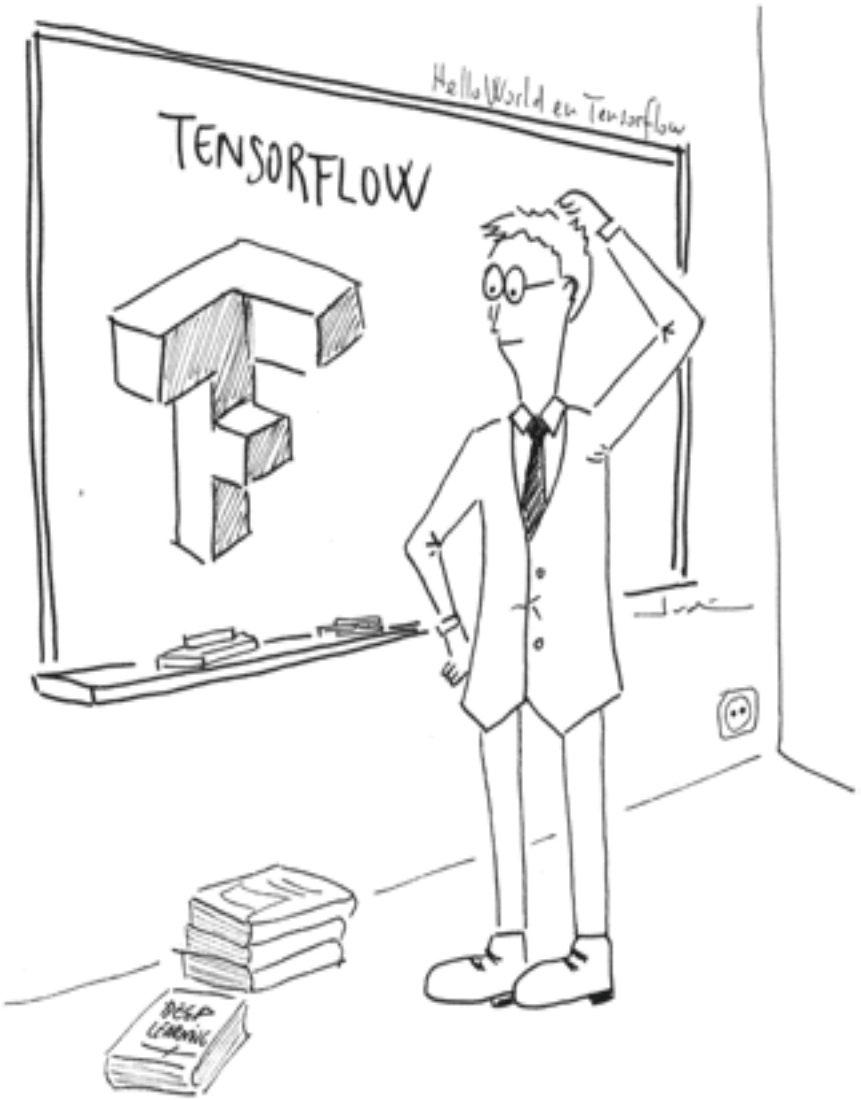
Hello World en TensorFlow

Patricia Becerra

8/03/2017

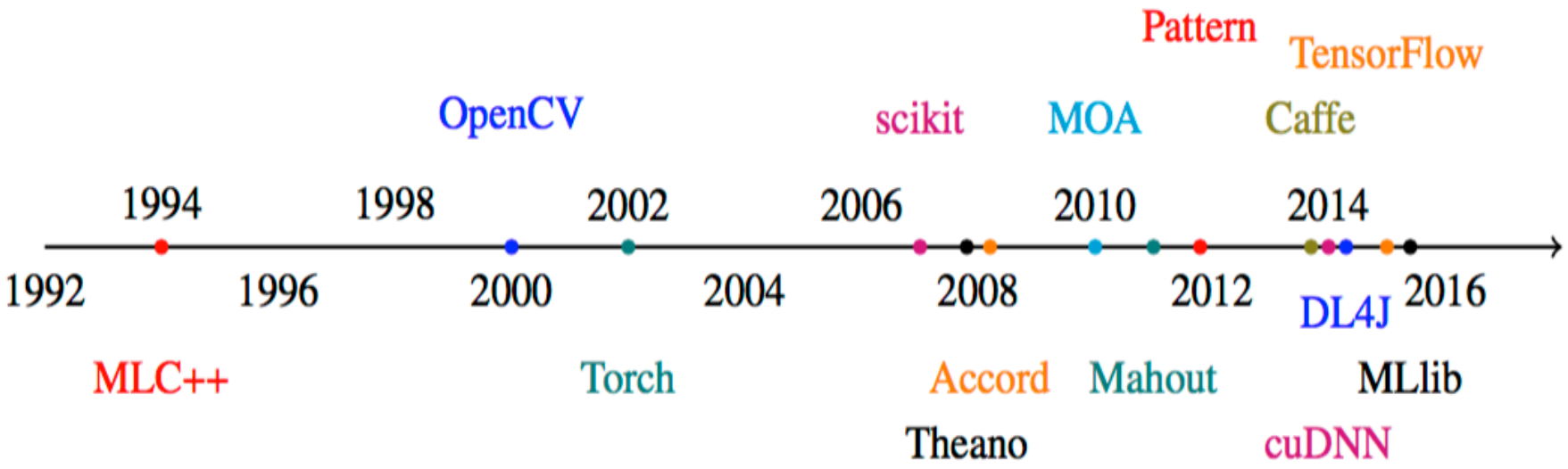


Que es TensorFlow

- Librería de aprendizaje automático (Machine Learning)
- Rápida, flexible y escalable.
- Licencia Apache 2.0.
- Código abierto de Google

TensorFlow

Machine Learning Libraries



A multidimensional array.

A graph of operations.

TensorFlow

Se ejecuta en

- CPUs , GPUs y TPU

Plataformas

- Linux de 64 bits , Mac OS X y Windows

Plataformas móviles

- Android o iOS.

TensorFlow

Para qué sirve TensorFlow

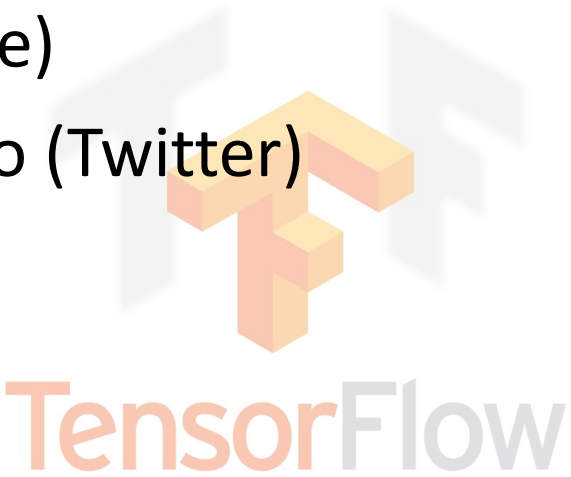
