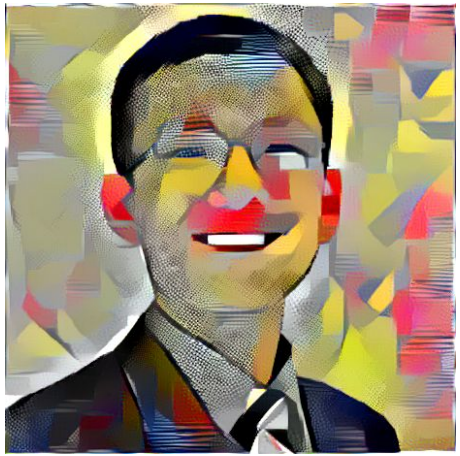
Getting up and running with TensorFlow

Create a Kaggle.com account

Sept. 18, 2017

Yufeng Guo
Developer Advocate,
Machine Learning
@YufengG
yufengg.com

Google Cloud Platform



Yufeng
yfg@google.com
@YufengG

These slides:
bit.ly/tf-workshop-aicon

Break at 10:30am

Create a Kaggle.com account

Overview of the Workshop

- Intro to TensorFlow, its ecosystem, cool st… f being done
  - *(& a little bit of keras)*
- TensorFlow's high l… canned' Estimators for Linea… eep Neural Networks
  - *Featuri… ashrooms and Fashion*
- "Wide & Deep" models with TensorFlow, and feature engineering
- Building Custom Estimators (for CNN models)

Create a Kaggle.com account

bit.ly/tf-workshop-aicon

bit.ly/tensorflow-workshop

These slides: bit.ly/tf-workshop-aicon

Create a Kaggle.com account

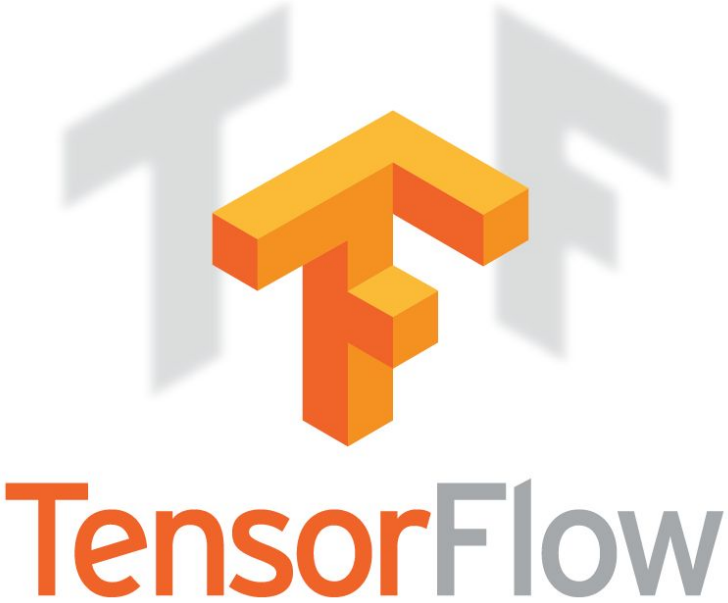
Install TensorFlow & Jupyter: www.tensorflow.org/install/

```
pip install tensorflow pandas jupyter
```

Clone GitHub Repo: bit.ly/tensorflow-workshop

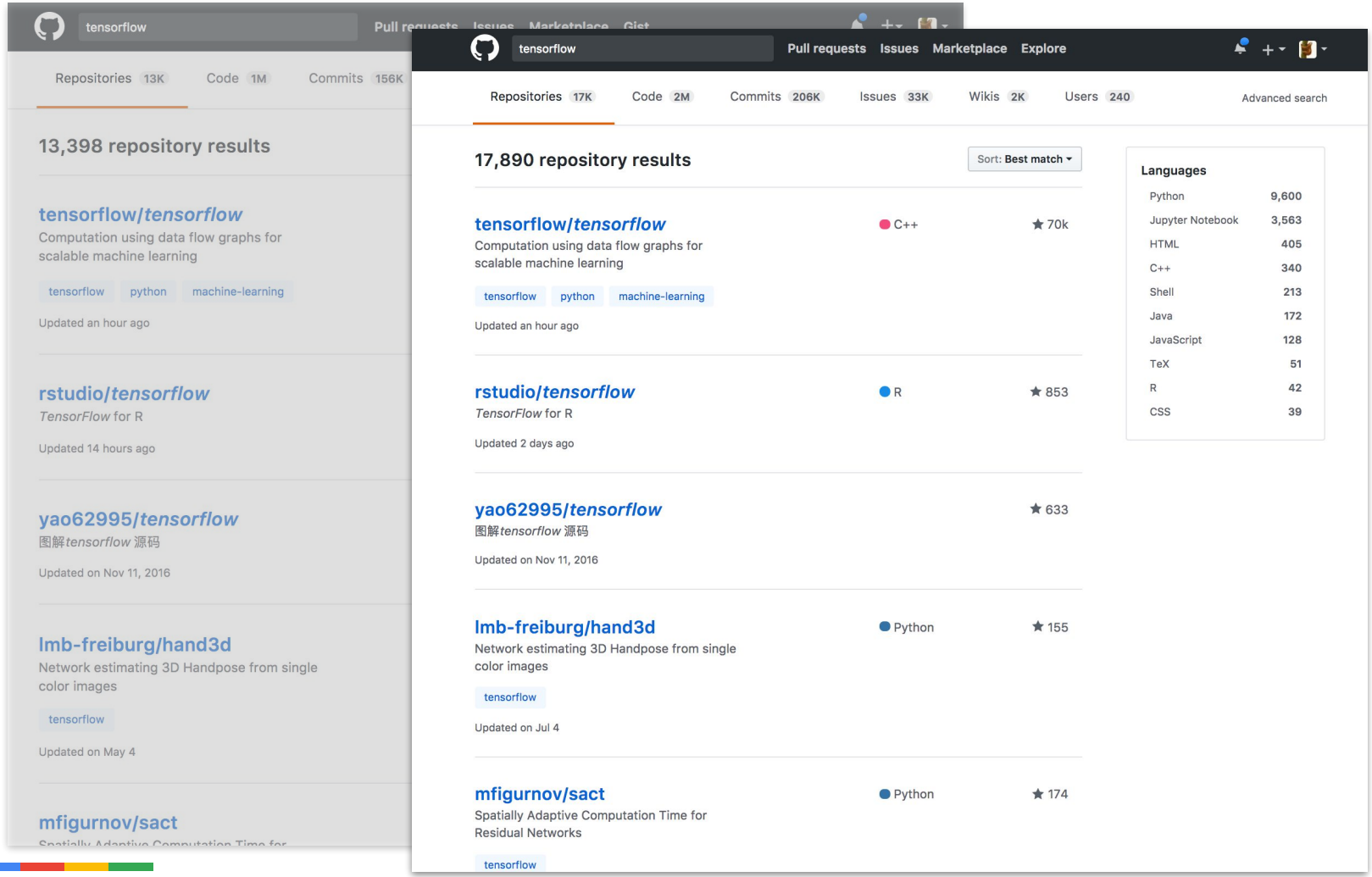
Optional: download the '**Fashion-MNIST**' files: https://github.com/zalandoresearch/fashion-mnist#get-the-data

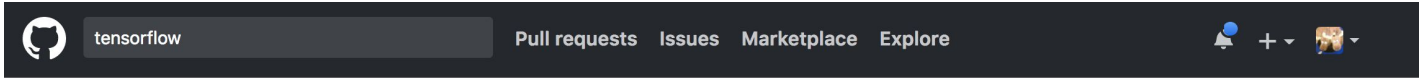
TensorFlow Research, TensorFlow Fun



tensorflow.org

- Fast, flexible, and scalable open-source machine learning library
- For research and production
- Apache 2.0 license


tensorflow
Pull requests
Issues
Marketplace
Gist
Repositories 13K
Code 1M
Commits 156K
13,398 repository results
tensorflow/tensorflow
Computation using data flow graphs for scalable machine learning
tensorflow
python
machine-learning
Updated an hour ago
rstudio/tensorflow
TensorFlow for R
Updated 14 hours ago
yao62995/tensorflow
图解tensorflow 源码
Updated on Nov 11, 2016
lmb-freiburg/hand3d
Network estimating 3D Handpose from single color images
tensorflow
Updated on May 4
mfigurnov/sact
tensorflow
Pull requests
Issues
Marketplace
Explore
Repositories 17K
Code 2M
Commits 206K
Issues 33K
Wikis 2K
Users 240
Advanced search
17,890 repository results
Sort: Best match
tensorflow/tensorflow
Computation using data flow graphs for scalable machine learning
C++
★ 70k
tensorflow
python
machine-learning
Updated an hour ago
rstudio/tensorflow
TensorFlow for R
R
★ 853
Updated 2 days ago
yao62995/tensorflow
图解tensorflow 源码
★ 633
Updated on Nov 11, 2016
lmb-freiburg/hand3d
Network estimating 3D Handpose from single color images
Python
★ 155
tensorflow
Updated on Jul 4
mfigurnov/sact
Spatially Adaptive Computation Time for Residual Networks
Python
★ 174
tensorflow
Languages
Python 9,600
Jupyter Notebook 3,563
HTML 405
C++ 340
Shell 213
Java 172
JavaScript 128
TeX 51
R 42
CSS 39



Repositories 31K
Code 4M
Commits 412K
Issues 56K
Topics 175
Wikis 3K
Users 420

Languages

Python	17,069
Jupyter Notebook	6,421
HTML	653
C++	567
Java	409
Shell	304
JavaScript	233
TeX	68
R	64
C	62

Advanced search Cheat sheet

Tensorflow

TensorFlow is an open source software library for numerical computation.

See topic

31,723 repository results

Sort: Most stars

tensorflow/*tensorflow*
Computation using data flow graphs for scalable machine learning
C++ ★ 95.7k
tensorflow python machine-learning
Apache-2.0 license Updated 43 minutes ago

tensorflow/models
Models and examples built with *TensorFlow*
Python ★ 32.7k
Apache-2.0 license Updated 3 hours ago

keras-team/keras
Deep Learning for humans
Python ★ 28.1k
deep-learning tensorflow python
Updated 7 hours ago 10 issues need help

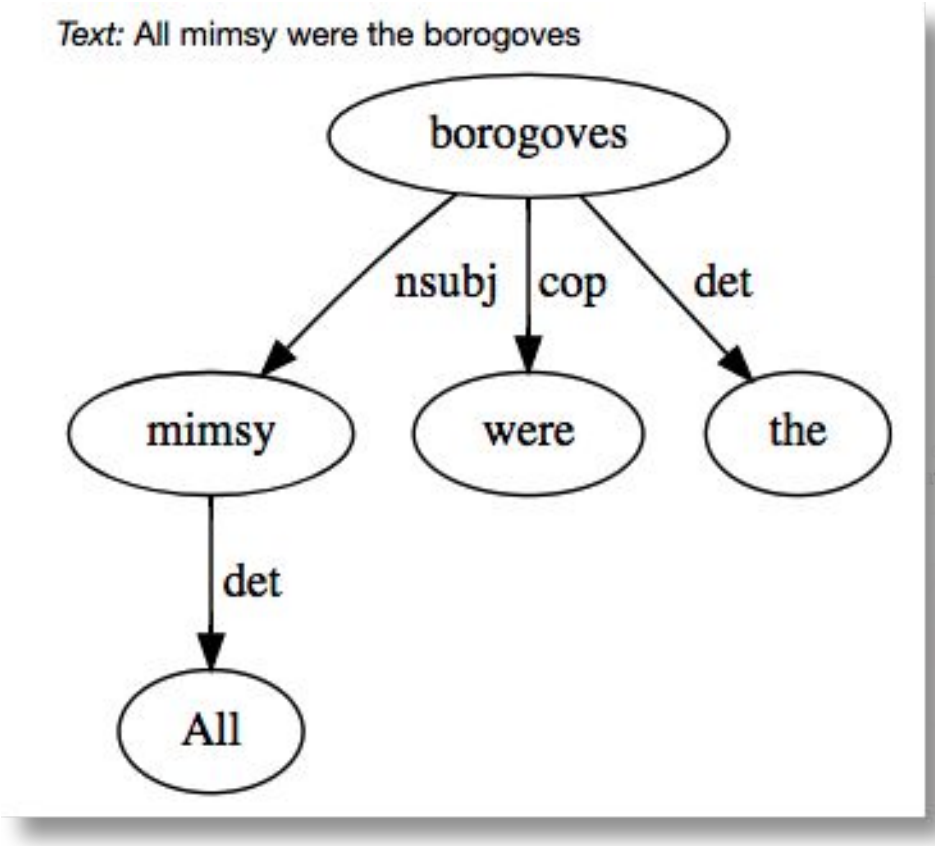
Open Source Models

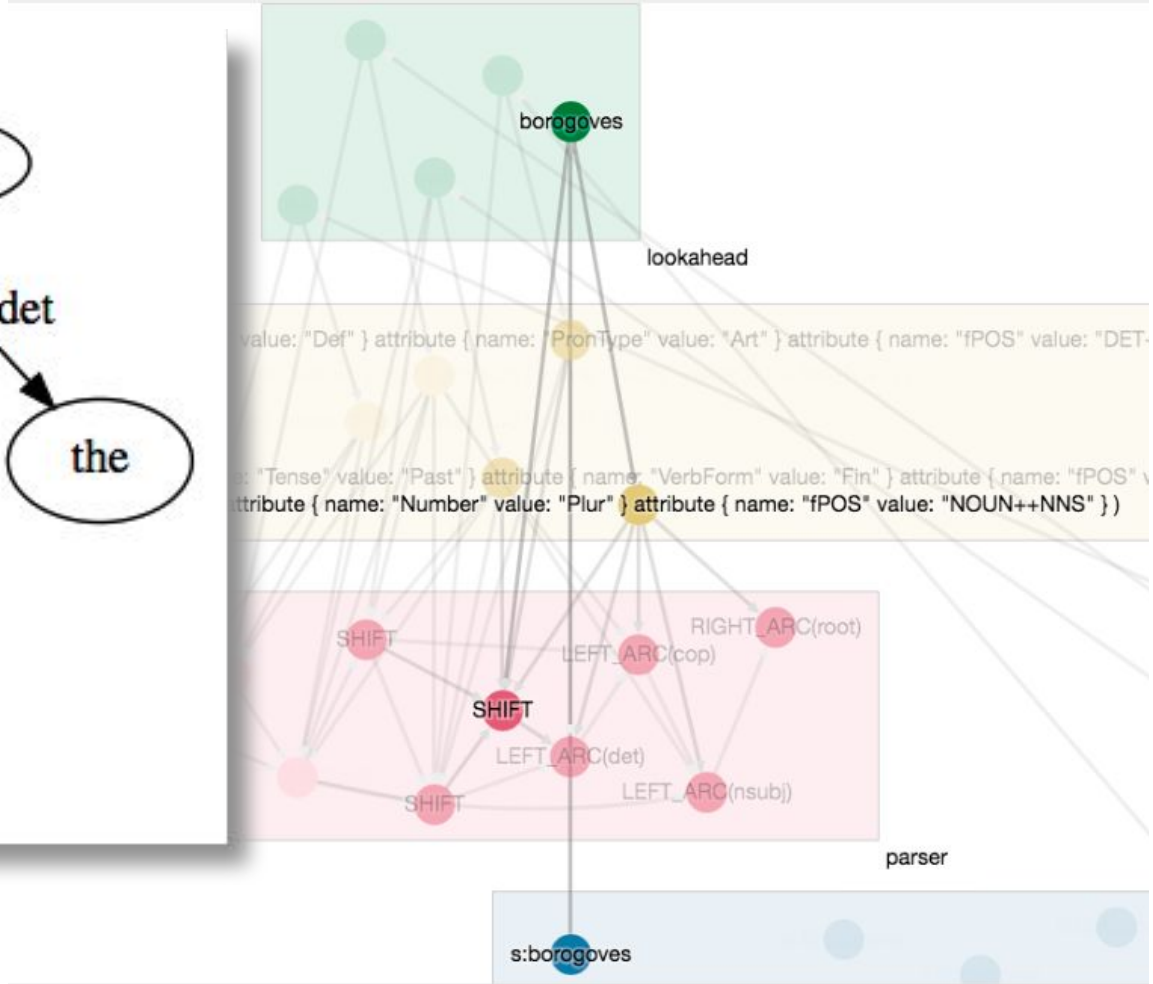
github.com/tensorflow/models

(many include trained weights, and support transfer learning)

SyntaxNet and ParseySaurus

research.googleblog.com/2017/03/an-upgrade-to-syntaxnet-new-models-and.html





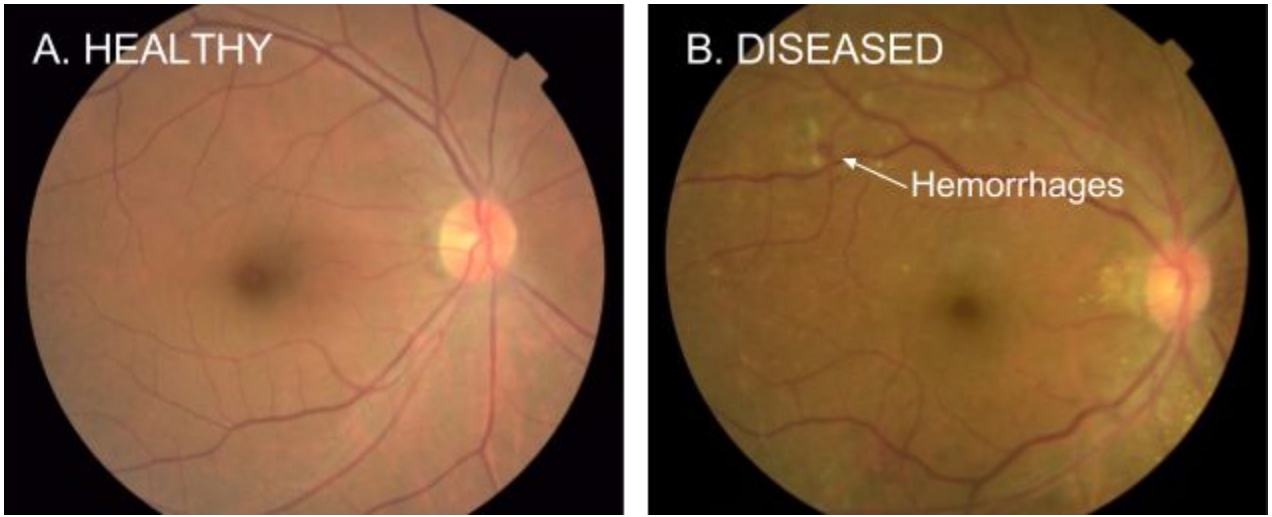
bit.ly/tf-workshop-aicon

Inception: Image classification

An Alaskan Malamute (left) and a Siberian Husky (right). Images from Wikipedia.

https://research.googleblog.com/2016/08/improving-inception-and-image.html

Detection of Diabetic Eye Disease



https://research.googleblog.com/2016/11/deep-learning-for-detection-of-diabetic.html

