

Tutoriales: Machine Learning

Profesor: Lihki Rubio

Departamento de Matemáticas y Estadística

Universidad del Norte de Barranquilla

1. **Instalación de Miniconda:**
https://www.youtube.com/watch?v=oHHbsMfyNR4&ab_channel=OpenSourceOptions
2. **Instalación de VS Code:**
https://www.youtube.com/watch?v=CPmQwlycfGI&ab_channel=AmitThinks
3. **Creación de Entornos Virtuales Miniconda:**
https://www.youtube.com/watch?v=ErXtLCxsknE&ab_channel=MachineMantra
4. **Entorno virtual para Machine Learning:**
 - Después de comprender **como se crean los entornos virtuales en Miniconda**, debe crear un entorno virtual para el curso de **Machine Learning**, el cual puede nombrar como **"ml_venv"** (a manera de ejemplo), o utilizar el nombre que sea de su agrado.
 - El proceso es basado en el **ítem 3** y requiere de los siguientes comandos, los cuales deberían **ejecutar desde la consola Powershell de Miniconda**. Este entorno debe ser creado con la versión **3.9 de Python**, para que **no existan mayores conflictos con la ejecución de algoritmos utilizados en el curso**.
 - Proceso
 - `conda create --name ml_venv python=3.9`
 - `conda activate ml_venv`
 - Recuerde que puede desactivar cualquier entorno activado utilizando la siguiente orden en la consola Powershell de Miniconda
 - `conda deactivate`
 - Durante las secciones de **Redes Neuronales Artificiales** se realizarán ejemplos aplicados a **Procesamiento de Imágenes y Lenguaje Natural** usando **TensorFlow**. Este es un segundo motivo para crear este entorno con **Python 3.9**.
5. El curso ha sido creado basado en el **sistema operativo Windows**, debido a que es el **más usado por los estudiantes**. De ser solicitado, también serán suministrados, **tutoriales adecuados, para aquellos estudiantes que usen Linux o macOS**.
6. Desafortunadamente ninguna laptop **MacOS** cuenta con tarjetas gráficas de alto rendimiento del tipo **NVIDIA RTX, GTX**. Además, **TensorFlow** desde sus inicios **no ha ofrecido soporte para MacOS**.

Install TensorFlow with pip

This guide is for the latest stable version of TensorFlow. For the preview build (*nightly*), use the pip package named `tf-nightly`. Refer to [these tables](#) for older TensorFlow version requirements. For the CPU-only build, use the pip package named `tensorflow-cpu`.

Here are the quick versions of the install commands. Scroll down for the step-by-step instructions.

Linux **MacOS** Windows Native Windows WSL2 CPU Nightly

```
# There is currently no official GPU support for MacOS.
python3 -m pip install tensorflow
# Verify install:
python3 -c "import tensorflow as tf; print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

- Por otra parte, los **computadores gamers (del tipo: ROG, Alienware, MSI, Legion, Omen, Nitro)** actuales con **Windows o Linux**, si cuentan con estas tarjetas, las cuales son de suma importancia, para el **entrenamiento local de algoritmos de Deep Learning**.
- Si usted cuenta con una computadora MacOS, o un computador con Windows sin GPU dedicada, puede usar TensorFlow**, pero, el entrenamiento de modelos de Deep Learning, **solo usará los CPU cores que tenga disponible** en su máquina. Los algoritmos funcionarán, pero el tiempo de ejecución será razonablemente más elevado.
- Claramente, la versión gratuita de **Google Colab**, puede funcionar correctamente, y **ejecutar los algoritmos de Deep Learning que se estudiarán en clases**. Sin embargo, el **tiempo de ejecución puede ser 12 veces mas lento que usar la CPU de su propia computadora**. Sin embargo, pagar la **suscripción a Google Colab Pro**, puede entregar más recursos (CPUs/GPUs cores) para ejecutar algoritmos de Deep Learning.

Parameter	Google Colab
No. CPU Cores	2
CPU Family	Haswell
Available RAM	12GB (upgradable to 26.75GB)

- A manera de ejemplo comparativo, para mayor comprensión de las ventajas de las tarjetas gráficas de PC gamers, en términos de número de núcleos de trabajo, actualmente, un **MacBook en promedio cuenta con 12 CPU cores**. Por otro lado, por ejemplo, un **ASUS ROG Zephyrus G16**, por ejemplo **¡cuenta con 14 CPU Cores y una GPU (Tarjeta gráfica) RTX 4070 con 7680 Cuda cores para Deep Learning!!**. La computadora usada para el diseño de este curso cuenta con una **GTX 1070 Max-Q's 8GB GDDR5 con 1920 CUDA cores** (especificaciones: <https://es.msi.com/Laptop/GS65-Stealth-Thin-8RF/Specification>)
- Solo CPU**: Procederemos a crear un **entorno virtual para TensorFlow**. Si cuenta **solo con CPU (MacOS o Portátil con Windows/Linux (No-Gamer))**. Solo debe ejecutar, **teniendo activado el ambiente "ml_venv"** la siguiente línea de comando

- `pip install --upgrade pip`
- `pip install tensorflow`