Permite construir, entrenar y ejecutar redes neuronales *Deep Learning* de manera ágil

TensorFlow

Dónde podemos ver TensorFlow?

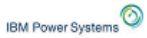
Reconocimiento de

- Voz (Google Now)
- Imágenes (Google Photos/Facebook)
- Texto (Google Translate)
- Análisis de sentimiento (Twitter)
- Spam de gmail
- Búsquedas (Google)
- Google prediction



TensorFlow

Companies using TensorFlow



TensorFlow

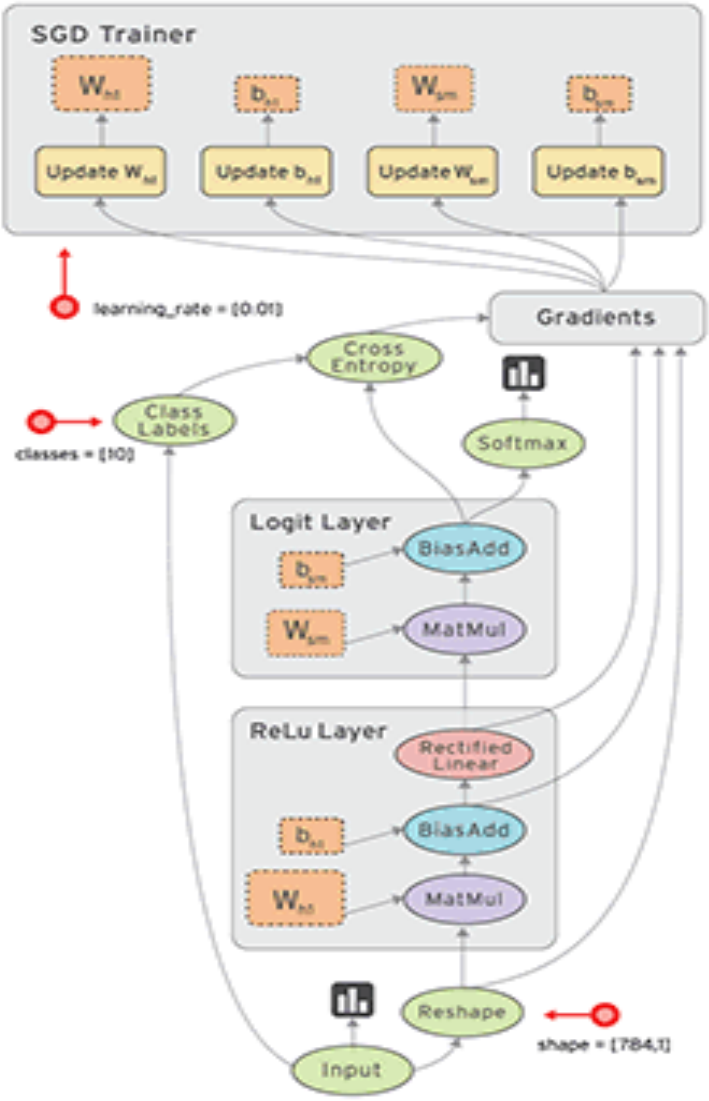
Ventajas

- Permite ejecutar la operación de entrenamiento simultáneamente en los GPUs

- Permite seleccionar una operación concreta para ejecutarla en un dispositivo determinados

