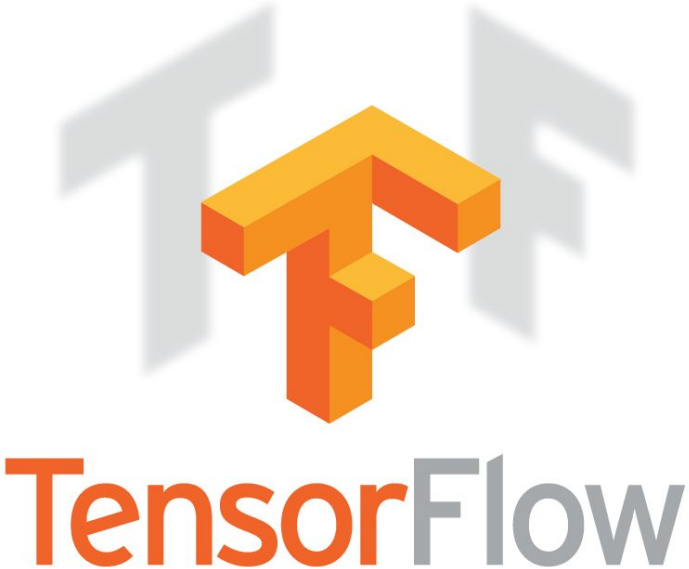
TensorFlow Tutorial

Deep Learning School 2016

Sherry Q. Moore

Have you installed TensorFlow?

https://github.com/sherrym/tf-tutorial/



- **Open source** Machine Learning library
- Especially useful for **Deep Learning**
- For research **and** production
- **Apache 2.0** license

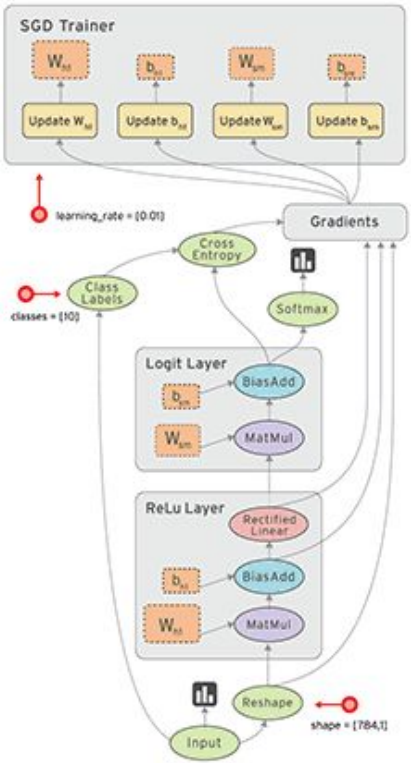
A multidimensional array.

A graph of operations.

Data Flow Graphs

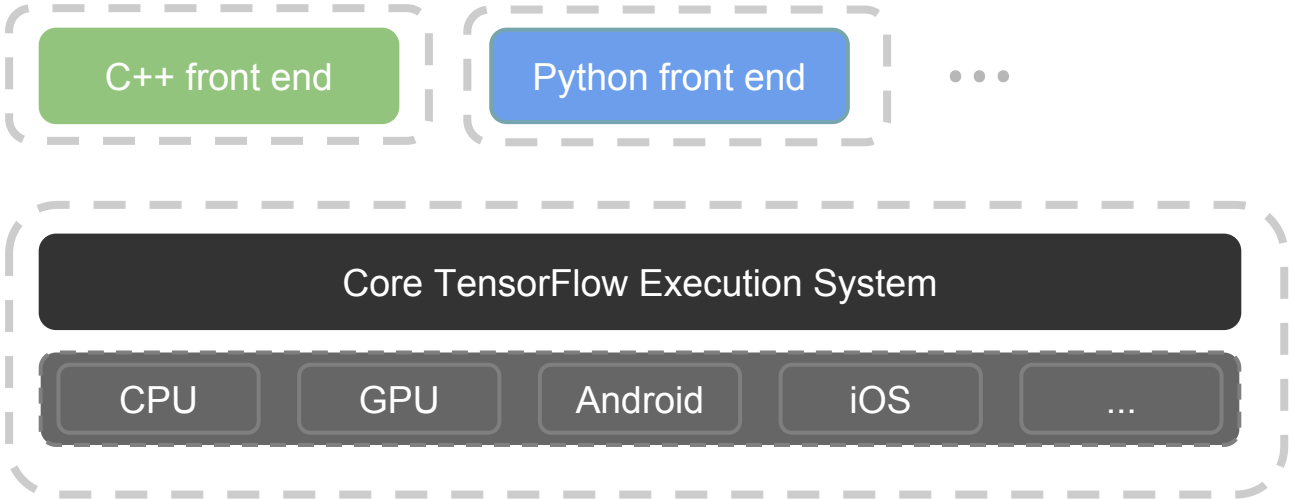
Computation is defined as a directed acyclic graph (DAG) to optimize an objective function

