Machine Learning: TensorFlow

Speaker Qualifications

- Specialize in next-generation technologies
- Author of O'Reilly Videos on Hypermedia, Linking Data, Security and Encryption
- Author of 'Resource-Oriented Architecture Patterns for Webs of Data'
- Teaches and speaks internationally about REST, Semantic Web, Data Science, Machine Learning, GPU Computing, Security, Visualization, Architecture
- Worked in Defense, Finance, Retail, Hospitality, Video Game, Health Care, Telecommunications and Publishing Industries
- International Pop Recording Artist

Agenda

- Introduction
- TensorFlow Basics
- 'Try TensorFlow'
- Fun with TensorFlow

Introduction

About TensorFlow

- Open source library for numerical computation from Google Brain Team

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research
- Data flow graph-based

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research
- Data flow graph-based
- Nodes represent operations

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research
- Data flow graph-based
- Nodes represent operations
- Tensors flow through the nodes

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research
- Data flow graph-based
- Nodes represent operations
- Tensors flow through the nodes
- Leverages CPUs and/or GPUs

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research
- Data flow graph-based
- Nodes represent operations
- Tensors flow through the nodes
- Leverages CPUs and/or GPUs
- Scalable across computer clusters

About TensorFlow

- Open source library for numerical computation from Google Brain Team
- Version 1.0 was just released
- Primarily for machine learning and deep neural network research
- Data flow graph-based
- Nodes represent operations
- Tensors flow through the nodes
- Leverages CPUs and/or GPUs
- Scalable across computer clusters
- Ecosystem for graph visualization, serving production models, etc.

"TensorFlow's primary purpose is not to provide out-of-the-box machine learning solutions. Instead, TensorFlow provides an extensive suite of functions and classes that allow users to define models from scratch mathematically. This allows users with the appropriate technical background to create customized, flexible models quickly and intuitively. Additionally, while TensorFlow does have extensive support for ML-specific functionality, it is just as well suited to performing complex mathematical computations."

Source: TensorFlow For Machine Intelligence: A hands-on introduction to learning algorithms, Abrahams et al.

When should I use it?

When should I use it?

- Experimenting with new machine learning architectures and approaches

When should I use it?

- Experimenting with new machine learning architectures and approaches
- Operationalizing your experiments

When should I use it?

- Experimenting with new machine learning architectures and approaches
- Operationalizing your experiments
- Iterating on various architectures

When should I use it?

- Experimenting with new machine learning architectures and approaches
- Operationalizing your experiments
- Iterating on various architectures
- Large-scale distributed models

When should I use it?

- Experimenting with new machine learning architectures and approaches
- Operationalizing your experiments
- Iterating on various architectures
- Large-scale distributed models
- Building models for mobile/embedded systems

What is a Tensor?

What is a Tensor?

- An n-dimensional matrix

What is a Tensor?

- An n-dimensional matrix
- `3` (Rank 0)

What is a Tensor?

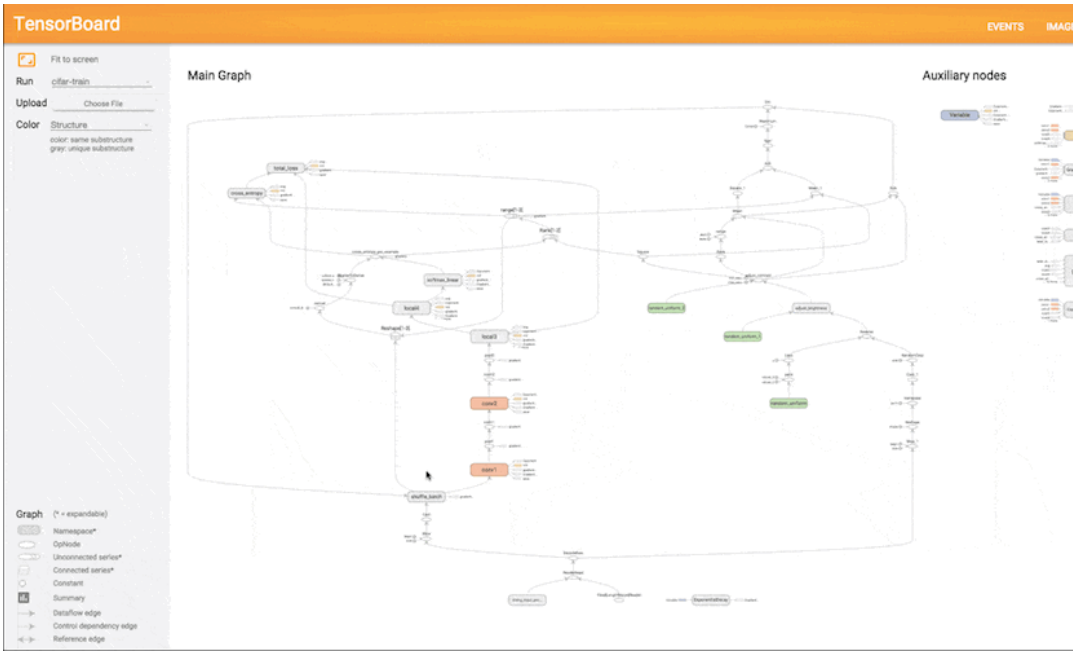
- An n-dimensional matrix
- `3` (Rank 0)
- `[1.,2.,3.]` (Rank 1 - Shape [3])

What is a Tensor?