- Utilizar varios GPU simultáneamente para resolver un mismo problema (construir un modelo y repartir el trabajo entre varias GPUs)

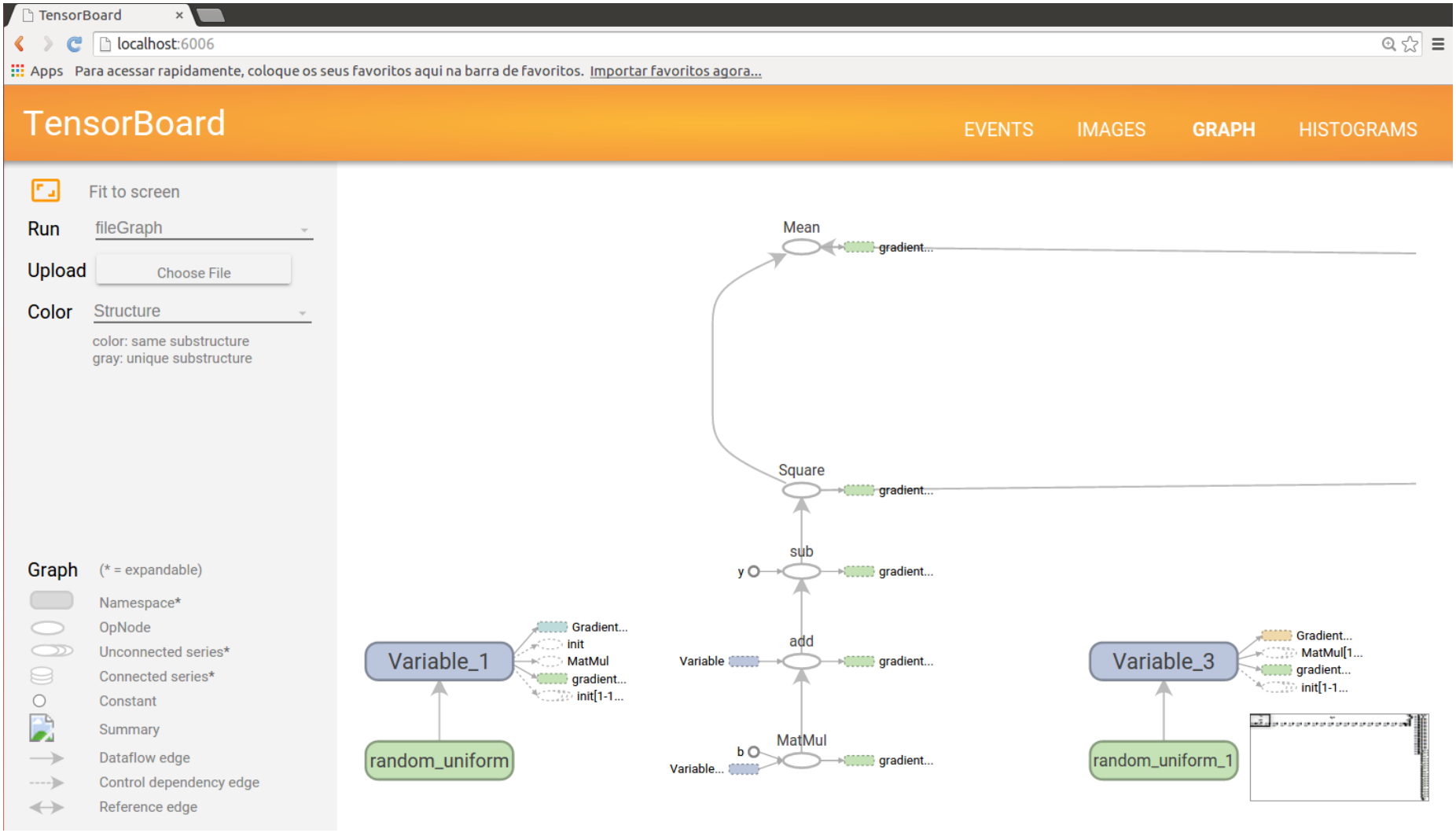
TensorFlow Computation graph



TensorFlow

TensorBoard

Visualizador de la información que se generan durante la ejecución de TensorFlow en tiempo real