- Graph is defined in high-level language (Python)
- Graph is compiled and optimized
- Graph is executed (in parts or fully) on available low level devices (CPU, GPU)
- Data (tensors) flow through the graph
- TensorFlow can compute gradients automatically



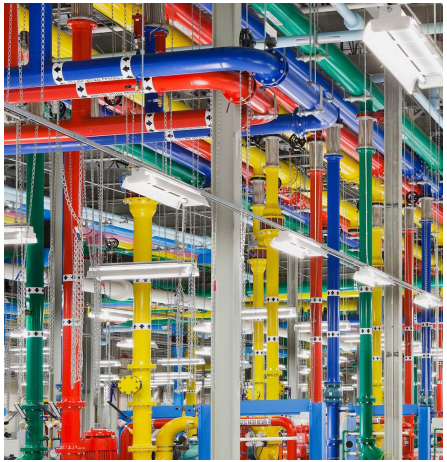
Architecture

- Core in C++
- Different front ends
  - Python and C++ today, community may add more



Portable & Scalable

Your laptop



Datacenters

Android

iOS

Raspberry Pi

Models on TensorFlow

https://github.com/tensorflow/models

- Inception https://github.com/tensorflow/models/tree/master/inception
- Show and Tell: A Neural Image Caption Generator
  https://github.com/tensorflow/models/tree/master/im2txt
- Language Model on One Billion Word
  https://github.com/tensorflow/models/tree/master/lm_1b
- SyntaxNet: Neural Models of Syntax
  https://github.com/tensorflow/models/tree/master/syntaxnet
- Resnet on Cifar10 and Cifar100
  https://github.com/tensorflow/models/tree/master/resnet
- Sequence-to-Sequence with Attention Model for Text Summarization
  https://github.com/tensorflow/models/tree/master/textsum

High Level Libraries

https://github.com/tensorflow/models

- TensorFlow-Slim image classification library https://github.com/tensorflow/models/tree/master/slim
- Image Compression with Neural Networks https://github.com/tensorflow/models/tree/master/compression
- Autoencoders https://github.com/tensorflow/models/tree/master/autoencoder
- Swivel https://github.com/tensorflow/models/blob/master/swivel
- Spatial Transformer Network https://github.com/tensorflow/models/tree/master/transformer

How is TensorFlow used at Google?