- An n-dimensional matrix
- `3` (Rank 0)
- `[1.,2.,3.]` (Rank 1 - Shape [3])
- `[[1.,2.,3.], [4.,5.,6.]]` (Rank 2 - Shape[2, 3])

What is a Tensor?

- An n-dimensional matrix
- `3` (Rank 0)
- `[1.,2.,3.]` (Rank 1 - Shape [3])
- `[[1.,2.,3.], [4.,5.,6.]]` (Rank 2 - Shape[2, 3])
- `[[[1.,2.,3.]], [[7.,8.,9.]]]` (Rank 3 - Shape[2, 1, 3])



Credit: TensorBoard: Graph Visualization

https://www.tensorflow.org/install/

TensorFlow Basics

https://www.tensorflow.org/get_started/get_started

```
$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

https://www.tensorflow.org/get_started/get_started


```
$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import tensorflow as tf
```

https://www.tensorflow.org/get_started/get_started

```
$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import tensorflow as tf
```

```
>>> node1 = tf.constant(3.0, tf.float32)
>>> node2 = tf.constant(4.0)
>>> print(node1, node2)
(<tf.Tensor 'Const:0' shape=() dtype=float32>,
 <tf.Tensor 'Const_1:0' shape=() dtype=float32>)
```

https://www.tensorflow.org/get_started/get_started


```
>>> sess = tf.Session()
2017-03-02 19:16:33.627295: I tensorflow/core/common_runtime/gpu/gpu_device.cc:885
Found device 0 with properties:
name: GeForce GTX 1080
major: 6 minor: 1 memoryClockRate (GHz) 1.771
pciBusID 0000:01:00.0
Total memory: 7.92GiB
Free memory: 5.01GiB
2017-03-02 19:16:33.627417: W tensorflow/stream_executor/cuda/cuda_driver.cc:485]
2017-03-02 19:16:33.749592: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc
2017-03-02 19:16:33.749891: I tensorflow/core/common_runtime/gpu/gpu_device.cc:885
Found device 1 with properties:
name: GeForce GTX 1080
major: 6 minor: 1 memoryClockRate (GHz) 1.771
pciBusID 0000:02:00.0
Total memory: 7.92GiB
Free memory: 6.00GiB
2017-03-02 19:16:33.752272: I tensorflow/core/common_runtime/gpu/gpu_device.cc:906
2017-03-02 19:16:33.752282: I tensorflow/core/common_runtime/gpu/gpu_device.cc:916
2017-03-02 19:16:33.752285: I tensorflow/core/common_runtime/gpu/gpu_device.cc:916
2017-03-02 19:16:33.752675: I tensorflow/core/common_runtime/gpu/gpu_device.cc:975
Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1080,
pci bus id: 0000:01:00.0)
2017-03-02 19:16:33.752682: I tensorflow/core/common_runtime/gpu/gpu_device.cc:975
Creating TensorFlow device (/gpu:1) -> (device: 1, name: GeForce GTX 1080,
pci bus id: 0000:02:00.0)
```

https://www.tensorflow.org/get_started/get_started

```
>>> print(sess.run([node1, node2]))
[3.0, 4.0]
```

https://www.tensorflow.org/get_started/get_started

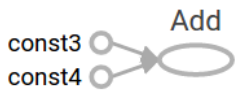

```
>>> print(sess.run([node1, node2]))
[3.0, 4.0]
```

```
>>> node3 = tf.add(node1, node2)
>>> print("node3: ", node3)
('node3: ', <tf.Tensor 'Add:0' shape=() dtype=float32>)
```

https://www.tensorflow.org/get_started/get_started

```
>>> print(sess.run([node1, node2]))
[3.0, 4.0]
```

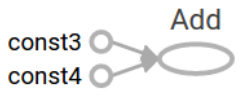
```
>>> node3 = tf.add(node1, node2)
>>> print("node3: ", node3)
('node3: ', <tf.Tensor 'Add:0' shape=() dtype=float32>)
```



https://www.tensorflow.org/get_started/get_started


```
>>> print(sess.run([node1, node2]))
[3.0, 4.0]
```

```
>>> node3 = tf.add(node1, node2)
>>> print("node3: ", node3)
('node3: ', <tf.Tensor 'Add:0' shape=() dtype=float32>)
```

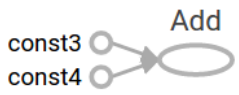



```
>>> print("sess.run(node3): ",sess.run(node3))
('sess.run(node3): ', 7.0)
```

https://www.tensorflow.org/get_started/get_started

```
>>> print(sess.run([node1, node2]))
[3.0, 4.0]
```

```
>>> node3 = tf.add(node1, node2)
>>> print("node3: ", node3)
('node3: ', <tf.Tensor 'Add:0' shape=() dtype=float32>)
```




