Google Cloud

TensorFlow Wide & Deep

Data Classification the Easy Way

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

Machine Learning
is
using many ***examples***
to ***answer questions***

Training

many

examples

Prediction

answer

questions

Training

many

examples

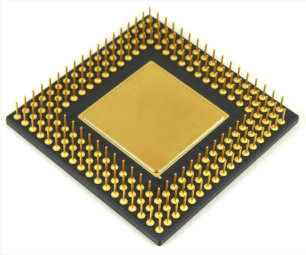
Prediction

answer questions

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

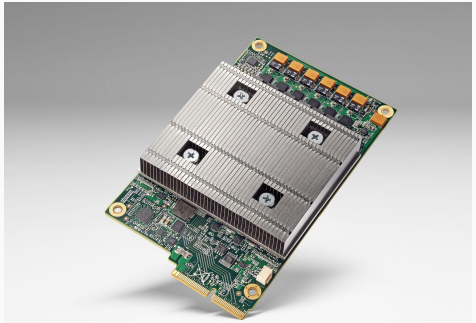
https://research.googleblog.com/2016/11/celebrating-tensorflows-first-year.html

TensorFlow Supports Many Platforms...



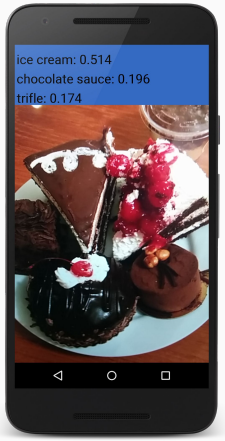
CPU

GPU

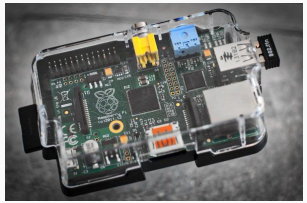


TPU

Android



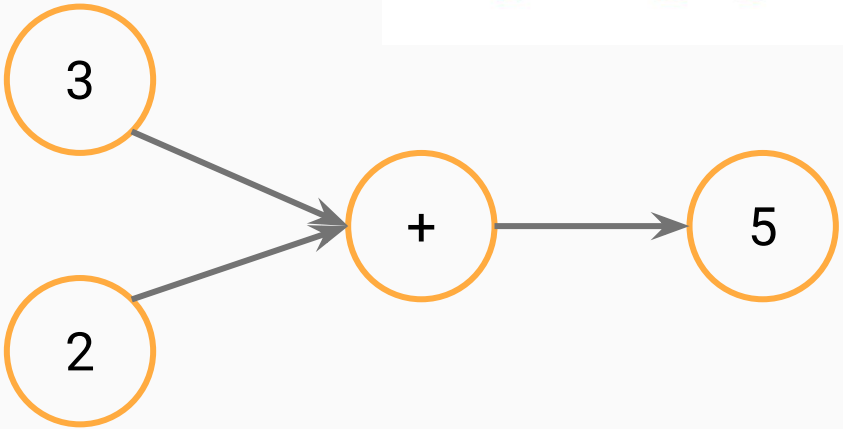
iOS



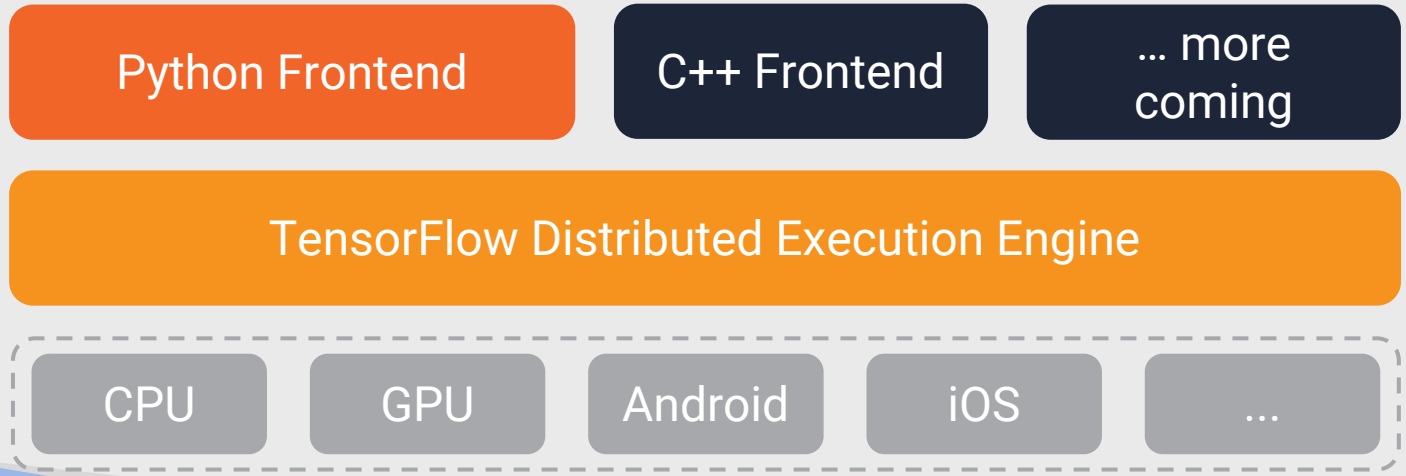
Raspberry Pi

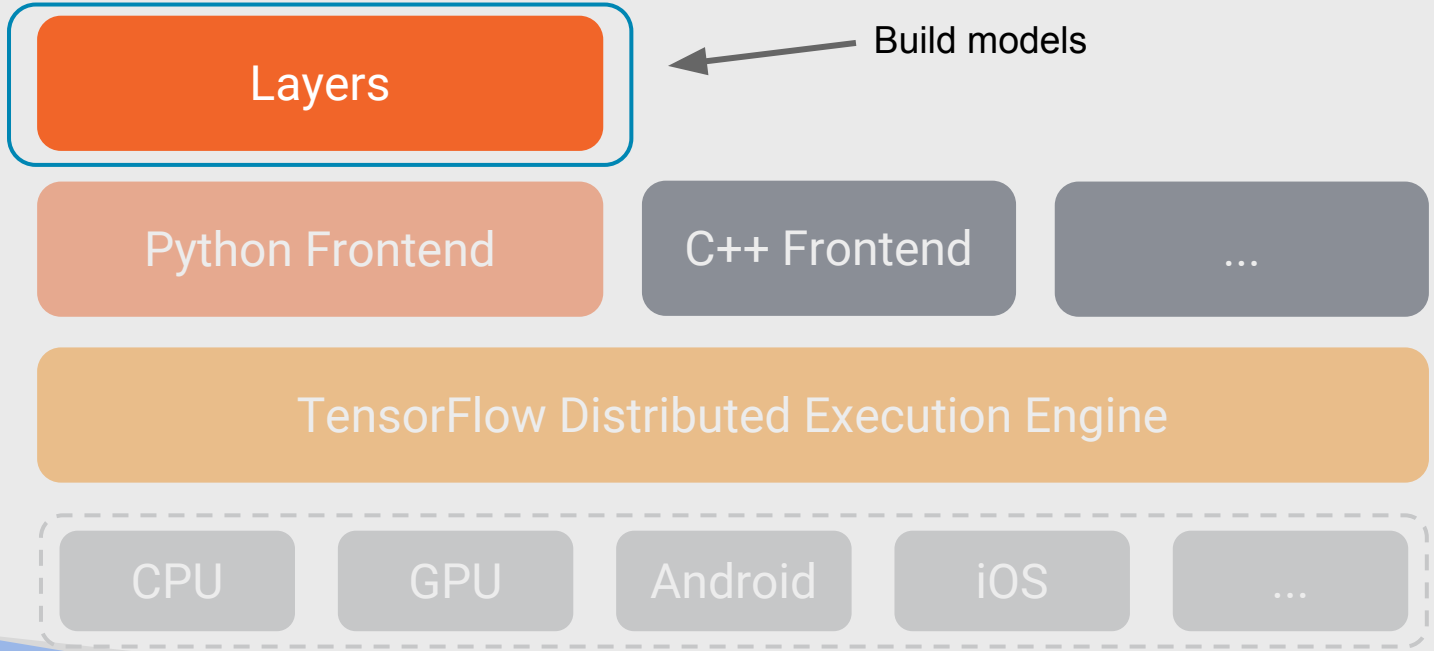
A multidimensional array.