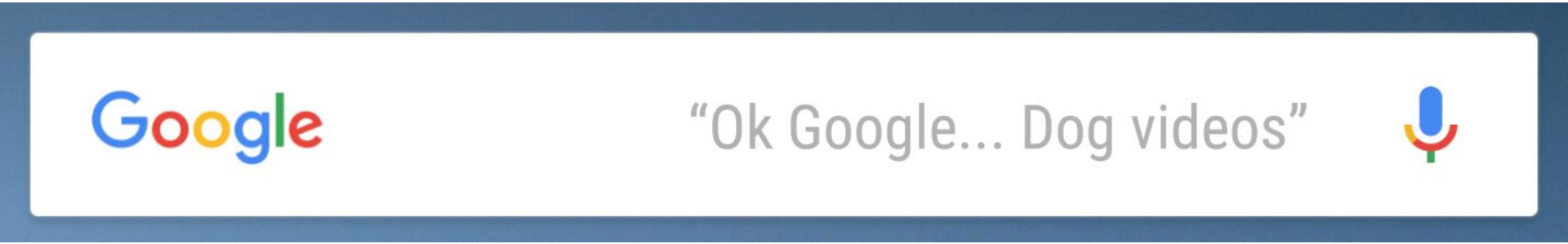
Recognizing images with Inception

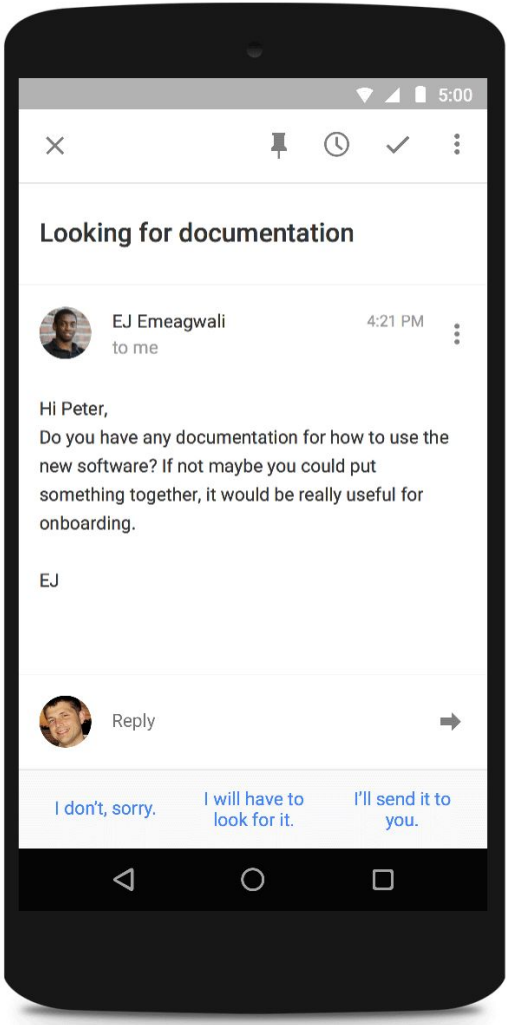
spotted salamander

fire salamander

retriever

https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html


Google
“Ok Google... Dog videos”



Smart reply in Inbox by Gmail

10%

of all responses sent on mobile (2016/02/01)

AlphaGo

BBC News Sport Weather iPlayer TV Radio More Search

NEWS Find local news

Home UK World Business Politics Tech Science Health Education Entertainment & Arts More

Technology

Google achieves AI 'breakthrough' at Go

An artificial intelligence program developed by Google beats Europe's top player at the ancient Chinese game of Go, about a decade earlier than expected.

27 January 2016 | Technology

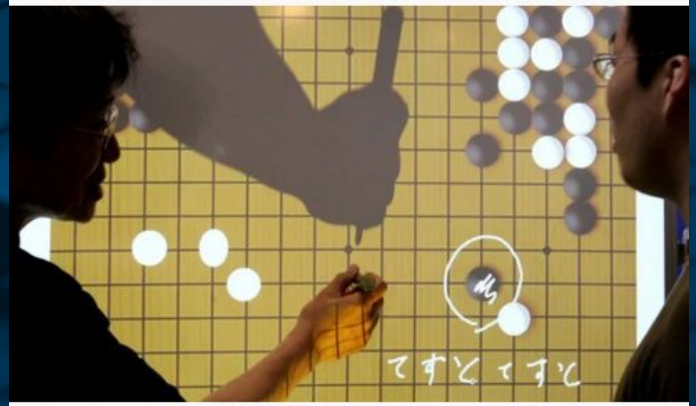
How did they do it?

