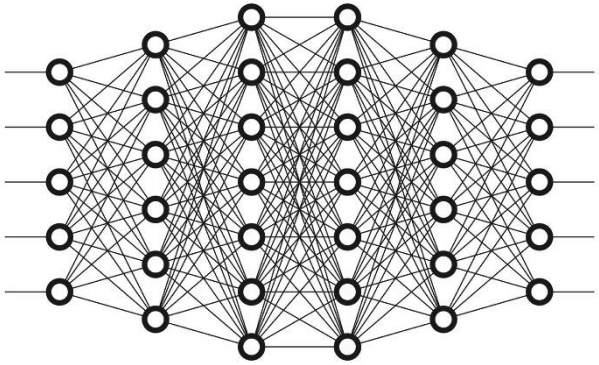
Deep Learning aplicado al análisis de señales e imágenes

INTRODUCCIÓN A TENSORFLOW

¿Qué es TensorFlow?

- **TensorFlow (TF)** es un framework desarrollado y mantenido por **Google** que permite la **ejecución** de **operaciones matemáticas**, mediante diagramas de flujo de datos, de una forma optimizada en una **CPU o GPU**.

- En nuestro caso **estamos más interesados en** la **GPU**, ya que es la única forma que tenemos de **entrenar una red neuronal** en un intervalo de **tiempo permisible**.

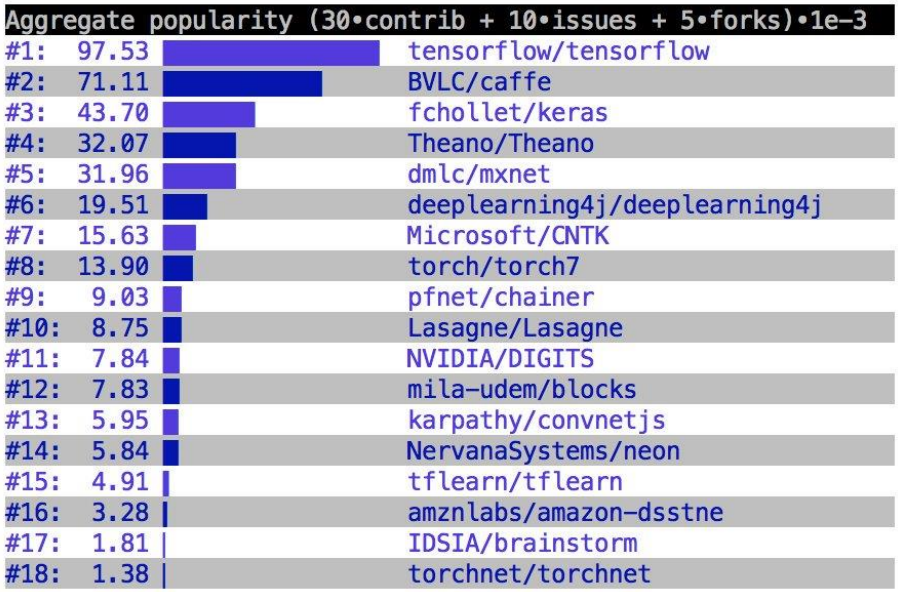


TensorFlow

¿Por qué TensorFlow?

- Necesidad de **código optimizado** y **ampliamente validado** (por ejem. para el backpropagation). Debido a estos dos aspectos, cuando se desean entrenar redes neuronales se hace uso de **librerías de más alto nivel**.

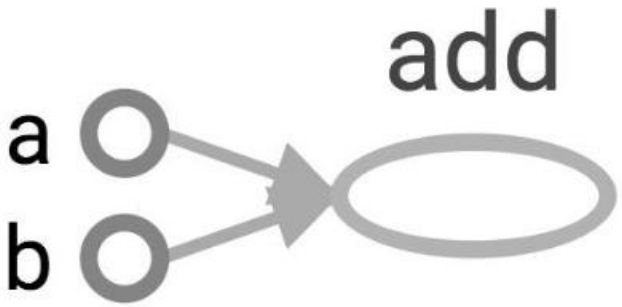
- TensorFlow es **flexible**, **escalable** y su amplia popularidad hace que sea una librería que esta **en constante actualización** y **mantenimiento**.



Características de TensorFlow

- TensorFlow utiliza **tensores** para **realizar** las **operaciones**. Los tensores no son más que **contenedores** de **datos**.