Skin Cancer Image Classification

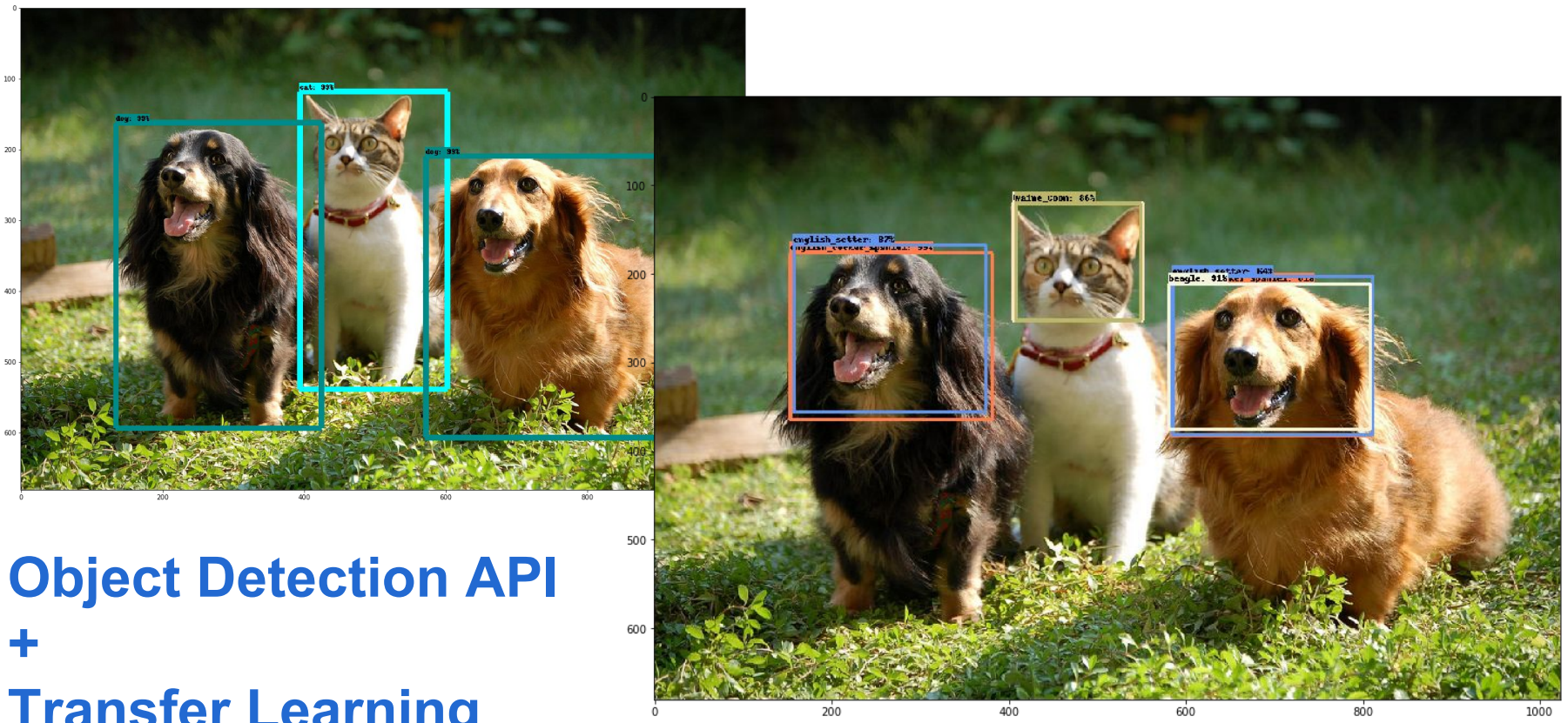
http://www.nature.com/nature/journal/v542/n7639/full/nature21056.html

TensorFlow Object Detection API

research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html

TensorFlow **Object Detection API**

bit.ly/tf-workshop-aicon

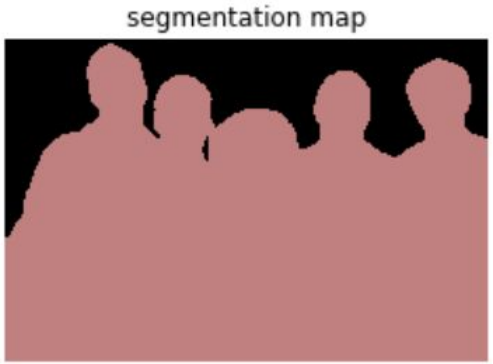


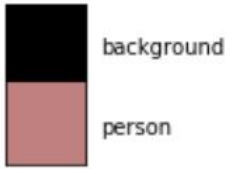
Object Detection API
+
Transfer Learning

bit.ly/tf-workshop-aicon

https://goo.gl/images/xf45oH, by https://www.flickr.com/photos/raneko/

TensorFlow **Object Detection API**
Now with **more detection!**



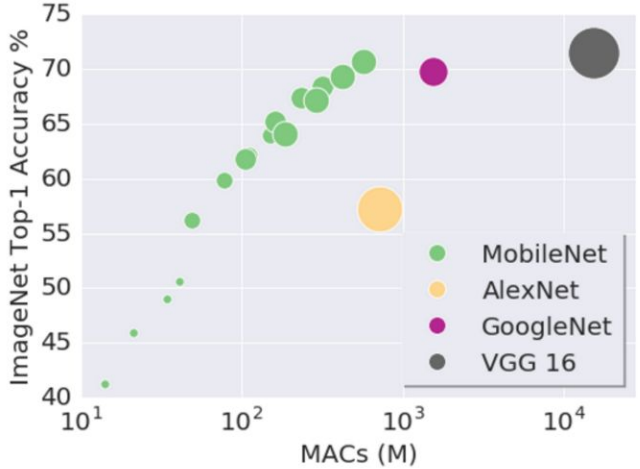


https://research.googleblog.com/2018/03/semantic-image-segmentation-with.html

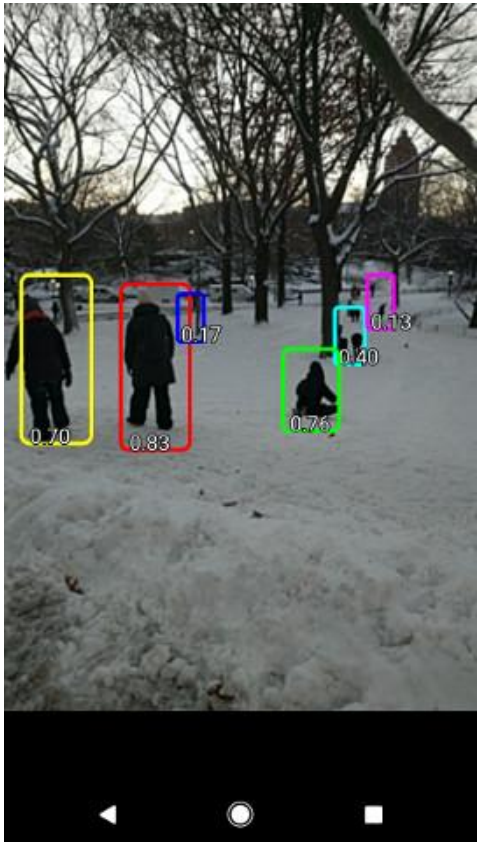
Real-time Video Segmentation (on mobile!)

https://research.googleblog.com/2018/03/mobile-real-time-video-segmentation.html

MobileNets: Open-Source Models for Efficient On-Device Vision

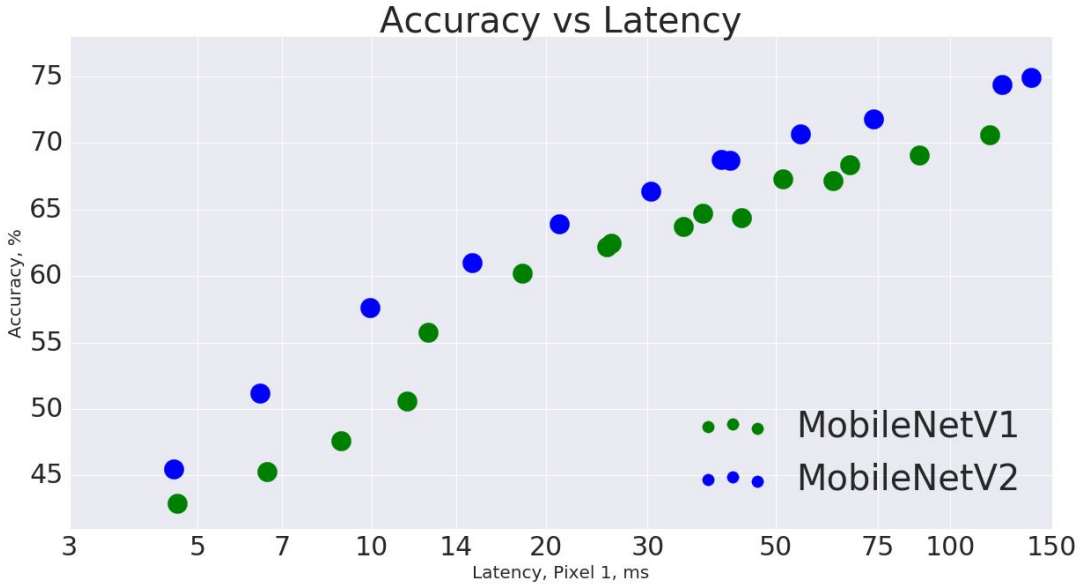


https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html



One of the newer Android demos, TF Detect, uses a **MobileNet** model trained using the **Tensorflow Object Detection API**.

bit.ly/tf-workshop-aicon



- **2x fewer** operations
- **30% fewer** parameters
- **30-40% faster** on a Google Pixel phone
- What about accuracy? **Higher**!

MobileNet v2!

https://research.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html

Shared Research in TensorFlow

Inception: tensorflow.org/tutorials/image_recognition

Object Detection API: research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html, **https://cloud.google.com/blog/big-data/2017/09/performing-prediction-with-tensorflow-object-detection-models-on-google-cloud-machine-learning-engine**

ParseySaurus: research.googleblog.com/2017/03/an-upgrade-to-syntaxnet-new-models-and.html

MobileNets: https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html

Show and Tell https://research.googleblog.com/2016/09/show-and-tell-image-captioning-open.html

Translation https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

Summarization https://research.googleblog.com/2016/08/text-summarization-with-tensorflow.html

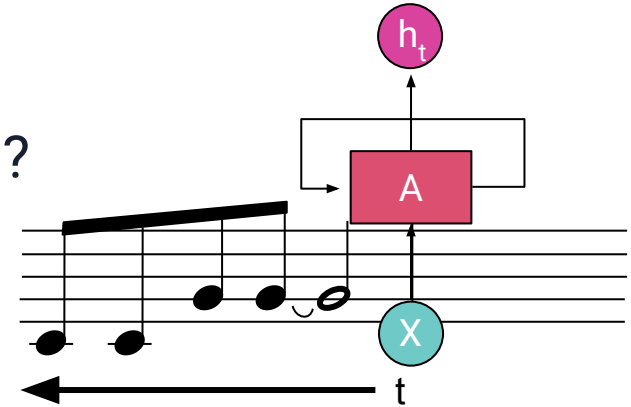
Pathology https://research.googleblog.com/2017/03/assisting-pathologists-in-detecting.html

Many more: https://github.com/tensorflow/models/

Project Magenta

Can ML generate compelling media?

- Music
- Images and Video
- Text (Jokes, Stories)
- Interestingness/Surprise
- Structure learning
- Measure success



magenta.tensorflow.org

Style Transfer

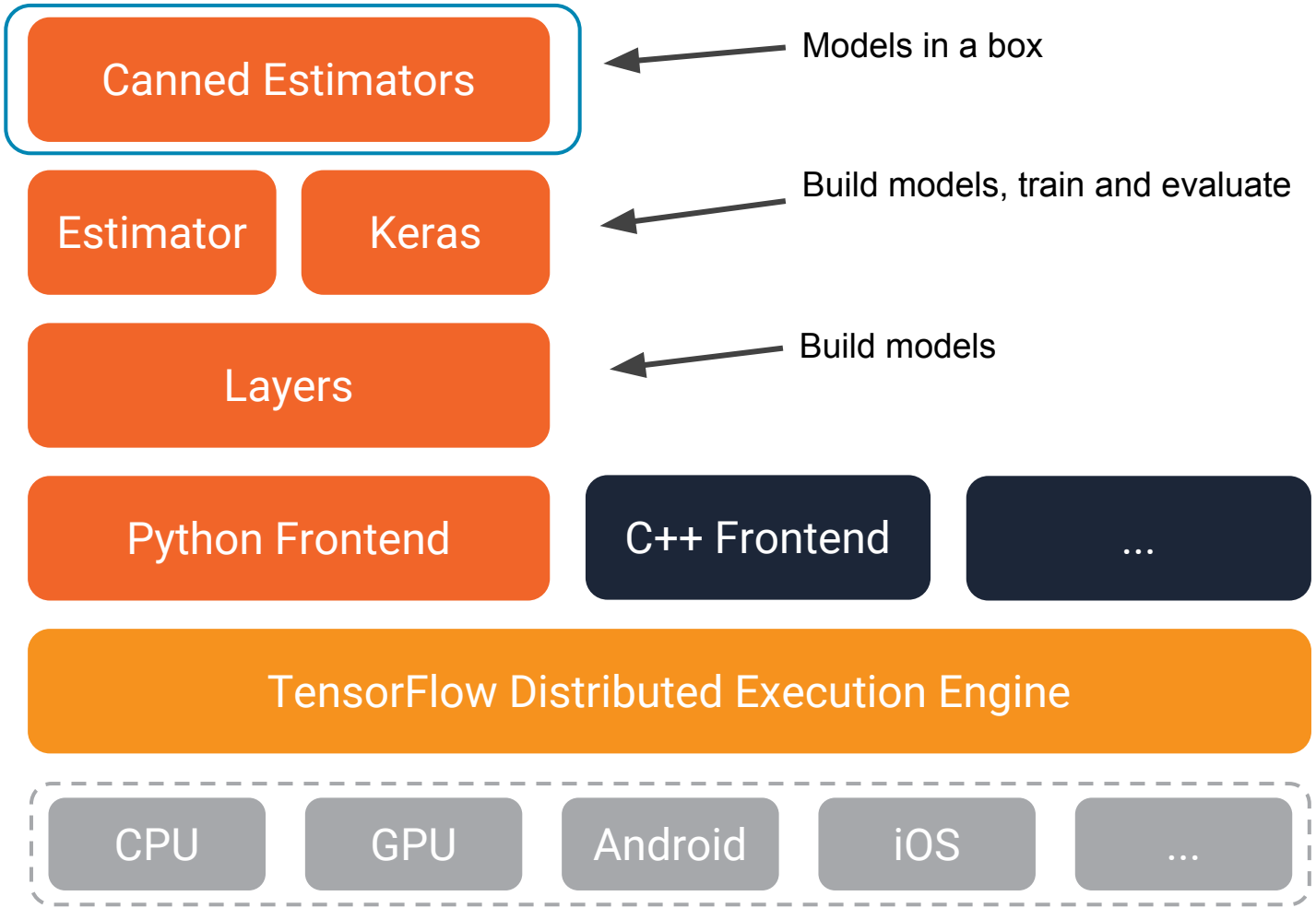
goo.gl/WPJtVe

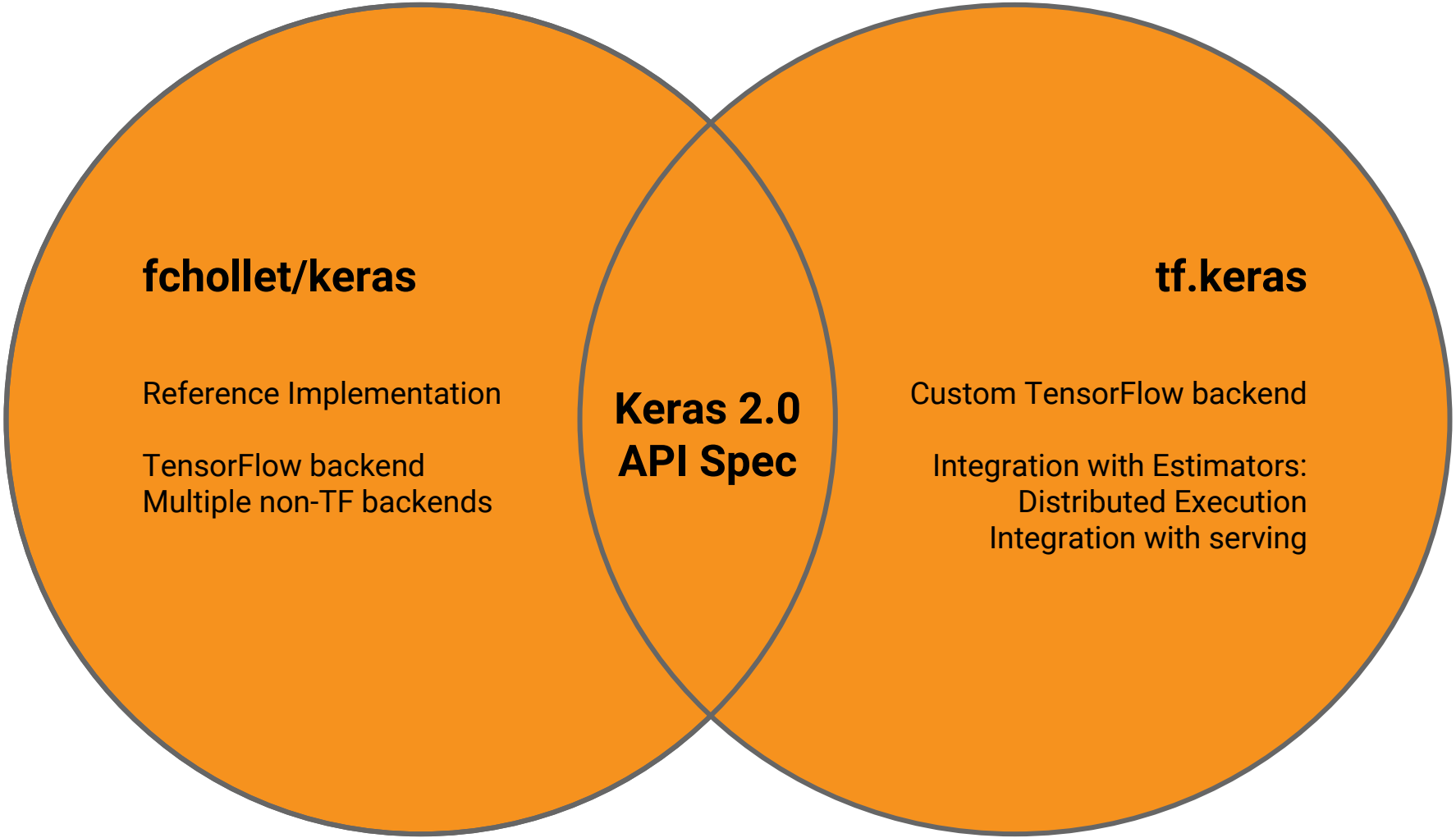
TensorFlow and the OSS community

TensorFlow was open-sourced in large part to allow the community to improve it with contributions.

- The TensorFlow team has set up processes to manage pull requests, review and route issues filed, and answer Stack Overflow and mailing list questions.
- So far, we've had more than 790 external contributors add to the code, from small documentation fixes to large additions

TensorFlow’s High-level APIs

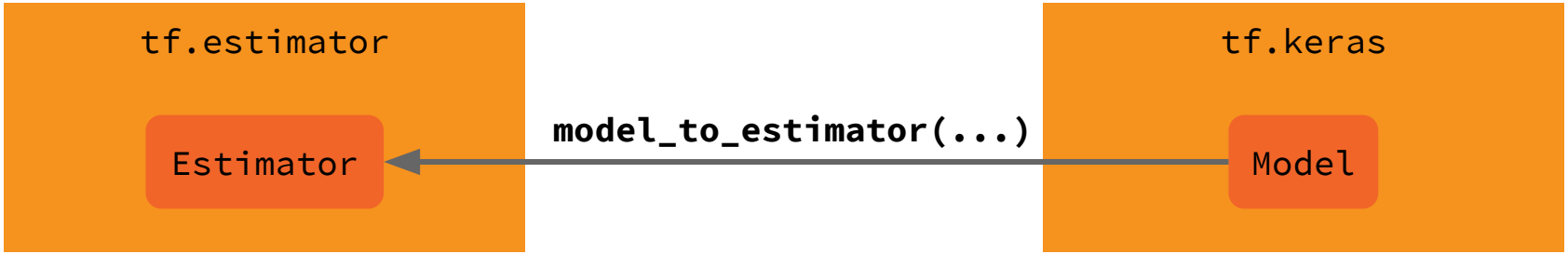

Canned Estimators
Models in a box
Estimator
Keras
Build models, train and evaluate
Layers
Build models
Python Frontend
C++ Frontend
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
...


fchollet/keras
Reference Implementation
TensorFlow backend
Multiple non-TF backends
Keras 2.0
API Spec
tf.keras
Custom TensorFlow backend
Integration with Estimators:
Distributed Execution
Integration with serving

`tf.layers` and `tf.keras`

- `tf.keras` implements Keras API spec *exactly*
- `tf.layers.Dense` contains the implementation for `tf.keras.Dense`
- `tf.keras` is as thin a wrapper as possible