What is the game Go?

Facebook trains AI to beat humans at Go

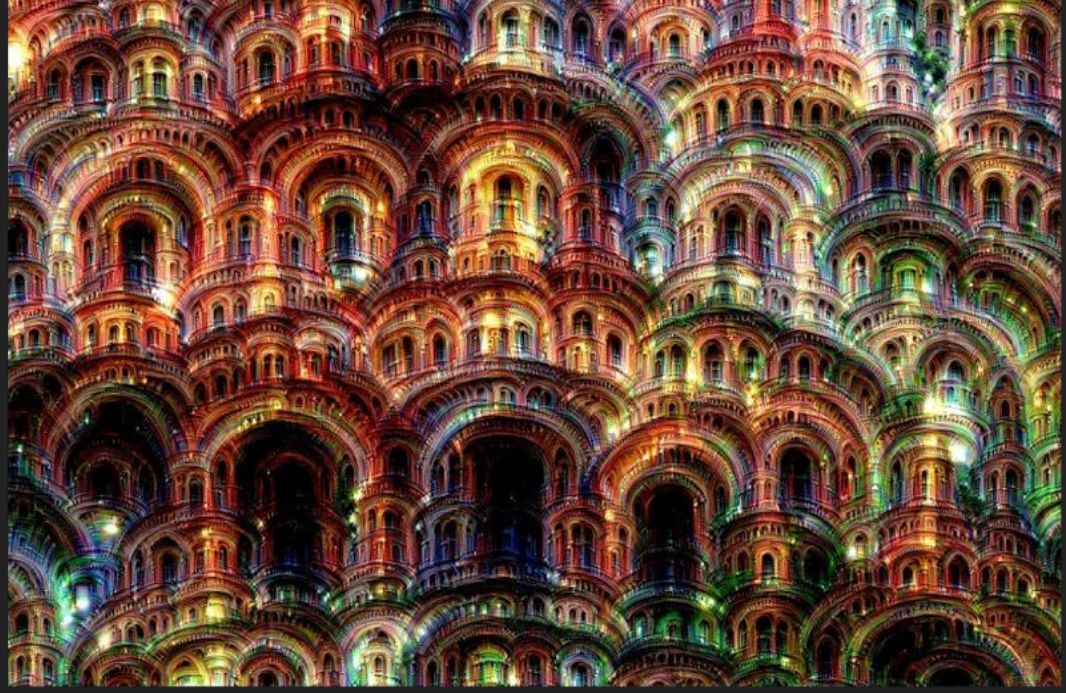
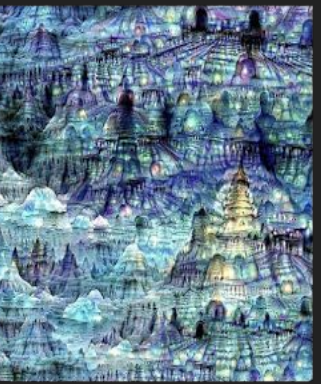
Google's AI just cracked the game that supposedly no computer could beat

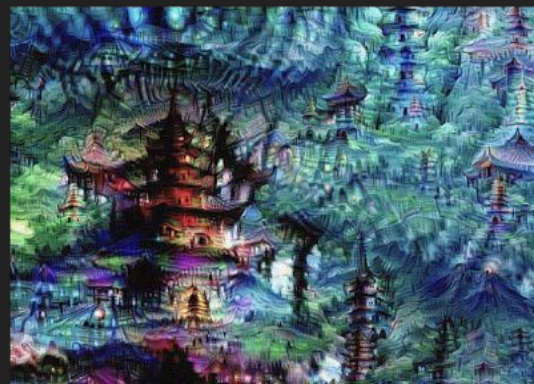
By Mike Murphy | January 27, 2016



Going up. (Reuters/Kiyoshi Ota)

Computers have slowly started to encroach on activities we previously believed only the brilliantly sophisticated human brain could handle. IBM's Deep Blue supercomputer beat Grand Master Garry Kasparov at chess in 1997, and in 2011 IBM's Watson beat former human winners at the quiz game *Jeopardy*. But the ancient board game Go has long been one of the major goals of artificial intelligence research. It's understood to be one of the most difficult games for computers to handle due to the sheer number of possible moves a player can make at any given point. Until now, that is.