```
>>> print("sess.run(node3): ",sess.run(node3))
('sess.run(node3): ', 7.0)
```

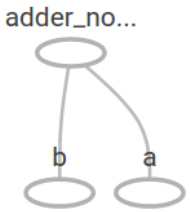
```
>>> print("sess.run(node1 + node2): ", sess.run(node1+node2))
('sess.run(node1 + node2): ', 7.0)
```

https://www.tensorflow.org/get_started/get_started


```
>>> a = tf.placeholder(tf.float32)
>>> b = tf.placeholder(tf.float32)
>>> adder_node = a + b
```

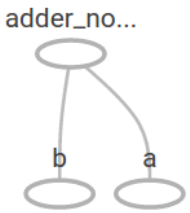
https://www.tensorflow.org/get_started/get_started

```
>>> a = tf.placeholder(tf.float32)
>>> b = tf.placeholder(tf.float32)
>>> adder_node = a + b
```



https://www.tensorflow.org/get_started/get_started


```
>>> a = tf.placeholder(tf.float32)
>>> b = tf.placeholder(tf.float32)
>>> adder_node = a + b
```

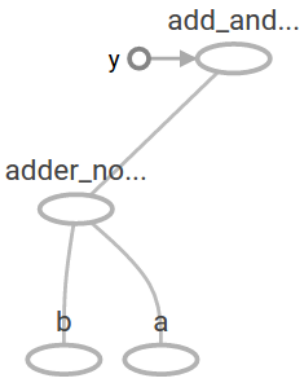

```
>>> print(sess.run(adder_node, {a: 3, b:4.5}))
7.5
>>> print(sess.run(adder_node, {a: [1,3], b: [2, 4]}))
[ 3.  7.]
```

https://www.tensorflow.org/get_started/get_started

```
>>> add_and_triple = adder_node * 3.
```

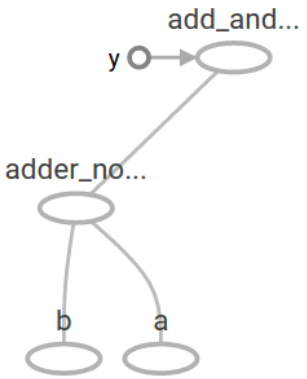
https://www.tensorflow.org/get_started/get_started


```
>>> add_and_triple = adder_node * 3.
```



https://www.tensorflow.org/get_started/get_started

```
>>> add_and_triple = adder_node * 3.
```




```
>>> print(sess.run(add_and_triple, {a: 3, b:4.5}))
22.5
```

https://www.tensorflow.org/get_started/get_started

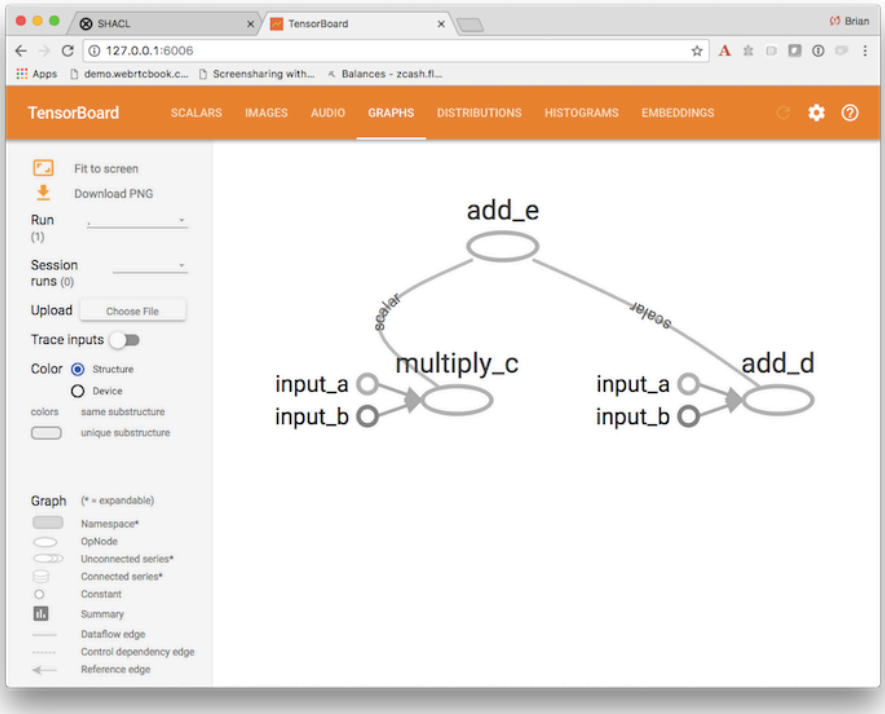
TensorBoard!

```
>>> import tensorflow as tf
>>> a = tf.constant(5, name="input_a")
>>> b = tf.constant(3, name="input_b")
>>> c = tf.multiply(a, b, name="multiply_c")
>>> d = tf.add(a,b, name="add_d")
>>> e = tf.add(c,d, name="add_e")
>>> sess = tf.Session()
>>> output = sess.run(e)
>>> writer = tf.train.SummaryWriter('./my_graph', sess.graph)
>>> writer.close()
>>> sess.close()
```

TensorBoard!