```
python3 -c "import tensorflow as tf;
print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

Esta última orden ejecutada en la terminal debe arrojar un tensor, en el caso de que TensorFlow haya sido instalado correctamente.

12. **Solo GPU (Gamer Laptop):** Si usted cuenta con una **GPU dedicada en su Gaming Laptop**, debe sacarle todo el provecho a los **recursos con los que cuenta su máquina para Deep Learning**. Por lo tanto, para crear un **ambiente virtual de TensorFlow para su GPU** debe seguir los siguientes pasos. Antes verifique los **requerimientos que aparecen en la página de TensorFlow:** <https://www.tensorflow.org/install/pip#windows-native>

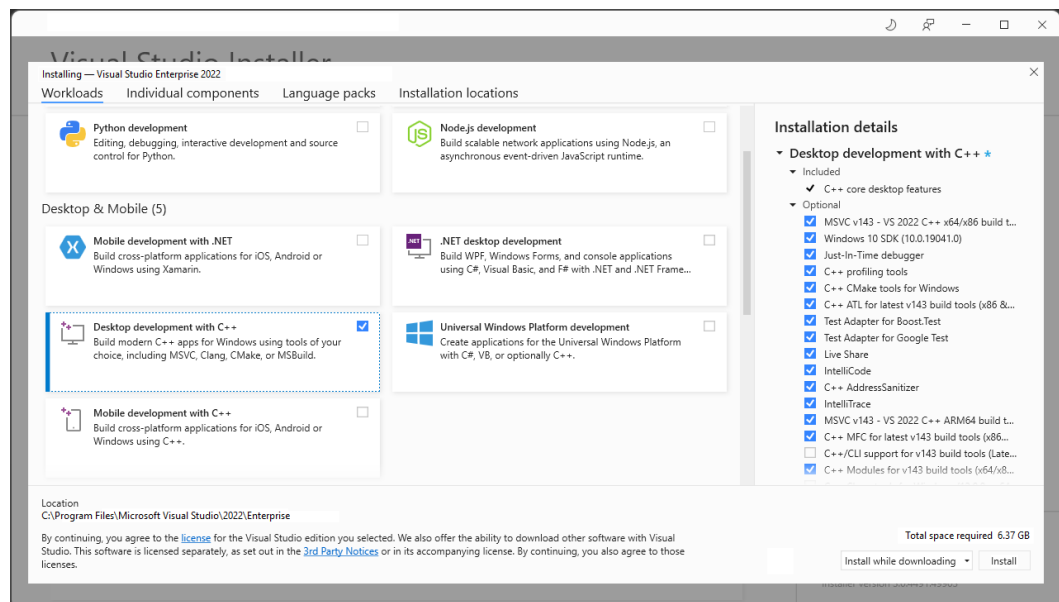
Software requirements

- Python 3.8–3.11
- pip version 19.0 or higher for Linux (requires `manylinux2014` support) and Windows. pip version 20.3 or higher for macOS.
- Windows Native Requires [Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019](#)

The following NVIDIA® software are only required for GPU support.

- [NVIDIA® GPU drivers](#) version 450.80.02 or higher.
- [CUDA® Toolkit 11.8](#).
- [cuDNN SDK 8.6.0](#).
- (Optional) [TensorRT](#) to improve latency and throughput for inference.