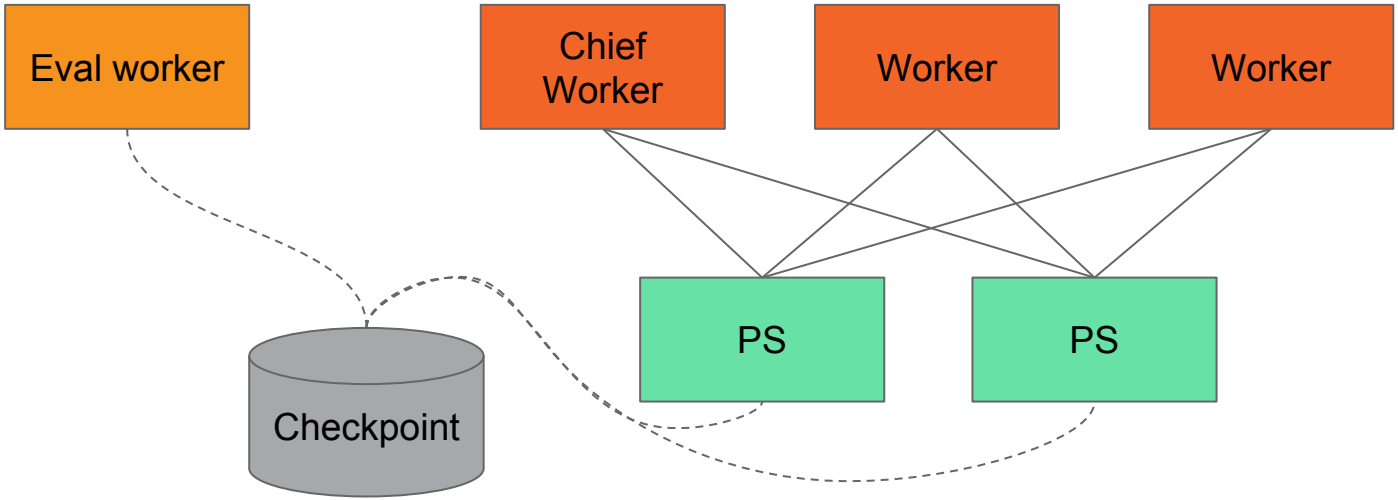
Keras/Estimator Integration



Estimators

- Implement:
  - training loop
  - checkpointing
  - Built-in integration with TensorBoard
  - Export to tensorflow/serving

- Supports complex use cases, custom inference, evaluation, etc.
- Complete implementations of standard models (linear, DNN, …)


Eval worker
Chief Worker
Worker
Worker
PS
PS
Checkpoint

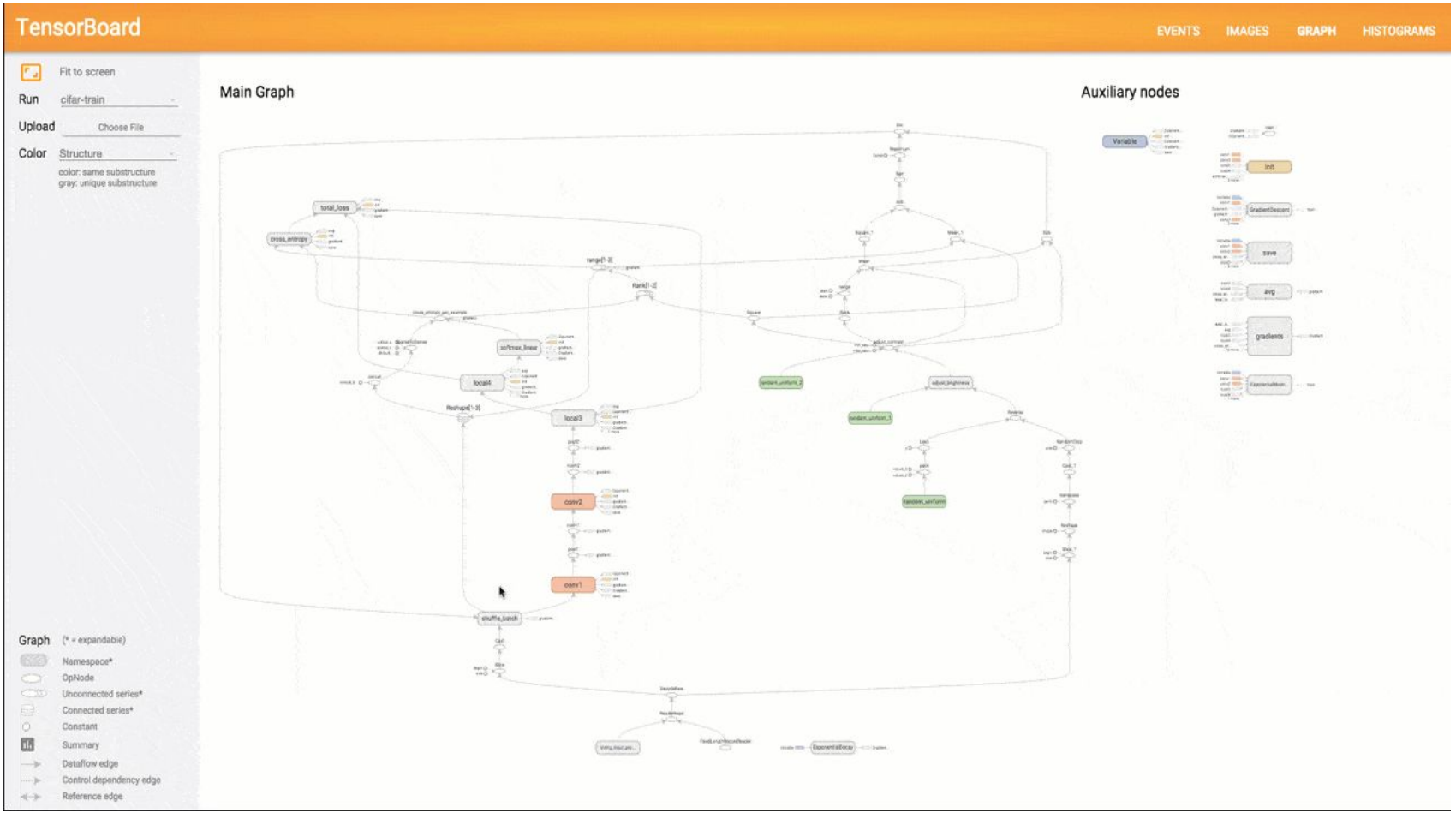
Estimators

Ready for distributed execution

- Recovery after worker failure
- Distributed device placement
- Accelerator support

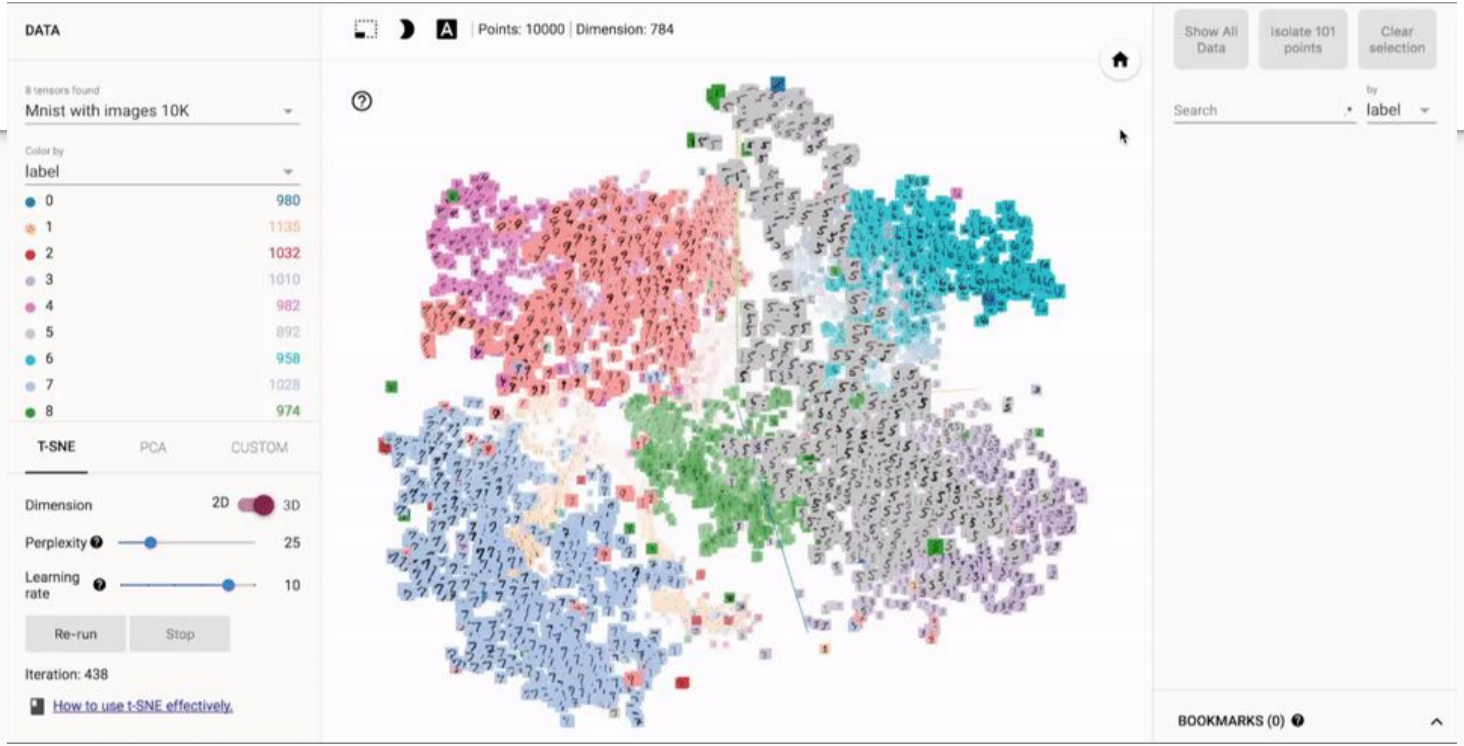
Run local or distributed; run on different devices, without changing your model.

The TensorFlow ecosystem...

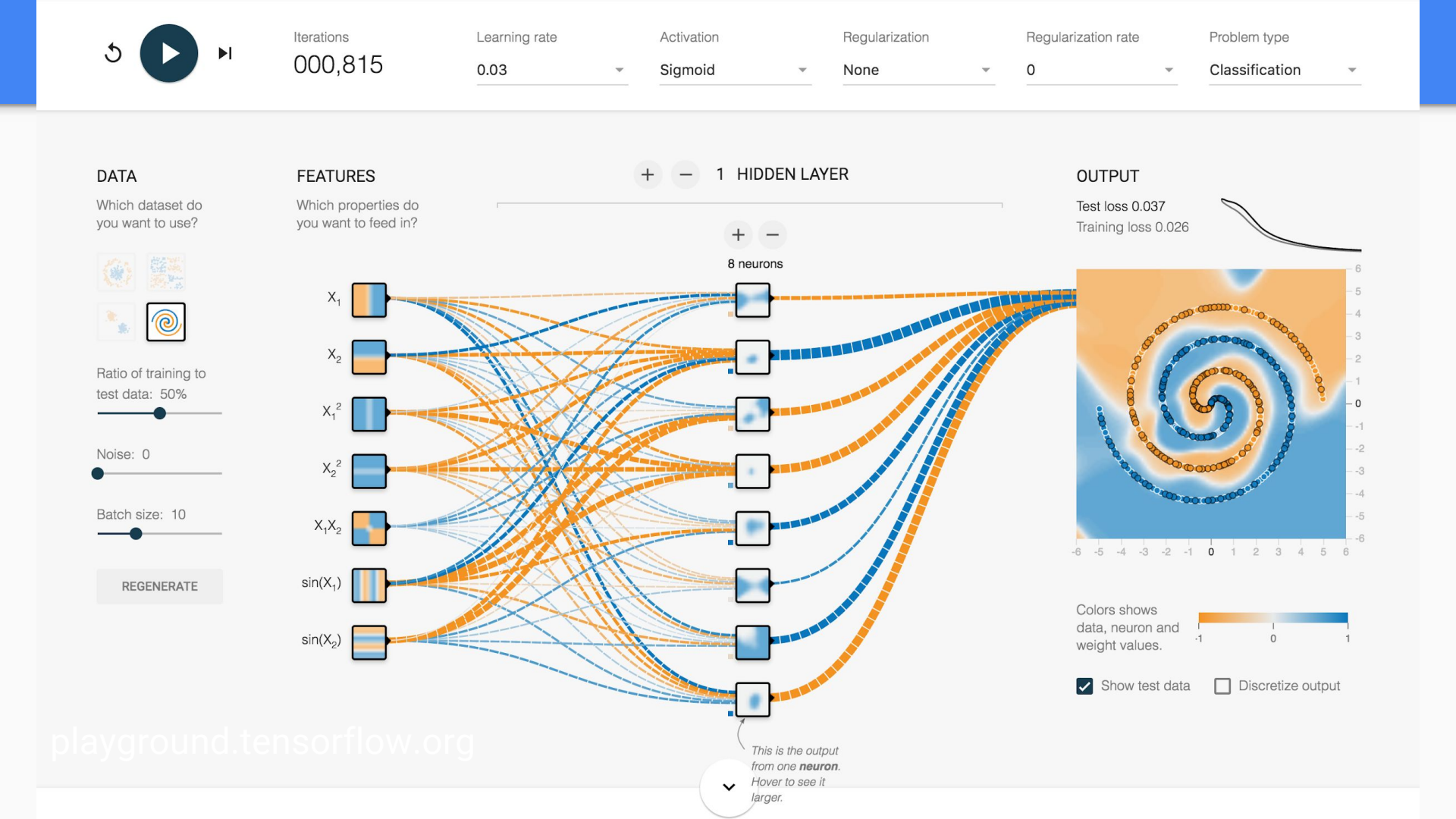


bit.ly/tf-workshop-aicon

TensorBoard



https://www.tensorflow.org/versions/r0.12/how_tos/embedding_viz/index.html


Iterations
000,815
Learning rate
0.03
Activation
Sigmoid
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
8 neurons
X1
X2
X1²
X2²
X1X2
sin(X1)
sin(X2)
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.037
Training loss 0.026
Colors shows data, neuron and weight values.
-1
0
1
Show test data
Discretize output
playground.tensorflow.org

Using Estimators: LinearClassifier and DNNClassifier

These slides: bit.ly/tf-workshop-aicon

Create a Kaggle.com account

Go to **kaggle.com/yufengg/kernels**

OR

Install TensorFlow & Jupyter: www.tensorflow.org/install/

```
pip install tensorflow pandas jupyter
```

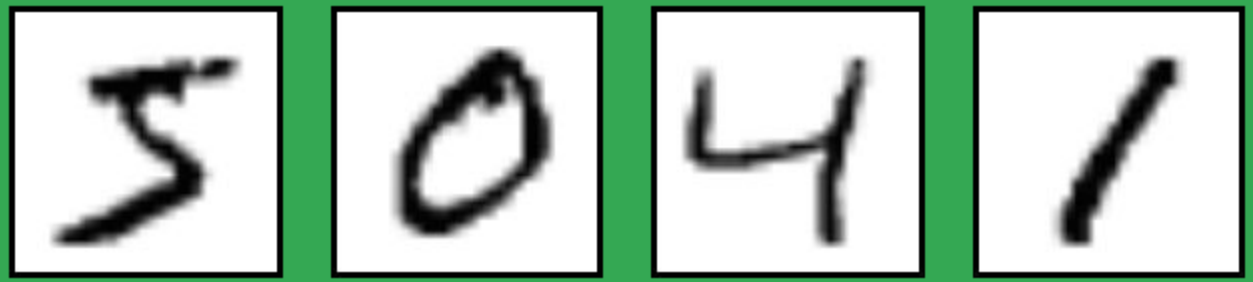
Clone GitHub Repo: bit.ly/tensorflow-workshop

Optional: download the '**Fashion-MNIST**' files:
github.com/zalandoresearch/fashion-mnist#get-the-data

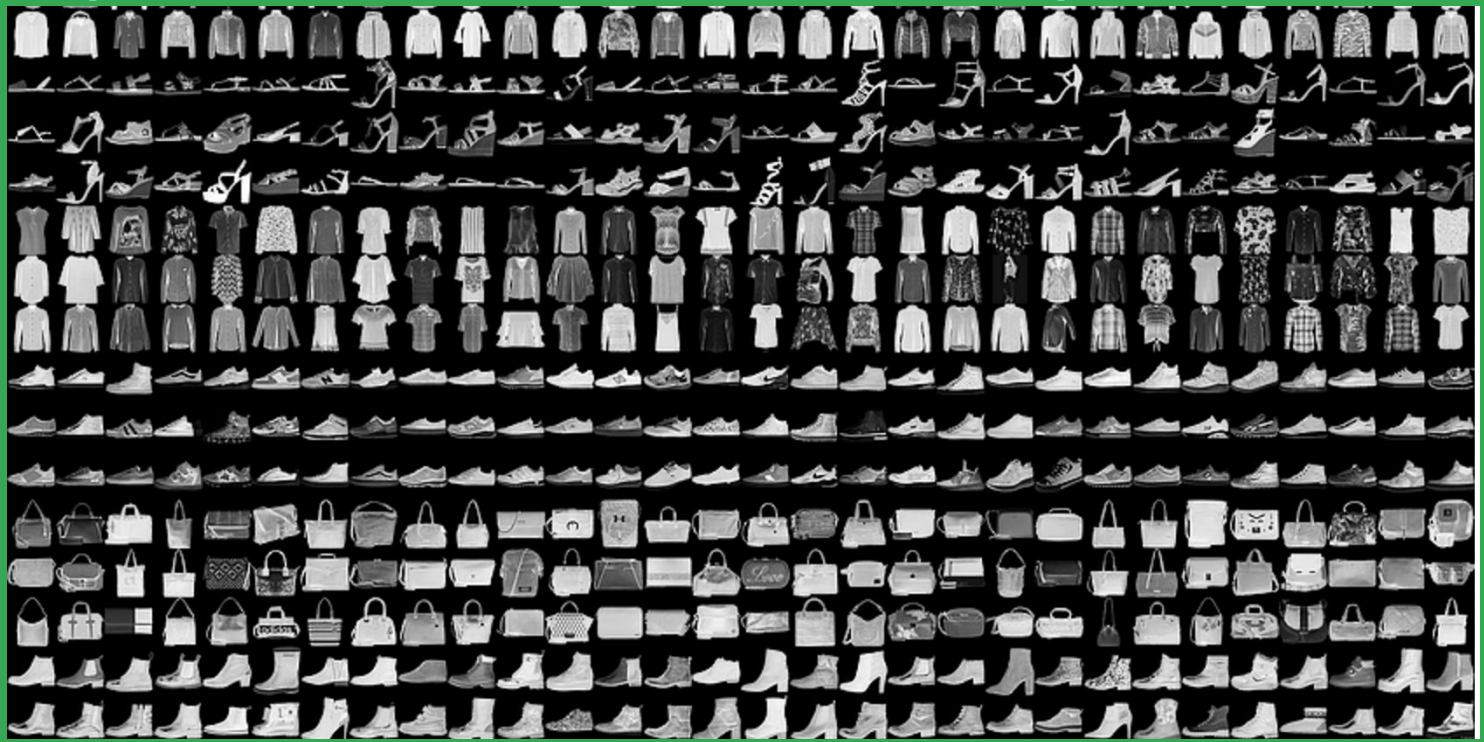
Estimator workflow

```
model =
  tf.estimator.LinearClassifier(feature_columns, … )

model.train(train_input_fn, steps)

model.evaluate(eval_input_fn)

model.predict(predict_input_fn)

model.export_savedmodel(serving_input_receiver_fn)
```

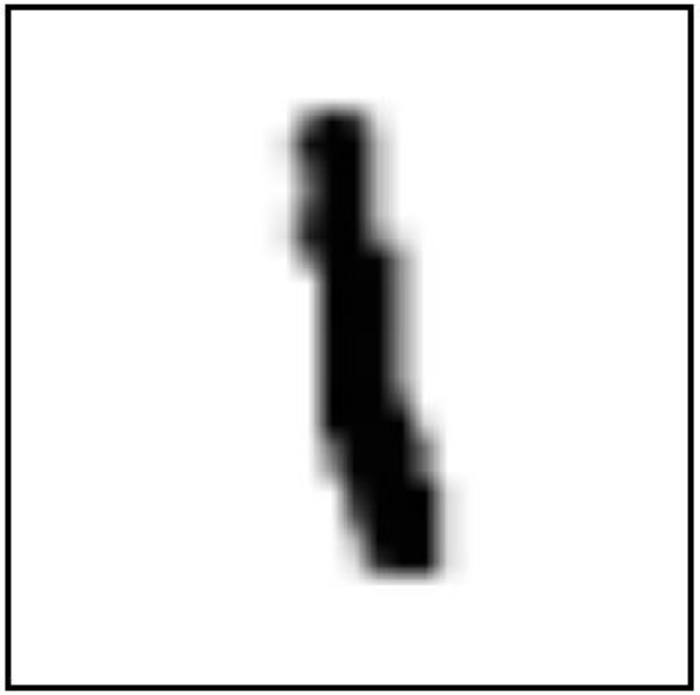
MNIST: The “Hello World” Of ML



fashion-MNIST:
Making MNIST fashionable again



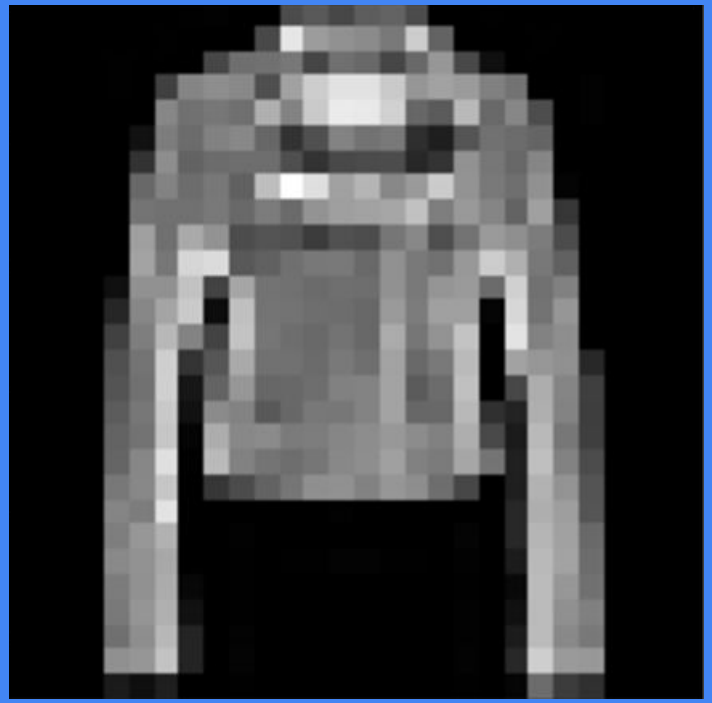
Computer Vision -- MNIST



≈

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Computer Fashion -- fashion-MNIST