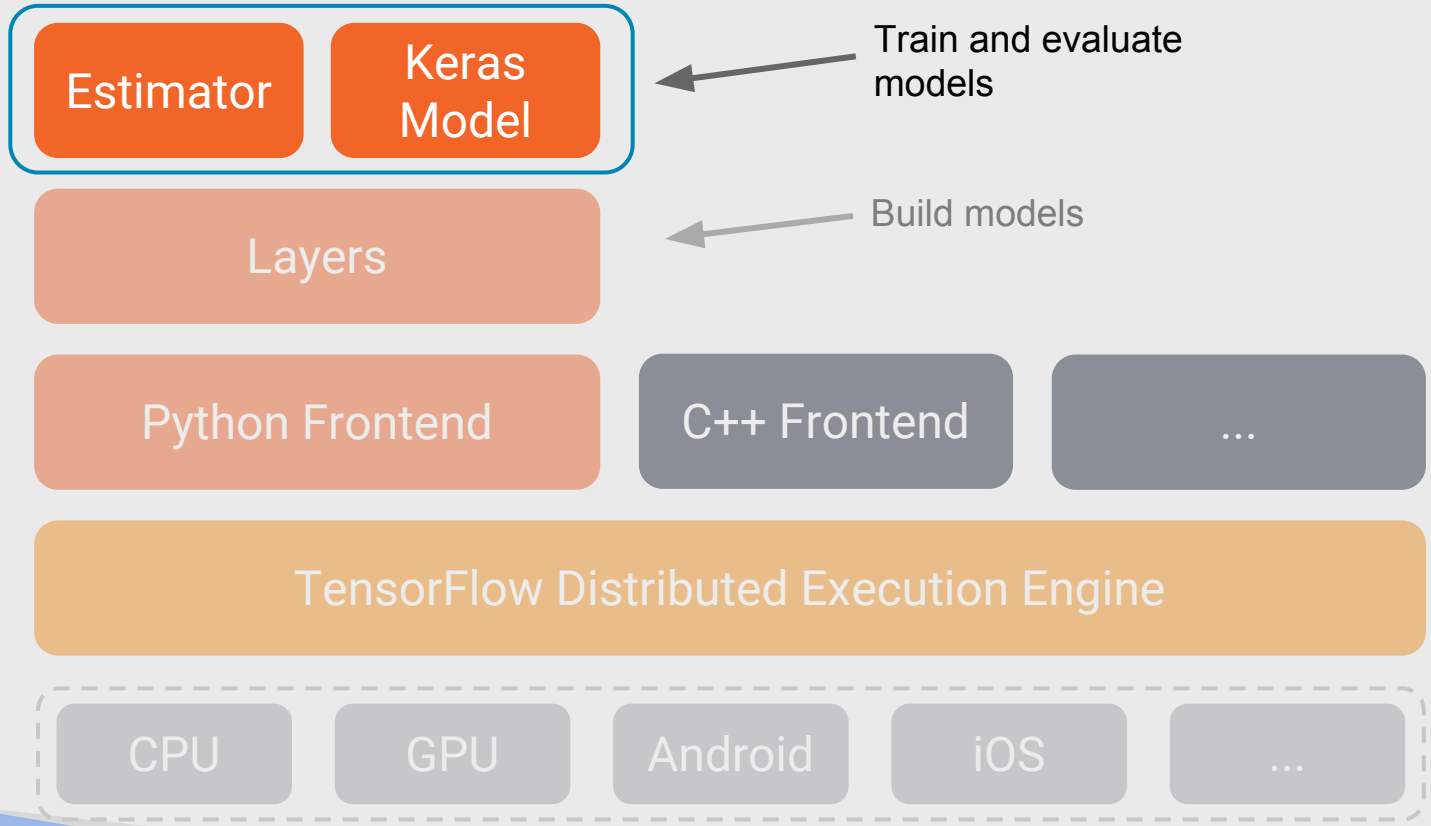
TensorFlow

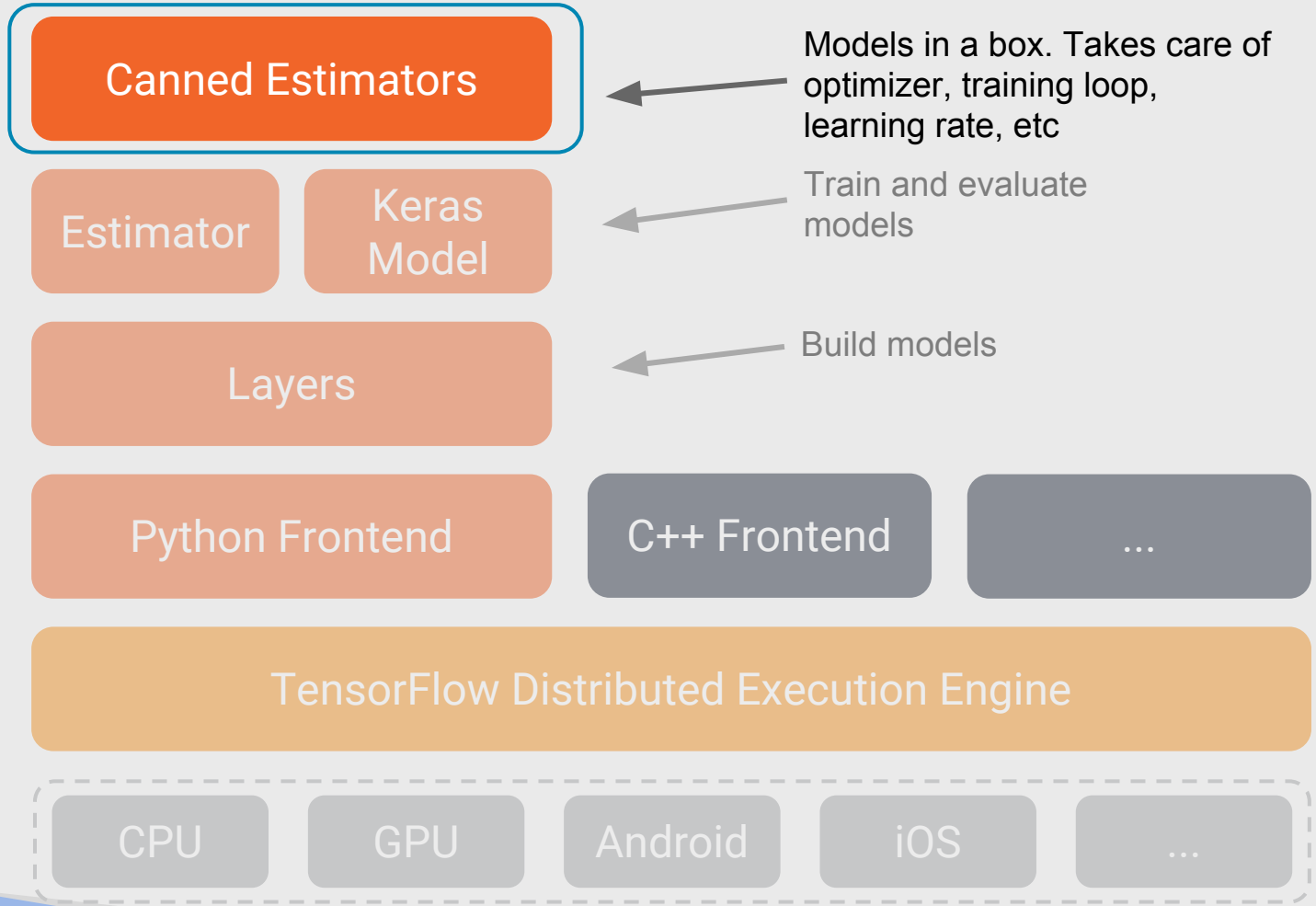


A graph of operations.


Python Frontend
C++ Frontend
… more coming
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
…


Layers
Build models
Python Frontend
C++ Frontend
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
...


Estimator
Keras Model
Train and evaluate models
Layers
Build models
Python Frontend
C++ Frontend
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
...


Canned Estimators
Models in a box. Takes care of optimizer, training loop, learning rate, etc
Estimator
Keras Model
Train and evaluate models
Layers
Build models
Python Frontend
C++ Frontend
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
...

A *quick* peek at some code

```
area = numeric_column("square_foot"),
rooms = numeric_column("num_rooms"),
zip_code = categorical_column_with_hash_bucket("zip_code", 10000)

classifier = DNNClassifier(
    feature_columns=[area, rooms, embedding_column(zip_code, 8)],
    hidden_units=[1024, 512, 256, 128])

classifier.train(train_input_fn)

results = classifier.evaluate(eval_input_fn)

print(results)
```

```
area = real_valued_column("square_foot"),
rooms = real_valued_column("num_rooms"),
zip_code = sparse_column_with_integerized_feature("zip_code",
10000)

classifier = DNNClassifier(
    feature_columns=[area, rooms, embedding_column(zip_code, 8)],
    hidden_units=[1024, 512, 256, 128])

classifier.fit(train_input_fn)

results = classifier.evaluate(eval_input_fn)

print(results)
```

```
classifier = DNNLinearCombinedRegressor(
    linear_feature_columns=linear_columns_list,
    linear_optimizer=tf.train.FtrlOptimizer(learning_rate=0.01,
                                            l2_regularization_strength=0.1),
    dnn_feature_columns=deep_columns_list
    dnn_optimizer=tf.train.AdagradOptimizer(learning_rate=0.01,
                                            initial_accumulator_value=0.1),
    dnn_activation_fn=tf.nn.relu,
    dnn_dropout = 0.5,
    gradient_clip_norm=0.1,
    hidden_units=[1024, 512, 256, 128])

classifier.fit(train_input_fn)
classifier.evaluate(eval_input_fn)
```

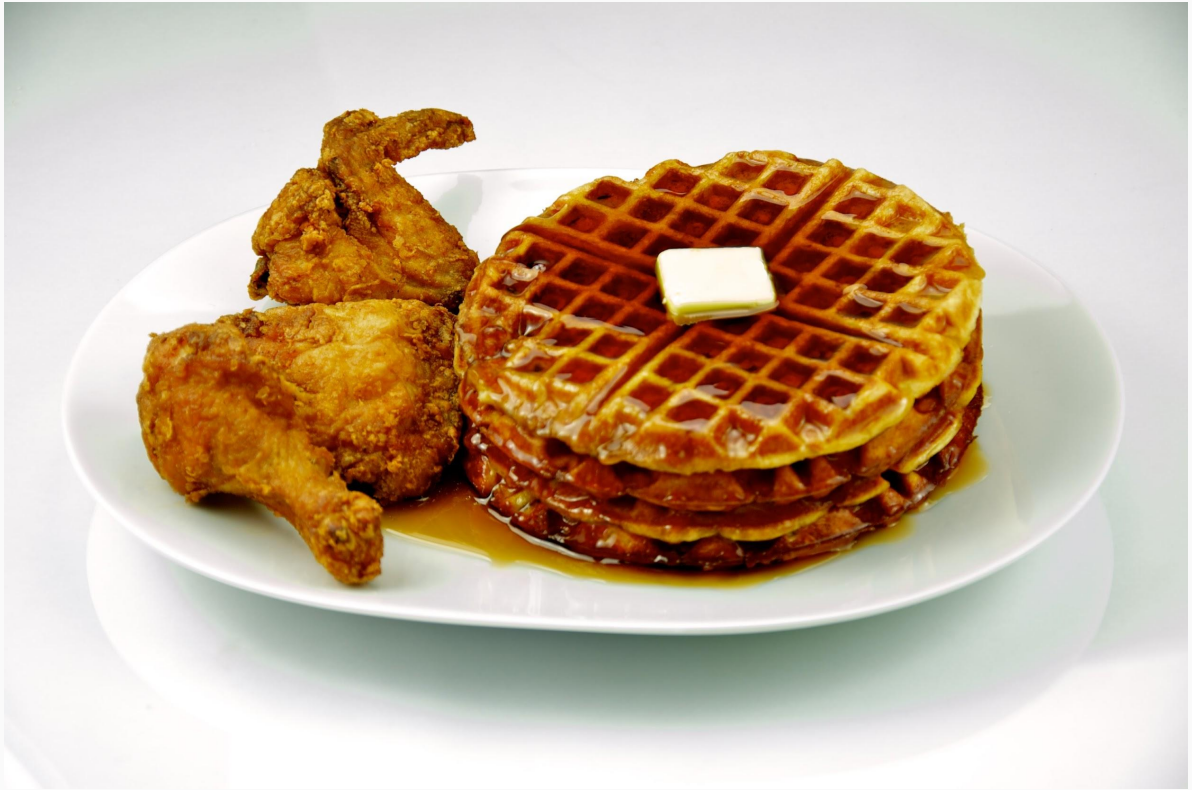
<Storytime>

Motivation - a "magical" food app

Just **Launch** and Iterate!

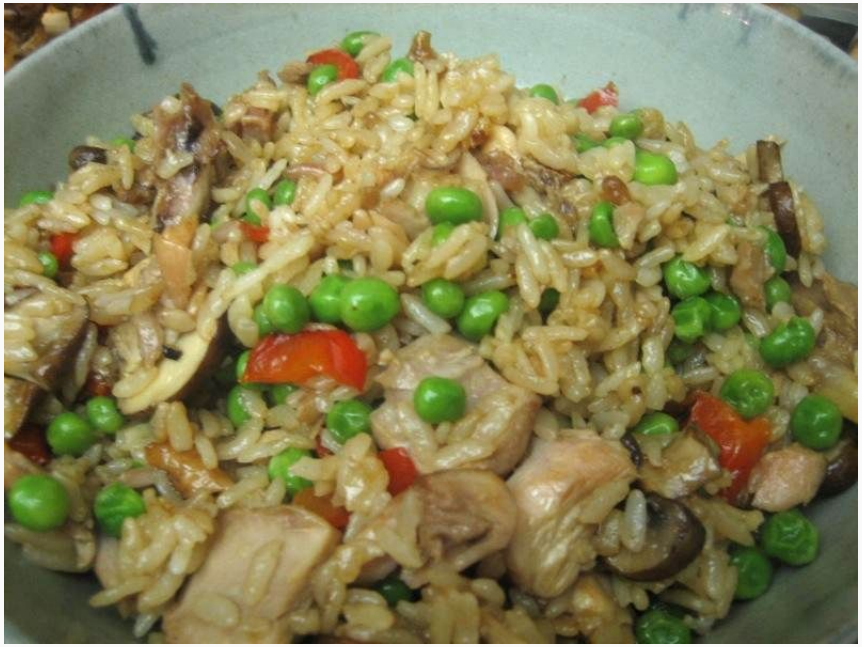
https://upload.wikimedia.org/wikipedia/commons/2/2c/Fried-Chicken-Set.jpg

