Regresión Lineal. Introducción a Keras.

**Requisitos**

Tener una [instalación de python](http://facundoq.github.io/guides/python_install) con las librerías **keras**, **jupyter** y **pandas**. No utilizar python 3.8 porque tiene problemas con Keras; 3.7 y 3.6 si funcionan.

**Objetivos**

Estos ejercicios tienen como objetivo que te familiarices con la librería Keras implementando modelos simples de Regresión Lineal.

Para comenzar, en los dos primeros ejercicios, no utilizaremos Keras, pero ofrecemos una implementación muy simple y didáctica de regresión lineal con una variable de entrada (ejercicio 1) o múltiples (ejercicio 2). Si bien ya proveemos la implementación del descenso de gradiente y el cálculo de derivadas, recomendamos estudiar brevemente las mismas.

En el ejercicio 3, haremos exactamente lo mismo que en el 2 pero utilizando Keras. El objetivo de esta progresión es comprender mejor cómo funciona Keras internamente.

**Preparación**

Desde una terminal con python en el path, ubicarse en el directorio de la práctica donde se encuentran los archivos con extensión “.ipynb” (notebooks) y escribir: “jupyter notebook”. Desde el entorno web, acceder a la dirección del servidor (generalmente <http://localhost:8888/>) para abrir los archivos. Opcionalmente, si lanzás jupyter notebook desde Anaconda Navigator, copiá los archivos de la práctica a la carpeta raíz del notebook.

**Ejercicio 0 ( repaso)** En este ejercicio deberás implementar la función de *predicción* o función *forward* de un modelo de regresión lineal con 1 variable de entrada y 1 de salida. Verifica que la implementación sea correcta con los casos de prueba, y luego que el modelo entrene correctamente.

*Archivo*: Regresion Lineal Univariada.ipynb

Igual que el ejercicio 1, pero con un modelo con múltiples variables de entrada y solo una de salida.

*Archivo*: Regresion Lineal.ipynb

**Ejercicio 1** El código de este ejercicio reimplementa el modelo del ejercicio 2 pero con la librería Keras. Si bien no hay nada que implementar, ejecutá el código y comparalo con el anterior. Probá también distintas combinaciones de la tasa de aprendizaje (lr), la cantidad de iteraciones (epochs) y el tamaño del batch (batch\_size) para ver su efecto en el aprendizaje.

*Archivo*: Regresión Lineal con Keras.ipynb

**Ejercicio 2 (opcional)** Utilizando datos de países obtenidos de la [OMS](https://www.who.int/es), vamos a predecir la esperanza de vida promedio de sus habitantes. Dichos datos se encuentran en el archivo **who\_life\_expectancy.csv**. Podés encontrar más información sobre el conjunto de datos en su [sitio de origen](https://www.kaggle.com/kumarajarshi/life-expectancy-who).

1. Preprocesamiento: Quitar la columna “País” para simplificar el análisis. Además, hay otra columna del conjunto de datos no es numérica. Convertirla a numérica, ya sea utilizando un script de python o utilizando un programa de planilla de cálculo.
2. Normalizar los valores de las variables numéricas.
3. Escribí un modelo de Regresión Lineal en Keras para predecir la expectativa de vida. Para ayudarte, podés hacer una copia del notebook del ejercicio 3, y adaptarlo para cargar el archivo **who\_life\_expectancy.csv**.
4. Una vez entrenado el modelo, indicá cuál es su error MSE (Mean Squared Error o Error Cuadrático Medio). Interpretarlo en términos de la expectativa de vida ¿qué tan bien predice el modelo?
5. Podemos analizar qué variables utiliza el modelo para realizar la predicción en base a los valores de sus pesos. Analizá los pesos del modelo para determinar cuáles son las variables correspondientes que más impacto tienen en la predicción. Poniéndose en lugar de un estadista, ¿cuáles de estas variables es más importante influenciar si se busca aumentar la esperanza de vida?
6. Podría argumentarse que algunas variables deberían quitarse, debido a que tienen una relación muy directa con la variable a predecir. En particular, la mortalidad adulta e infantil están demasiado relacionadas con la expectativa de vida, y entonces el modelo no aprende algo útil, ya que no son variables que un estadista pueda cambiar directamente. Quitá esas variables del conjunto de datos y repetí el análisis. ¿Cuáles son las variables relevantes ahora?

Código para imprimir los pesos, asumiendo que la variable **model** tiene el modelo:

for layer in model.layers:

print(layer.get\_config(), layer.get\_weights())