Machine Learning

Google Cloud Platform Fundamentals: Big Data and Machine Learning

Version #1.1

Google Cloud

Notes:

75 minutes lecture + 45 minutes labs

In the previous chapter, we looked at ways to migrate your workloads to GCP. Those -- relational databases, Hadoop ecoysystem -- are probably things you are doing already, and we wanted to make it easy to move those workloads to GCP. Hence, managed MySQL and Managed Hadoop. While those are good, as we discussed in Chapter 1, it is not taking full advantage of what the Cloud has to offer. In this chapter, we will look at more transformational use cases. These are going to be about *changing* how you compute with GCP. More transformational use cases. What kinds of transformational use cases?

Notes:

1. Introduction
Overview of GCP as a whole, but with emphasis on the data-handling aspects of the platform

- GCP, GCP Big Data
- Usage scenarios
- Create an account on GCP

2. Foundation of GCP
Compute and Storage with a focus on their value in data ingest, storage, and federated analysis

- Compute Engine
- Cloud Storage
- Start GCE instance
- Upload data to GCS

3. Data analytics on the Cloud
Common use cases that Google manages for you and for which there is an easy migration path to the Cloud

- Cloud SQL
- Dataproc
- Import data into and query Cloud SQL
- Machine Learning with Dataproc

In the morning, we will complete Modules 1 and 2 and get halfway through Module 3.

4a. Scaling data analysis
Change how you compute, not just where you compute with GCP

- Datalab
- Datastore, Big Table
- BigQuery

5. TensorFlow
Change how you compute, not just where you compute with GCP

- TensorFlow
- Datalab instance
- BigQuery
- Demand forecasting with ML

6. Data processing architectures
Scaleable, reliable data processing on GCP

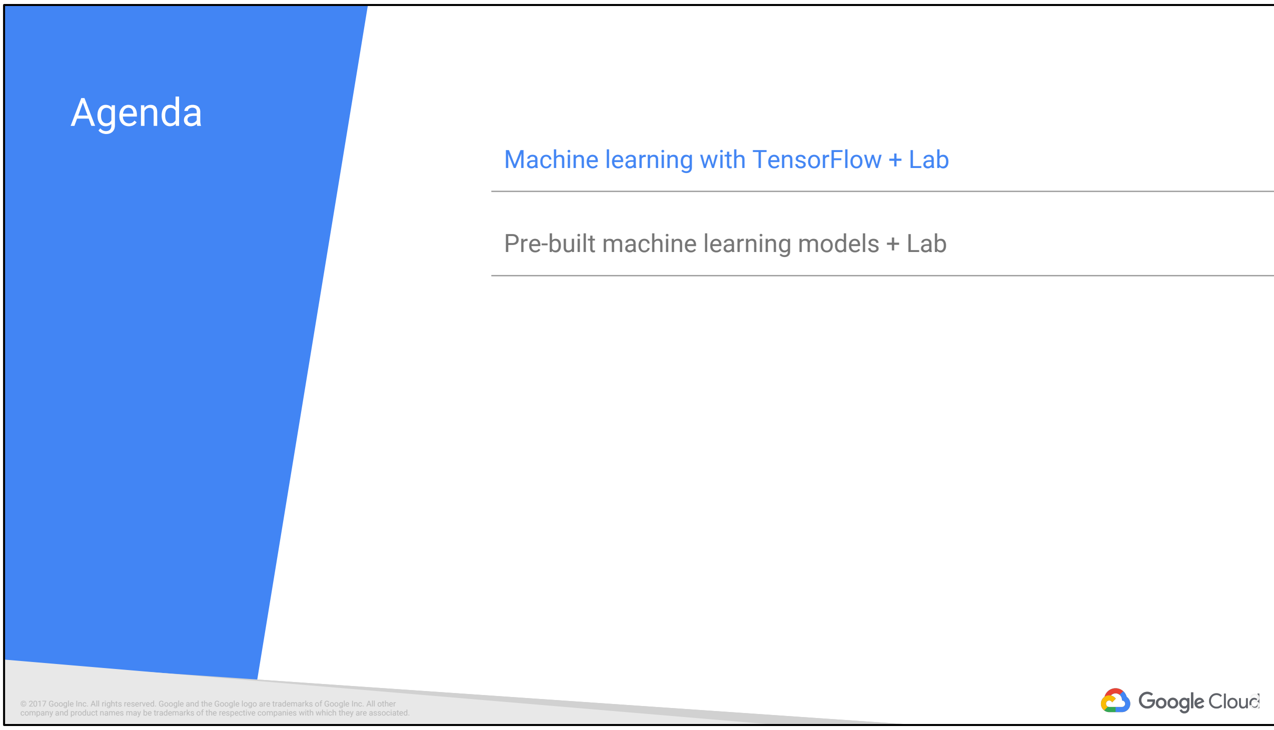
- Pub/Sub
- Dataflow

7. Summary
Course summary

- Resources

Please feel free to use the appendixes for self-study.
In the morning, we will get halfway through Module 3.

Please feel free to use the appendixes for self-study.



Notes:

In this chapter, we look at building custom machine learning models using TensorFlow.
In some cases though, you can take advantage of pre-built ML models, and simply apply them.

