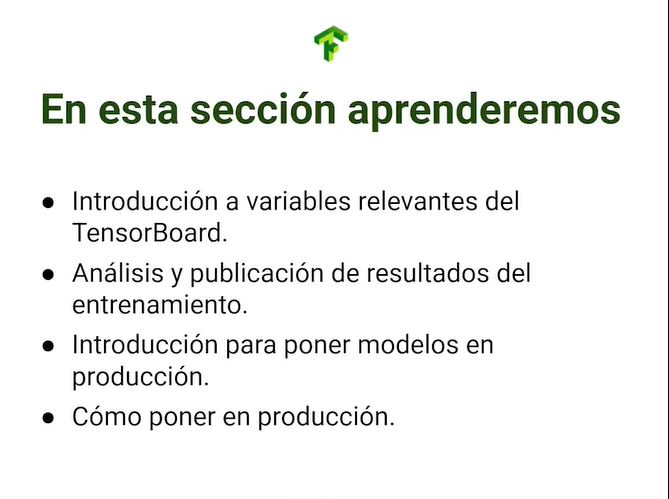
A través de los módulos hemos explorado gran parte del ciclo de vida de los algoritmos de deep learning, desde la carga y limpieza de datos hasta la creación y optimización de los mismos, culminando con la implementación de modelos preentrenados desde Keras Applications y TensorHub.

Ahora indagaremos sobre TensorBoard, una herramienta que nos permitirá publicar los resultados de nuestros modelos a la comunidad.

A través de las siguientes sesiones, comprenderemos qué es Tensorboard, sus variables relevantes, cómo analizar y publicar nuestros resultados, además de algunas nociones para escalar nuestros modelos a producción.



**Introducción a TensorBoard**

TensorBoard es una herramienta de TensorFlow creada para la visualización de resultados, nos permite ver de manera gráfica la composición, estructura, variables relevantes y demás insights de nuestro modelo y nos ofrece la oportunidad de debuggearlo para obtener las mejores configuraciones posibles.

Importaremos el callback de TensorBoard y el módulo nativo de time de Python.

from tensorflow.python.keras.callbacks import TensorBoard

from **time** import **time**

Generaremos un modelo con nuestra función generadora genérica y lo compilaremos con la configuración usual.

model\_tensorboard = get\_model()

model\_tensorboard.compile(optimizer = "adam", loss = "categorical\_crossentropy", metrics = ["accuracy"])

Antes de entrenar el modelo crearemos el callback de TensorBoard al que le especificaremos la ruta donde deberá guardar los registros.

callback\_tensorboard = TensorBoard(log\_dir = f"logs/{time()}")

Entrenaremos el modelo agregando el TensorBoard como callback.

history\_tensorboard = model\_tensorboard.fit(

train\_generator,

epochs = 20,

callbacks = [callback\_tensorboard],

validation\_data = validation\_generator

)

Con esto tenemos todos los ingredientes necesarios para explorar nuestros resultados.

**Contribución creada por** Sebastián Franco Gómez.