```
>>> import tensorflow as tf
>>> a = tf.constant(5, name="input_a")
>>> b = tf.constant(3, name="input_b")
>>> c = tf.multiply(a, b, name="multiply_c")
>>> d = tf.add(a,b, name="add_d")
>>> e = tf.add(c,d, name="add_e")
>>> sess = tf.Session()
>>> output = sess.run(e)
>>> writer = tf.train.SummaryWriter('./my_graph', sess.graph)
>>> writer.close()
>>> sess.close()
```

```
> tensorboard --logdir=my_graph/
Starting TensorBoard 39 on port 6006
(You can navigate to http://127.0.0.1:6006)
```

TensorFlow Data Types

Type	Description	Type	Description
tf.float32	32-bit floating point	tf.uint8	8-bit unsigned integer
tf.float64	64-bit floating point	tf.string	String
tf.int8	8-bit signed integer	tf.bool	Boolean
tf.int16	16-bit signed integer	tf.complex64	32-bit floating point real 32-bit floating point imaginary
tf.int32	32-bit signed integer	tf.qint8	8-bit signed integer
tf.int64	64-bit signed integer	tf.qint32	32-bit signed integer


```
>>> a = tf.placeholder(tf.int16)
>>> b = tf.placeholder(tf.int16)
>>> add = tf.add(a, b)
>>> mul = tf.multiply(a, b)
>>> sess.run(add, feed_dict={a: 2, b: 3})
5
>>> sess.run(mul, feed_dict={a: 2, b: 3})
6
```

```
>>> a = tf.placeholder(tf.int16)
>>> b = tf.placeholder(tf.int16)
>>> add = tf.add(a, b)
>>> mul = tf.multiply(a, b)
>>> sess.run(add, feed_dict={a: 2, b: 3})
5
>>> sess.run(mul, feed_dict={a: 2, b: 3})
6
```

```
>>> matrix1 = tf.constant([[3., 3.]])
>>> matrix2 = tf.constant([[2.],[2.]])
>>> product = tf.matmul(matrix1, matrix2)
>>> result = sess.run(product)
>>> print(result)
[[ 12.]]
```



```
>>> W = tf.Variable([.3], tf.float32)
>>> b = tf.Variable([-.3], tf.float32)
>>> x = tf.placeholder(tf.float32)
>>> linear_model = W * x + b
```

https://www.tensorflow.org/get_started/get_started

```
>>> W = tf.Variable([.3], tf.float32)
>>> b = tf.Variable([-.3], tf.float32)
>>> x = tf.placeholder(tf.float32)
>>> linear_model = W * x + b
```

```
>>> init = tf.global_variables_initializer()
>>> sess.run(init)
```

https://www.tensorflow.org/get_started/get_started


```
>>> W = tf.Variable([.3], tf.float32)
>>> b = tf.Variable([-.3], tf.float32)
>>> x = tf.placeholder(tf.float32)
>>> linear_model = W * x + b
```

```
>>> init = tf.global_variables_initializer()
>>> sess.run(init)
```


```
>>> print(sess.run(linear_model, {x:[1,2,3,4]}))
[ 0.          0.30000001  0.60000002  0.90000004]
```

https://www.tensorflow.org/get_started/get_started

```
>>> W = tf.Variable([.3], tf.float32)
>>> b = tf.Variable([-.3], tf.float32)
>>> x = tf.placeholder(tf.float32)
>>> linear_model = W * x + b
```

```
>>> init = tf.global_variables_initializer()
>>> sess.run(init)
```

```
>>> print(sess.run(linear_model, {x:[1,2,3,4]}))
[ 0.          0.30000001  0.60000002  0.90000004]
```

```
>>> y = tf.placeholder(tf.float32)
>>> squared_deltas = tf.square(linear_model - y)
>>> loss = tf.reduce_sum(squared_deltas)
>>> print(sess.run(loss, {x:[1,2,3,4], y:[0,-1,-2,-3]}))
23.66
```

https://www.tensorflow.org/get_started/get_started


```
>>> W = tf.Variable([.3], tf.float32)
>>> b = tf.Variable([-.3], tf.float32)
>>> x = tf.placeholder(tf.float32)
>>> linear_model = W * x + b
```

```
>>> init = tf.global_variables_initializer()
>>> sess.run(init)
```

```
>>> print(sess.run(linear_model, {x:[1,2,3,4]}))
[ 0.          0.30000001  0.60000002  0.90000004]
```

```
>>> y = tf.placeholder(tf.float32)
>>> squared_deltas = tf.square(linear_model - y)
>>> loss = tf.reduce_sum(squared_deltas)
>>> print(sess.run(loss, {x:[1,2,3,4], y:[0,-1,-2,-3]}))
23.66
```