TensorBoard
localhost:6006
Apps Para acessar rapidamente, coloque os seus favoritos aqui na barra de favoritos. Importar favoritos agora...
TensorBoard
EVENTS IMAGES GRAPH HISTOGRAMS
Fit to screen
Run fileGraph
Upload Choose File
Color Structure
color: same substructure
gray: unique substructure
Graph (* = expandable)
Namespace*
OpNode
Unconnected series*
Connected series*
Constant
Summary
Dataflow edge
Control dependency edge
Reference edge
Mean
gradient...
Square
gradient...
sub
y
gradient...
add
Variable
gradient...
MatMul
b
Variable...
gradient...
Variable_1
Gradient...
init
MatMul
gradient...
init[1-1...
random_uniform
Variable_3
Gradient...
MatMul[1...
gradient...
init[1-1...
random_uniform_1

TensorFlow

Instalación de TensorFlow

API Documentation

TensorFlow has APIs available in several languages both for constructing and executing a TensorFlow graph. The Python API is at present the most complete and the easiest to use, but other language APIs may be easier to integrate into projects and may offer some performance advantages in graph execution.

A word of caution: the APIs in languages other than Python are not yet covered by the API stability promises.

- Python
- C++
- Java
- Go

https://storage.googleapis.com/tensorflow/mac/tensorflow-0.5.0-py2-none-any.whl

TensorFlow

Instalación de TensorFlow

- Requiere tener Python 2.7

- Usar el entorno virtual Virtualenv para no sobreescribir versiones pre-existentes de paquetes Python requeridos por TensorFlow

TensorFlow

Instalación de TensorFlow

1.- Instalar Pip y Virtualenv :

```
# Ubuntu/Linux 64-bit
$ sudo apt-get install python-pip python-dev python-virtualenv

# Mac OS X
$ sudo easy_install pip
$ sudo pip install --upgrade virtualenv
```

TensorFlow

Instalación de TensorFlow

2.- Crear un entorno virtual Virtualenven el directorio ~/tensorflow:

```
$ virtualenv --system-site-packages ~/tensorflow
```

TensorFlow

Instalación de TensorFlow

3.- Activar el Virtualenv

```
$ source ~/tensorflow/bin/activate # si se usa bash
$ source ~/tensorflow/bin/activate.csh # si se usa csh
(tensorflow)$
```

Instalación de TensorFlow

Finalmente, cuando se ha terminado, se debe desactivar el entorno virtual con:

```
(tensorflow)$ deactivate
```

Funciones de TensorFlow

Operación matemática	Descripción
tf.add	Calcula la suma
tf.sub	Calcula la resta
tf.mul	Calcula la multiplicación
tf.div	Calcula la división
tf.mod	Retorna el módulo
tf.abs	Retorna el valor absoluto
tf.neg	Retorna el valor negativo
tf.sign	Retorna el signo
tf.inv	Retorna el inverso
tf.square	Calcula el cuadrado
tf.round	Retorna el valor entero más próximo
tf.sqrt	Calcula la raíz cuadrada
tf.pow	Calcula la potencia
tf.exp	Calcula el exponencial
tf.log	Calcula el logaritmo
tf.maximum	Retorna el máximo
tf.minimum	Retorna el mínimo
tf.cos	Calcula el coseno
tf.sin	Calcula el seno

TensorFlow

Cómo funciona TensorFlow

1. Construir el algoritmo: especificar todo las variables y parámetros
2. Crear la sesión para ejecutar los cálculos
3. Inicializar las variables
4. Ejecutar los cálculos

TensorFlow

Cómo funciona TensorFlow

```
import tensorflow as tf
x = tf.placeholder(shape=[None],
                   dtype=tf.float32,
                   name='x')
W = tf.get_variable(shape=[], name='W')
b = tf.get_variable(shape=[], name='b')
y = W * x + b

with tf.Session() as sess:

  sess.run(tf.initialize_all_variables())

  print(sess.run(y, feed_dict={x: x_in}))
```

Build the graph

Prepare execution environment

Initialize variables

Run the computation (usually often)

TensorFlow

Primer código en TensorFlow

```
import tensorflow as tf

hello = tf.constant('Hello World!')
session = tf.Session()
print session.run(hello)
```

TensorFlow

Primer código en TensorFlow

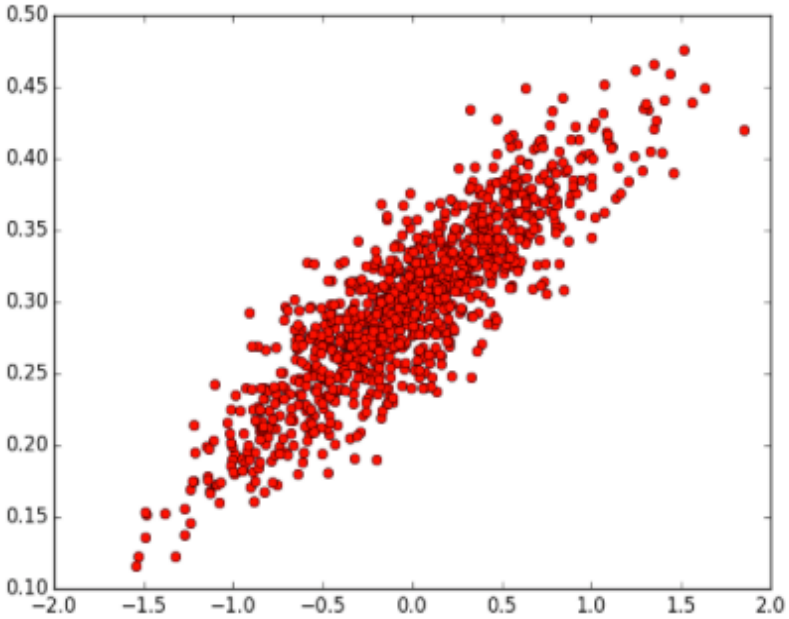
```
import tensorflow as tf

a = tf.placeholder("float")
b = tf.placeholder("float")

y = tf.mul(a, b)

sess = tf.Session()

print sess.run(y, feed_dict={a: 3, b: 3})
```