Just **Launch** and Iterate



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

Just **Launch** and Iterate

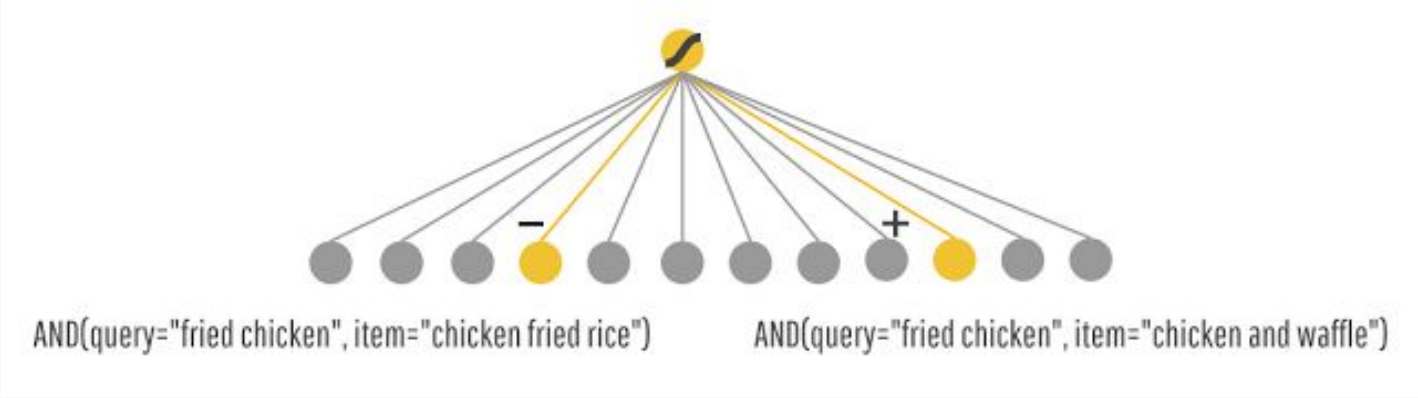


… naive character matching

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

v2.0: **memorize** all the things!

- Train a linear TF model

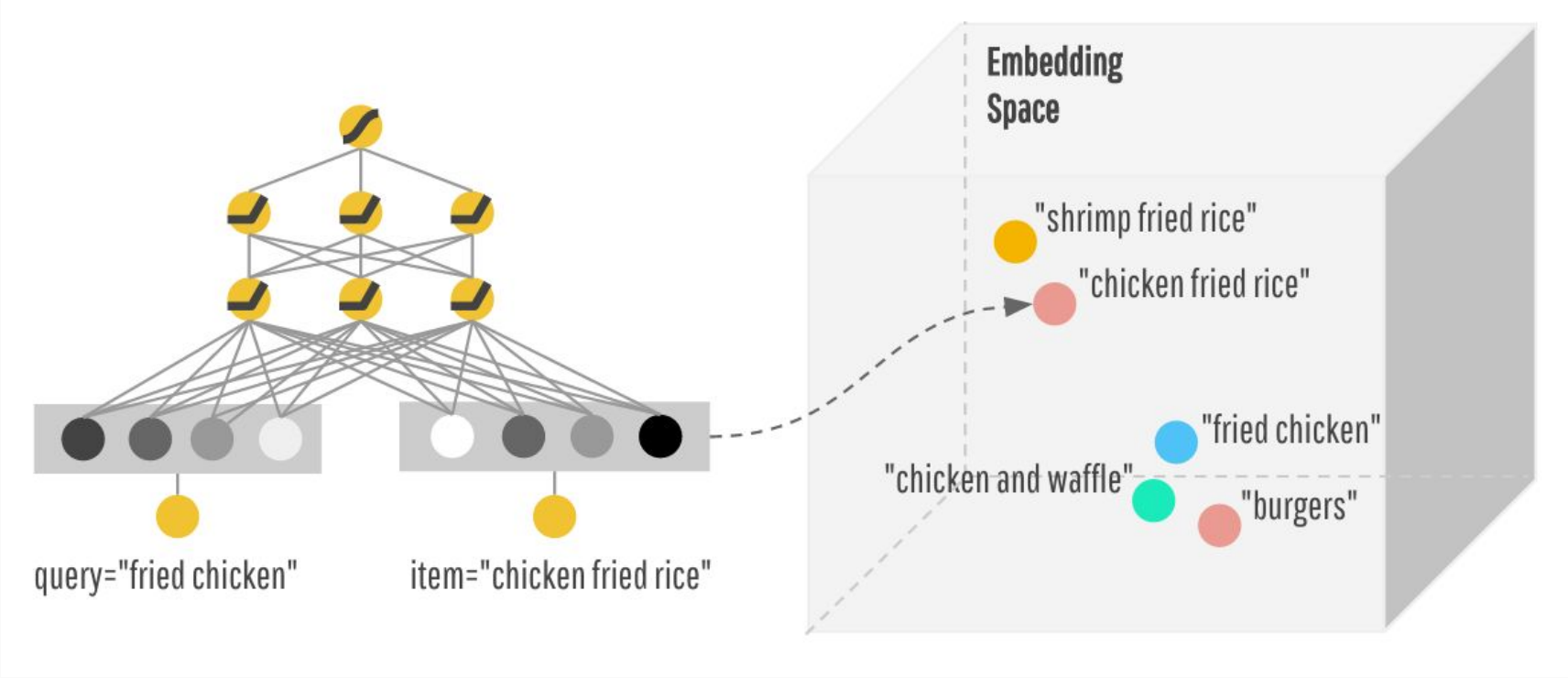


- Your app is gaining traction!

Problem: Your users are bored!

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

v3.0: More generalized recommendations for all



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

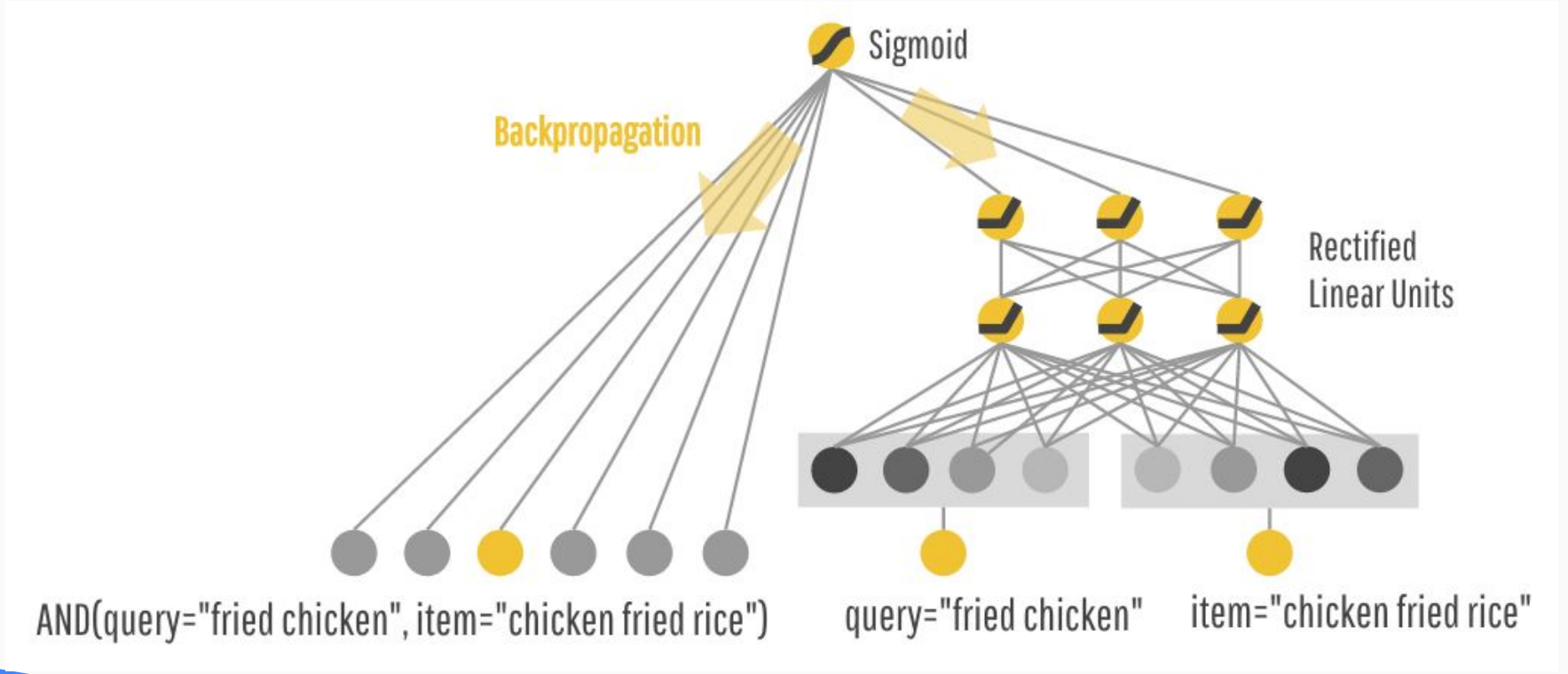
iced decaf latte with nonfat milk

Two types of requests:

specific and general

How to balance this?

v4.0: Why not both?

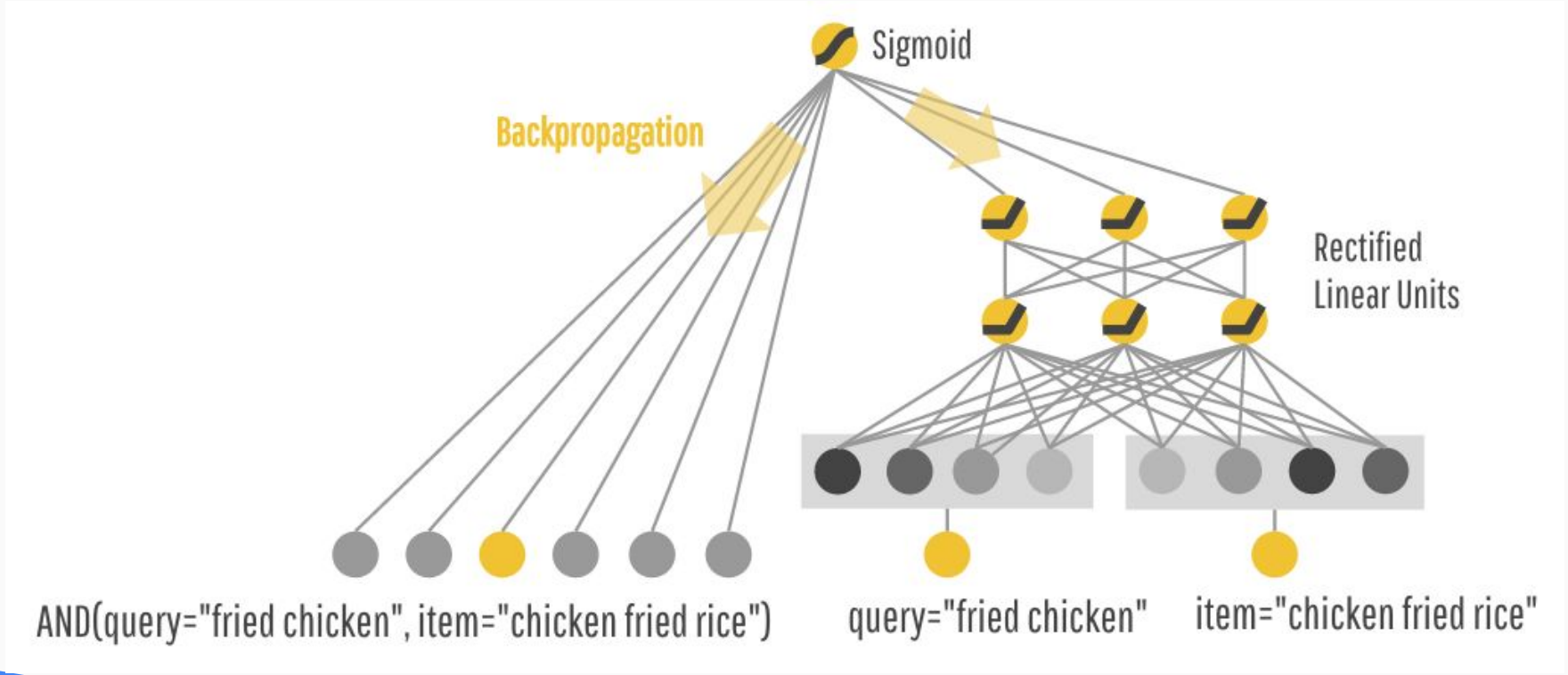


Wide & Deep

memorization
relevance

generalization
diversity

Wide and Deep



</Storytime>

Meet our dataset: US Census Data

- **Task**: predict whether the household has an annual income of over $50K
- Over **32k** training 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). Not 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.
gender	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.**