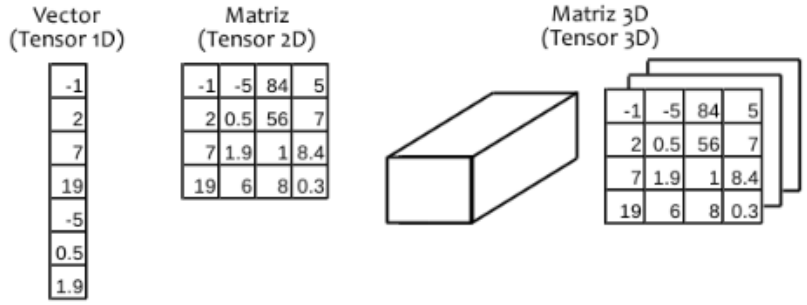
- En TensorFlow, **primero se definen las operaciones** a realizar (construimos el grafo), **y luego se ejecutan** (se ejecuta el grafo).



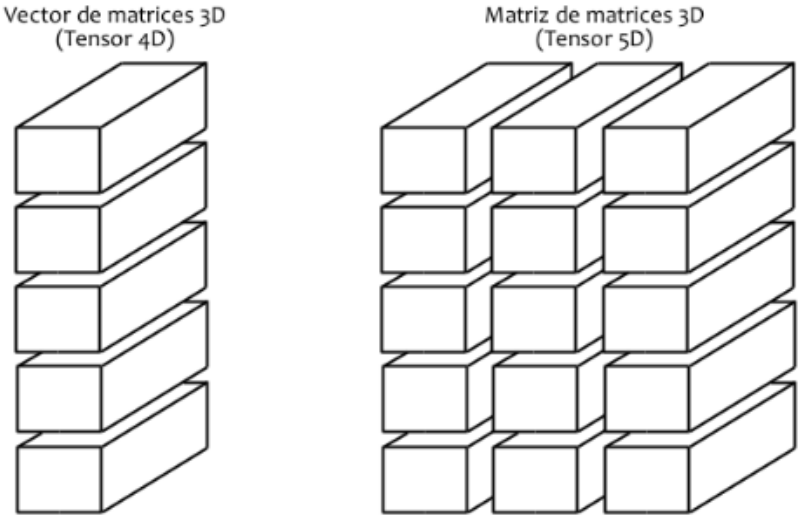
- Permite ejecutar el código implementado **paralelamente** en **una** o **varias GPUs**, a elección del usuario.

Tensores

- En el ámbito de la inteligencia artificial, los **tensores** se pueden entender simplemente como **contenedores de números**.



Tensores 3D: utilizados en **series temporales**
Tensores 4D: utilizados con **imágenes**
Tensores 5D: utilizados con **videos**

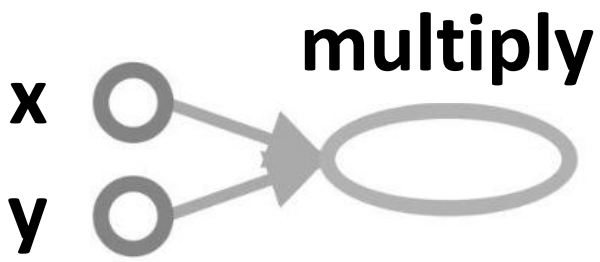


- Si queremos almacenar **64 imágenes RGB** de **224x224 píxels**, necesitaremos un vector de matrices 3D, o lo que es lo mismo, un tensor 4D. ¿**Dimensiones del tensor**?

(64, 224, 224, 3)

Nuestro primer grafo

- Los **nodos** en el **grafo** representan **operaciones matemáticas**, mientras que las **conexiones o links del grafo** representan los **conjuntos de datos multidimensionales** (**tensores**).