TensorFlow

TensorFlow

Regresión Lineal en TensorFlow

```
import numpy as np

num_puntos = 1000
conjunto_puntos = []
for i in xrange(num_puntos):
     x1= np.random.normal(0.0, 0.55)
     y1= x1 * 0.1 + 0.3 + np.random.normal(0.0, 0.03)
     conjunto_puntos.append([x1, y1])

x_data = [v[0] for v in conjunto_puntos]
y_data = [v[1] for v in conjunto_puntos]

import matplotlib.pyplot as plt

#Graphic display
plt.plot(x_data, y_data, 'ro')
plt.legend()
plt.show()
```



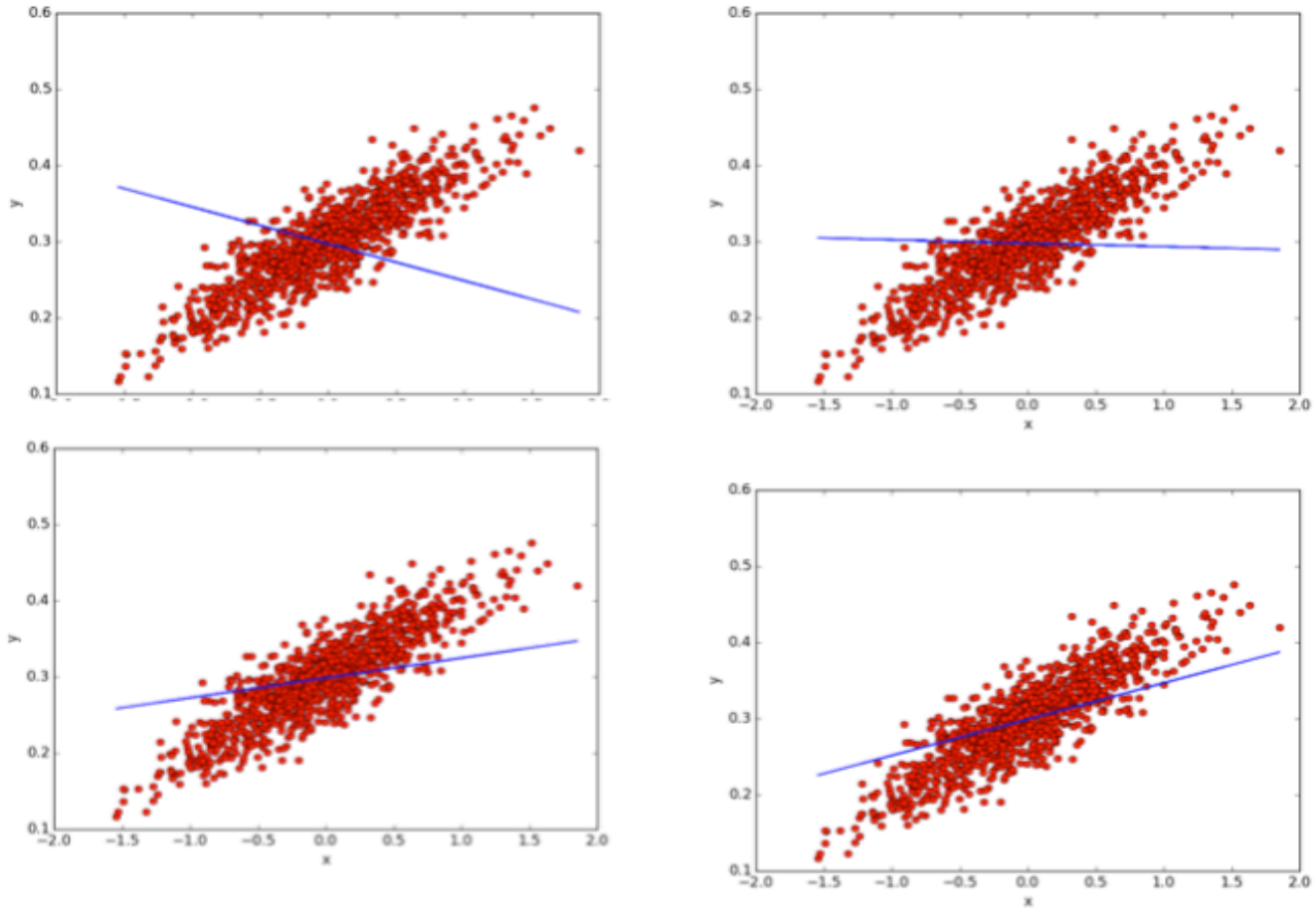
TensorFlow

Regresión Lineal en TensorFlow

```
import tensorflow as tf

W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
b = tf.Variable(tf.zeros([1]))
y = W * x_data + b

loss = tf.reduce_mean(tf.square(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

init = tf.initialize_all_variables()

sess = tf.Session()
sess.run(init)

for step in xrange(16):
        sess.run(train)
        print(step, sess.run(W), sess.run(b))
        #Graphic display
        plt.plot(x_data, y_data, 'ro')
        plt.plot(x_data, sess.run(W) * x_data + sess.run(b))
        plt.xlabel('x')
        plt.xlim(-2,2)
        plt.ylim(0.1,0.6)
        plt.ylabel('y')
        plt.legend()
        plt.show()
```