Wide & Deep

memorization	generalization
relevance	diversity
Sparse	Dense/Real
Categorical	Continuous

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

```
df = pandas.read_csv(filename,
       header=None, names=COLUMNS)
labels = df["labels"]


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

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

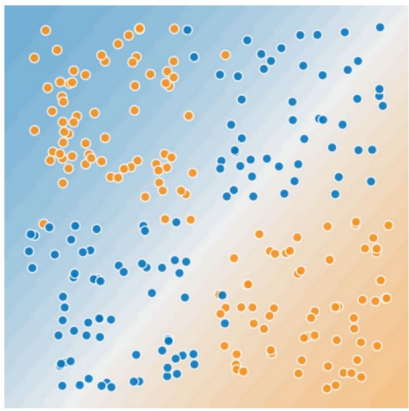
```
# Sparse base columns.
gender = categorical_column_with_vocabulary_list(
    key="gender", vocabulary_list=["female", "male"])
education = categorical_column_with_hash_bucket(
    key="education", hash_bucket_size=1000)
...
# Continuous base columns.
age = numeric_column("age")
...
```

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=generate_input_fn(train_file),
        steps=1000)
results = m.evaluate(
     input_fn=generate_input_fn(test_file))

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

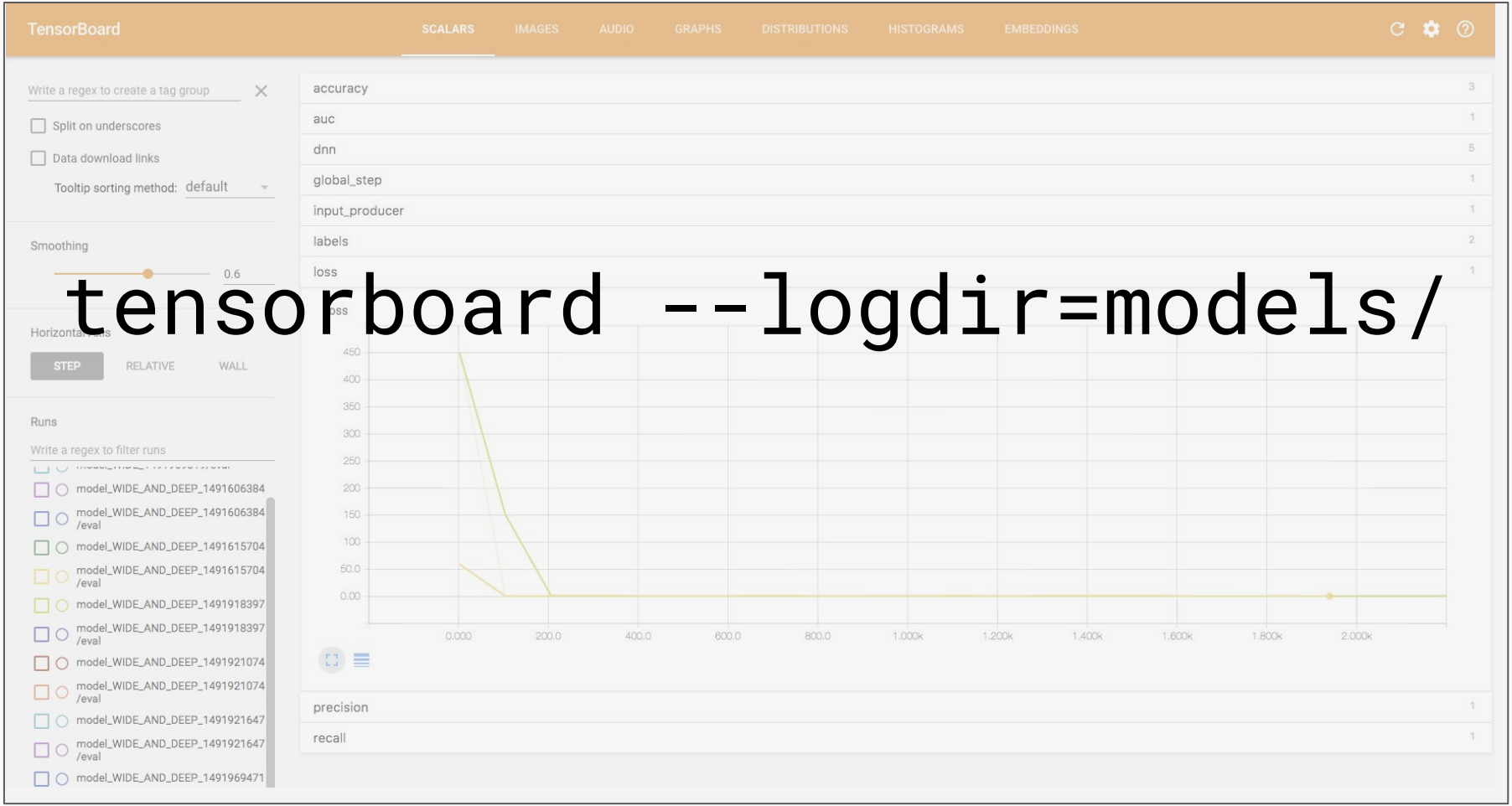
Make Predictions!