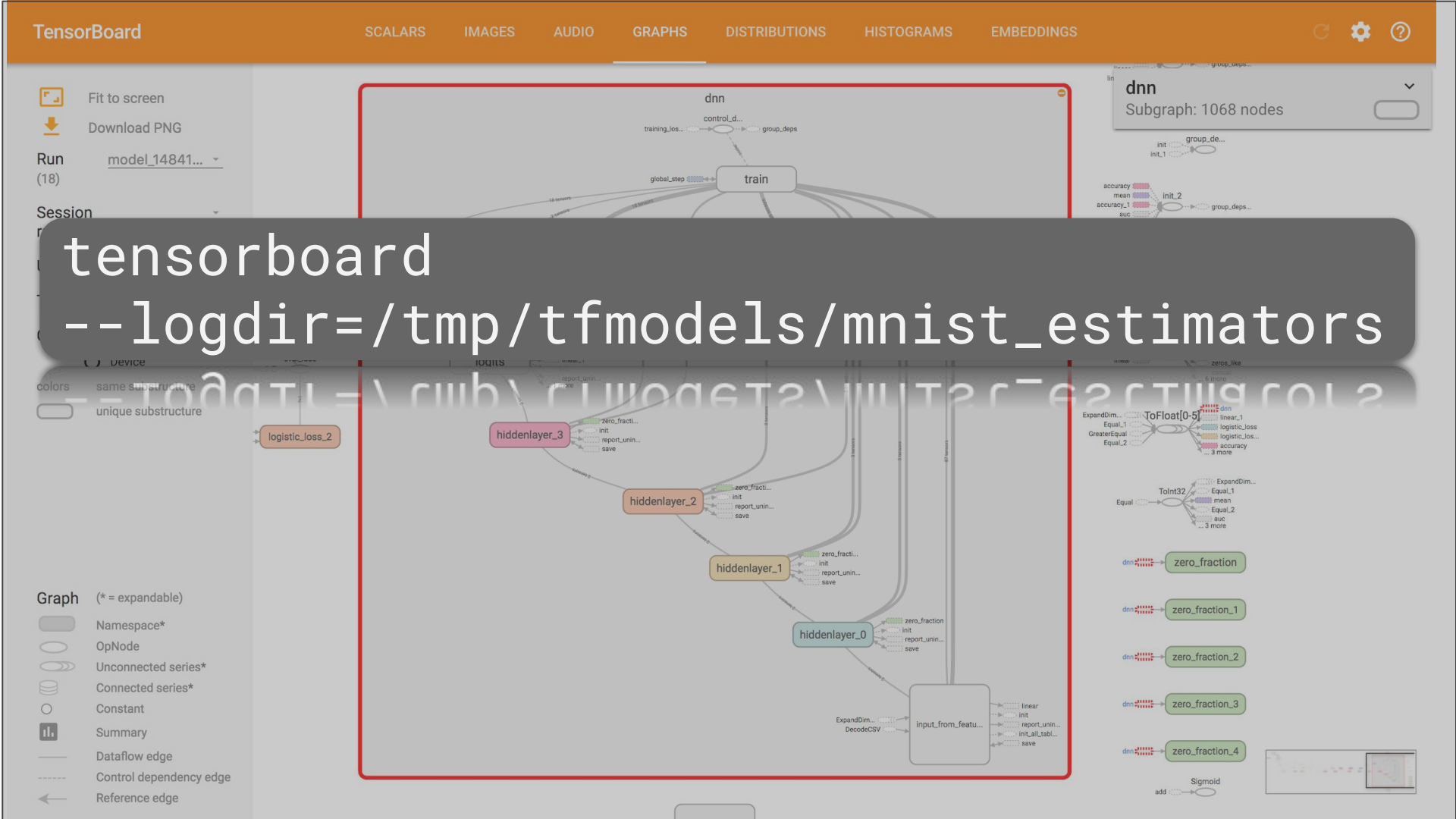


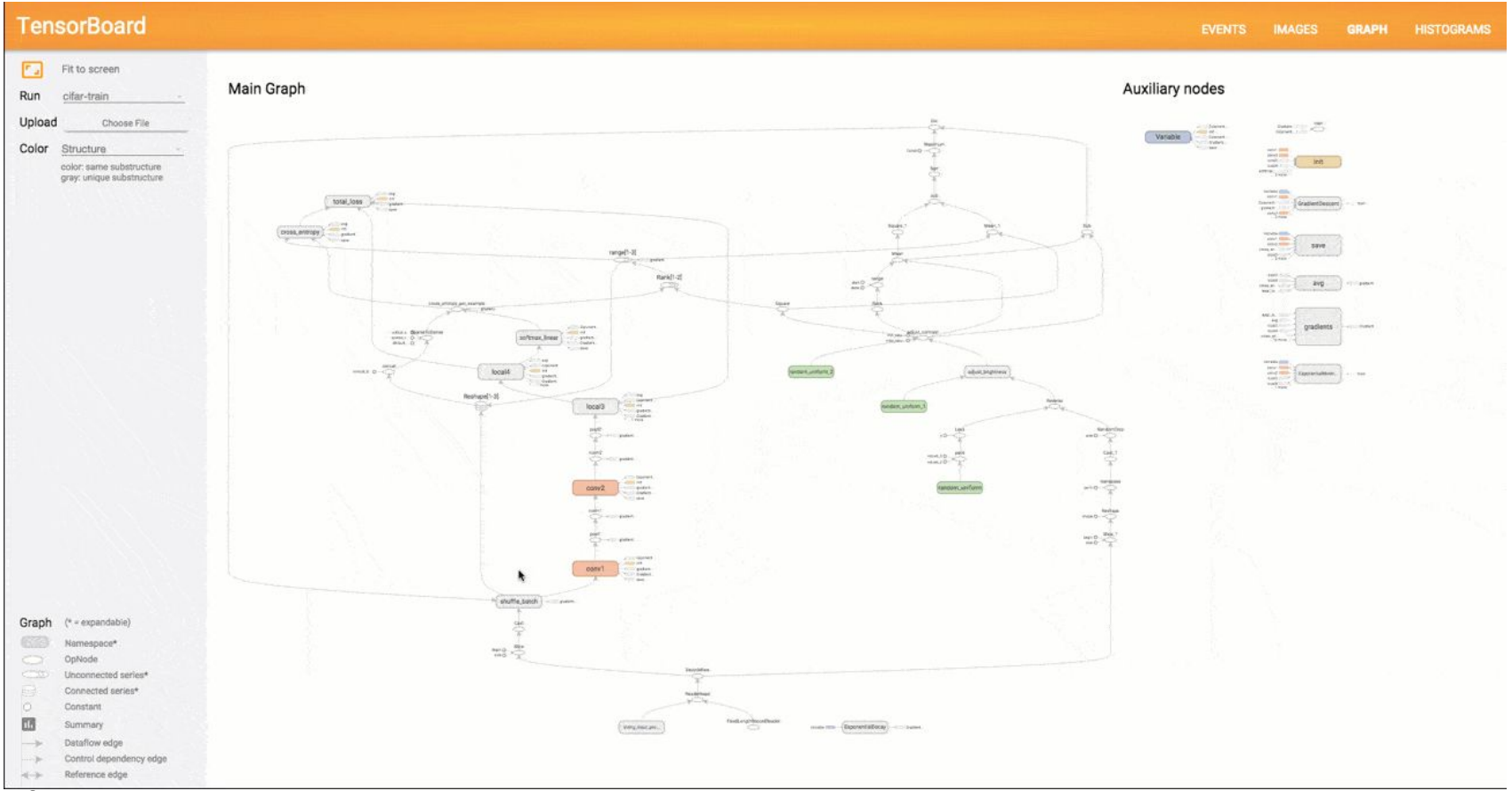
Lab: using TensorFlow's Estimator APIs

Fork my notebook:

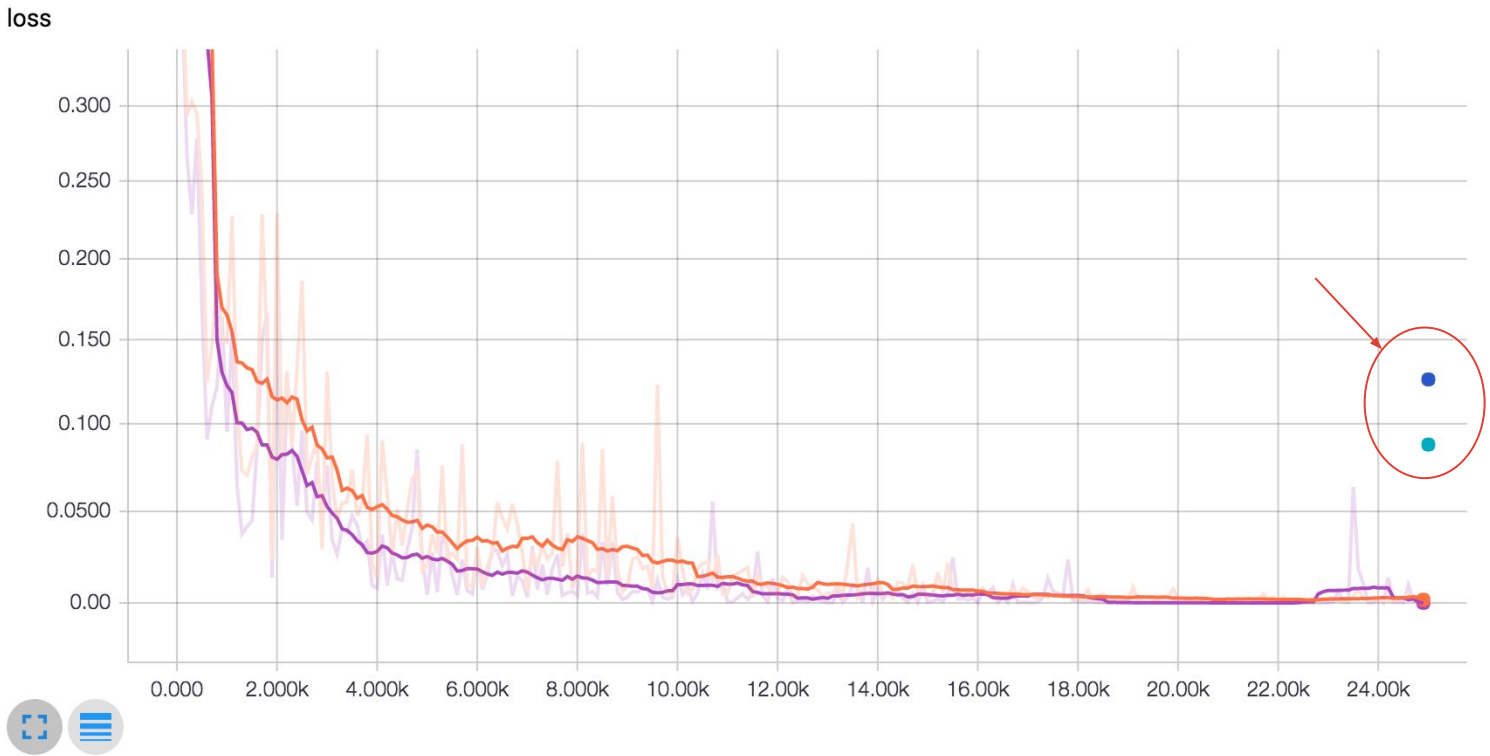
`kaggle.com/yufengg/fashion-mnist`

Wait, how do I fork a notebook???


TensorBoard
SCALARS
IMAGES
AUDIO
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
EMBEDDINGS
Fit to screen
Download PNG
Run
model_14841...
(18)
Session
dnn
Subgraph: 1068 nodes
tensorboard
--logdir=/tmp/tfmodels/mnist_estimators
Graph
(* = expandable)
Namespace*
OpNode
Unconnected series*
Connected series*
Constant
Summary
Dataflow edge
Control dependency edge
Reference edge
unique substructure
train
hiddenlayer_3
hiddenlayer_2
hiddenlayer_1
hiddenlayer_0
logistic_loss_2
zero_fraction
zero_fraction_1
zero_fraction_2
zero_fraction_3
zero_fraction_4



bit.ly/tf-workshop-aicon



Comparing two different optimizers for DNNClassifier

Show data download links

Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

0.831

Horizontal Axis

STEP RELATIVE WALL

Runs

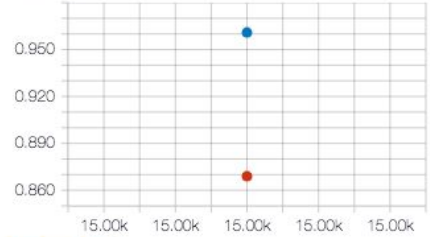
Write a regex to filter runs

- fashion/.
- fashion/eval
- reg/.
- reg/eval

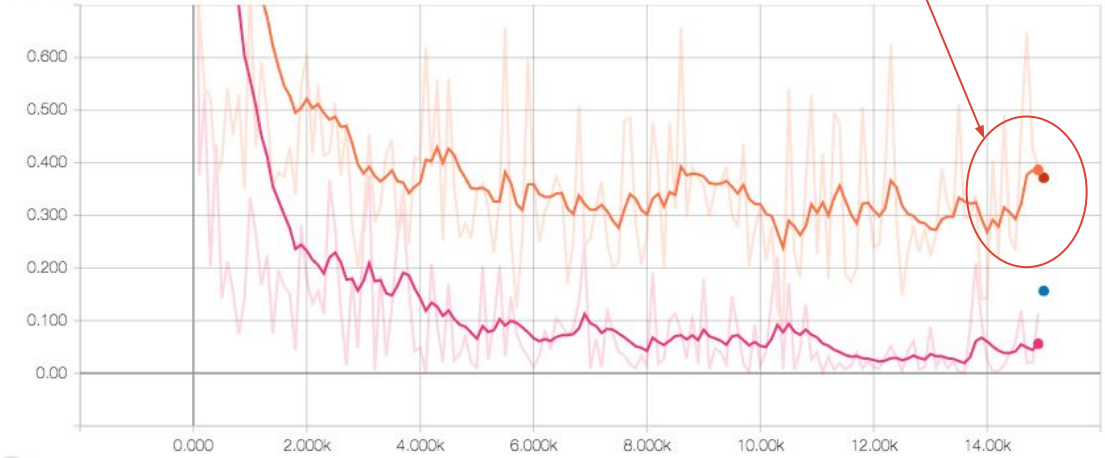
.*

Tags matching /.*/ (all tags) 8

accuracy



average_loss



fashion-mnist

What’s TensorFlow made of?

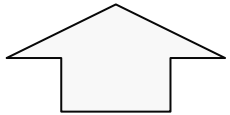
A multidimensional array.

A graph of operations.

Tensors - generalized **matrices**

Tensors have a Shape that's described with a *vector*.

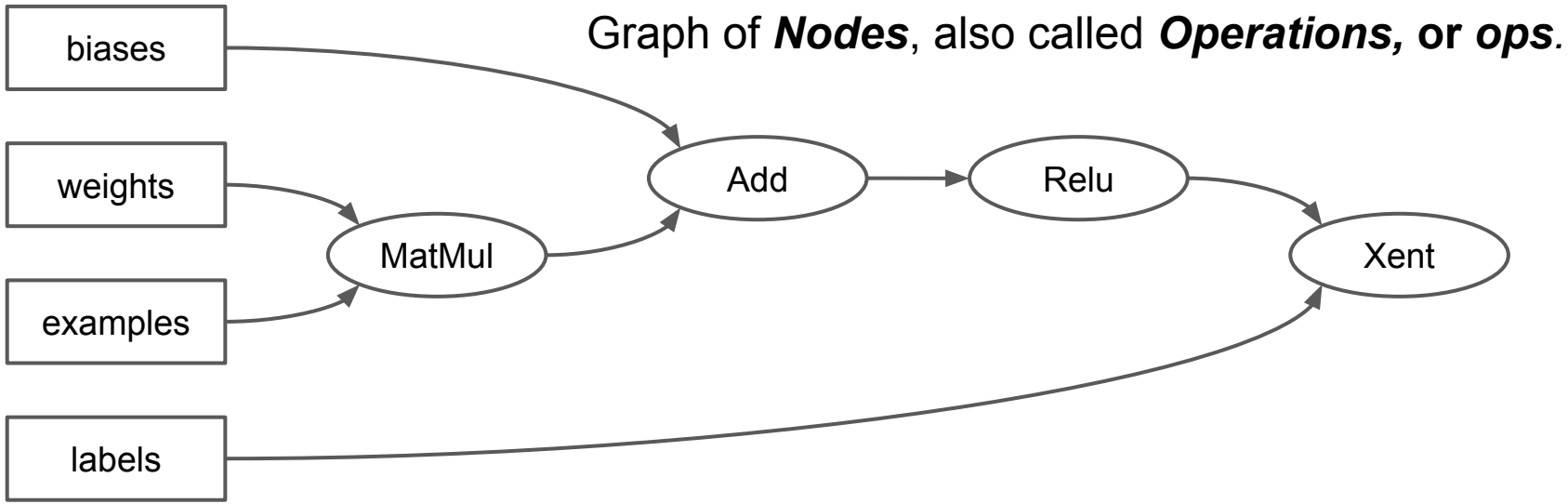
[10000, 256, 256, 3]



- **10000** Images
- Each Image has **256 Rows**
- Each Row has **256 Pixels**
- Each Pixel has **3 channels** (RGB)