```
>>> fixW = tf.assign(W, [-1.])
>>> fixb = tf.assign(b, [1.])
>>> sess.run([fixW, fixb])
[array([-1.], dtype=float32), array([ 1.], dtype=float32)]
>>> print(sess.run(loss, {x:[1,2,3,4], y:[0,-1,-2,-3]}))
0.0
```

https://www.tensorflow.org/get_started/get_started

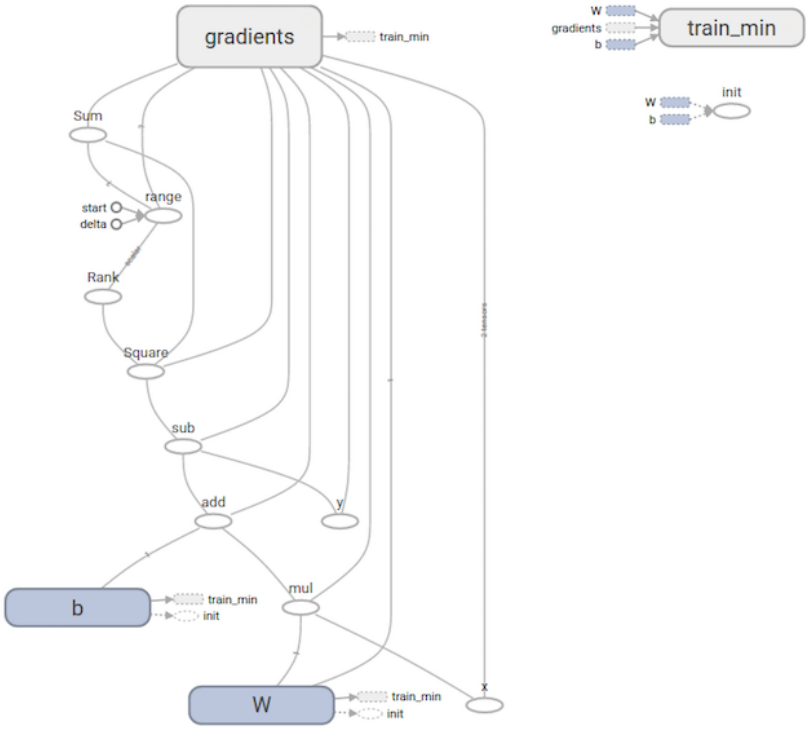
```
>>> optimizer = tf.train.GradientDescentOptimizer(0.01)
>>> train = optimizer.minimize(loss)
```

https://www.tensorflow.org/get_started/get_started


```
>>> optimizer = tf.train.GradientDescentOptimizer(0.01)
>>> train = optimizer.minimize(loss)
```

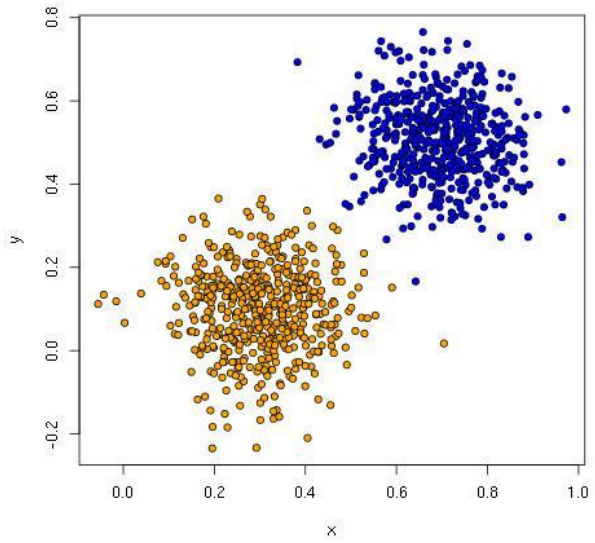

```
>>> sess.run(init) # reset values to incorrect defaults.
>>> for i in range(1000):
...   sess.run(train, {x:[1,2,3,4], y:[0,-1,-2,-3]})
...