Lab#1 Linear Regression Model

https://github.com/sherrym/tf-tutorial/blob/master/1_linear_regression_model.ipynb

What are we trying to do?

Mystery equation: `y = 0.1 * x + 0.3 + noise`

Model: `y = W * x + b`

Objective: Given enough `(x, y)` value samples, figure out the value of `W` and `b`.

1_linear_regression_model.ipynb

1. Import libraries (cell 1.1)
2. Create input data (cell 1.2)
3. Build inference graph (cell 1.3)
    a. Create Variables to hold weights and biases.
    b. Create Operations that produce logistic outputs.
4. Build training graph (cell 1.4)
    a. Loss
    b. Optimizer
    c. Train_op: Operation that minimizes Loss
5. Create session and run initialization (cell 1.5)
6. Perform training (cell 1.6)

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/api_docs/python/index.md

Exercises to run

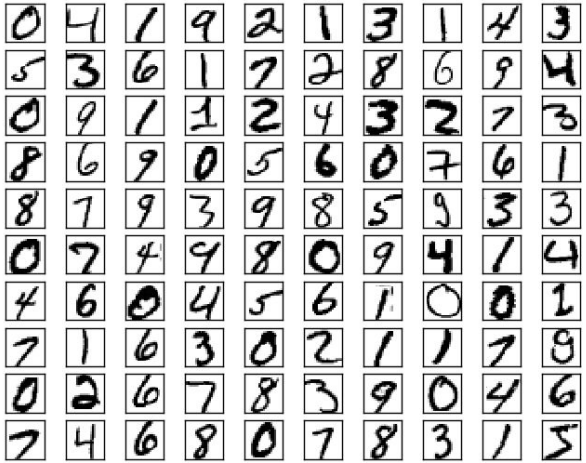
1. Run all the cells
2. Uncomment 1.5 and see the graph
3. Uncomment 1.8 and compare values
4. Try to train a different linear function
    - If you can't visualize the values, try changing pylab.ylim()
5. Try a different initialization function.
    - tf.random_normal
6. Try a different optimizer
    - tf.AdagradOptimizer

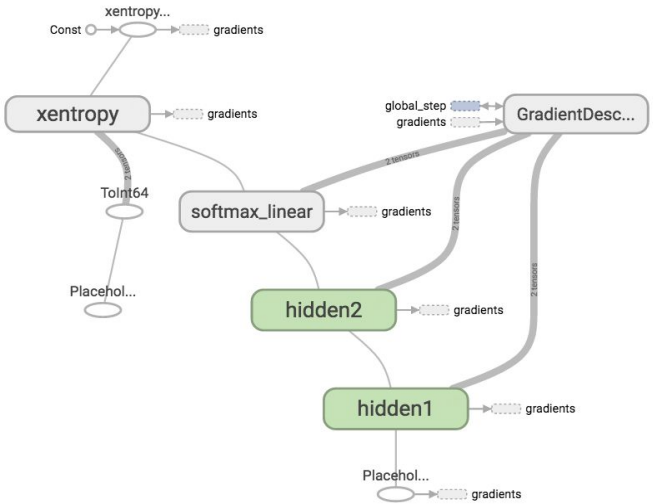
https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/api_docs/python/index.md

