

Práctica 6.

Introducción a Keras.

Deep learning

*Máster en Big Data*

*Curso 2019-2020*

***Data Science & Advanced Analytics***

*Montse Llos i Bombardó*

*Keras*

Keras es una librería de Python (también en R) que proporciona de manera limpia y sencilla la creación de una gama de modelos de Deep Learning encima de otras librerías TensorFlow, Theano o CNTK. Keras fue desarrollado y es mantenido por [François Chollet](https://twitter.com/fchollet), un ingeniero de Google. Su código ha sido liberado bajo la licencia permisiva del MIT.

Más información en <https://torres.ai/ca/primeros-pasos-en-keras/>

*Instalación de Keras en Python*

Depende de qué versión de Python tengamos instalado deberemos instalar la librería de Keras adecuada.

Aquí encontrarás más información sobre versiones de Keras. Ahora con Tensorflow es sufciente.

<https://www.linkedin.com/pulse/keras-y-tensorflow-deep-learning-en-estado-puro-montse-llos/?trackingId=1UXMGpW0XY2ymk%2BZ48pMew%3D%3D>

Para instalar la versión 2.3 de Keras en nuestro ordenador, necesitamos tener la versión 3.7 de Python en 64 bits (no soportable en máquinas de 32 bits).

python –version

Debemos saber si nuestro ordenador tiene alguna tarjeta GPU ya que la instalación difiere.

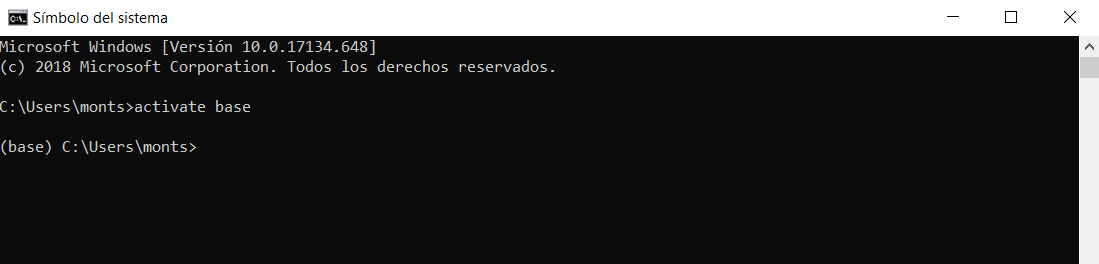
**Instalación manual**

Para instalar Keras, debemos entrar en entorno vía command prompt

activate <nombre entorno>

(si tenemos un único entorno, el de defecto es base)

Así abrimos el entorno de conda



y ejecutar

conda install tensorflow=2.0

Para cerrar entorno de conda

deactivate

Más información en: <https://www.tensorflow.org/install/pip>

En el campus encontraréis un cheat sheet de instrucciones con Conda.

**Instalación desde Anaconda**

También podemos entrar al entorno desde Anaconda Prompt o desde Anaconda Navigator activando el entorno donde queremos instalar Tensorflow.

**Instalación semi manual**

Mediante los scripts que nos podemos bajar del campus según tengamos GPU o no en nuestro ordenador.

conda env create -v -f tensorflow.yml

Para ver los pasos ir a: <https://github.com/jeffheaton/t81_558_deep_learning/blob/master/t81_558_class_01_1_overview.ipynb>

**Enlazar con Jupyter Notebook**

Debemos enlazar el nuevo entorno creado (en este caso llamado tensorflow) con Jupyter Notebook para tomar este entorno como kernel de Jupyter

python -m ipykernel install --user --name tensorflow --display-name "Python 3.7 (tensorflow)"

**Desde Google Colab (opción más fácil)**

Si trabajamos sobre el entorno cloud de Google (<https://colab.research.google.com/>) no es necesario instalar nada