>>> print(sess.run([W, b]))
[array([-0.9999969], dtype=float32), array([ 0.99999082], dtype=float32)]
```

https://www.tensorflow.org/get_started/get_started

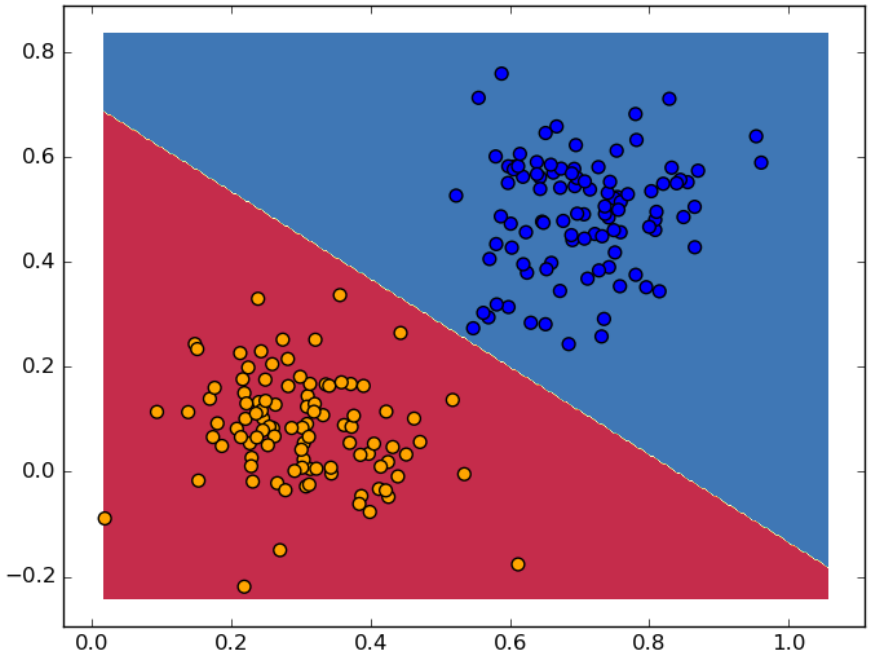


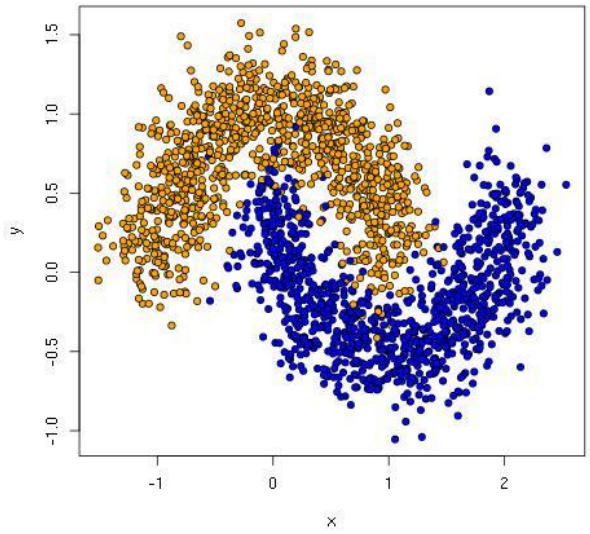
'Try TensorFlow'

http://bcomposes.com/2015/11/26/simple-end-to-end-tensorflow-examples/



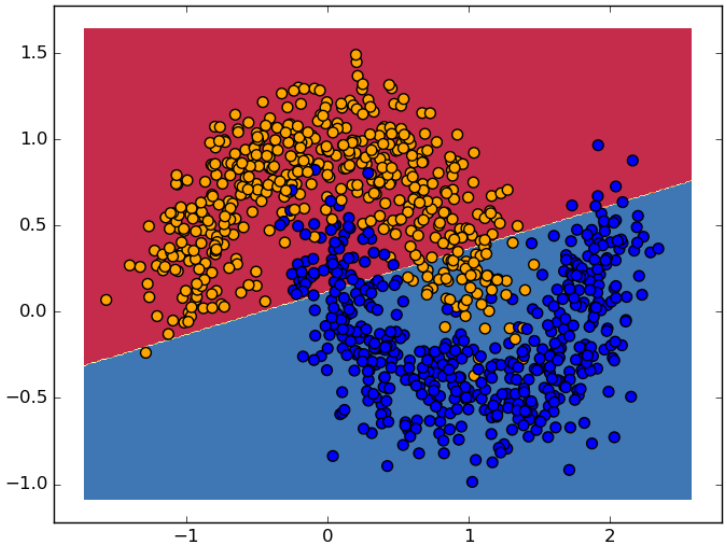
```
$ python softmax.py --train simdata/linear_data_train.csv
--test simdata/linear_data_eval.csv --num_epochs 5 --verbose True
Initialized!

Training.
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49

Weight matrix.
[[-1.87038445  1.87038457]
 [-2.23716712  2.23716712]]

Bias vector.
[ 1.57296884 -1.57296848]

Applying model to first test instance.
Point = [[ 0.14756215  0.24351828]]
Wx+b =  [[ 0.7521798  -0.75217938]]
softmax(Wx+b) =  [[ 0.81822371  0.18177626]]