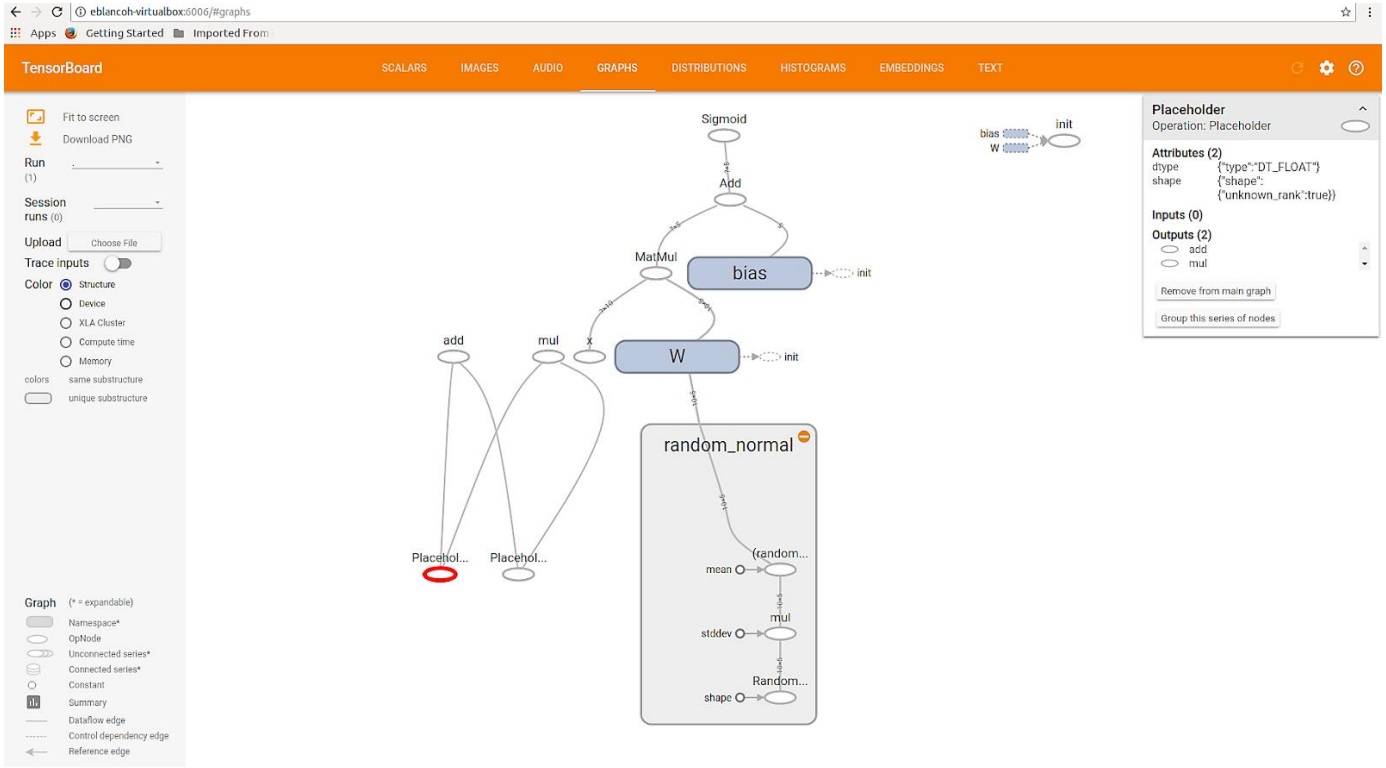

```
import tensorflow as tf

# Definición del grafo
x = tf.constant(6)
y = tf.constant(8)
result = tf.multiply(x, y)

# Ejecución del grafo
sess = tf.Session()
output = sess.run(result)
print(output)
>>>> 48
```

Grafos complejos

- Lógicamente cuando se desarrolla un **algoritmo** el **grafo resultante suele ser bastante complejo**. Si se desea ver dicho grafo se puede emplear la herramienta **tensorBoard**.



Variable vs Placeholder

- Las **variables** son **definidas e inicializadas con un tensor de determinas dimensiones (*static shape*)**.

- El **contenido de las mismas puede ir actualizándose en tiempo de ejecución**, por ejemplo, en el proceso de actualización de los pesos de una red neuronal

- Los **placeholders** no son más que **reservas de espacios de memoria**. Dichas reservas de memoria se pueden realizar definiendo las dimensiones del tensor que contendrá el placeholder o sin definirlas, puesto que se trata de **reserva de memoria dinámica (*dinamic shape*)**.

- El **contenido de un placeholder** se **rellena en tiempo de ejecución** (i.e. cuando ejecutamos **tf.Session.run()**).

- Se utiliza mayoritariamente **para alimentar al grafo con datos de entrada**. En un placeholder almacenariamos el dataset para entrenar una red neuronal.

- A diferencia de las variables, los **placeholders no pueden modificar** su **contenido en tiempo de ejecución**.

Placeholder

```
import tensorflow as tf

# Definición grafo 1
x1 = tf.placeholder(tf.int32, shape=[4])
print (x1.get_shape())
>>>>(4, )

# Definición grafo 2
x2 = tf.placeholder(tf.int32)
print (x2.get_shape())
>>>>(?, )
```

Placeholder

```
import tensorflow as tf

# Definición grafo
x = tf.placeholder(tf.int32, shape=[4])
print (x.get_shape())
>>>>(4, )
y, _ = elimina2elementos(x)
print (y.get_shape())
>>>>(?, )
```

Placeholder

```
import tensorflow as tf

# Definición grafo
x = tf.placeholder(tf.int32, shape=[4])
print (x.get_shape())
>>>>(4, )
y, _ = elimina2elementos(x)
print (y.get_shape())
>>>>(?, )

# Ejecución del grafo
sess = tf.Session()
print (sess.run(y, feed_dict={x: [0, 1, 1, 3]}).shape)
sess.close()
>>>>(2, )
```

Optimización empleando SGD

```
import tensorflow as tf

# Definición grafo
# Variables
x = tf.Variable(value, dtype=tf.float32)
y = f(x)
# Instanciamos el optimizador pasandole como parámetro el
learning rate
opt = tf.train.GradientDescentOptimizer(lr)
train = opt.minimize(y) # Minimizamos función
init = tf.initializers.global_variables() # Inicializamos
las variables

# Ejecutamos los grafos
with tf.Session() as session:
    session.run(init)
    for step in range(10):
      session.run(train)
```

TensorFlow 2.0

- Nueva *release* de TensorFlow en **octubre de 2019**.