Lab#2 MNIST

https://github.com/sherrym/tf-tutorial/blob/master/2_mnist.ipynb

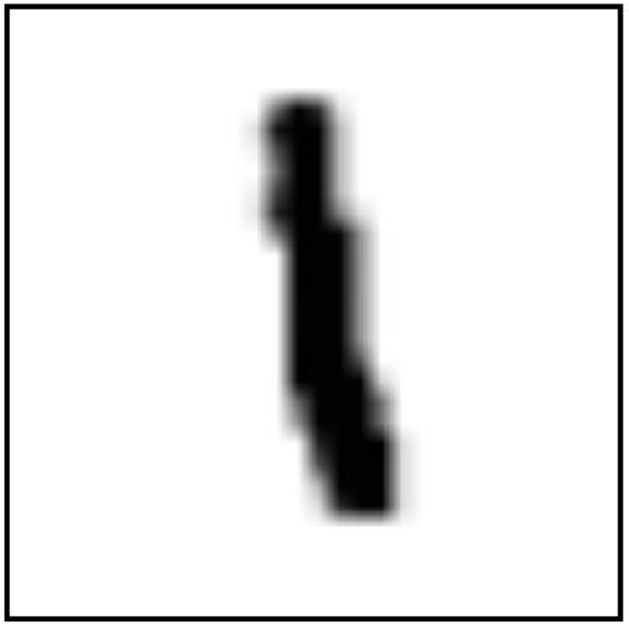
What are we trying to do?





Objective: Given enough images and labels, figure out the weights and biases so the model can correctly identify digits.

What we see



What the computer “sees”

≈

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	.6	.8	0	0	0	0	0	0
0	0	0	0	0	0	.7	1	0	0	0	0	0	0
0	0	0	0	0	0	.7	1	0	0	0	0	0	0
0	0	0	0	0	0	.5	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.7	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	.9	1	.1	0	0	0	0
0	0	0	0	0	0	0	.3	1	.1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

2_mnist.ipynb

1. Import libraries (cell 2.1)
2. Define a set of constants for convenience (cell 2.2)
3. Get input data (cell 2.3)
4. Build inference graph (cell 2.4)
5. Build training graph (cell 2.5)
6. Build the complete graph for training, and saving checkpoints (cell 2.6)
7. Run training for MAX_STEPS and save checkpoint at the end (cell 2.7)
8. Run evaluation based on the saved checkpoint (cell 2.8)

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/api_docs/python/index.md

Exercises to run

1. Run all the cells
2. Uncomment the graph writing to see what the graph looks like.
3. Print out loss value every 100 steps. What do you notice?
4. Save checkpoints every 500 steps.
5. Run evaluation with different checkpoints. What do you notice?
6. Run evaluation on the complete validation set.
7. Build evaluation graph from scratch instead of importing from meta graph.

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/api_docs/python/index.md

Bottomline: TensorFlow is for Machine Learning, from lab to production.

Thanks and have fun!