Accuracy: 1.0
```



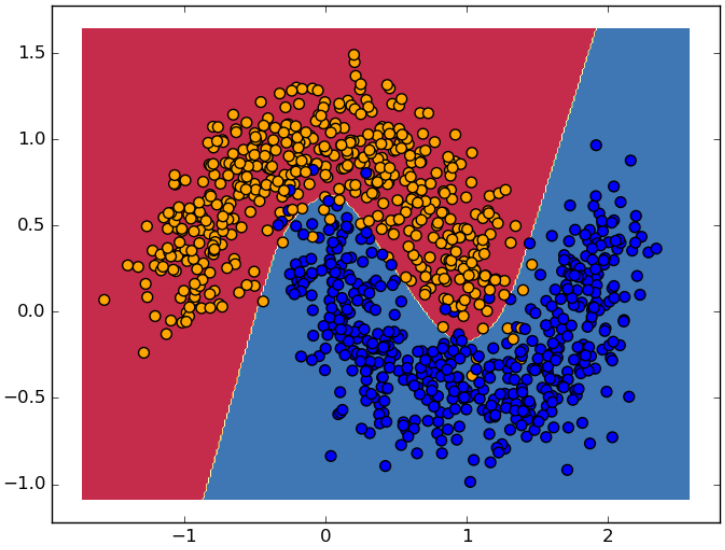

```
$ python softmax.py --train simdata/moon_data_train.csv
--test simdata/moon_data_eval.csv --num_epochs 100
Accuracy: 0.861
```

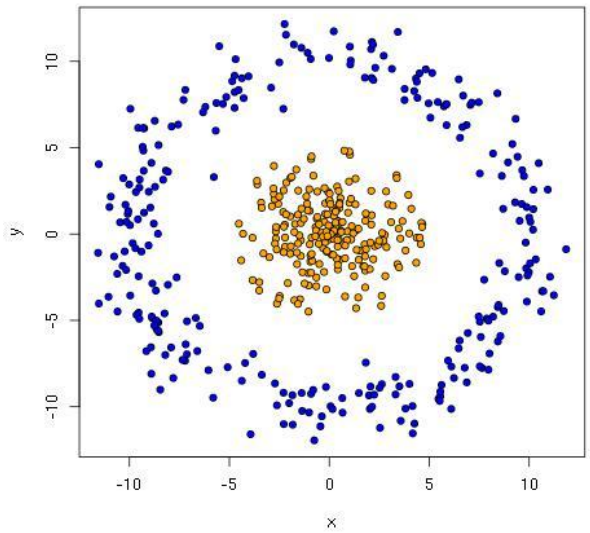
```
$ python softmax.py --train simdata/moon_data_train.csv
--test simdata/moon_data_eval.csv --num_epochs 100
Accuracy: 0.861
```




```
$ python hidden.py --train simdata/moon_data_train.csv
--test simdata/moon_data_eval.csv --num_epochs 100 --num_hidden 5
Accuracy: 0.971
```

```
$ python hidden.py --train simdata/moon_data_train.csv
--test simdata/moon_data_eval.csv --num_epochs 100 --num_hidden 5
Accuracy: 0.971
```

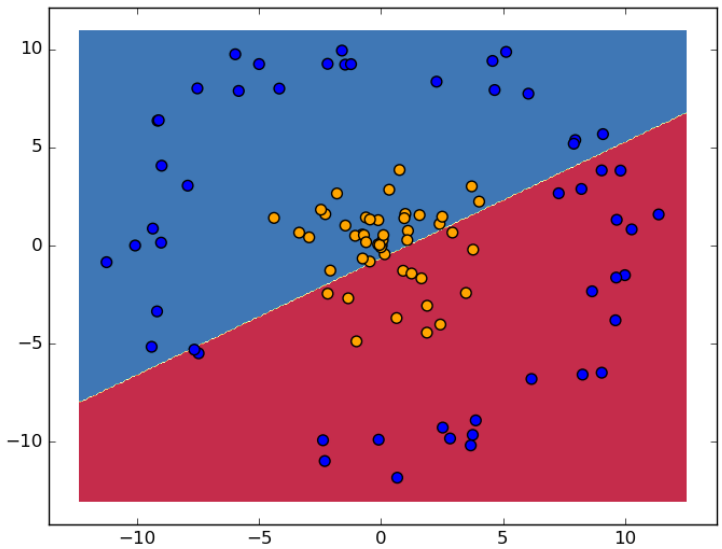




```
$ python softmax.py --train simdata/saturn_data_train.csv
--test simdata/saturn_data_eval.csv --num_epochs 100
Accuracy: 0.43
```



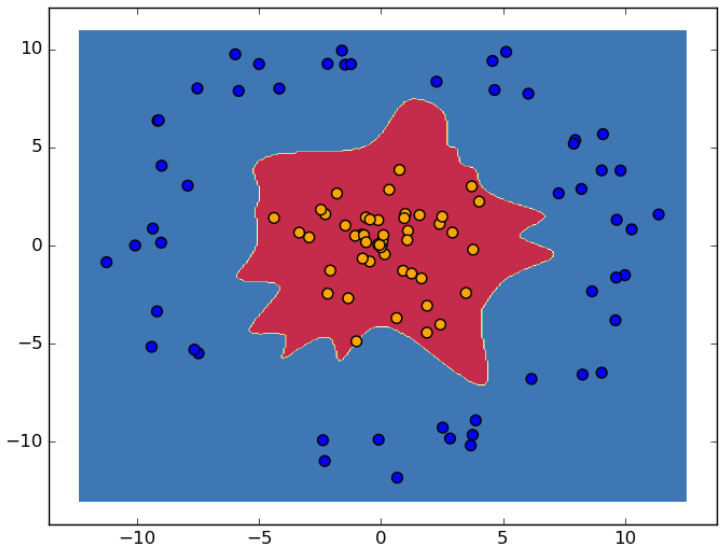
```
$ python softmax.py --train simdata/saturn_data_train.csv
--test simdata/saturn_data_eval.csv --num_epochs 100
Accuracy: 0.43
```

```
$ python hidden.py --train simdata/saturn_data_train.csv
--test simdata/saturn_data_eval.csv --num_epochs 100 --num_hidden 15
Accuracy: 1.0
```