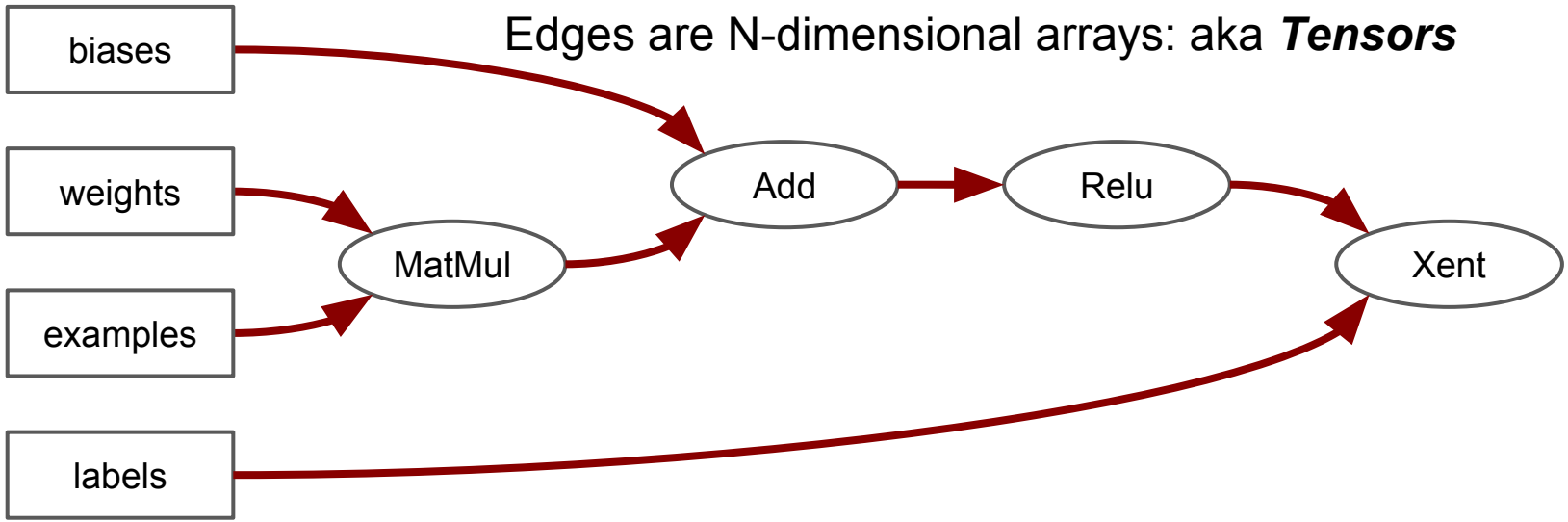
bit.ly/tf-workshop-aicon

Computation is a dataflow graph



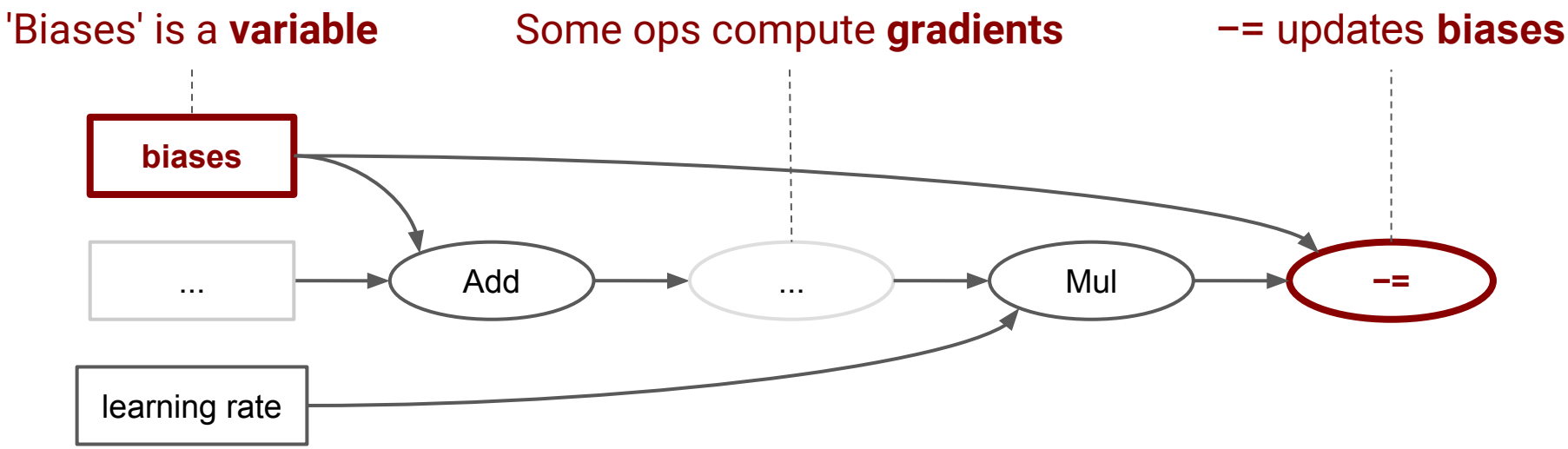
Computation is a dataflow graph

with tensors



Computation is a dataflow graph

with state



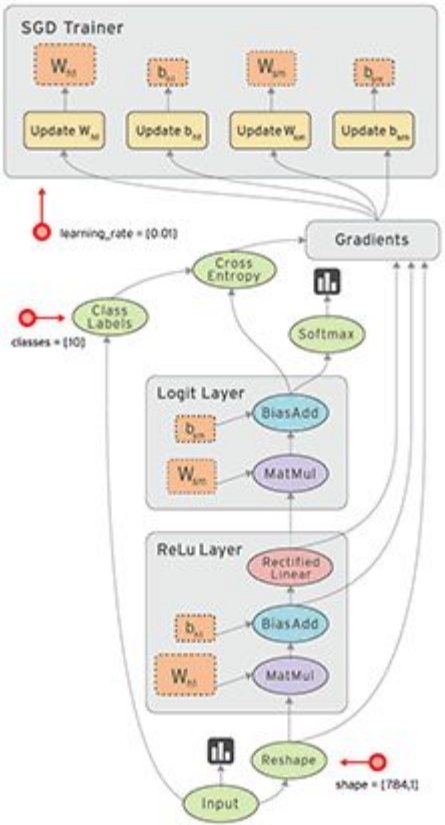
TensorFlow

A multidimensional array.

A graph of operations.

Operates over **tensors**: *n-dimensional arrays*
Using a **flow graph**: *data flow computation framework*

- Flexible, intuitive construction
- Support for threads, queues, and asynchronous computation; **distributed runtime**
- **automatic differentiation**
- Train on CPUs, GPUs, ...and coming soon, **TPUS**...
- Run wherever you like



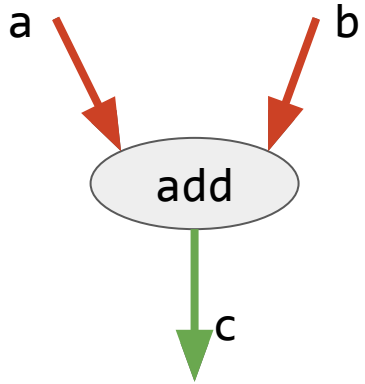
bit.ly/tf-workshop-aicon

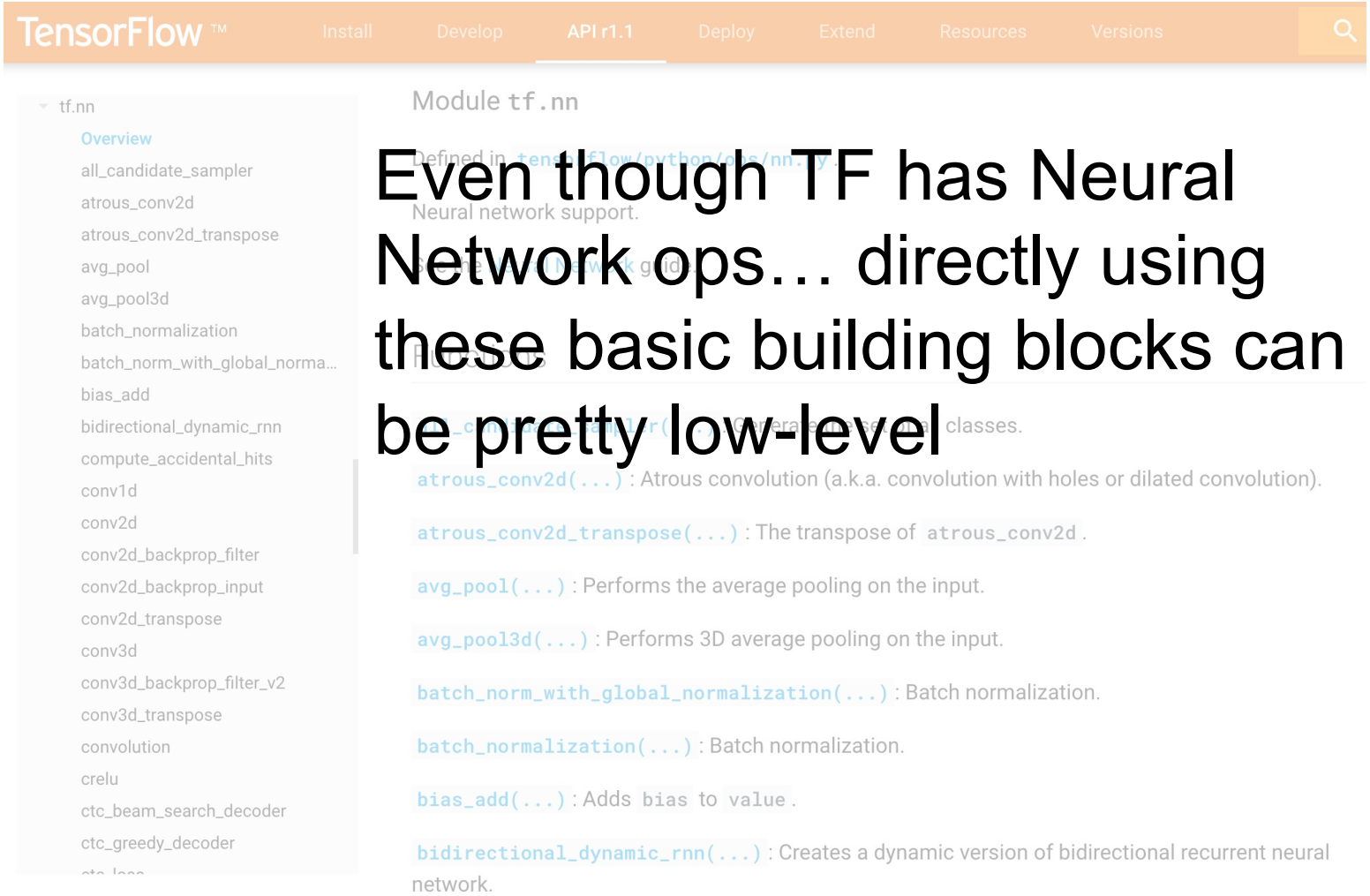
Building a computation graph, then executing it:

- **Portability**: the graph can be executed immediately, or saved to use later, and it can run on multiple platforms: CPUs, GPUs, TPUs, mobile, embedded.
  - Deploy to production without having to depend on any of the code that built the graph, only the runtime necessary to execute it.
- **Transformable** and **Optimizable**:
  - produce a more optimal version of the graph for a given platform.
  - perform memory or compute optimizations, and trade off between them.
- Support for **distributed execution**.

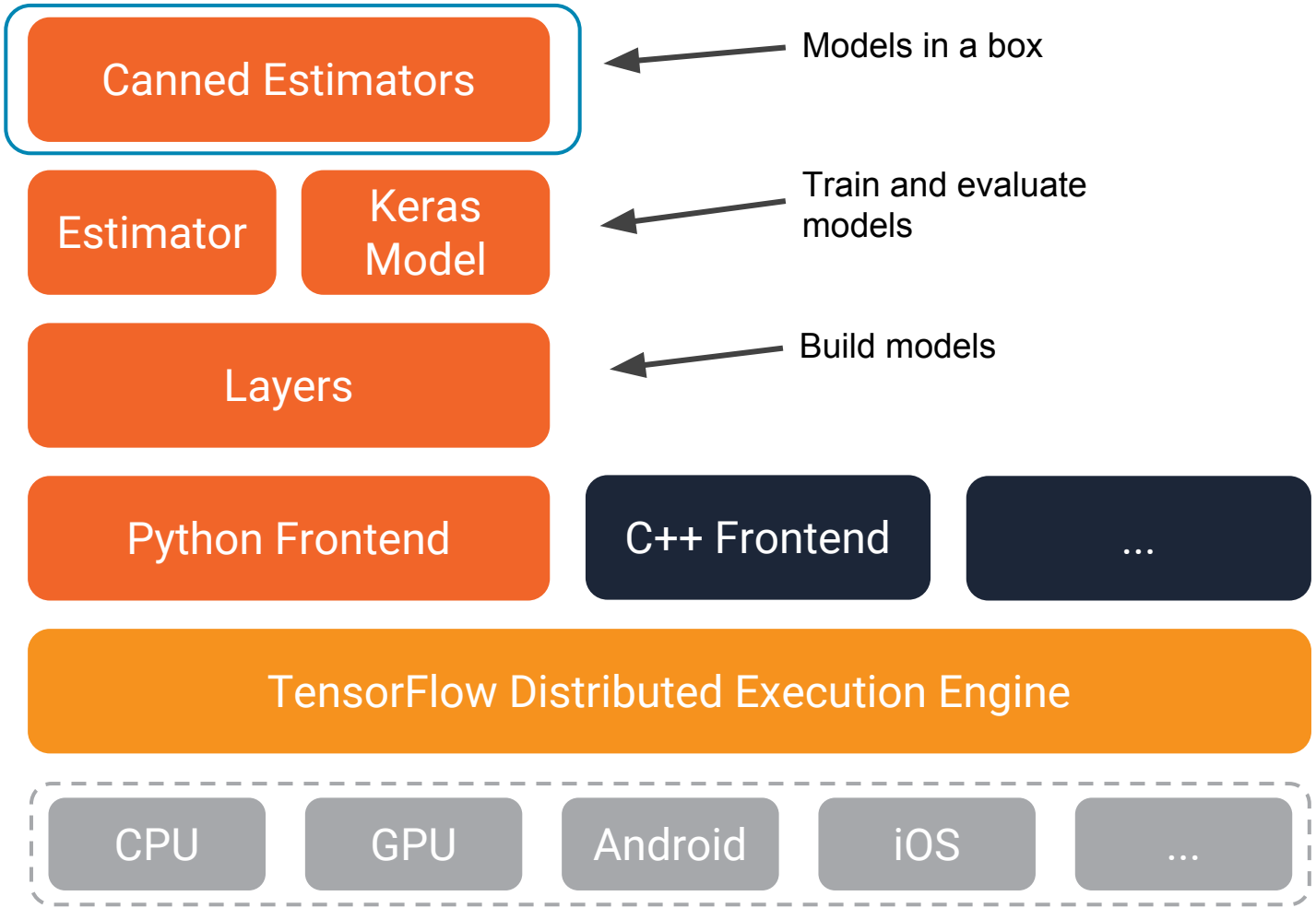
Build a graph; *then* run it.

```
...
# build graph
c = tf.add(a, b)

...

with tf.Session() as session:
    # run graph
    value_of_c = session.run(c, {a=1, b=2})
```





Even though TF has Neural Network ops… directly using these basic building blocks can be pretty low-level


Canned Estimators
Models in a box
Estimator
Keras Model
Train and evaluate models
Layers
Build models
Python Frontend
C++ Frontend
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
...

Many levels of abstraction

Choose the right one for you:

- Layers, losses, metrics
- Training/Eval loop functions
- Estimator (BaseEstimator)
  - Any model you want, but must separate input from the rest of the model
- Predefined estimators
  - `LinearClassifier, DNNClassifier, DNNRegressor, …` **`DNNLinearCombinedClassifier`**
  - Limited configuration options: feature columns, metrics.

These slides: bit.ly/tf-workshop-aicon

Create a Kaggle.com account

Go to **kaggle.com/yufengg/kernels**

OR

Install TensorFlow & Jupyter: www.tensorflow.org/install/

```
pip install tensorflow pandas jupyter
```

Clone GitHub Repo: bit.ly/tensorflow-workshop

Optional: download the '**Fashion-MNIST**' files:
github.com/zalandoresearch/fashion-mnist#get-the-data

Using the TensorFlow High-level APIs with tf.data

kaggle.com/yufengg/mushrooms-with-datasets-and-estimators

Too much to type?
Try this instead

kaggle.com/yufengg/kernels

But first, let's look at Facets

The `Dataset` interface

Data sources and functional transformations

Create a `Dataset` from one or more `tf.Tensor` objects:

```
Dataset.from_tensors((features, labels))
Dataset.from_tensor_slices((features, labels))
TextLineDataset(filenames)
```

The `Dataset` interface

Data sources and functional transformations

Or create a `Dataset` from another `Dataset`:

```
dataset.map(lambda x: tf.decode_jpeg(x))
dataset.repeat(NUM_EPOCHS)
dataset.batch(BATCH_SIZE)
```

...and many more.

The `Dataset` interface

Data sources and functional transformations

Or create a `Dataset` from a Python generator:

```
def generator():
  while True:
    yield ...

dataset.from_generator(generator, tf.int32)
```

Let's go to the land of the mushrooms!

kaggle.com/yufengg/mushrooms-with-datasets-and-estimators

“Wide & Deep”:
Using the TensorFlow High-level APIs and feature engineering

Motivation - a "magical" food app

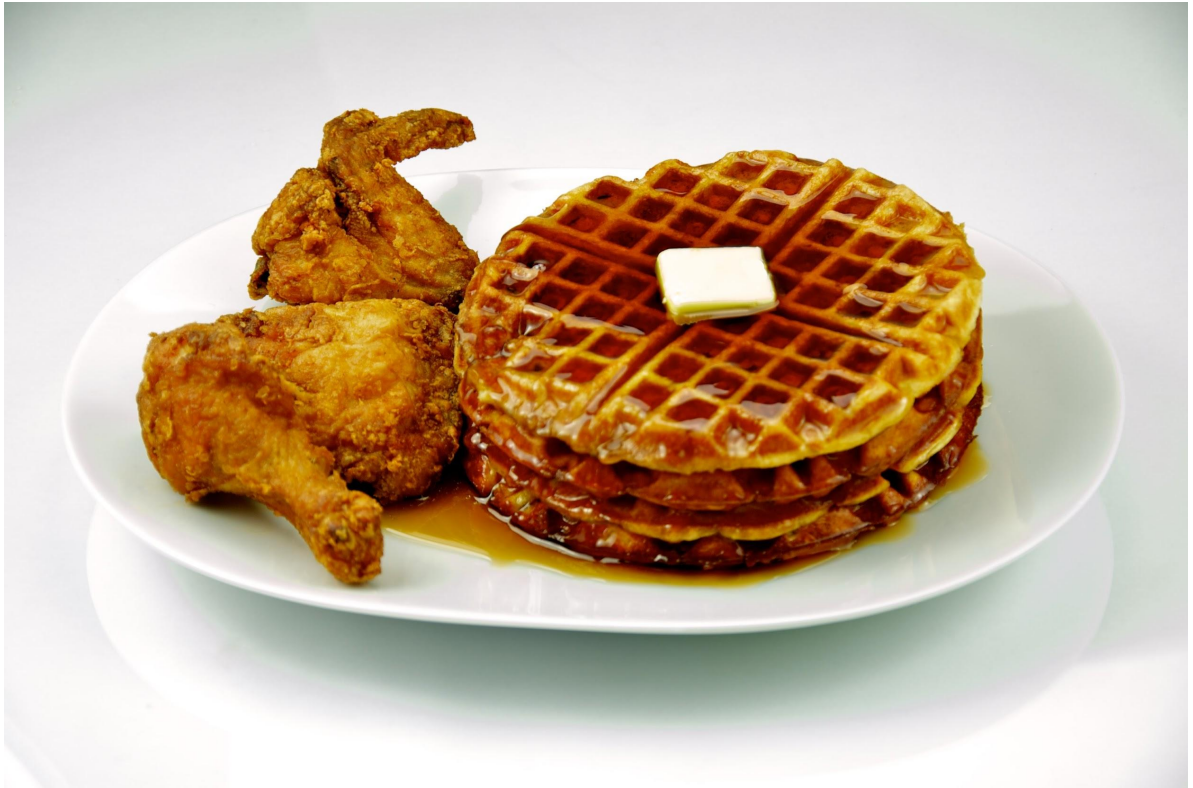
bit.ly/tf-workshop-aicon

Launch and Iterate!

https://upload.wikimedia.org/wikipedia/commons/2/2c/Fried-Chicken-Set.jpg

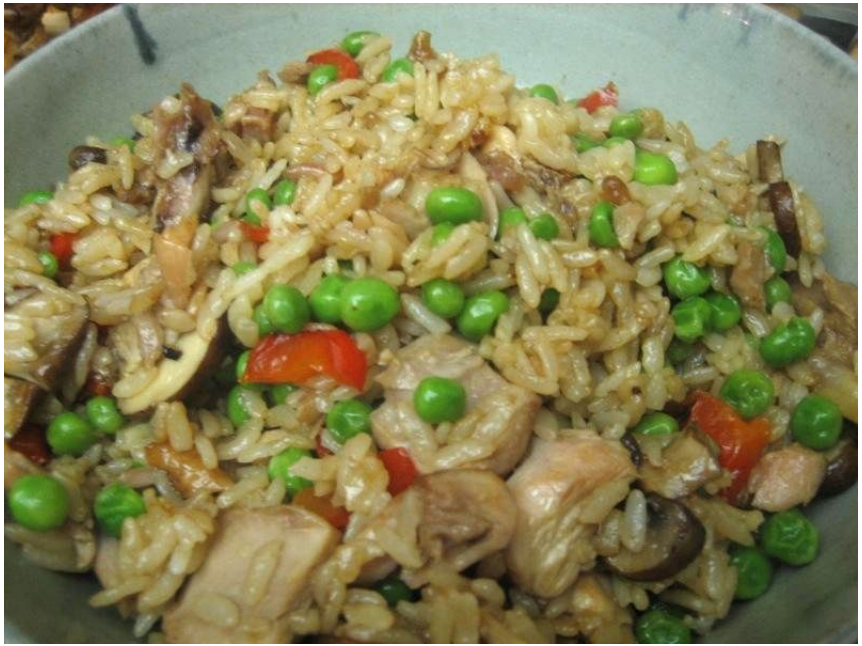
bit.ly/tf-workshop-aicon

Launch and Iterate!



https://upload.wikimedia.org/wikipedia/commons/f/fd/Chicken_and_Waffles_201_-_Evan_Swigart.jpg

Launch and Iterate!