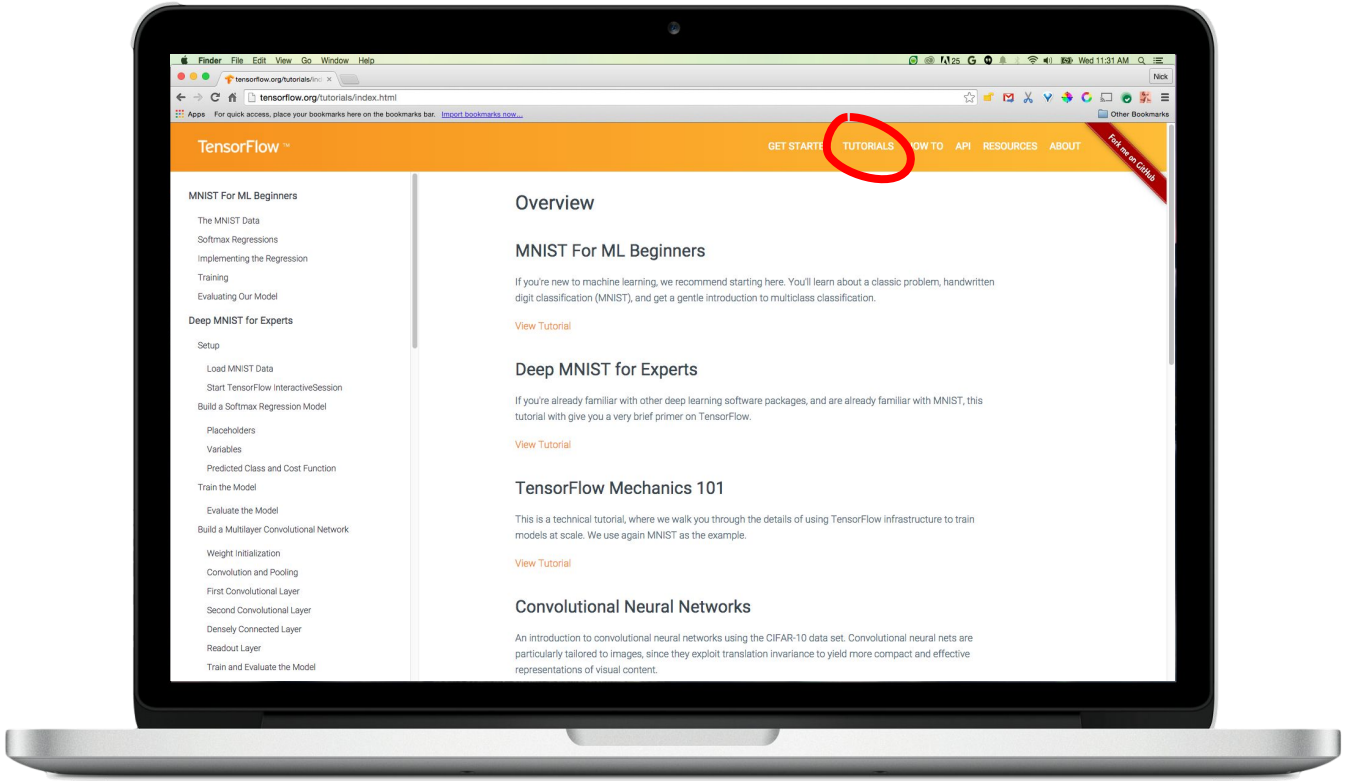
Sherry Q. Moore

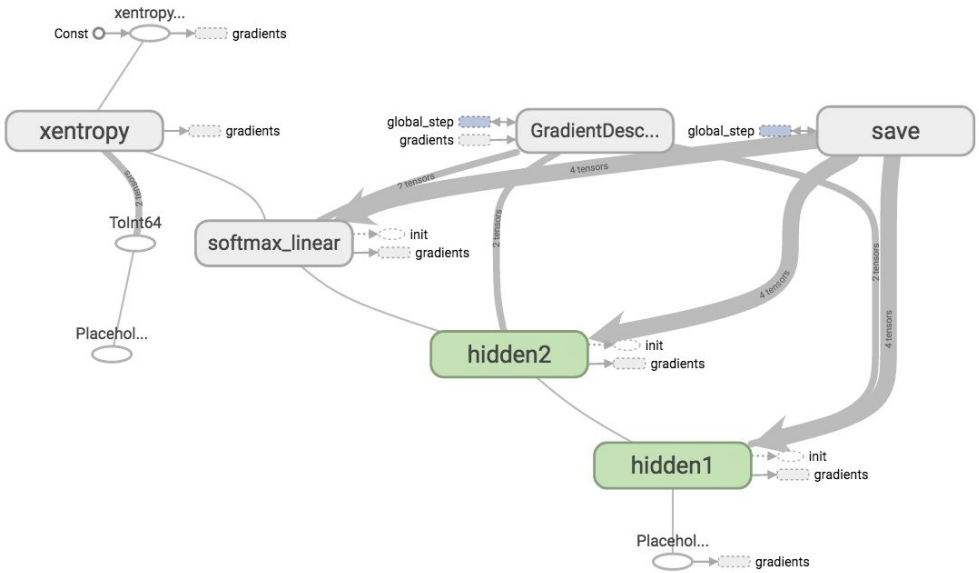
Backup Slides


TensorFlow

TensorFlow Playground Demo

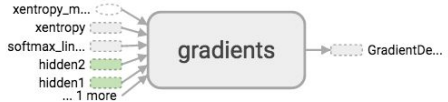
Lots of tutorials at tensorflow.org




xentropy...
Const
gradients
xentropy
gradients
ToInt64
Placehol...
softmax_linear
init
gradients
global_step
gradients
GradientDesc...
global_step
save
2 tensors
4 tensors
2 tensors
4 tensors
2 tensors
4 tensors
hidden2
init
gradients
hidden1
init
gradients
Placehol...
gradients


global_step
init
GradientDe...
save


hidden1
hidden2
softmax_lin...
global_step
init


xentropy_m...
xentropy
softmax_lin...
hidden2
hidden1
... 1 more
gradients
GradientDe...

Exercise: Evaluation on the whole set

```
# Build graph to count correct answers.
def mnist_evaluation(logits, labels):
  """Evaluate the quality of the logits at predicting the label.
  Args:
    logits: Logits tensor, float - [BATCH_SIZE, NUM_CLASSES].
    labels: Labels tensor, int32 - [BATCH_SIZE], with values in the
      range [0, NUM_CLASSES).
  Returns:
    A scalar int32 tensor with the number of examples (out of batch_size)
    that were predicted correctly.
  """
  # For a classifier model, we can use the in_top_k Op.
  # It returns a bool tensor with shape [batch_size] that is true for
  # the examples where the label is in the top k (here k=1)
  # of all logits for that example.
  correct = tf.nn.in_top_k(logits, labels, 1)
  # Return the number of true entries.
  return tf.reduce_sum(tf.cast(correct, tf.int32))
```

```