```
$ python hidden.py --train simdata/saturn_data_train.csv
--test simdata/saturn_data_eval.csv --num_epochs 100 --num_hidden 15
Accuracy: 1.0
```



Fun With TensorFlow

https://www.tensorflow.org/tutorials/image_recognition

```
> git clone https://github.com/tensorflow/models.git tensorflow-models
> cd tensorflow-models/tutorials/image/imagenet
```

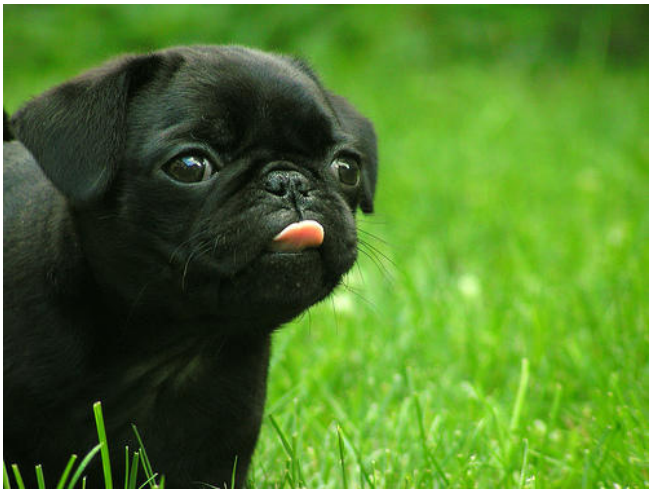


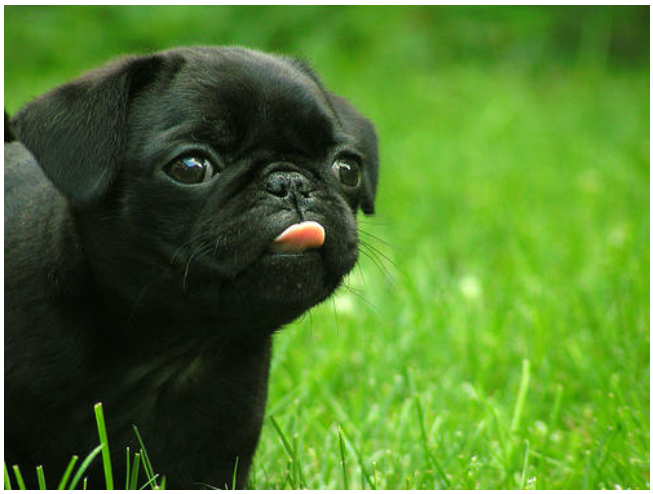
```
> git clone https://github.com/tensorflow/models.git tensorflow-models
> cd tensorflow-models/tutorials/image/imagenet
```

```
> git clone https://github.com/tensorflow/models.git tensorflow-models
> cd tensorflow-models/tutorials/image/imagenet
```

```
> python classify_image.py
>> Downloading inception-2015-12-05.tgz 100.0%
Successfully downloaded inception-2015-12-05.tgz 88931400 bytes.
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalizatio
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 = 0.003
custard apple (score = 0.00149)
earthstar (score = 0.00127)
```


```
> python classify_image.py --image_file 712033765.jpg
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalizatio
flamingo (score = 0.84379)
spoonbill (score = 0.00857)
black swan, Cygnus atratus (score = 0.00188)
goose (score = 0.00135)
black stork, Ciconia nigra (score = 0.00132)
```

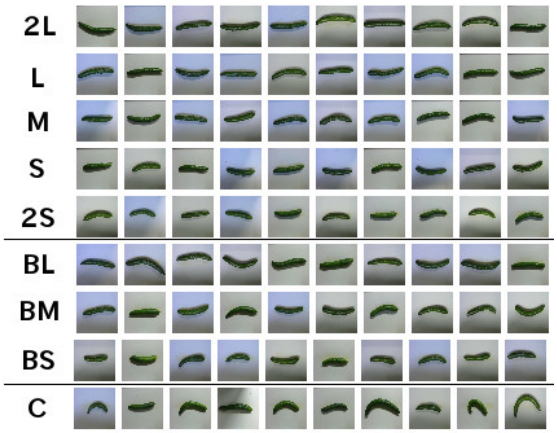





```
> python classify_image.py --image_file pug.jpeg
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalizatio
pug, pug-dog (score = 0.71793)
Brabancon griffon (score = 0.22177)
French bulldog (score = 0.01942)
Pekinese, Pekingese, Peke (score = 0.00133)
Boston bull, Boston terrier (score = 0.00065)
```


```
> python classify_image.py --image_file Beggars.jpg
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalizatio
Norwich terrier (score = 0.86040)
Norfolk terrier (score = 0.09727)
Australian terrier (score = 0.01304)
Lakeland terrier (score = 0.00186)
cairn, cairn terrier (score = 0.00178)
```


Sorting Cucumbers



http://workpiles.com/2016/06/ccb9-prototype2-recognition_system/

https://youtu.be/oZikw5k_2FM?t=55