```
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(train_file))
```

To the code!

bit.ly/widendeep-census



tensorboard --logdir=models/

@YufengG

TensorBoard SCALARS GRAPHS DISTRIBUTIONS HISTOGRAMS PROJECTOR INACTIVE

Show data download links

Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

0.6

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

.

eval

TOGGLE ALL RUNS

models/model_WIDE_AND_DEEP_1508857607

auc 1

auc_precision_recall 1

average_loss 1

dnn 5

enqueue_input 1

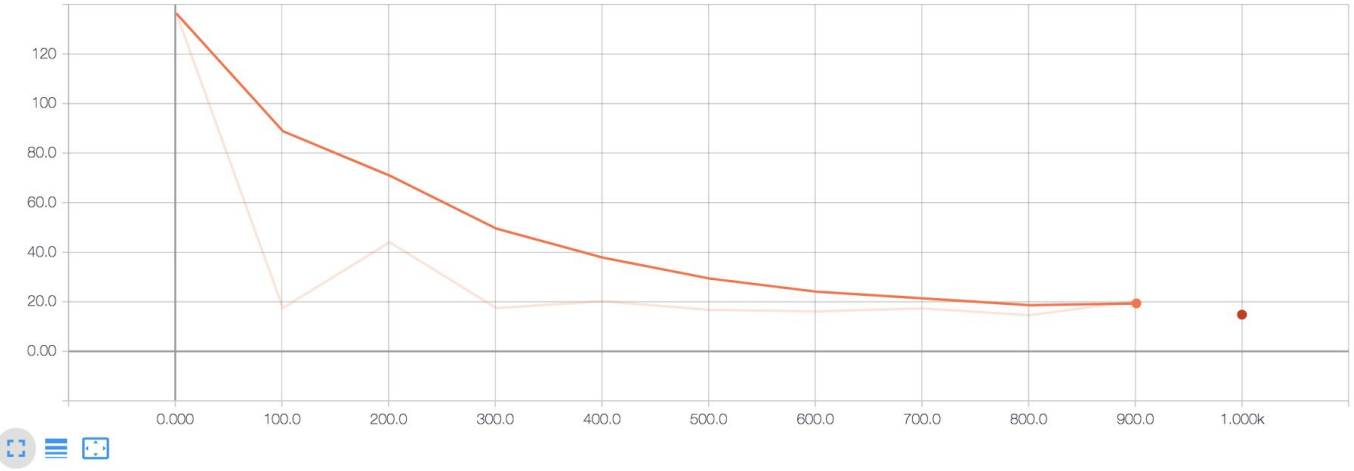
global_step 1

label 1

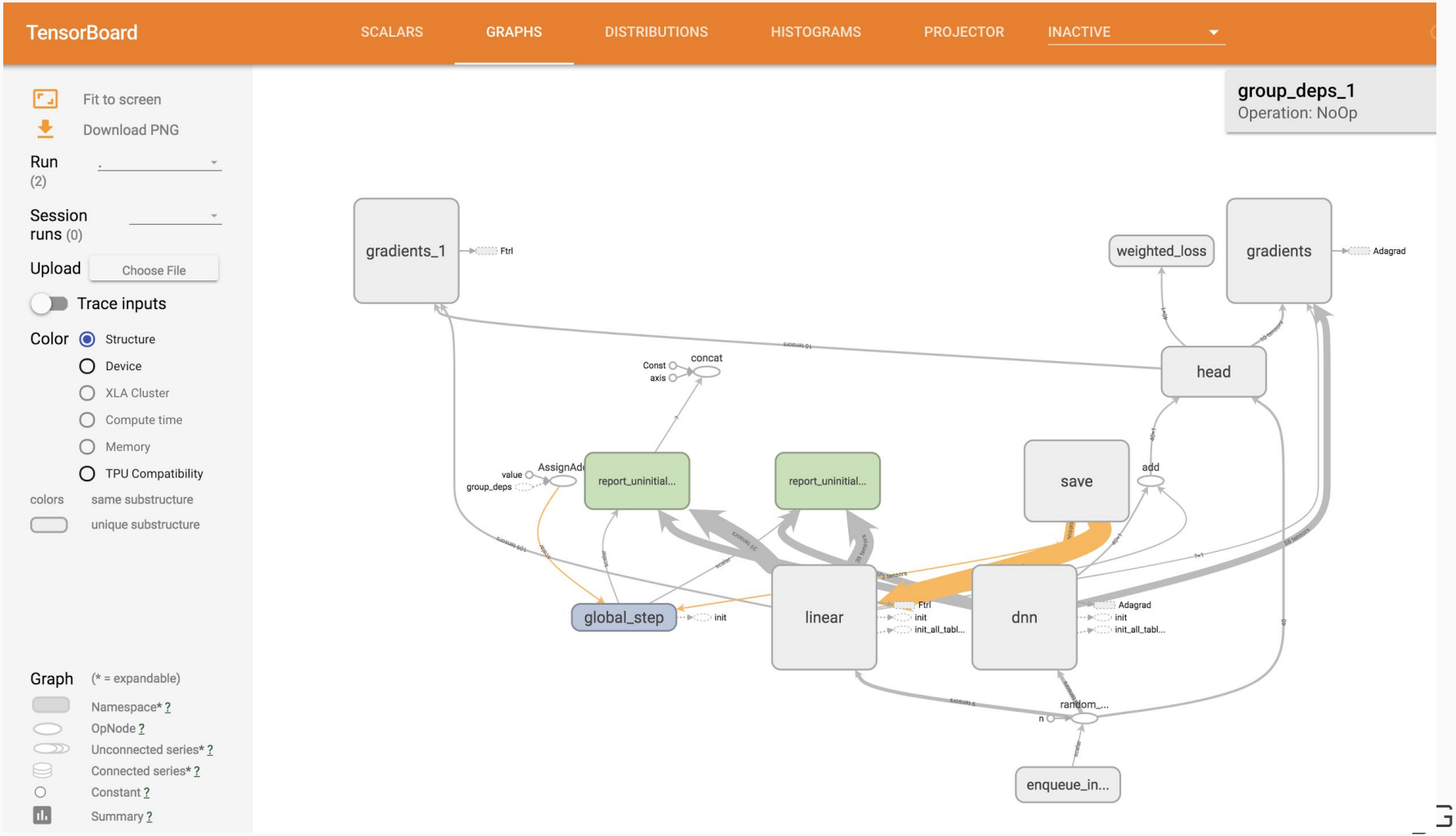
linear 1

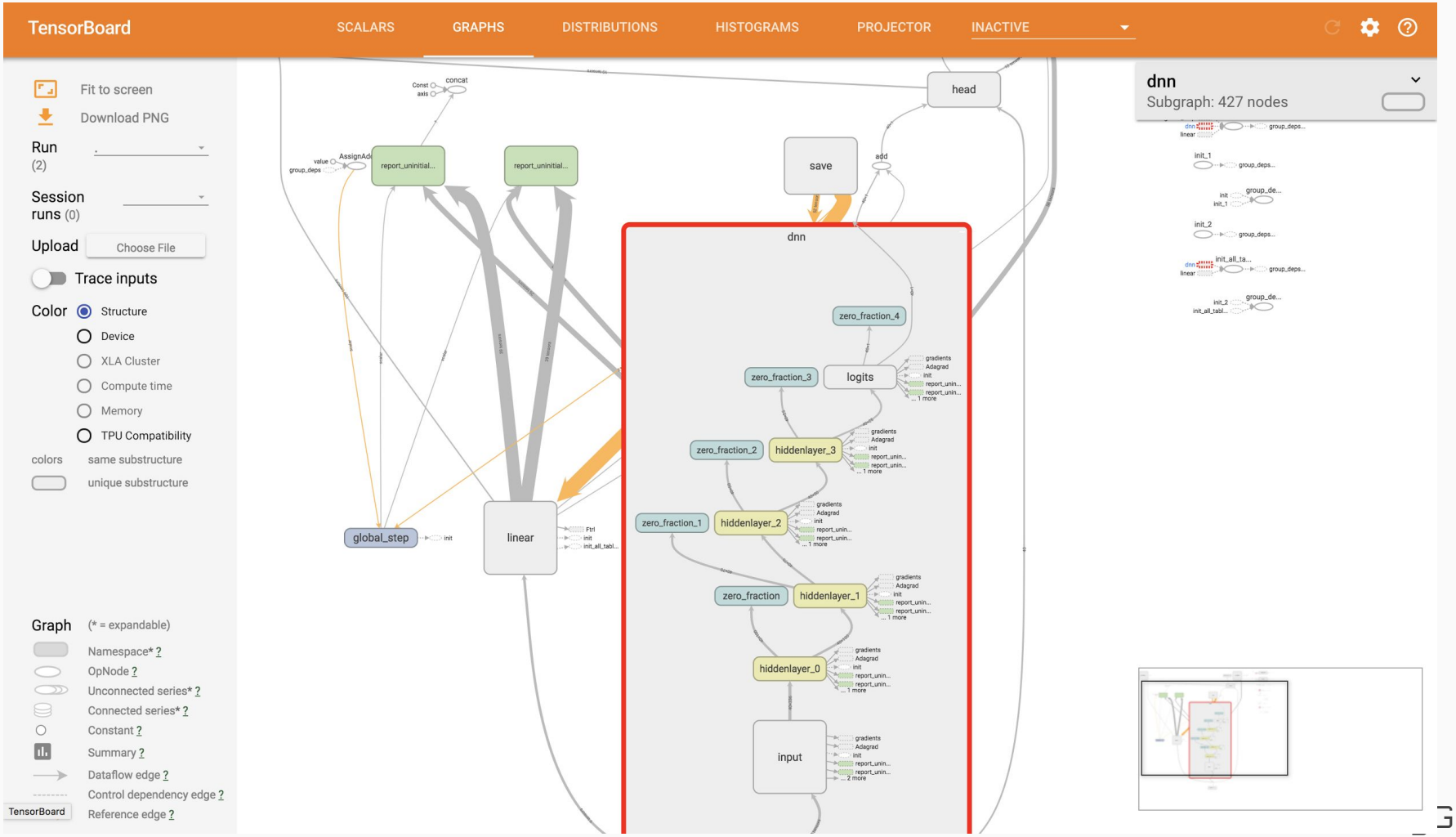
loss 1

loss



prediction 1


TensorBoard
SCALARS
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
PROJECTOR
INACTIVE
Fit to screen
Download PNG
Run (2)
Session runs (0)
Upload
Choose File
Trace inputs
Color
Structure
Device
XLA Cluster
Compute time
Memory
TPU Compatibility
colors
same substructure
unique substructure
Graph (* = expandable)
Namespace* ?
OpNode ?
Unconnected series* ?
Connected series* ?
Constant ?
Summary ?
group_deps_1
Operation: NoOp
gradients_1
Ftrl
weighted_loss
gradients
Adagrad
head
concat
Const
axis
value
group_deps
AssignAd
report_uninitial...
report_uninitial...
save
add
global_step
init
linear
Ftrl
init
init_all_tabl...
dnn
Adagrad
init
init_all_tabl...
random_...
n
enqueue_in...


TensorBoard
SCALARS
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
PROJECTOR
INACTIVE
Fit to screen
Download PNG
Run (2)
Session runs (0)
Upload
Choose File
Trace inputs
Color
Structure
Device
XLA Cluster
Compute time
Memory
TPU Compatibility
colors
same substructure
unique substructure
Graph
(* = expandable)
Namespace* ?
OpNode ?
Unconnected series* ?
Connected series* ?
Constant ?
Summary ?
Dataflow edge ?
Control dependency edge ?
Reference edge ?
TensorBoard
concat
head
save
add
report_uninitial...
global_step
linear
dnn
zero_fraction_4
zero_fraction_3
logits
zero_fraction_2
hiddenlayer_3
zero_fraction_1
hiddenlayer_2
zero_fraction
hiddenlayer_1
hiddenlayer_0
input
dnn
Subgraph: 427 nodes

Training

many

examples

Prediction

answer questions

Export your model for prediction

```
:
```

```
def column_to_dtype(column):
    if column in CATEGORICAL_COLUMNS:
        return tf.string
    else:
        return tf.float32

feature_spec = {
    column: tf.FixedLenFeature(shape=[1], dtype=column_to_dtype(column))
        for column in FEATURE_COLUMNS
}
serving_fn = tf.estimator.export.build_parsing_serving_input_receiver_fn(feature_spec)
m.export_savedmodel(export_dir_base=model_dir + '/export',
                            serving_input_receiver_fn=serving_fn)
```

```
INFO:tensorflow:Restoring parameters from models/model_WIDE_AND_DEEP_1508857607/model.ckpt-1000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: models/model_WIDE_AND_DEEP_1508857607/export/1508857859/saved_model.pb
```

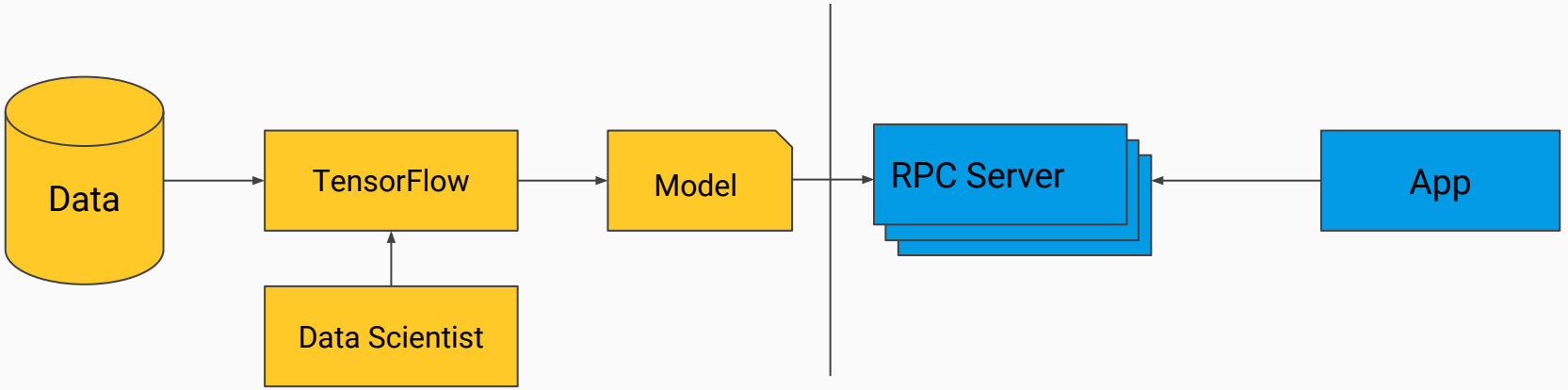
```
:
```

```
'models/model_WIDE_AND_DEEP_1508857607/export/1508857859'
```

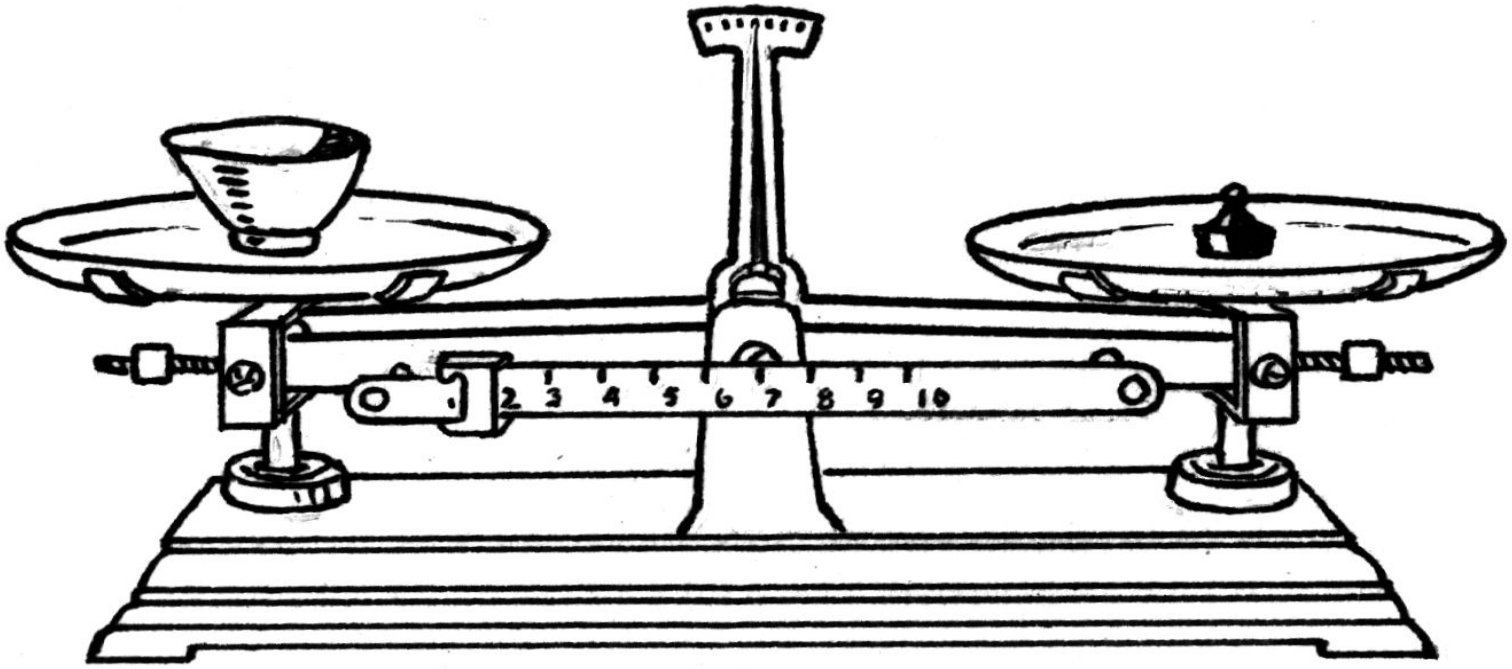
1. yufengg@yufengg-macbookpro6.roam.corp.google.com (192.168.54.177) - byobu (tmux)