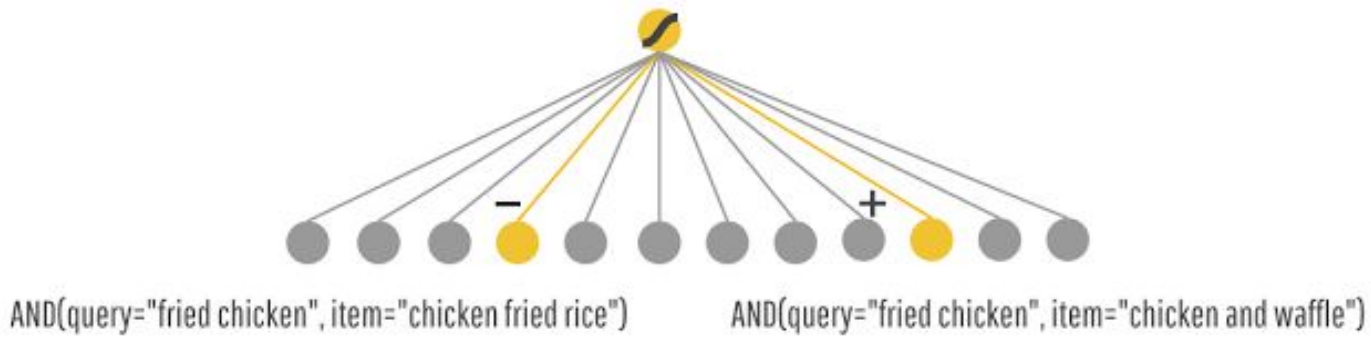
… naive character matching

https://www.flickr.com/photos/elsiehui/9575196704
https://www.flickr.com/photos/patrickwoodward/2208031777

bit.ly/tf-workshop-aicon

v2.0: memorize all the things

- Train a linear TF model



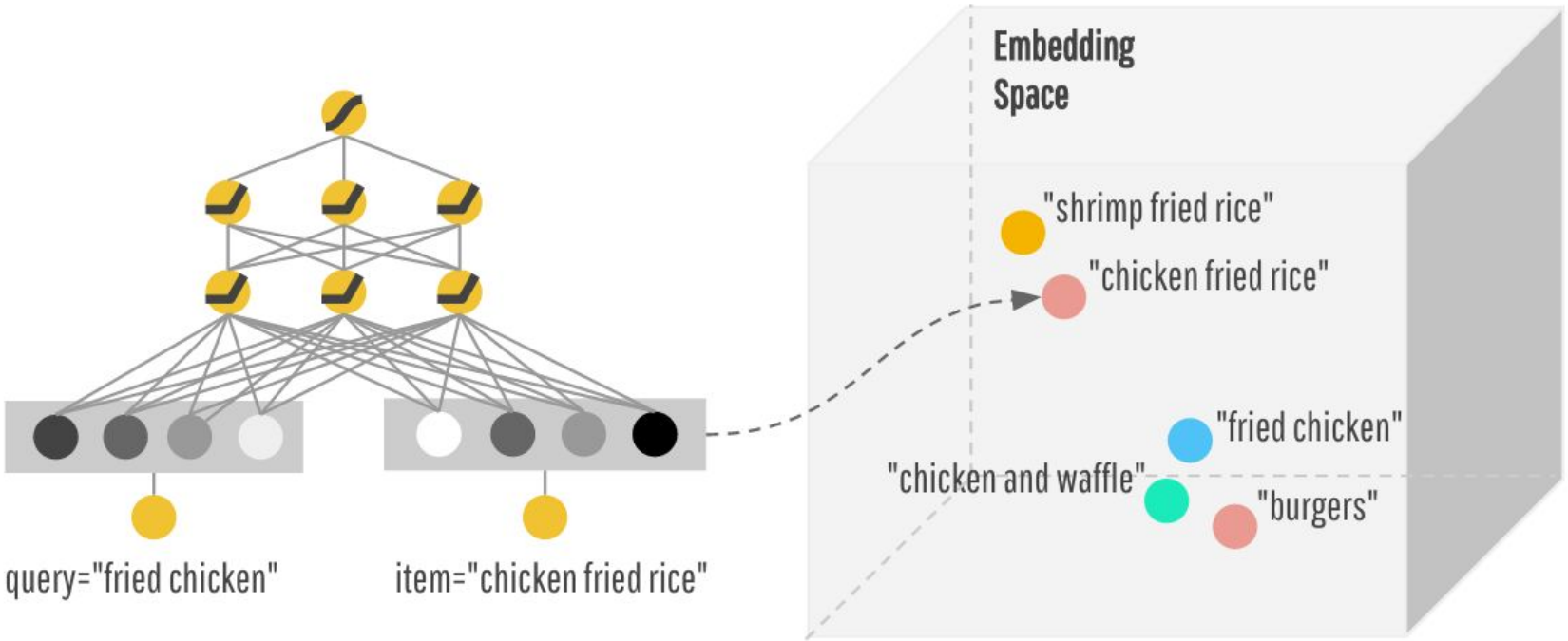
- Your app is gaining traction!

Problem: Your users are bored!

https://upload.wikimedia.org/wikipedia/commons/f/fd/Chicken_and_Waffles_201_-_Evan_Swigart.jpg

bit.ly/tf-workshop-aicon

v3.0: More generalized recommendations for all



bit.ly/tf-workshop-aicon

No good deed goes unpunished

- Some recommendations are "too general"
  - Irrelevant dishes are being sent
- Your users are still picky 😥

No good deed goes unpunished

!=

hot latte with whole milk

iced decaf latte with nonfat milk

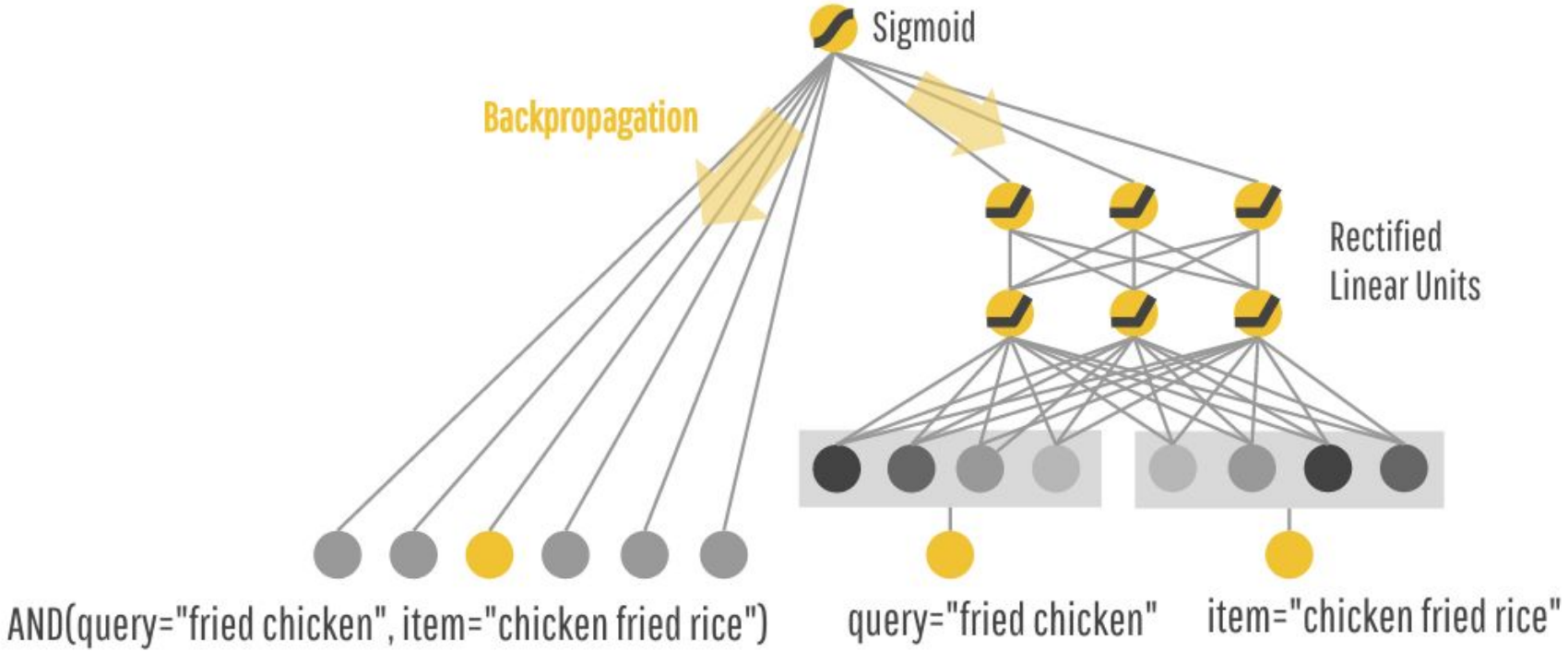
bit.ly/tf-workshop-aicon

Two types of requests:

specific and general

How to balance this?

v4.0: Why not both?



bit.ly/tf-workshop-aicon

Wide & Deep

memorization
relevance

generalization
diversity

Lab:
Wide and Deep:
Using TensorFlow's high-level APIs

kaggle.com/yufengg/census-with-w-d-estimators

These slides: bit.ly/tf-workshop-aicon

Create a Kaggle.com account

Go to **kaggle.com/yufengg/kernels**

OR

Install TensorFlow & Jupyter: www.tensorflow.org/install/

```
pip install tensorflow pandas jupyter
```

Clone GitHub Repo: bit.ly/tensorflow-workshop

Optional: download the '**Fashion-MNIST**' files:
github.com/zalandoresearch/fashion-mnist#get-the-data

Meet our dataset: US Census Data

- Just as exciting as chicken and waffles
- **Task**: predict the probability that the individual has an annual income of over 50K dollars
- Over 32k examples
- Extracted from the 1994 US Census by Barry Becker.

Meet our dataset: US Census Data

Column Name	Type	Description
age	Continuous	The age of the individual
workclass	Categorical	The type of employer the individual has (government, military, private, etc.).
fnlwgt	Continuous	The number of people the census takers believe that observation represents (sample weight). This variable will not be used.
education	Categorical	The highest level of education achieved for that individual.
education_num	Continuous	The highest level of education in numerical form.
marital_status	Categorical	Marital status of the individual.

Meet our dataset: US Census Data

Column Name	Type	Description
occupation	Categorical	The occupation of the individual.
relationship	Categorical	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race	Categorical	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex	Categorical	Female, Male.
capital_gain	Continuous	Capital gains recorded.
capital_loss	Continuous	Capital Losses recorded.

Meet our dataset: US Census Data

Column Name	Type	Description
hours_per_week	Continuous	Hours worked per week.
native_country	Categorical	Country of origin of the individual.
income_bracket	**Categorical**	**">50K" or "<=50K", meaning whether the person makes more than $50,000 annually.**

The return of Facets

Typical structure

- Load data
- Set up feature columns
- Create your model
- Run the training loop (fit the model)
- Evaluate your model's accuracy (and other metrics)
- (optional) Predict new examples

Estimators high-level structure

```
feature_columns = tf.feature_column.numeric_column(
    "pixels", shape=784)
model = tf.estimator.DNNClassifier(
        [layer2_hidden_units, layer1_hidden_units],
        feature_columns=feature_columns,
        n_classes=2
)
model.train(input_fn=training_input_fn, steps=1000)
model.evaluate(input_fn=evaluate_input_fn)
model.predict(input_fn=predict_input_fn)
```

To the code!

kaggle.com/yufengg/census-with-w-d-estimators

Typical structure

- **Load data**
- Set up feature columns
- Create your model
- Run the training loop (fit the model)
- Evaluate your model's accuracy (and other metrics)
- (optional) Predict new examples

Input data format

```
features = {
  'hours_per_week': array([16, 45, 50], dtype=int32),

  'relationship': SparseTensorValue(indices=array([[0, 0],[1, 0],[2, 0]]), values=array([' Not-in-family', ' Husband', ' Not-in-family'], dtype=object), shape=array([3, 1])),

  'sex': SparseTensorValue(indices=array([[0, 0],[1, 0],[2, 0]]), values=array([' Female', ' Male', ' Female'], dtype=object), shape=array([3, 1])),

  'age': array([49, 52, 31], dtype=int32)
...
}
labels = [0 1 1]
```

bit.ly/tf-workshop-aicon

Input data format

```
df = pandas.read_csv(filename)
labels = df["labels"]

return tf.estimator.inputs.pandas_input_fn(
      x=df, y=labels)
```

bit.ly/tf-workshop-aicon

Typical structure

- Load data
- **Set up feature columns**
- Create your model
- Run the training loop (fit the model)
- Evaluate your model's accuracy (and other metrics)
- (optional) Predict new examples

Feature columns

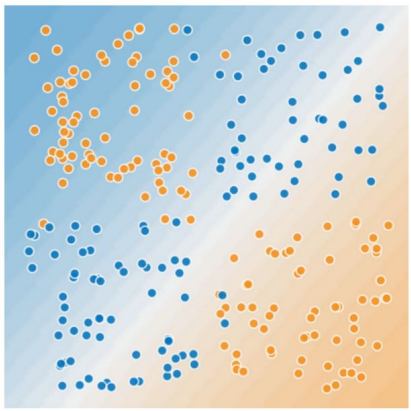
```
# Sparse base columns.
sex = categorical_column_with_vocabulary_list(
    key="sex", vocabulary_list=["female", "male"])
education = categorical_column_with_hash_bucket(
    key="education", hash_bucket_size=1000)
...
# Continuous base columns.
age = numeric_column("age")
...
```

Feature columns continued

```
# Transformations.
age_buckets = bucketized_column(
    age, boundaries=[18, 25, 30, 35, 40, 45, 50, 55, 60, 65])
education_occupation =crossed_column(
    ["education", "occupation"], hash_bucket_size=int(1e4))
...
# embeddings for deep learning
embedding_column(workclass, dimension=8)
```

Feature columns continued


```
# Transformations.
age_buckets = layers.bucketized_column(
    age, boundaries=[18, 25, 30, 35, 40, 45, 50, 55, 60, 65])
education_occupation = layers.crossed_column(
    [education, occupation], hash_bucket_size=int(1e4))
...
# embeddi                                    ing
layers.emb                                   kclas
```



Typical structure

- Load data
- Set up feature columns
- **Create your model**
- Run the training loop (fit the model)
- Evaluate your model's accuracy (and other metrics)
- (optional) Predict new examples

Make the model (Estimator)

```
m = tf.estimator.DNNLinearCombinedClassifier(
      model_dir=model_dir,
      linear_feature_columns=wide_columns,
      dnn_feature_columns=deep_columns,
      dnn_hidden_units=[100, 70, 50, 25])
```

Typical structure

- Load data
- Set up feature columns
- Create your model
- **Run the training loop**
- **Evaluate your model's accuracy (and other metrics)**
- (optional) Predict new examples

Train and Evaluate

```
m.train(input_fn=train_input_fn)

results = m.evaluate(input_fn=eval_input_fn)


print('Accuracy: %s' % results['accuracy'])
```

Make Predictions!

```
predict_input_fn =
    tf.estimator.inputs.pandas_input_fn(
          x=data_predict, batch_size=1,
          num_epochs=1, shuffle=False)


predictions =
    m.predict(input_fn=predict_input_fn)
```

Checkpointing and reloading a trained model

- Everything is stored in the `model_dir` folder

```
m = tf.estimator.DNNLinearCombinedClassifier(
    model_dir=model_dir,
    linear_feature_columns=wide_columns,
    dnn_feature_columns=deep_columns,
    dnn_hidden_units=[100, 70, 50, 25])
```

- If you run multiple `train` operations on the same `Estimator` and supply the same directory, training will resume where it left off.

bit.ly/tensorflow-workshop

bit.ly/tf-workshop-aicon

Next: Tweak the TensorFlow model

Exploring the Wide & Deep Model's characteristics

Ideas: TensorFlow model tweaking

- Learning Rate
- Layers (# of layers, size of layers)
- Optimizer
- Feature crosses, buckets, embedding
- … your dataset???

Building Custom Estimators for CNN models

Fashion-MNIST revisited

(Maybe CNNs will let us do better with ‘Fashion-MNIST’)

Brief intro to convolutional neural nets (CNNs)

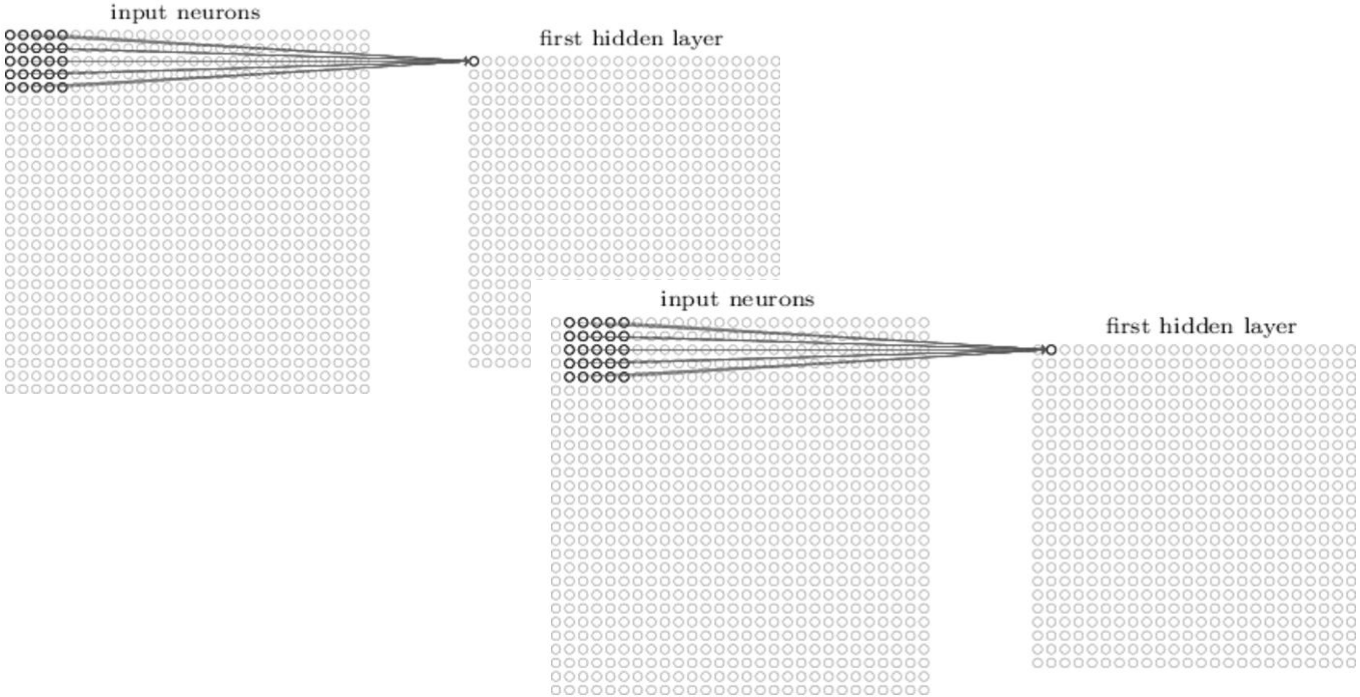
Convolutional neural networks

- A fully-connected NN does not take into account spatial structure of the images
- What if instead we use a NN that tries to leverage spatial structure?