- Debe instalar inicialmente **Visual Studio Community 2022** (<https://visualstudio.microsoft.com/es/downloads/>). Utilice el ejecutable descargado desde la página, asegúrese de **solo seleccionar en la instalación, todo relacionado con C++ (ver imagen)**. **CUDA TensorFlow requiere de C++ para funcionar correctamente.**



- Posteriormente **debe instalar el NVIDIA GPU driver que le corresponde a su tarjeta de video**. Verifique cual es la versión de su tarjeta de video, y descargue el Driver que le corresponde en la siguiente página (<https://www.nvidia.com/Download/index.aspx>). Adjunto imagen, a manera de ejemplo, de la descarga del **driver para la tarjeta de video usada en diseño del presente curso (GTX 1070)**. Haga click en “Search” para descargar el ejecutable requerido para la instalación. Luego de instalar el Driver, debe reiniciar su computadora.

NVIDIA Driver Downloads

Select from the dropdown list below to identify the appropriate driver for your NVIDIA product. [Help](#)

Product Type:

Product Series:

Product:

Operating System:

Download Type: ?

Language:

- Luego de haber instalado **Visual Studio Community C++ y el NVIDIA Driver**, puede realizar la instalación de TensorFlow en el entorno virtual de miniconda “**ml_venv**” creado en el ítem 4. Para esto, debe primeramente activar el entorno desde Miniconda PowerShell. Siga los siguientes pasos para instalar CUDA, cuDNN y TensorFlow

```

• conda activate ml_venv
• conda install -c conda-forge cudatoolkit=11.2
  cudnn=8.1.0
• pip install --upgrade pip
• pip install "tensorflow<2.11"
• python -c "import tensorflow as tf;
  print(tf.config.list_physical_devices('GPU'))"

```

La última línea de comandos **deberá arrojarle una lista de dispositivos GPU si la instalación de TensorFlow fue realizada correctamente**. En este caso, a manera de ejemplo, por tratarse de un **portátil con una sola GPU dedicada**, aparece solo una tarjeta. En el caso de los famosos **Clúster de GPUs** aparecerá una lista.

```

(base) PS C:\Users\rubio> conda activate ml_venv
(ml_tf) PS C:\Users\rubio> python -c "import tensorflow as tf; print(tf.config.list_physical_devices('GPU'))"
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
(ml_tf) PS C:\Users\rubio>

```

13. Luego de haber creado el **entorno virtual de Miniconda para TensorFlow**, debe agregarlo al editor **VS Code**. Para esto debe seguir los pasos del siguiente enlace (ver <https://code.visualstudio.com/docs/python/environments>)

14. Para que los algoritmos utilizados en el presente curso funcionen correctamente, debe **instalar dentro del entorno virtual “ml_venv”** el siguiente **archivo de requerimientos, nombrado (requirements.txt)**. Siga los siguientes pasos

- `conda activate ml_venv`
- `pip install -r requirements.txt`

Si por algún motivo, alguna librería no es instalada, **puede eliminar el nombre de esta librería del archivo “requirements.txt” y seguir con la instalación mediante el uso de “pip install -r requirements.txt”**. Descargue el archivo “requirements.txt” del siguiente link: https://github.com/lihkir/Data/blob/main/requirementsml_mathstatun/requirements202330/requirements.txt

15. Seguir el siguiente tutorial para configurar un **virtual environment para crear informes usando Jupyter-Book**: https://www.youtube.com/watch?v=IZ2FHTkyaMU&ab_channel=ChrisHoldgraf
Antes de proceder con los pasos de **creación y publicación de informes por medio de Jupyter Book**, debe instalar en su maquina el **cliente de git** <https://gitforwindows.org/> y **GitHub Desktop** <https://desktop.github.com/> para facilitar la creación de repositorios. Desde GitHub Desktop debe **crear un repositorio para su informe**, en este caso este será nombrado: **jbook_ml202330**. Los siguientes pasos deben tenerse en cuenta. Cada orden debe ser ejecutada desde la consola Powershell de Miniconda:

- `mkdir jbook_ml202330`
- `cd jbook_ml202330`
- `conda activate ml_venv`
- `pip install -U jupyter-book`
- `jupyter-book create docs`
- `cd docs`
- `code notebook`
- `pip freeze > requirements.txt`
- `cd ..`
- `jupyter-book build docs/`
- `python3 -m pip install -U ghp-import`
- `git init`
- `git remote add origin https://github.com/lihkir/jbook_ml202330.git`
- `git checkout -b main`
- `git pull https://github.com/lihkir/ jbook_ml202330 main`
- `ghp-import -n -p -c thelinkofmygithubrepo.com -f directoryroot/docs/_build/html`
- [End: GitHub Pages/Check it out here!](#)

16. Utilizar **Lipmap** (<https://www.litmaps.com/>), **Tree of Science** (<https://tos.coreofscience.com/>) y **Web of Science** (<https://www.webofscience.com/wos/woscc/basic-search>) para realizar una **revisión de literatura** relacionada con un **tema de su interés** en el cual se haga uso de **Machine Learning**. Debe **proponer por lo menos dos artículos, en los que se propongan modelos mejorados de Machine Learning, los cuales no se han utilizado en la actualidad para resolver su**

problema de aplicación de interés. Debe implementar la metodología del artículo propuesto, para solucionar el problema de aplicación de su interés.

Machine Learning Model for MyInterestApplication	Improved Machine Learning Model for AnyOtherApplication	Improved Machine Learning Model for AnyOtherApplication
Model A1 (Paper A1)	Model B1 (Paper B1)	Model B1 (Paper B1) Not Listed At Models AJ (Paper AJ), J=1,2,...,10
Model A2 (Paper A2)	Model B2 (Paper B2)	Model B3 (Paper B3) Not Listed At Models AJ (Paper AJ), J=1,2,...,10
Model A3 (Paper A3)	Model B3 (Paper B3)	Model B5 (Paper B5) Not Listed At Models AJ (Paper AJ), J=1,2,...,10
Model A4 (Paper A4)	Model B4 (Paper B4)	Model B9 (Paper B9) Not Listed At Models AJ (Paper AJ), J=1,2,...,10
Model A5 (Paper A5)	Model B5 (Paper B5)	
Model A6 (Paper A6)	Model B6 (Paper B6)	
Model A7 (Paper A7)	Model B7 (Paper B7)	
Model A8 (Paper A8)	Model B8 (Paper B8)	
Model A9 (Paper A9)	Model B9 (Paper B9)	
Model A10 (Paper A10)	Model B10 (Paper B10)	

17. Deberá presentar un **informe en Jupyter-Book** (**Atención: Solo el link de la GitHub page creada mediante Jupyter-Book será recibido**). En el informe solicitado **debe agregar las siguientes secciones:**

- **Título:** Ser atractivo, de modo que describa el contenido del **informe** en forma específica, clara, exacta, breve, y concisa. Posibilitar que el lector identifique el tema con facilidad. Permitir una indización precisa del material.
- **Resumen:** Contexto, Propósito, Metodología, Resultados, y Conclusión. Una lista de Palabras-clave escogidas por él(los) autor(es) se coloca al final del resumen, precediendo el cuerpo del informe. Máximo 250 palabras.
- **Metodología:** La metodología debe contener lo siguiente:
 - Población y muestra (tiempo y espacio como selección de la muestra).
 - Variables (diccionario de características)
 - Técnicas (modelos de ML)
 - Diseño General (tipo de estudio).
 - Validez
 - Grupos de datos coherentes.
 - Automatizado: sistema utilizado para la obtención de datos (CSV, API, SQL query, etc,...).
 - Técnicas: Chi cuadrado, T-Students, Prueba Fisher, Autocorrelación, etc,....
- **Resultados y Discusión:** Describa de forma concisa y precisa los resultados experimentales, su interpretación, así como las conclusiones experimentales que pueden extraerse. Los resultados y sus implicaciones deben debatirse en el contexto más amplio posible y deben destacarse las limitaciones del trabajo. También pueden mencionarse futuras líneas de investigación.

18. La estructura de **selección de tema de investigación para informe final 40%** se presenta en la tabla de arriba.
19. La **sustentación del informe tendrá una valoración del 10% del curso** y será realizada en frente de **público experto e inexperto en temas de Machine Learning (history telling training)**.
20. **El 50% del curso corresponde a un parcial escrito** en el cual se evaluará **la teoría matemática/estadística asociada a los modelos de Machine Learning** estudiados durante el curso.