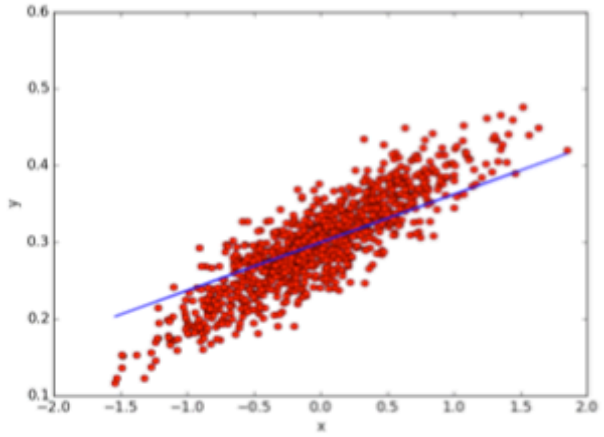
TensorFlow

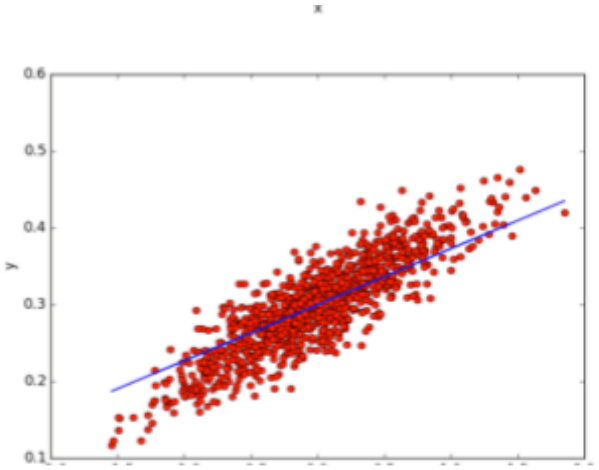
Regresión Lineal en TensorFlow

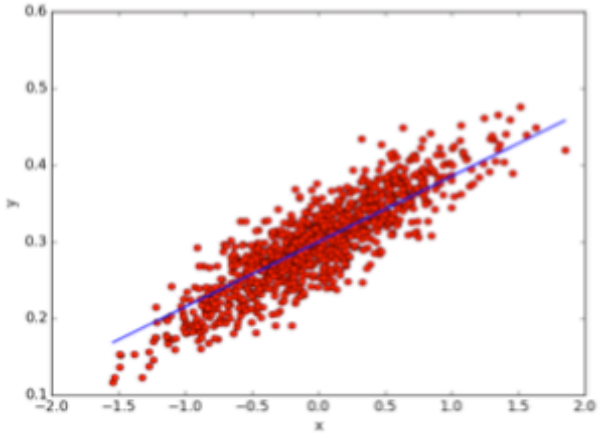


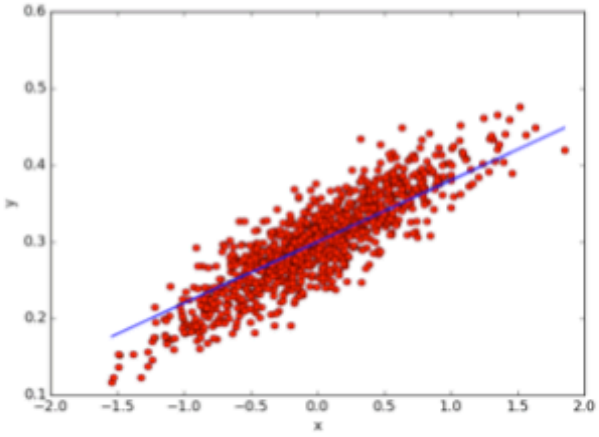
TensorFlow

Regresión Lineal en TensorFlow









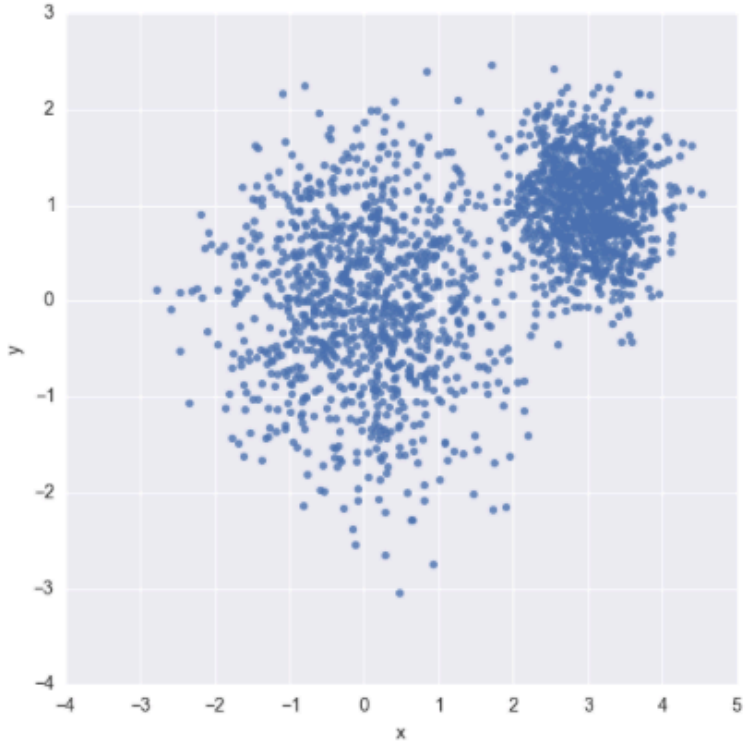
TensorFlow

TensorFlow

```
75 import tensorflow as tf
76 vectors=tf.constant(conjunto_puntos)
77 print "vectors"
78 print vectors
79 k=4
80 centroides=tf.Variable(tf.slice(tf.random_shuffle(vectors),[0,0],[k,-1]))
81 expanded_vectors=tf.expand_dims(vectors,0)
82 expanded_centroides=tf.expand_dims(centroides,1)
83 assignments=tf.argmin(tf.reduce_sum(tf.square(tf.sub(expanded_vectors,expanded_centroides)),2),0)
84 means=tf.concat(0,[tf.reduce_mean(tf.gather(vectors,tf.reshape(tf.where(tf.equal(assignments,c)),[1,-1])),
       reduction_indices=[1])for c in xrange(k)])
85 update_centroides=tf.assign(centroides,means)
86 init_op=tf.initialize_all_variables()
87 sess=tf.Session()
88 sess.run(init_op)
89 _,centroid_values,assignment_values=sess.run([update_centroides,centroides,assignments])
90
91
92 data={"x":[],"y":[],"cluster":[]}
93 for i in xrange(len(assignment_values)):
94     data["x"].append(conjunto_puntos[i][0])
95     data["y"].append(conjunto_puntos[i][1])
96     data["cluster"].append(assignment_values[i])
97 df=pd.DataFrame(data)
98 sns.lmplot("x","y",data=df,fit_reg=False,size=6,hue="cluster",legend=False)
99 plt.show()
100
```