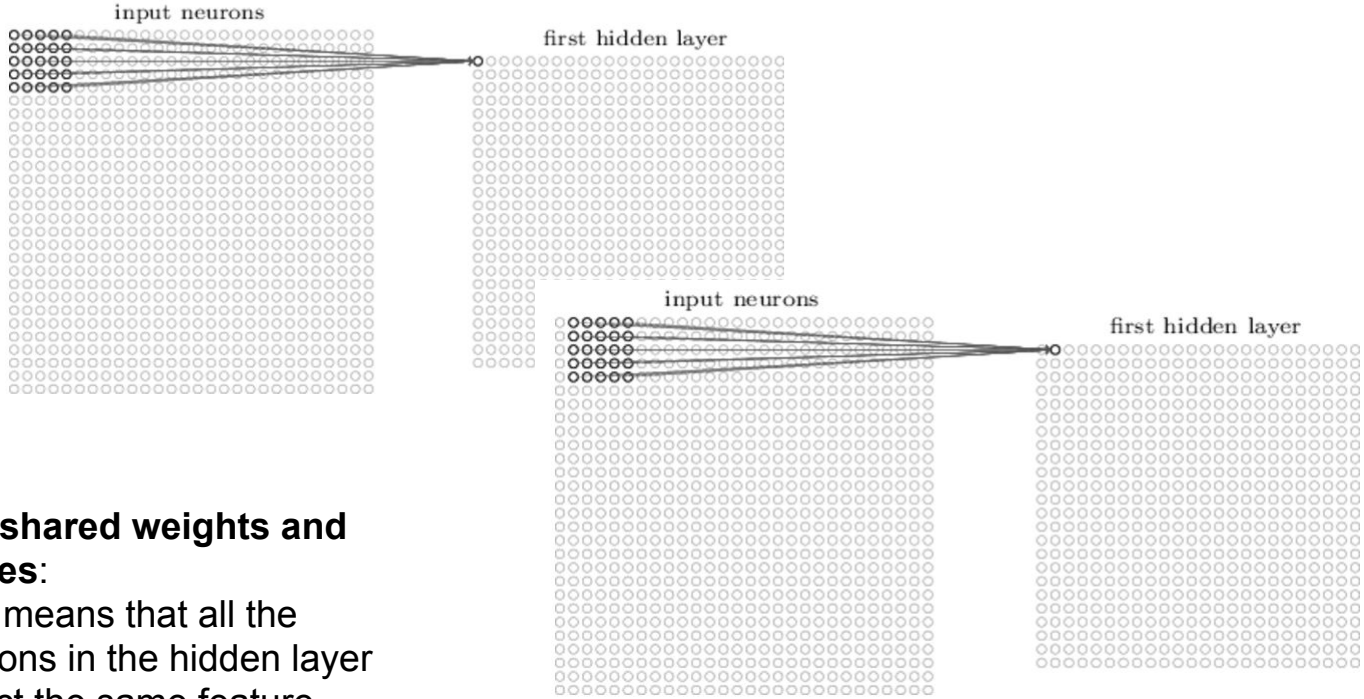
Convolutional neural networks

Three basic ideas:

- *local receptive fields*
- shared weights
- pooling



From: http://neuralnetworksanddeeplearning.com/chap6.html#introducing_convolutional_networks



Add **shared weights and biases**:
This means that all the neurons in the hidden layer detect the same feature

From: http://neuralnetworksanddeeplearning.com/chap6.html#introducing_convolutional_networks

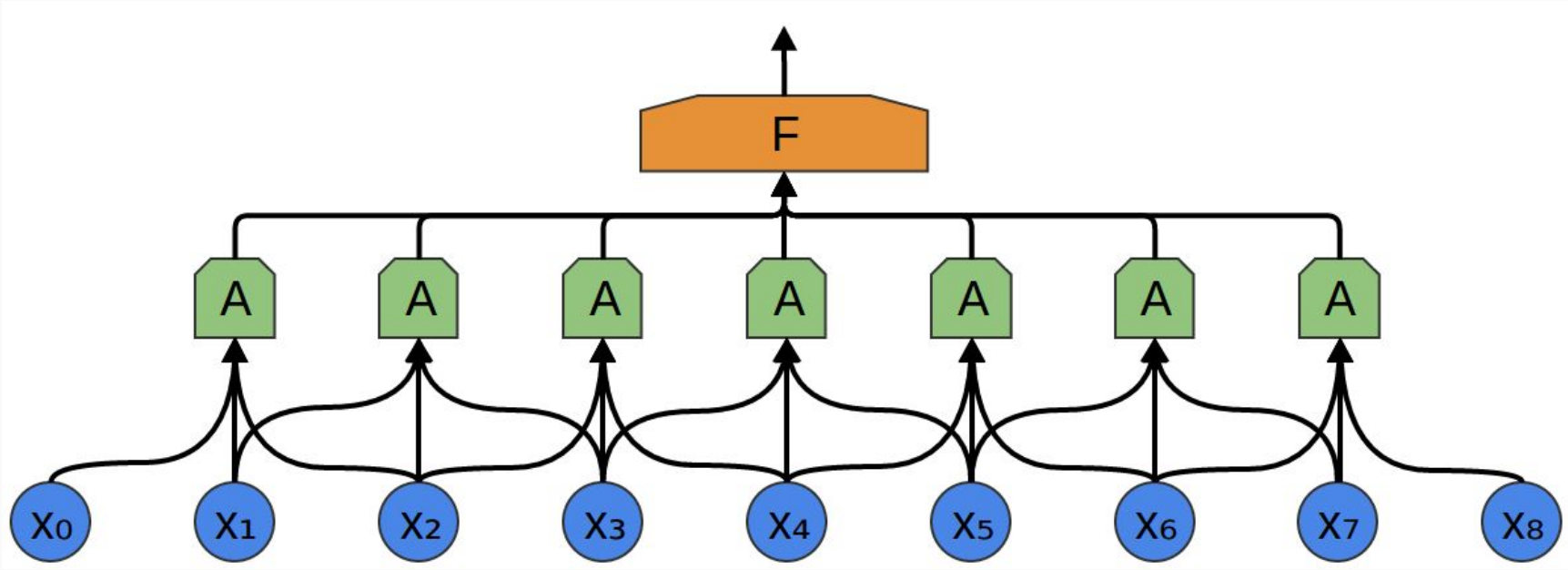
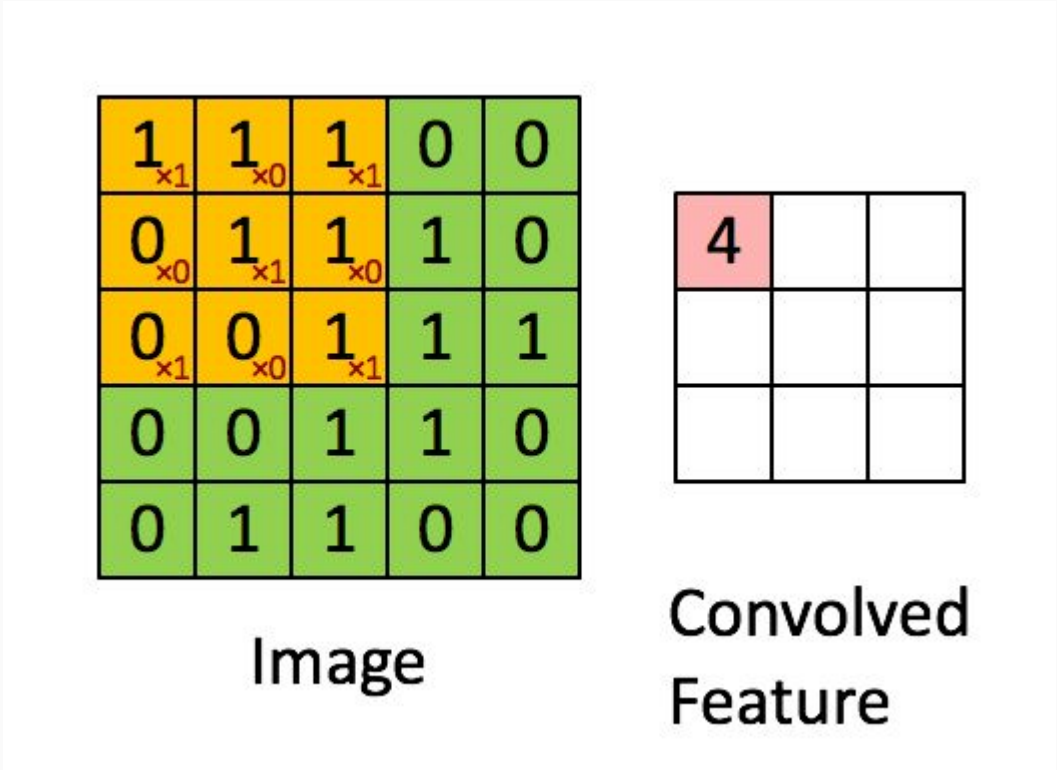
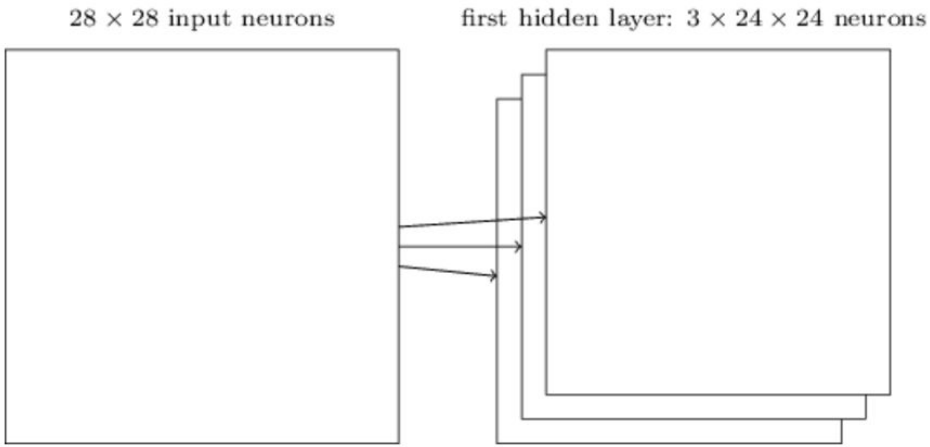


Image from: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/



Convolution with 3×3 Filter. Source:
http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution



Typically a convolutional layer will have multiple different *feature maps*.

From: http://neuralnetworksanddeeplearning.com/chap6.html#introducing_convolutional_networks

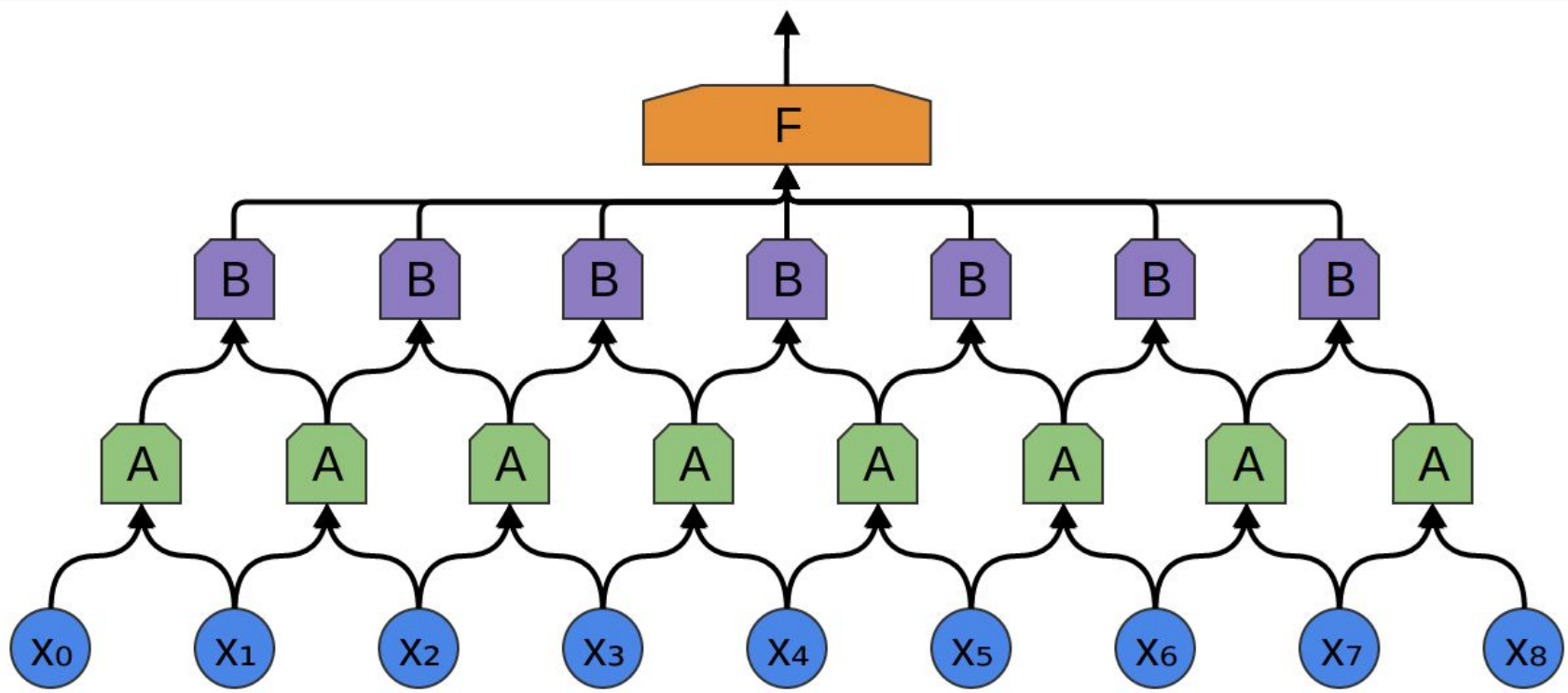


Image from: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

Often **max-pooling** layers are added

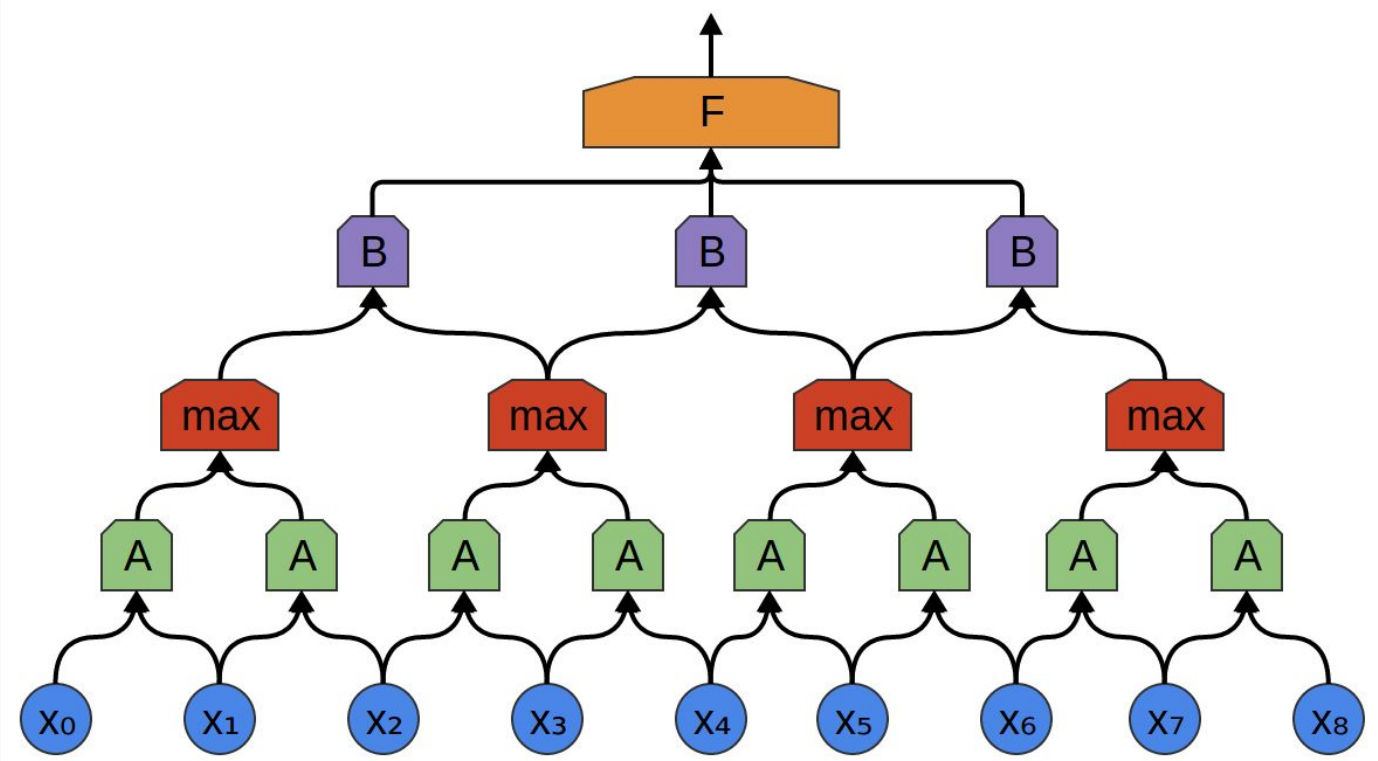
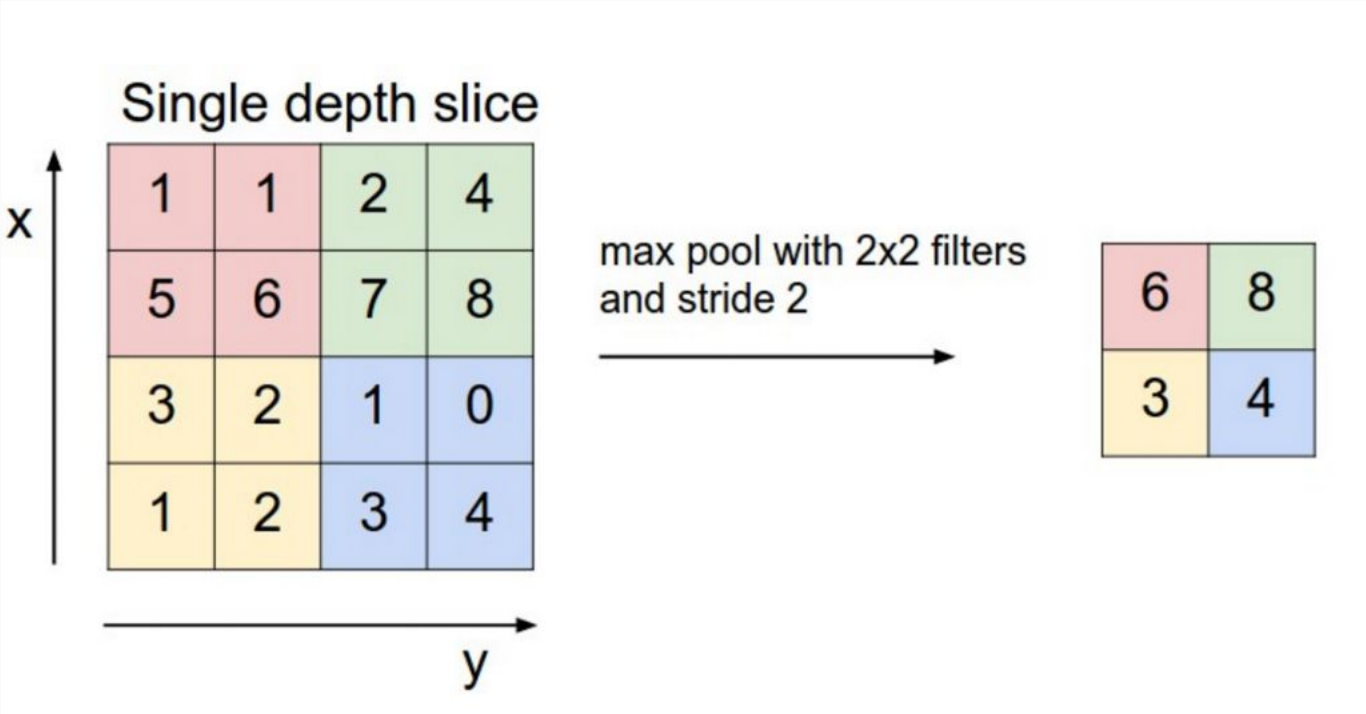


Image from: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/



Max pooling in CNN. Source: http://cs231n.github.io/convolutional-networks/#pool, via http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