```
(ml-py2) yufengg@yufengg-macbookpro6:~/code/github/tensorflow-workshop/workshop_sections/wide_n_deep
$ ls -l models/model_WIDE_AND_DEEP_1508857607/export/1508857859
total 928
-rw-r--r--  1 yufengg  eng   460K Oct 24 17:11 saved_model.pb
drwxr-xr-x  4 yufengg  eng   136B Oct 24 17:11 variables/
```

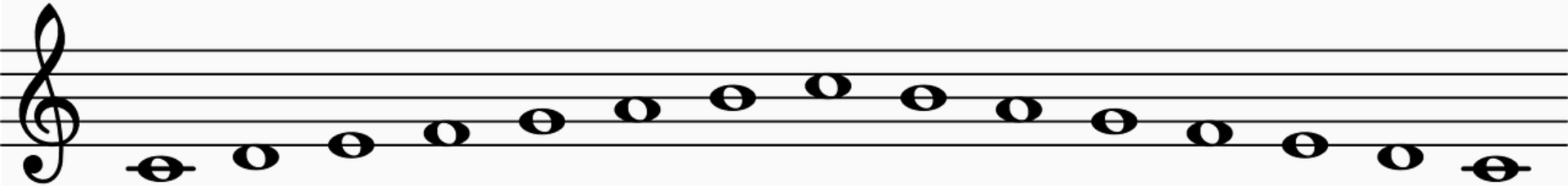
What is Serving?



Scale



Scale?



Scale

Cloud Machine Learning Engine

A managed service that enables you to build machine learning models, that work on any type of data, of any size.

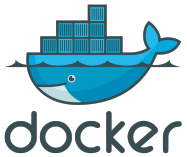
Cloud Machine Learning Engine

Managed **scalable machine learning**

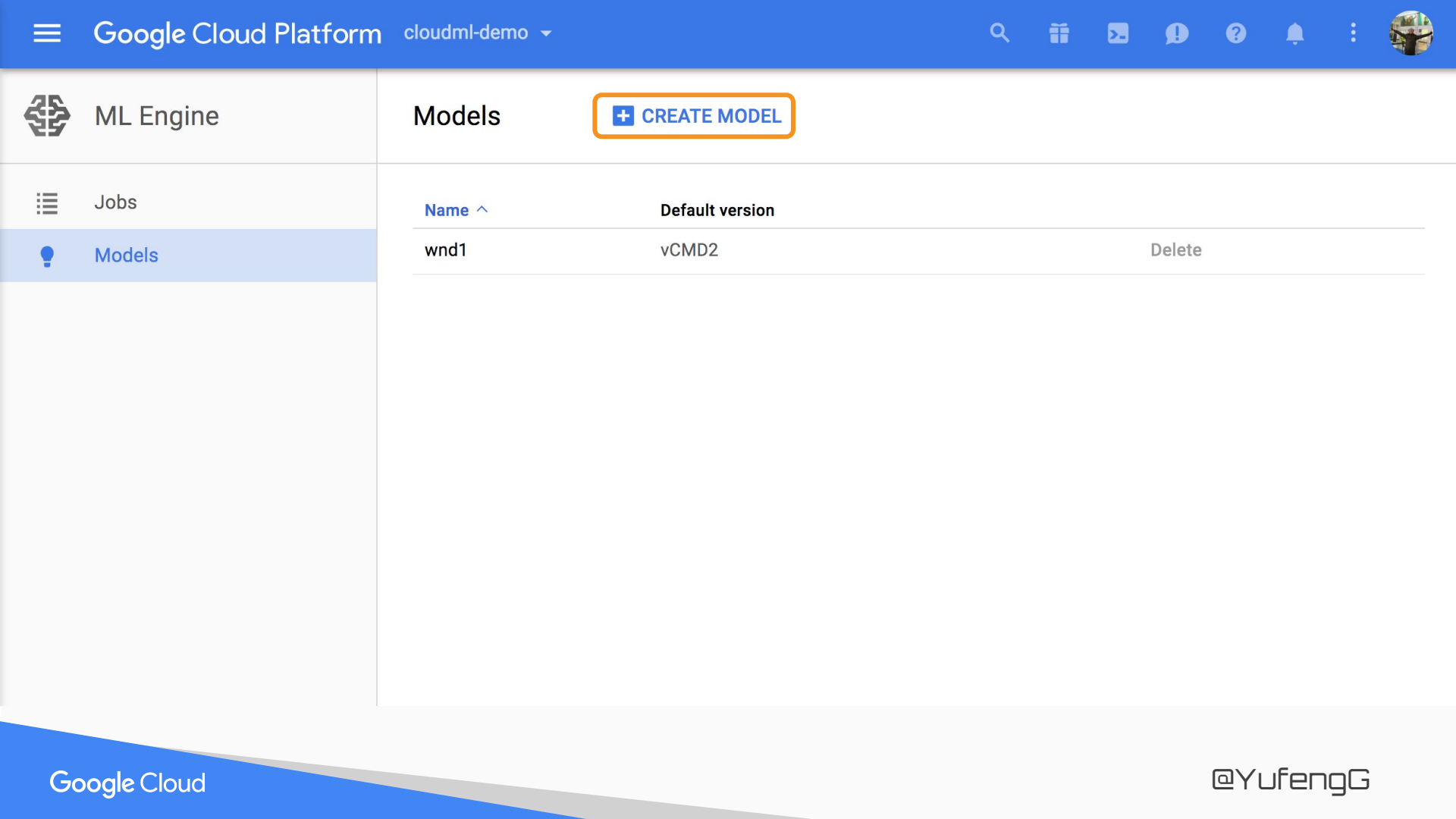
- TensorFlow
- Handles provisioning, scaling, and monitoring
- Integrated: Dataflow, Storage, Datalab
- Automatically tune models with HyperTune

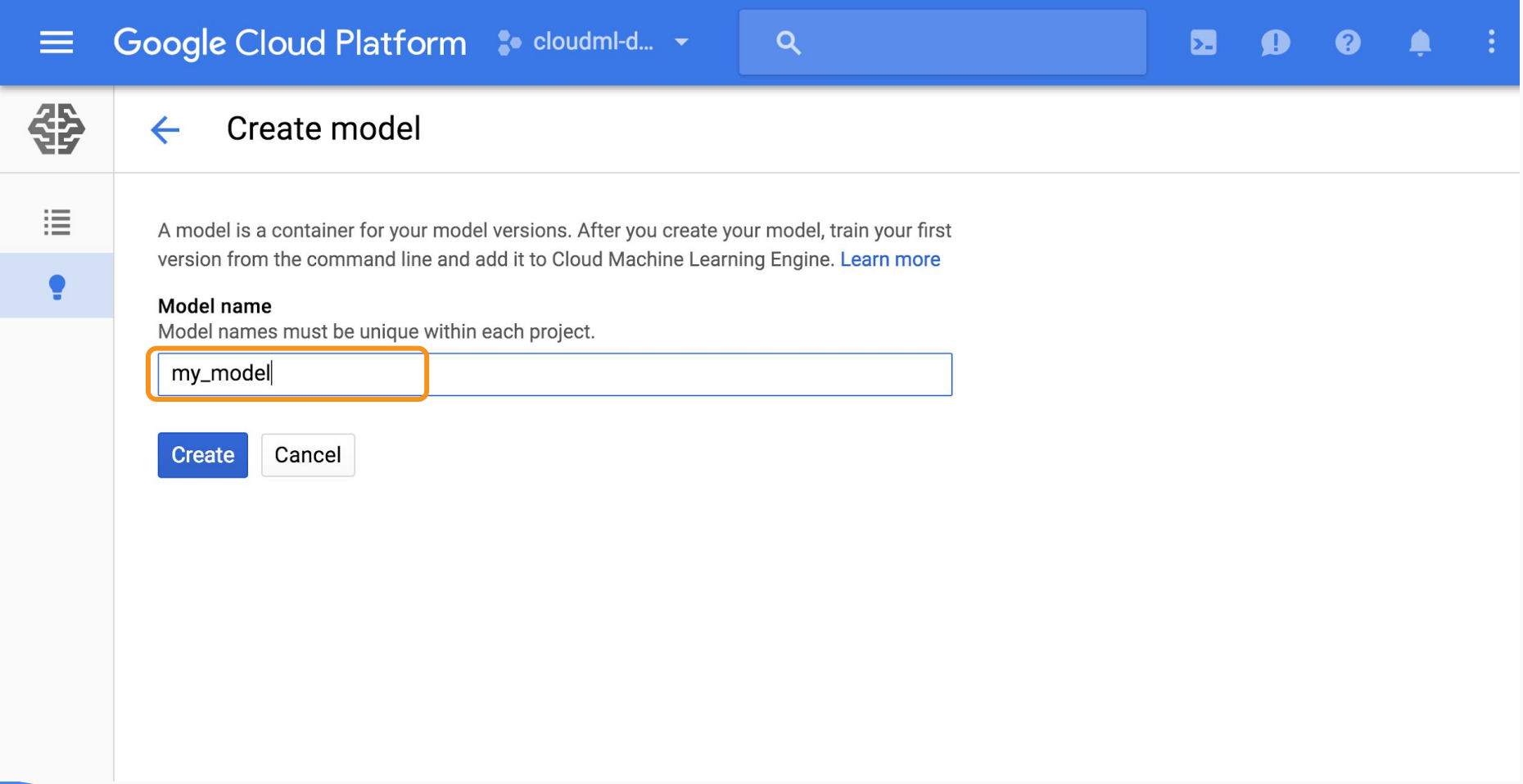
What is TensorFlow Serving?

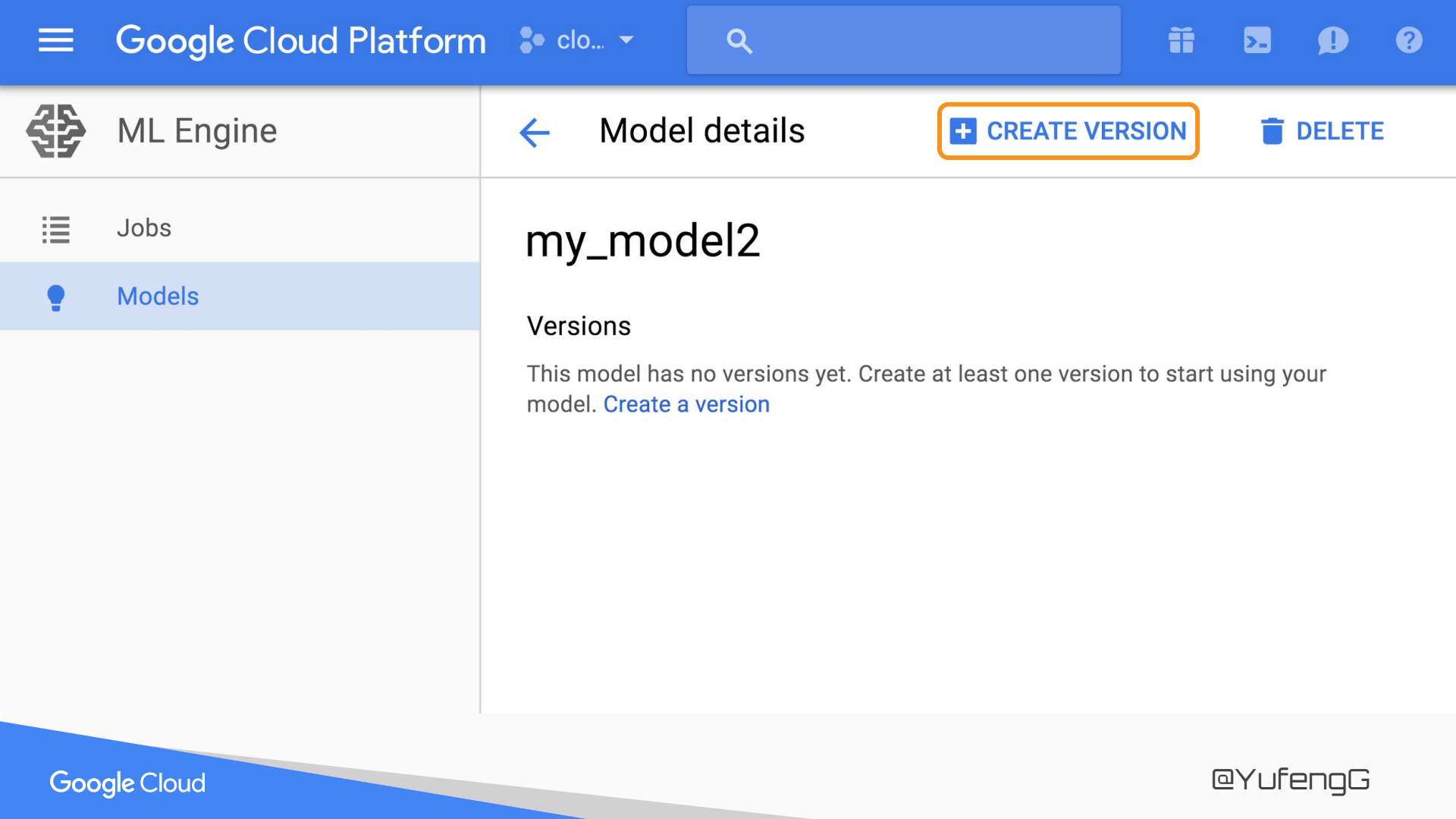
- C++ Libraries
  - TensorFlow model save / export formats
  - Generic core platform
- Binaries
  - Best practices out of the box
  - Docker containers, K8s tutorial
- Hosted Service across
  - Google Cloud ML Engine
  - Internal service

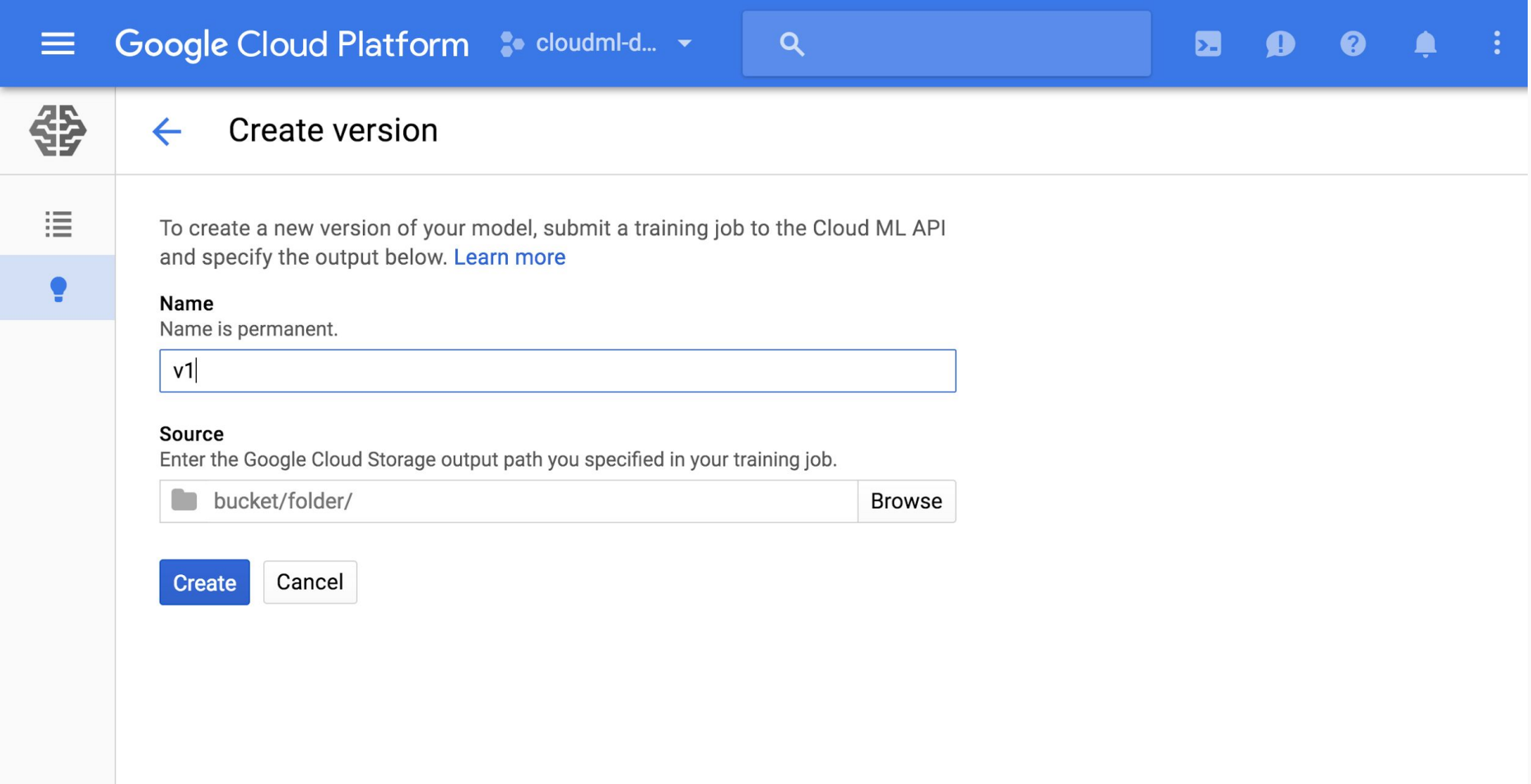


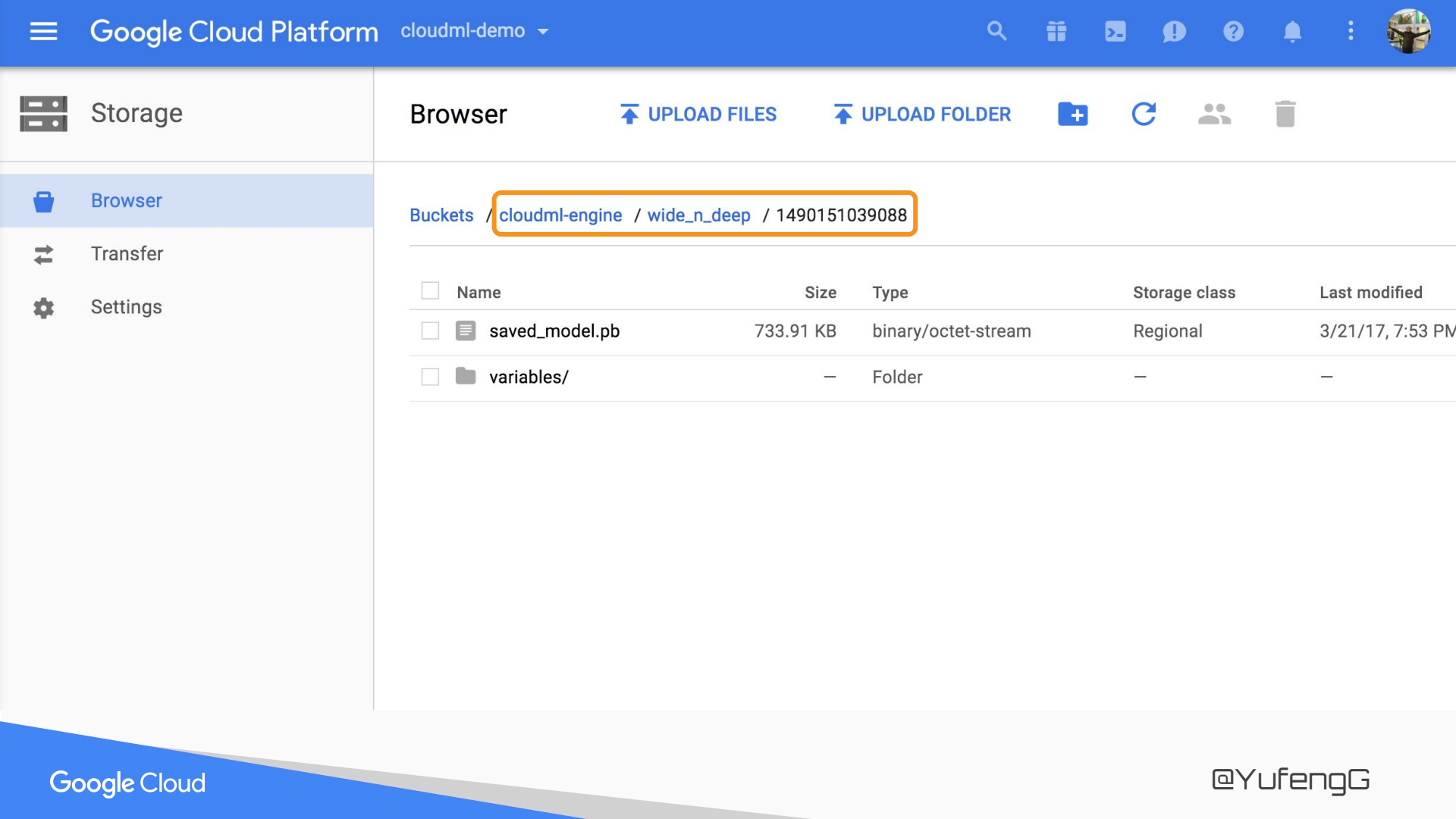
Model Creation

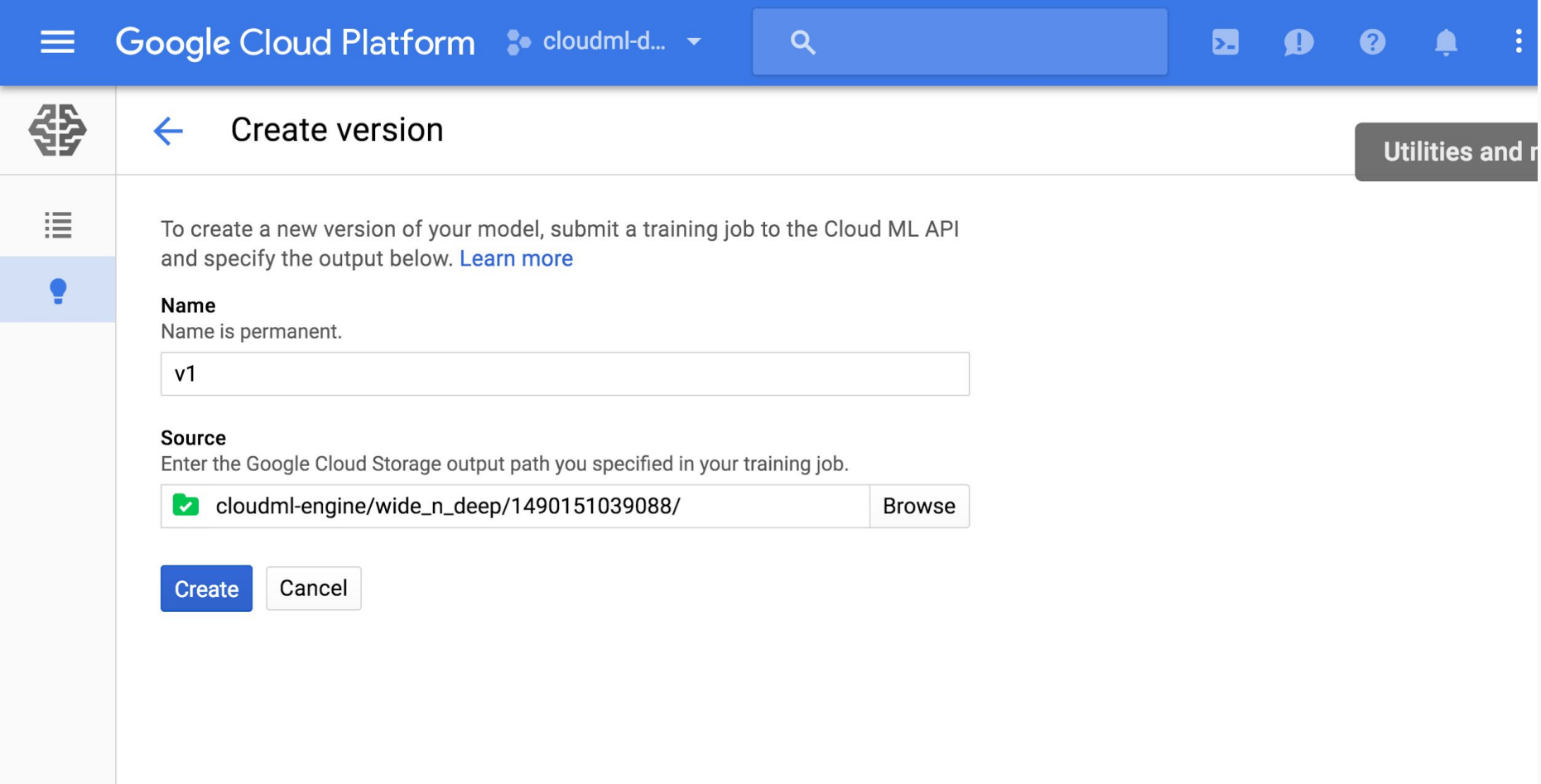

Google Cloud Platform
cloudml-demo
ML Engine
Jobs
Models
Models
CREATE MODEL
Name
Default version
wnd1
vCMD2
Delete

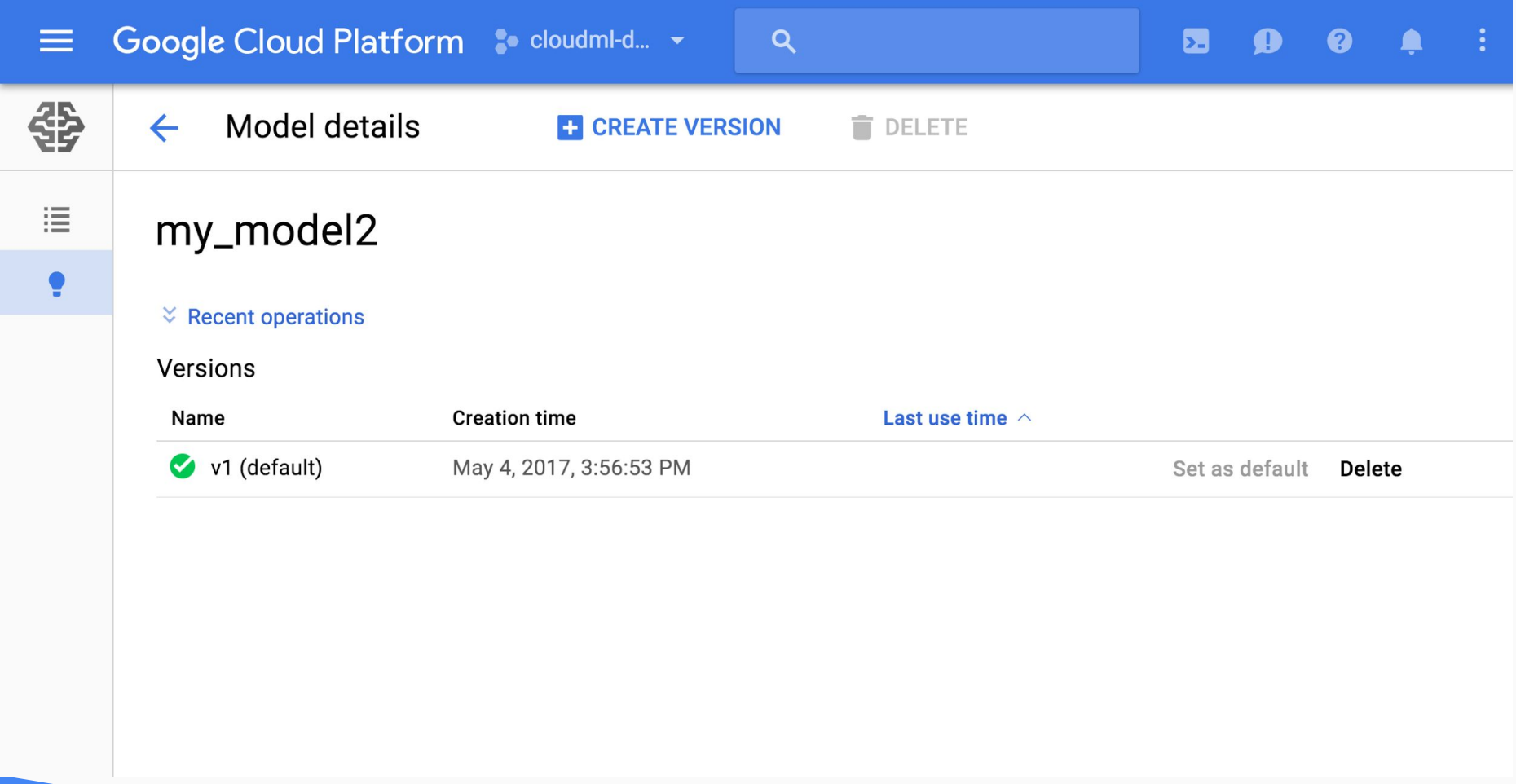

Google Cloud Platform
cloudml-d...
Create model