*Práctica 6.0*

Vamos a comprobar que Tensorflow está bien instalado

*# What version of Python do you have?*

**import** **sys**

**import** **tensorflow.keras**

**import** **pandas** **as** **pd**

**import** **sklearn** **as** **sk**

**import** **tensorflow** **as** **tf**

print(f"Tensor Flow Version: **{tf.\_\_version\_\_}**")

print(f"Keras Version: **{tensorflow.keras.\_\_version\_\_}**")

print()

print(f"Python **{sys.version}**")

print(f"Pandas **{pd.\_\_version\_\_}**")

print(f"Scikit-Learn **{sk.\_\_version\_\_}**")

print("GPU is", "available" **if** tf.test.is\_gpu\_available() **else** "NOT AVAILABLE")

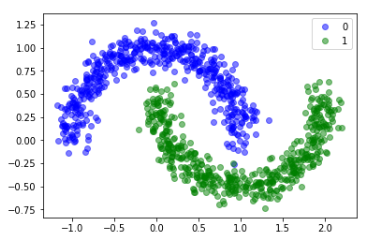
*Práctica 6.00*

Titanic again!

A partir de la práctica de Titanic resuelta en el Tema 4 de Introducción a Python mediante una regresión logística, vamos a cambiar el entrenamiento por una red diseñada con Keras.

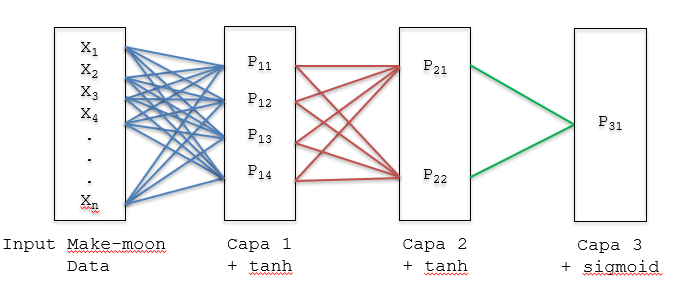
*Práctica 6.1*

Vamos a resolver con Keras un problema de clasificación de dos grupos conocidos en el mundo de Machine Learning: Make Moon dataset



Realizaremos una red neuronal sencilla, con una sola capa, que aprenda de estos datos usando las APIs de Keras.

Posteriormente montaremos nosotros una con la siguiente estructura



Evalúa el rendimiento de la nueva estructura versus la anterior

*Ejercicio 6.2*

Vamos a aprender las características que determinan que una diabetes se pueda manifestar para así poder predecir en futuros pacientes si tienen más disposición a padecer esta enfermedad

Los datos históricos que disponemos son los siguientes (diabetes.csv)

* Pregnancies:
  + Número de veces que ha estado embarazada
* Glucose:
  + Concentración de glucosa en plasma a 2 horas en una prueba oral de tolerancia a la glucosa
* BloodPressure:
  + Presión arterial diastólica (mm Hg)
* SkinThickness:
  + Grosor del pliegue de la piel del tríceps (mm)
* Insulin
  + insulina sérica de 2 horas (mu U/ml)
* IBMI:
  + Índice de masa corporal (peso en kg/(altura en m)^2)
* DiabetesPedigreeFunction:
  + Función de pedigrí de la diabetes
* Age:
  + Edad (años)

Y el campo Outcome será el campo a predecir, binario que indica si el paciente manifiesta diabetes o no.

Primero exploraremos los datos.

¿Deberíamos aplicar una normalización? ¿De qué tipo?

Prepara los datos en X e y para usar Deep Learning como método de aprendizaje

Monta una estructura en Keras para el aprendizaje de estos datos y evalúa el resultado.

El ejercicio se debe realizar en Python. Recuerda de comentar todos y cada uno de los pasos realizados.

Número límite de personas: 5

Entregables:

* Un fichero con el código Python
  + Nombre del fichero: <Apellidos+InicalNombres>\_Ejercicio\_6.2.py o .ipynb
  + Ejemplo si sois dos: *LLosM\_RooneyM\_Ejercicio\_6.2.ipynb*
* Un documento Pdf con nombre y foto de la persona que ha realizado el ejercicio y conclusiones del ejercicio
  + Nombre del fichero: <Apellidos+Nombres>\_Ejercicio\_6.2.pdf
  + Ejemplo si sois dos: *LLosM\_RooneyM\_Ejercicio\_6.2.pdf*

Fecha límite entrega: 23 marzo 2020 a las 24h

Subir al campus