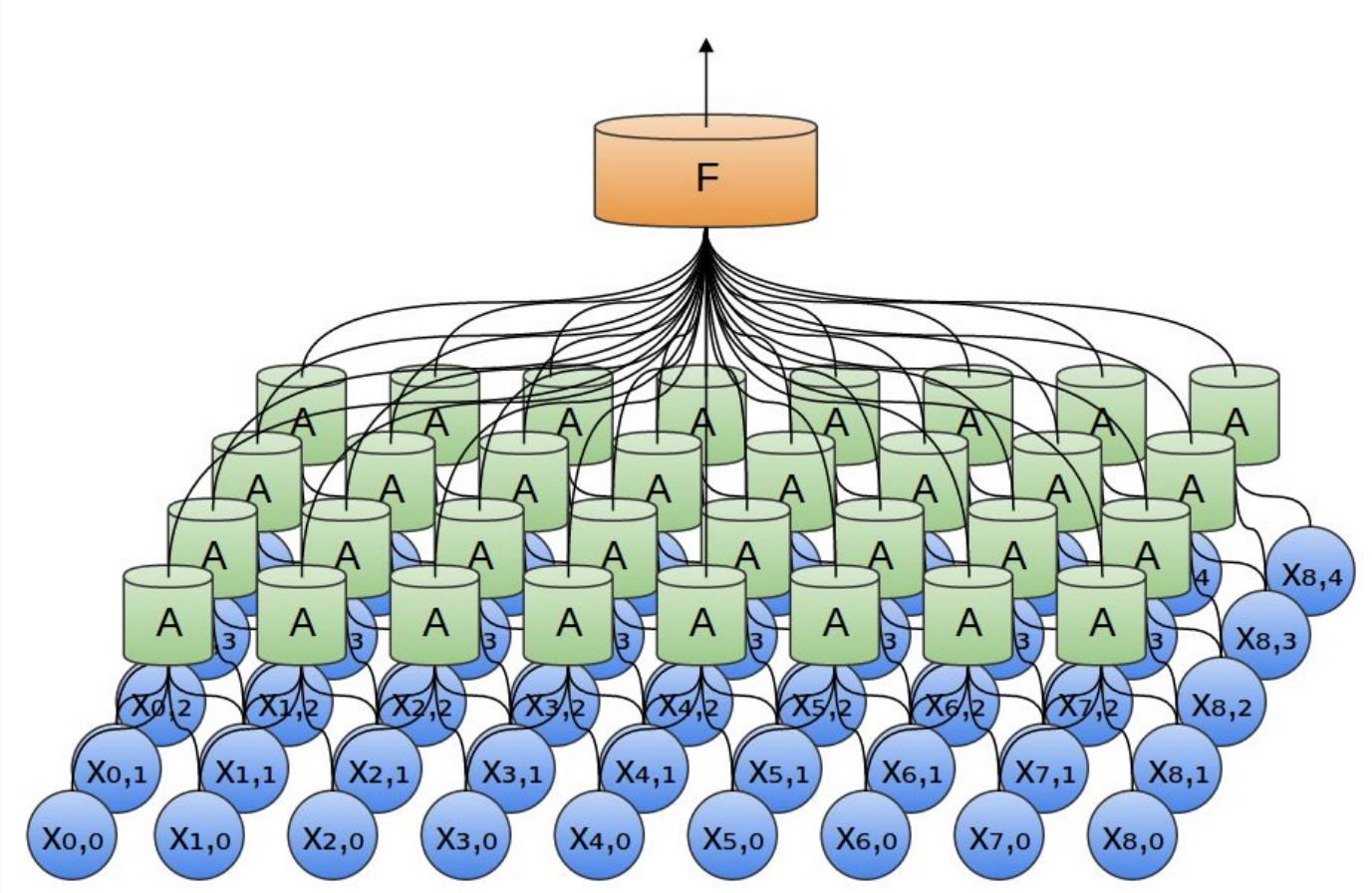


Image from: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

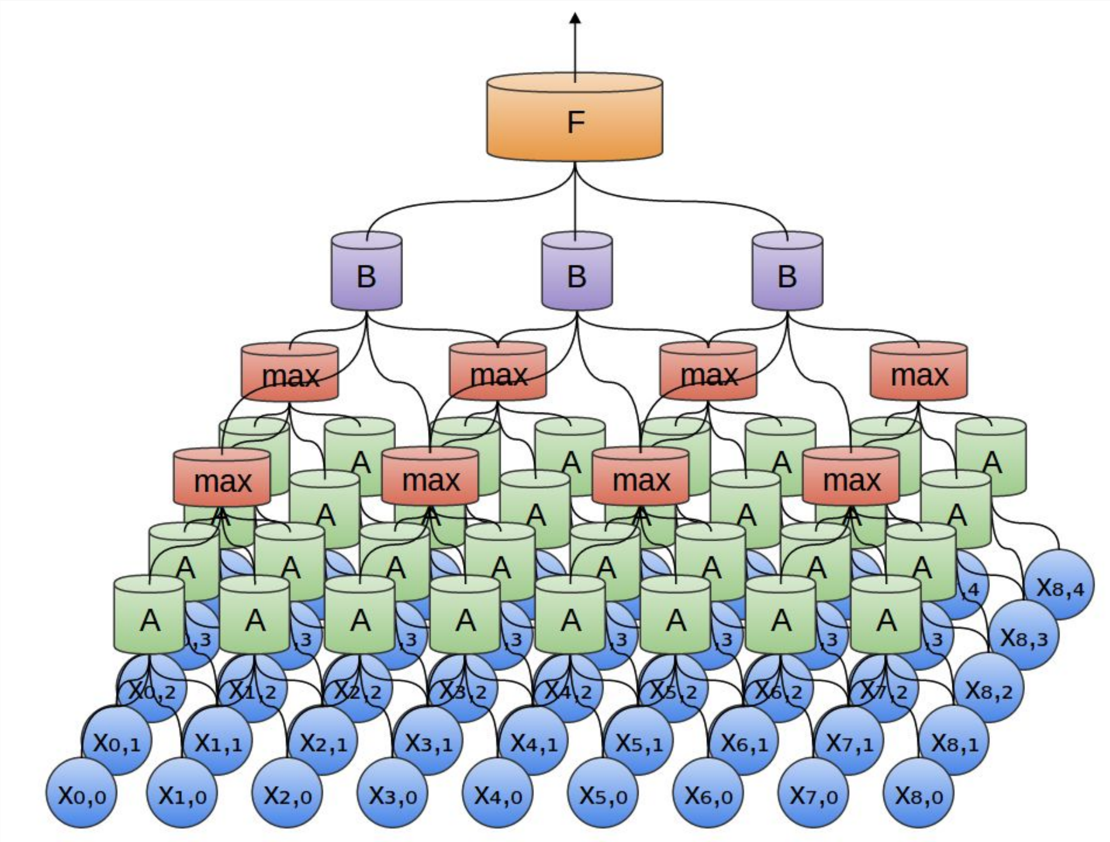
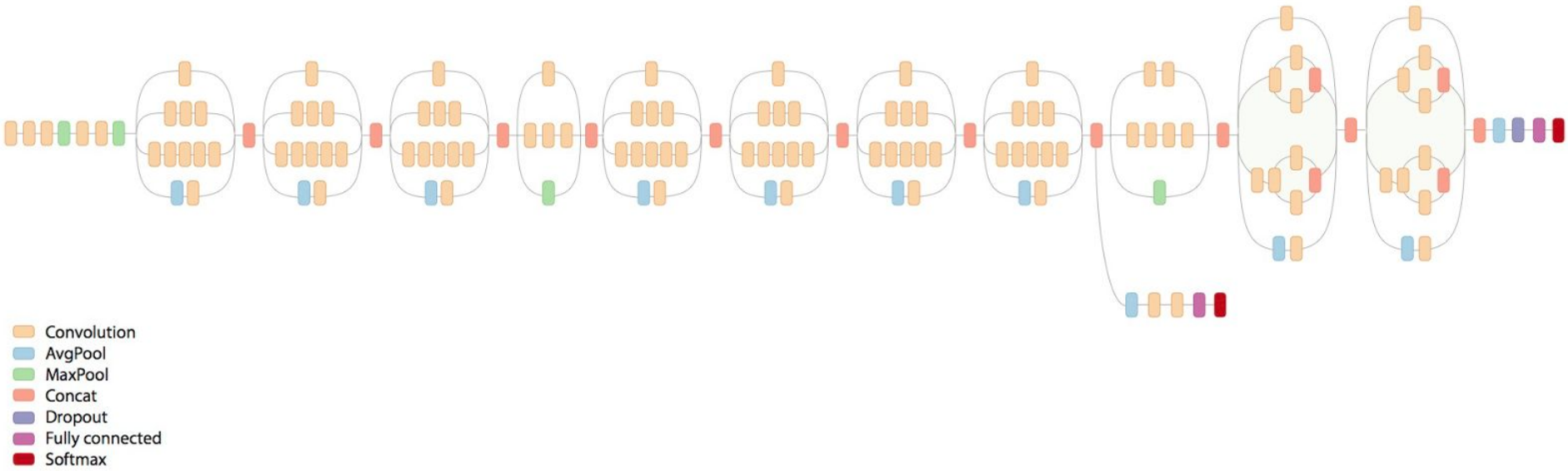


Image from: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/



From: http://googleresearch.blogspot.com/2016_03_01_archive.html

Let’s build a custom **Estimator** for a *CNN*-based model

Lab: CNN Custom Estimators

Workshop section:

`kaggle.com/yufengg/fashion-mnist-goes-cnn`

These slides: bit.ly/tf-workshop-aicon

Create a Kaggle.com account

Go to **kaggle.com/yufengg/kernels**

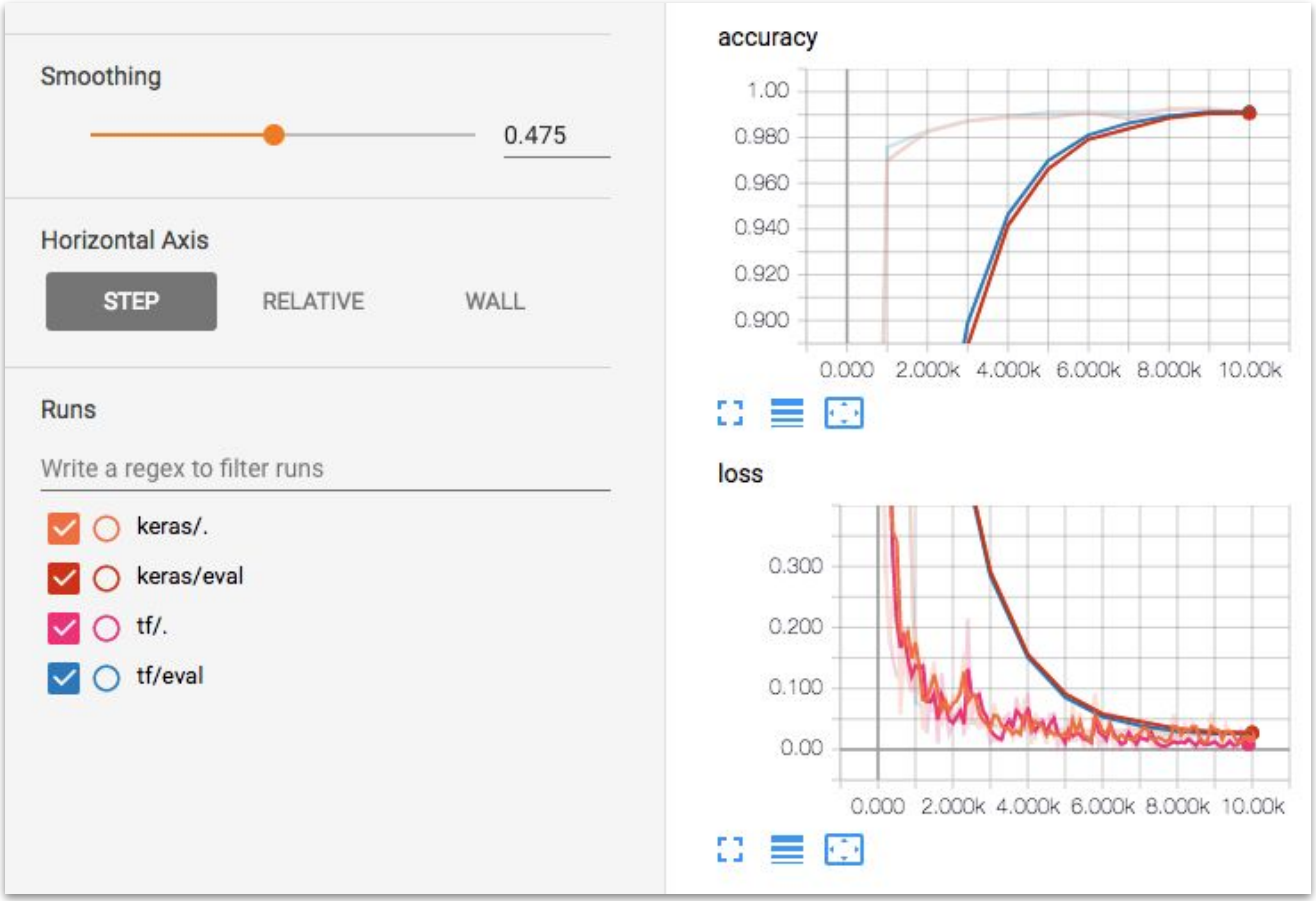
OR

Install TensorFlow & Jupyter: www.tensorflow.org/install/

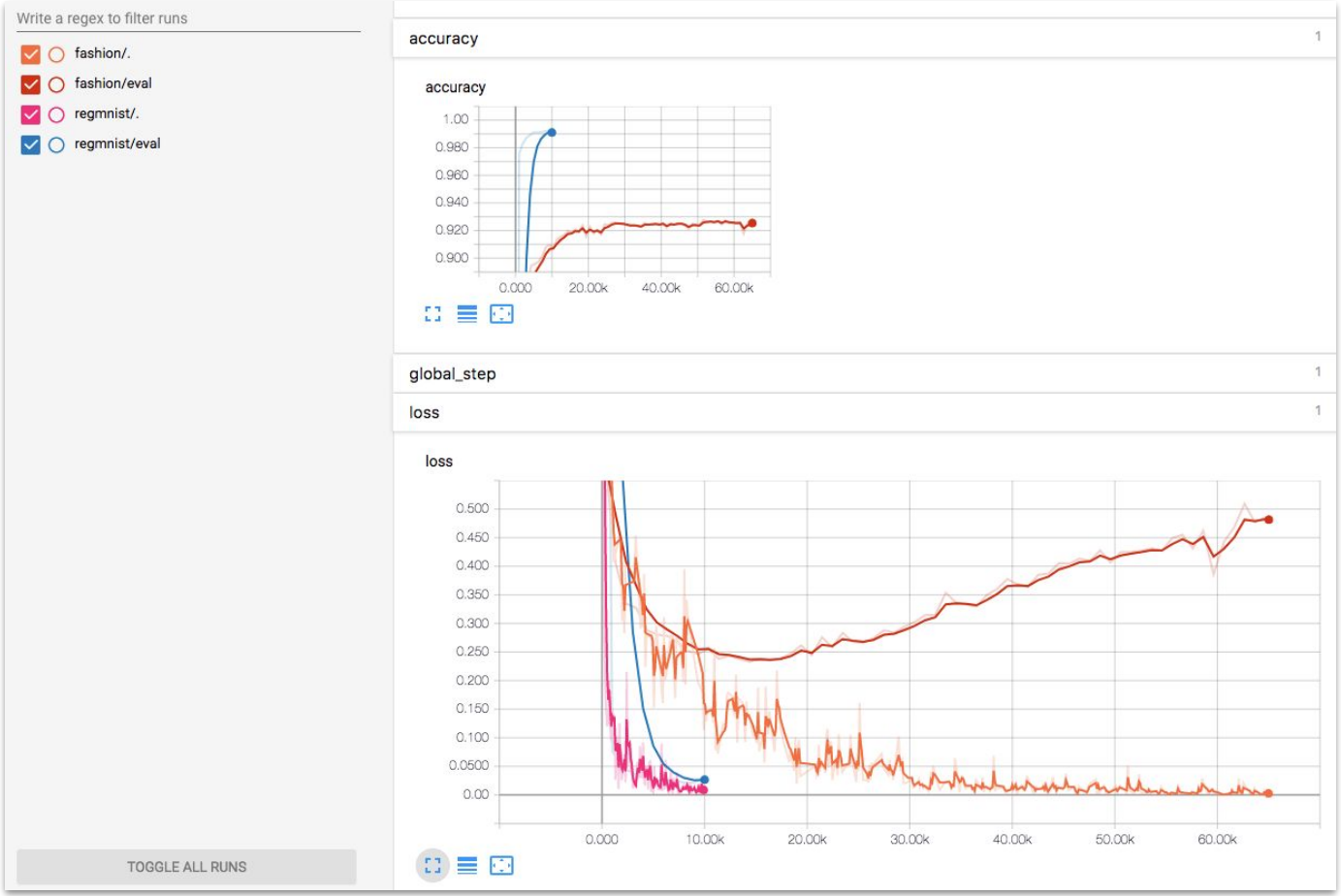
```
pip install tensorflow pandas jupyter
```

Clone GitHub Repo: bit.ly/tensorflow-workshop

Optional: download the '**Fashion-MNIST**' files:
github.com/zalandoresearch/fashion-mnist#get-the-data



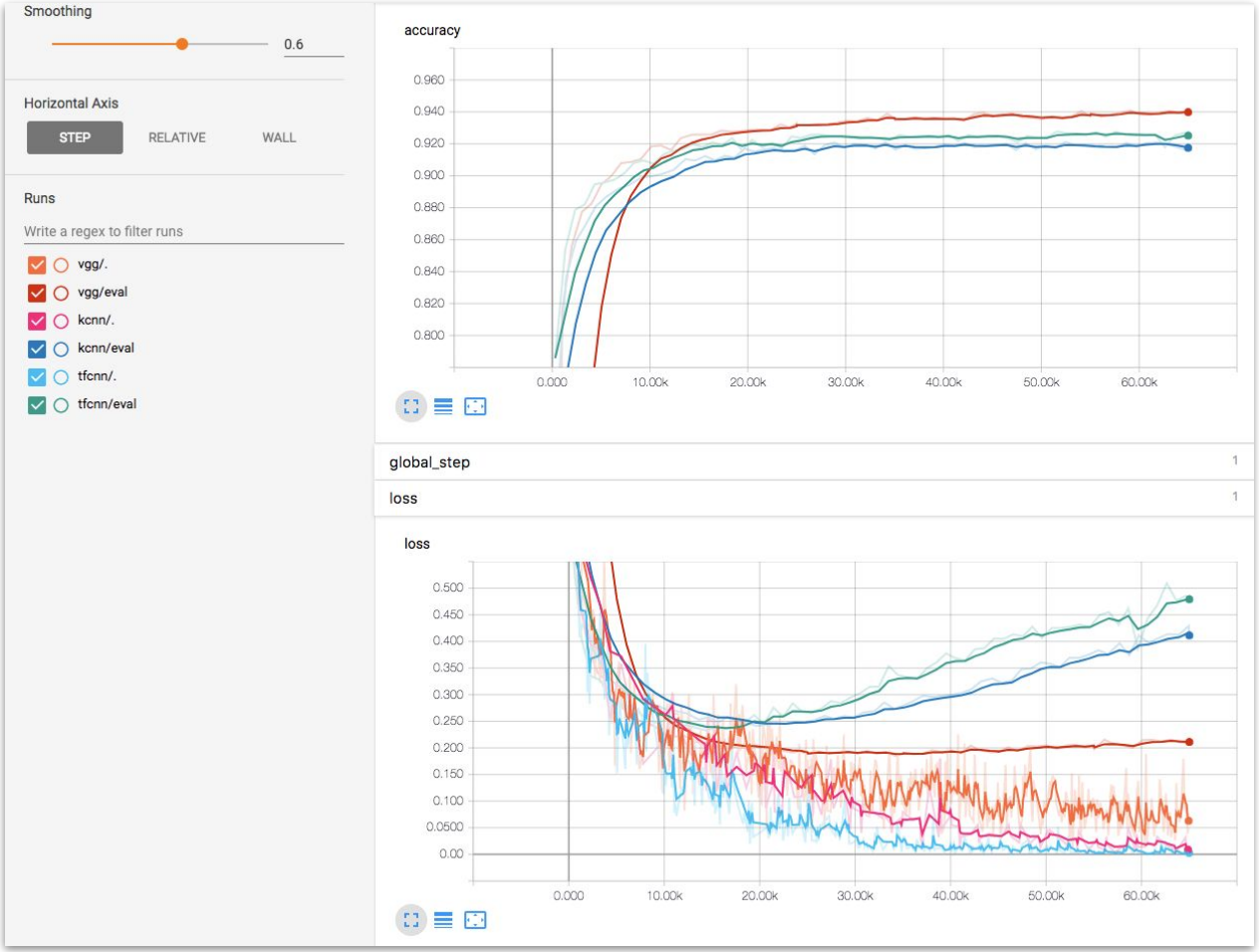
bit.ly/tf-workshop-aicon



Comparing ‘regular’ with Fashion MNIST

bit.ly/tf-workshop-aicon

Maybe there are better models?...



bit.ly/tf-workshop-aicon

Wrapup

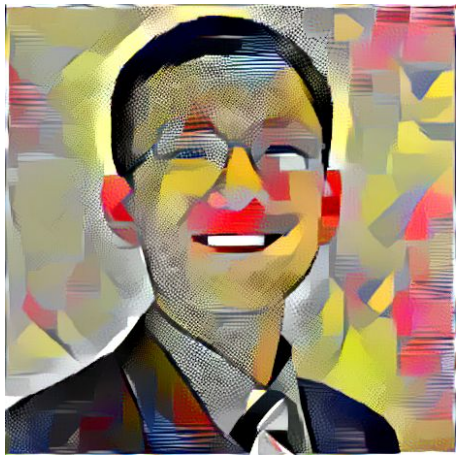
bit.ly/aiadventures

Exploring the art, science, and tools of machine learning

Thank you!

Rate this session:

bit.ly/tf-workshop-beij-ai18



Yufeng
yfg@google.com
@YufengG
yufengg.com

What next?

Rate this session:
bit.ly/tf-workshop-beij-ai18

Tutorials and code
tensorflow.org

TensorFlow Dev Summit
2017: https://goo.gl/bv9mm7
2018: tensorflow.org/dev-summit/

Intro to Deep Learning with TensorFlow
Udacity class goo.gl/iHssII

Stanford's CS231n
cs231n.github.io

TensorFlow Playground
goo.gl/mXhncM

These Slides
bit.ly/tf-workshop-aicon

AI Adventures (on YouTube)
bit.ly/aiadventures

Deep Learning **(Goodfellow, Bengio, Courville)**
http://www.deeplearningbook.org/

Chris Olah's blog
colah.github.io

Michael Nielsen's book
neuralnetworksanddeeplearning.com

end