- El cambio más notorio es la **desaparición** del concepto de **creación/ejecución** del **grafo.**

- Para ello se introduce el concepto de ***Eager execution***. Las instrucción se evalúan inmediatamente sin necesidad de crear primero el grafo y tener que ejecutarlo después.

- Únicamente se requiere de la función ***GradientTape*** para almacenar la información del *fordward pass* y computo del error para poder retropropagarlo.

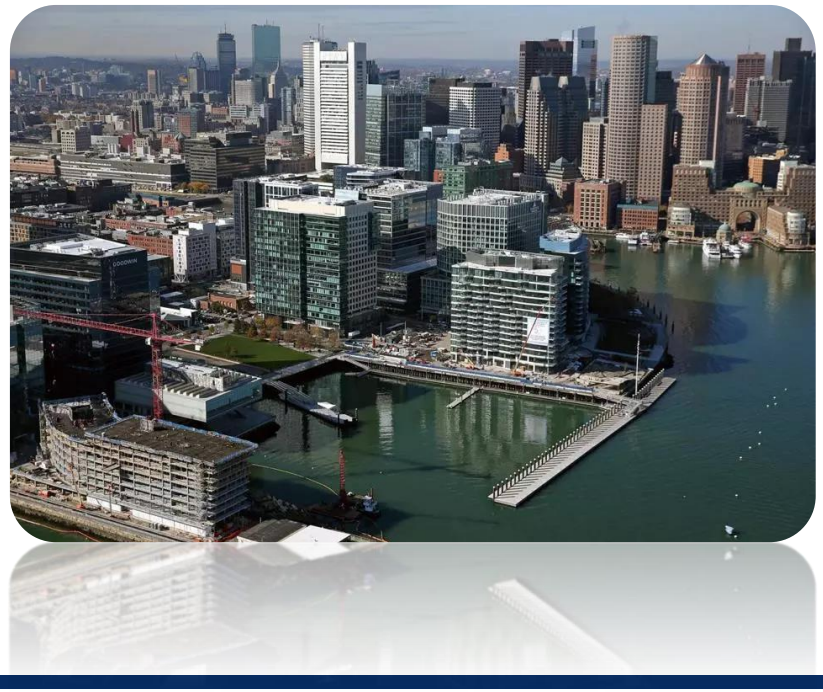
- Se pierden las nociones del **bajo nivel** (comprensión del funcionamiento de la red neuronal por dentro).

- Mejora en el **entrenamiento distribuido** entre varias GPUs y en el **entrenamiento** mediante **TPU**.

Boston dataset

- El **dataset Boston** es un set de datos utilizado para **predecir** el **valor** de una **vivienda** en Boston a partir de muestras pasadas.

- El conjunto de datos esta compuesto por **506 muestras** cada una de ellas definida por **13 características más el target** (valor de la vivienda).



Boston dataset

Nombre	Descripción
CRIM	Ratio crimen por capita del municipio.
ZN	Proporción de área de tierra residencial con más de 25,000 pies cuadrados.
INDUS	Proporción de negocios no minoristas por ciudad.
CHAS	Variable del río Charles (= 1 si el trecho delimita el río; 0 en caso contrario).
NOX	Concentración de óxidos de nitrógeno (partes por 10 millones).
RM	Número medio de habitaciones por vivienda.
AGE	Proporción de viviendas ocupadas por sus propietarios construidas antes de 1940.
DIS	Media ponderada de distancias a cinco centros de empleo de Boston.
RAD	Indice de accesibilidad a autopistas radiales.
TAX	Tasa de impuesto a la propiedad de valor total por $10,000.
PTRATIO	Proporción alumno-profesor por municipio.
B	1000 (Bk - 0,63) ^ 2 donde Bk es la proporción de personas negras por municipio.
LSTAT	Menor estado de la población (porcentaje).
MEDV	Valor medio de las viviendas ocupadas por sus propietarios en $ 1000.

Deep Learning aplicado al análisis de señales e imágenes

INTRODUCCIÓN A TENSORFLOW