A model is a container for your model versions. After you create your model, train your first version from the command line and add it to Cloud Machine Learning Engine. Learn more
Model name
Model names must be unique within each project.
my_model
Create
Cancel


Google Cloud Platform
clo...
ML Engine
Jobs
Models
Model details
CREATE VERSION
DELETE
my_model2
Versions
This model has no versions yet. Create at least one version to start using your model. Create a version
Google Cloud
@YufengG


Google Cloud Platform
cloudml-d...
Create version
To create a new version of your model, submit a training job to the Cloud ML API and specify the output below. Learn more
Name
Name is permanent.
v1
Source
Enter the Google Cloud Storage output path you specified in your training job.
bucket/folder/
Browse
Create
Cancel


Google Cloud Platform
cloudml-demo
Storage
Browser
Transfer
Settings
Browser
UPLOAD FILES
UPLOAD FOLDER
Buckets / cloudml-engine / wide_n_deep / 1490151039088
Name
Size
Type
Storage class
Last modified
saved_model.pb
733.91 KB
binary/octet-stream
Regional
3/21/17, 7:53 PM
variables/
–
Folder
–
–


Google Cloud Platform
cloudml-d...
Create version
Utilities and r
To create a new version of your model, submit a training job to the Cloud ML API and specify the output below. Learn more
Name
Name is permanent.
v1
Source
Enter the Google Cloud Storage output path you specified in your training job.
cloudml-engine/wide_n_deep/1490151039088/
Browse
Create
Cancel


Google Cloud Platform
cloudml-d...
Model details
CREATE VERSION
DELETE
my_model2
Recent operations
Versions
Name
Creation time
Last use time
v1 (default)
May 4, 2017, 3:56:53 PM
Set as default
Delete

