Temp  Perf  Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+
| 0   GeForce GTX 1060  WDDM  | 00000000:01:00.0 Off |          N/A       |
| N/A   38C    P8      11W /  N/A |  92MiB /  6144MiB     |        0%      Default |
+-----+-----+-----+-----+
+-----+
| Processes:                                                       GPU Memory |
|  GPU PID       Type    Process name                               Usage    |
+-----+-----+-----+-----+
|  0   11492     C+G     ...xperience\NVIDIA GeForce Experience.exe  N/A      |
+-----+

```

d. Versión instalada del driver: **425.31**.

5. Instalar Anaconda:

- Link: <https://www.anaconda.com/distribution/>
- Versión: Anaconda 2019.03 + Python 3.7

Anaconda 2019.03 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (662 MB)

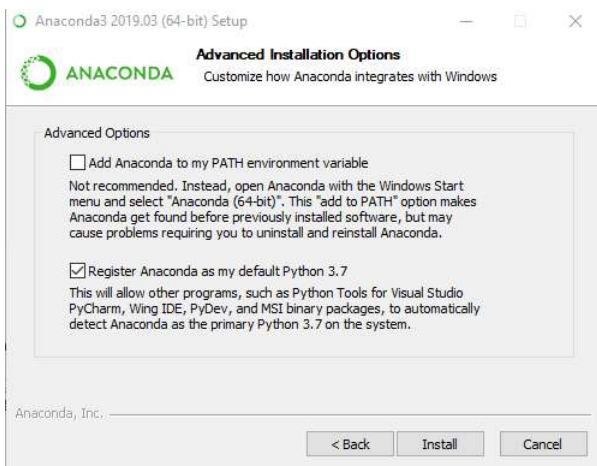
32-Bit Graphical Installer (546 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (587 MB)
32-Bit Graphical Installer (493 MB)

c. Seleccionar las siguientes opciones:



6. Crear "conda environment":

- Para crear el "environment" donde se va a usar TensorFlow se debe tener en cuenta que la versión de Python máxima soportada es la 3.6, tal y como se ve en la siguiente tabla:

GPU

Version	Python version
tensorflow_gpu-1.13.0	3.5-3.6
tensorflow_gpu-1.12.0	3.5-3.6

- Abir "Anaconda Prompt" y crear un nuevo ambiente indicando la versión de Python 3.6. Para efectos de esta guía se crea un "environment" llamado `tf_conda`.

```

Anaconda Prompt
(base) C:\Users\dloperab>conda create --name tf_conda python=3.6
WARNING: The conda.compat module is deprecated and will be removed in a future release.

```

- c. Verificar los "environments" creados en Anaconda:

```

(base) C:\Users\dloperab>conda env list
WARNING: The conda.compat module is deprecated and will be removed in a future release.
WARNING: The conda.compat module is deprecated and will be removed in a future release.
# conda environments:
#
base                    * C:\Users\dloperab\Anaconda3
tensorflow               C:\Users\dloperab\Anaconda3\envs\tensorflow

```

- d. Activar el "environment":

```

(base) C:\Users\dloperab>activate tf_conda
(tf_conda) C:\Users\dloperab>

```

7. Instalar TensorFlow con soporte GPU:

- a. Escribir el comando "conda install tensorflow-gpu" en "Anaconda Prompt":

```

Anaconda Prompt - conda install tensorflow-gpu
(tf_conda) C:\Users\dloperab>conda install tensorflow-gpu
WARNING: The conda.compat module is deprecated and will be removed in a future release.
Collecting package metadata: done

```

- b. Al terminar el punto anterior, se procede a probar la instalación de TensorFlow por medio del siguiente código en la consola:

```

Anaconda Prompt - conda install tensorflow-gpu - python
(tf_conda) C:\Users\dloperab>python
Python 3.6.8 [Anaconda, Inc.] (default, Feb 21 2019, 18:30:04) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import tensorflow as tf
>>> sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
2019-04-11 19:20:17.497910: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
2019-04-11 19:20:18.505442: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties:
  name: GeForce GTX 1060 major: 6 minor: 1 memoryClockRate(GHz): 1.733
  pciBusID: 0000:01:00:0
  totalMemory: 6.00GiB freeMemory: 4.97GiB
2019-04-11 19:20:18.515208: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0
2019-04-11 19:20:19.057796: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-04-11 19:20:19.063518: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0 0
2019-04-11 19:20:19.067370: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2019-04-11 19:20:19.070931: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 4716 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1060, pci bus id: 0000:01:00:0, compute capability: 6.1)
Device mapping:
/job:localhost/replica:0/task:0/device:GPU:0 -> device: 0, name: GeForce GTX 1060, pci bus id: 0000:01:00:0, compute capability: 6.1
2019-04-11 19:20:19.085673: I tensorflow/core/common_runtime/direct_session.cc:317] Device mapping:
/job:localhost/replica:0/task:0/device:GPU:0 -> device: 0, name: GeForce GTX 1060, pci bus id: 0000:01:00:0, compute capability: 6.1
>>>

```

8. Cierre:

Esta instalación "Auto" por medio de Anaconda evita muchos dolores de cabeza con los instaladores, versiones y configuraciones que se deben hacer manualmente con CUDA y cuDNN. Lo que hace Anaconda es que con el comando "conda install tensorflow-gpu" detecta los requisitos que necesita para poder instalar TensorFlow con soporte para GPU con base a la tarjeta de video que se tenga instalada.

Si se quiere indagar un poco más y conocer la versión de TensorFlow, CUDA y cuDNN que Anaconda ha instalado se puede realizar lo siguiente:

- Versión TensorFlow (1.13.1):

```

Anaconda Prompt - conda install tensorflow-gpu - python
(tf_conda) C:\Users\dloperab>python
Python 3.6.8 [Anaconda, Inc.] (default, Feb 21 2019, 18:30:04) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import tensorflow as tf
>>> tf.VERSION
'1.13.1'

```

- Versión CUDA (10.0.130) y cuDNN (7.3.1):
 - Ir a la ruta "<Carpetanaconda>\Anaconda3\pkgs":

dloperab > Anaconda3 > pkgs

Name	Date modified	Type	Size
cryptography-2.6.1-py27h7e1d8c1_0	4/11/2019 3:10 PM	File folder	
cuda-toolkit-10.0.130-0	4/11/2019 4:43 PM	File folder	
cudnn-7.3.1-cuda10.0_0	4/11/2019 4:27 PM	File folder	
curl-7.64.0-h2a8ff88b_2	4/11/2019 3:18 PM	File folder	

Referencias:

- Changing your hardware to achieve faster Deep Learning on your PC: <https://towardsdatascience.com/pimp-up-your-pc-for-machine-learning-d11f68c2815>
- Install TensorFlow with GPU Support on Windows 10 (without a full CUDA install): <https://www.pugetsystems.com/labs/hpc/Install-TensorFlow-with-GPU-Support-on-Windows-10-without-a-full-CUDA-install-1172/>
- Ubuntu 18.04: Install TensorFlow and Keras for Deep Learning: <https://www.pyimagesearch.com/2019/01/30/ubuntu-18-04-install-tensorflow-and-keras-for-deep-learning/>
- Tensorflow GPU Installation Made Easy: Use conda instead of pip: <https://towardsdatascience.com/tensorflow-gpu-installation-made-easy-use-conda-instead-of-pip-52e5249374bc>