TensorFlow

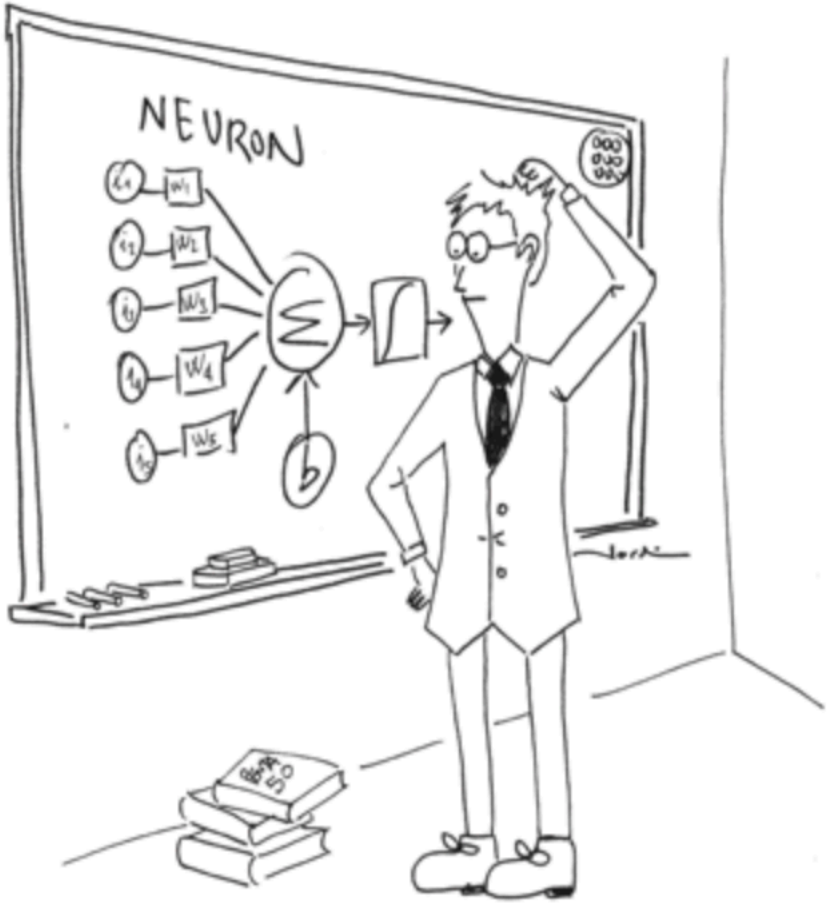
K-means en TensorFlow



1

2

TensorFlow



TensorFlow

Iris Flower data set in TensorFlow

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0
...	...	...	...	...
7.0	3.2	4.7	1.4	1
6.4	3.2	4.5	1.5	1
6.9	3.1	4.9	1.5	1
...	...	...	...	...
6.5	3.0	5.2	2.0	2
6.2	3.4	5.4	2.3	2
5.9	3.0	5.1	1.8	2

Iris setosa

Iris versicolor

Iris virginica

TensorFlow

Iris Flower data set in TensorFlow

```
import numpy as np
import tensorflow as tf
from tensorflow.contrib import learn

tf.logging.set_verbosity(tf.logging.INFO)


# Data sets
IRIS_TRAINING = os.path.join(os.path.dirname(__file__), "iris_training.csv")
IRIS_TEST = os.path.join(os.path.dirname(__file__), "iris_test.csv")

# Load datasets.
training_set = tf.contrib.learn.datasets.base.load_csv_with_header(
                                                                    filename=IRIS_TRAINING,
                                                                    target_dtype=np.int,
                                                                    features_dtype=np.float32)

test_set = tf.contrib.learn.datasets.base.load_csv_with_header(
                                                                filename=IRIS_TEST,
                                                                target_dtype=np.int,
                                                                features_dtype=np.float32)

# Specify that all features have real-value data
feature_columns = [tf.contrib.layers.real_valued_column("", dimension=4)]

# Build 3 layer DNN with 10, 20, 10 units respectively.
classifier = tf.contrib.learn.DNNClassifier(feature_columns=feature_columns,
                                            hidden_units=[10, 20, 10],
                                            n_classes=3,
                                            model_dir="/tmp/iris_model")

# Fit model.
classifier.fit(x=training_set.data,
               y=training_set.target,
               steps=2000)

# Evaluate accuracy.
accuracy_score = classifier.evaluate(x=test_set.data,
                                     y=test_set.target)["accuracy"]
print('Accuracy: {0:f}'.format(accuracy_score))

# Classify two new flower samples.
new_samples = np.array(
                      [[6.4, 3.2, 4.5, 1.5], [5.8, 3.1, 5.0, 1.7]], dtype=float)
y = list(classifier.predict(new_samples, as_iterable=True))
print('Predictions: {}'.format(str(y)))

```

TensorFlow

Iris Flower data set in TensorFlow

Sepal Length	Sepal Width	Petal Length	Petal Width
6.4	3.2	4.5	1.5
5.8	3.1	5.0	1.7

You can predict their species with the following code:

```
# Classify two new flower samples.
new_samples = np.array(
    [[6.4, 3.2, 4.5, 1.5], [5.8, 3.1, 5.0, 1.7]], dtype=float)
y = classifier.predict(new_samples)
print('Predictions: {}'.format(str(y)))
```

The `predict()` method returns an array of predictions, one for each sample:

```
Prediction: [1 2]
```

TensorFlow

Clasificación de Imágenes en TensorFlow

```
cd models/tutorials/image/imagenet
python classify_image.py
```

The above command will classify a supplied image of a panda bear.

If the model runs correctly, the script will produce the following output:

```
giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.88493)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00878)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score = 0.00317)
custard apple (score = 0.00149)
earthstar (score = 0.00127)
```

TensorFlow

Clasificación de Imágenes en TensorFlow

```
I tensorflow/examples/label_image/main.cc:200] military uniform (866): 0.647296
I tensorflow/examples/label_image/main.cc:200] suit (794): 0.0477196
I tensorflow/examples/label_image/main.cc:200] academic gown (896): 0.0232411
I tensorflow/examples/label_image/main.cc:200] bow tie (817): 0.0157356
I tensorflow/examples/label_image/main.cc:200] bolo tie (940): 0.0145024
```

In this case, we're using the default image of Admiral Grace Hopper, and you can see the network correctly identifies she's wearing a military uniform, with a high score of 0.6.

TensorFlow

Referencias

https://www.tensorflow.org

http://jorditorres.org/libro-hello-world-en-tensorflow/

TensorFlow

Google en realidad no ha liberado ninguno de sus modelos ya entrenados.