```
export MODEL_NAME='my_model'
gcloud ml-engine models --regions us-central1 create $MODEL_NAME
```

```
export MODEL_NAME='cloudwnd'
export VERSION_NAME='learn_runner_standard'
export DEPLOYMENT_SOURCE='gs://cloudml-engine/widendeep_yufeng
g_20170410_164903/model_WIDE_AND_DEEP_1491857627/export/Servo/
1491857907860'

$ gcloud ml-engine versions create $VERSION_NAME --model
$MODEL_NAME --origin $DEPLOYMENT_SOURCE
Creating version (this might take a few minutes)......
```

Instance Prediction

```
{
  "age": 25,
  "workclass": " Private",
  "education": " 11th",
  "education_num": 7,
  "marital_status": "
Never-married",
  "occupation": "
Machine-op-inspct",
  "relationship": "
Own-child",
  "race": " Black",
  "gender": " Male",
  "capital_gain": 0,
  "capital_loss": 0,
  "hours_per_week": 40,
  "native_country": "
United-States"
}
```

```
{
  "age": 42,
  "workclass": "
Self-emp-inc",
  "education": " HS-grad",
  "education_num": 9,
  "marital_status": "
Married-civ-spouse",
  "occupation": "
Exec-managerial",
  "relationship": " Husband",
  "race": " White",
  "gender": " Male",
  "capital_gain": 5178,
  "capital_loss": 0,
  "hours_per_week": 50,
  "native_country": "
United-States"
}
```

```
$ gcloud ml-engine predict --model wnd1 --version vCMD2 --json-instances census.json
CLASSES  LOGISTIC                 LOGITS                   PROBABILITIES
0        [0.0051437602378427 98]  [-5.2648138999938965]    [0.9948562383651733, 0.0051437602378427 98]
1        [0.8839852213859558]     [2.0307230949401855]     [0.1160147413611412, 0.8839852213859558]
```

```
PROBABILITIES
[0.9948562383651733, 0.0051437602378427 98]
[0.1160147413611412, 0.8839852213859558]
```

```
{
  "probabilities": [
    0.11601490527391434,
    0.8839850425720215
  ],
  "logits": [
    2.030721426010132
  ],
  "classes": 1,
  "logistic": [
    0.8839850425720215
  ]
}
```

```
{
  "probabilities": [
    0.9948562383651733,
    0.005143760237842798
  ],
  "logits": [
    -5.2648138999938965
  ],
  "classes": 0,
  "logistic": [
    0.005143760237842798
  ]
}
```

Training

many *examples*

Prediction

answer questions

Training

many *examples*

Prediction

answer questions

Training

many *examples*

Prediction

answer questions

bit.ly/ai-adventures

Exploring the art, science, and tools of machine learning

@YufengG @BretMcG

Thank you!

@YufengG
yufengg.com

Resources:

Cloud Machine Learning Engine
cloud.google.com/ml-engine

TensorFlow
tensorflow.org

More machine learning
bit.ly/ai-adventures
bit.ly/widendeep-census

The End