# Evaluation on the whole evaluation set.
with tf.Session(graph=tf.Graph()) as sess:
  saver = tf.train.import_meta_graph(
      os.path.join(TRAIN_DIR, "checkpoint-1999.meta"))
  saver.restore(
      sess, os.path.join(TRAIN_DIR, "checkpoint-1999"))

  print('Validation Data Eval:')
  # And run one epoch of eval.
  true_count = 0  # Counts the number of correct predictions.
  steps_per_epoch = data_sets.validation.num_examples
  num_examples = steps_per_epoch * BATCH_SIZE
  # Retrieve the Ops we 'remembered'.
  logits = tf.get_collection("logits")[0]
  images_placeholder = tf.get_collection("images")[0]
  labels_placeholder = tf.get_collection("labels")[0]

  # Add the Op to compare the logits to the labels during evaluation.
  eval_op = mnist_evaluation(logits, labels_placeholder)
```

```
for step in xrange(steps_per_epoch):
      images_feed, labels_feed =
data_sets.validation.next_batch(BATCH_SIZE)
      true_count += sess.run(eval_op,
                             feed_dict={images_placeholder:
images_feed,
                                        labels_placeholder: labels_feed})
  precision = true_count * 1.0 / num_examples
  print('  Num examples: %d  Num correct: %d  Precision @ 1:
%0.04f' %
        (num_examples, true_count, precision))
```

What's Next

tensorflow.org
github.com/tensorflow

Want to learn more?
Udacity class on Deep Learning, goo.gl/iHssII

Guides, codelabs, videos
MNIST for Beginners, goo.gl/tx8R2b
TF Learn Quickstart, goo.gl/uiefRn
TensorFlow for Poets, goo.gl/bVjFIL
ML Recipes, goo.gl/KewA03
TensorFlow and Deep Learning without a PhD, goo.gl/pHeXe7