TensorFlow is an open source library that underlies many Google products

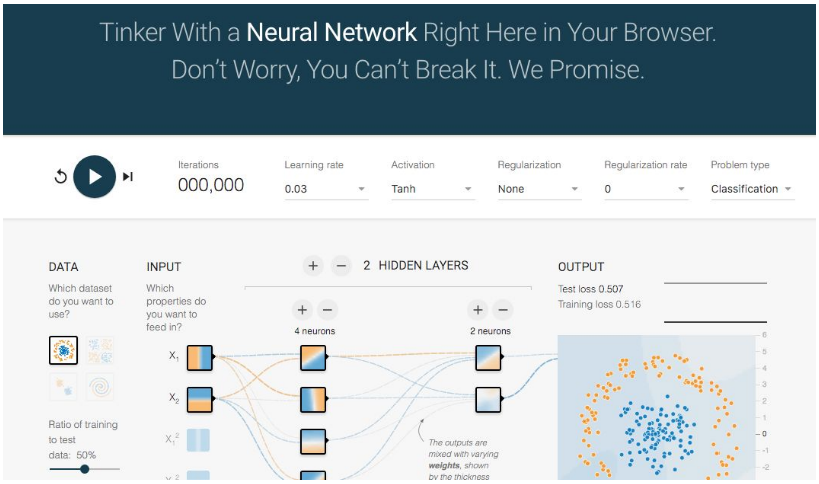
Google Cloud

Notes:
Tensorflow is:

- C++ engine and API
- Python API that talks to C++
- Deep-learning neural networks with auto-differentiation of objective functions

On the last point: Normally, you'd use an objective function - [actual - predicted]^2 - and provide the library with its derivative - 2*[actual-predicted] - so that it can do gradient descent. TensorFlow will let you define any objective function, even something like classification accuracy for which there is no close-form derivative, and it will still work.

Demo: Playing with neural networks to learn what they are



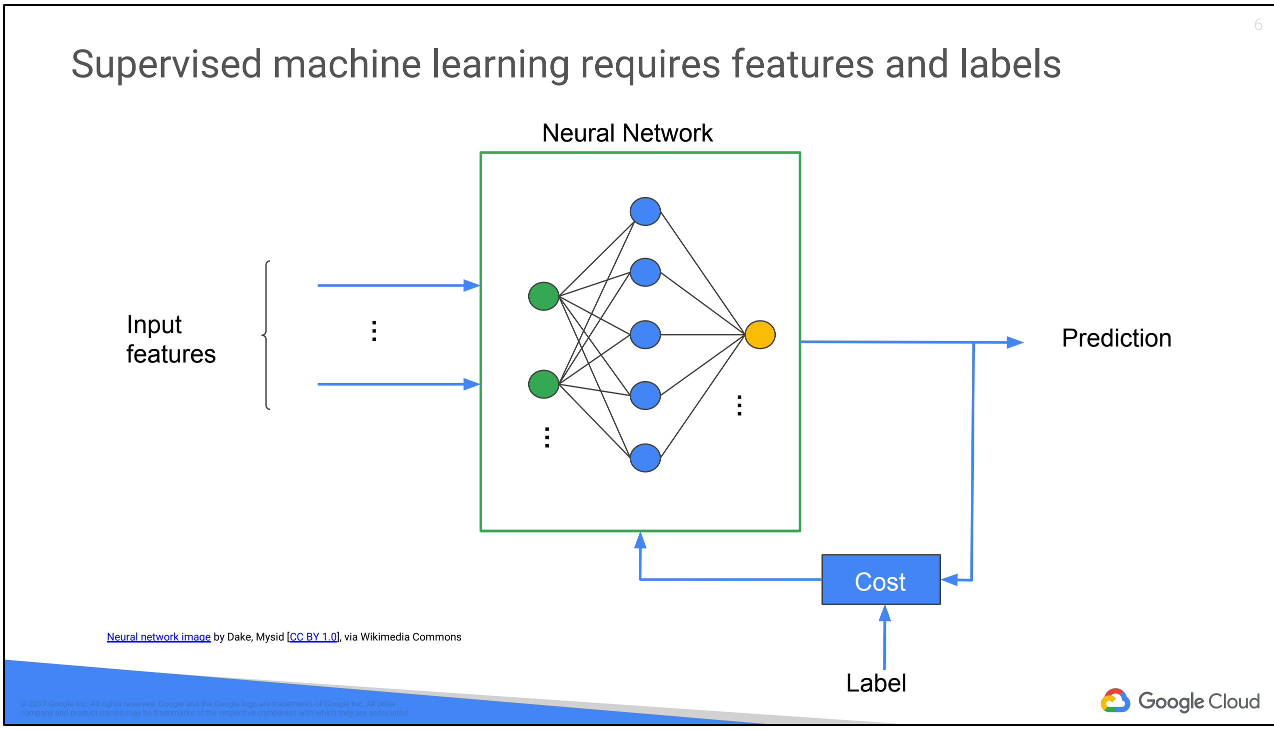
http://playground.tensorflow.org/

Google Cloud

Notes:

Click the link and use it to explain inputs, outputs, hidden layers, and so on.

A tensor is a multi-dimensional array. In the taxicab data, all the weather inputs and taxi data together form the vector of inputs. As we start layering these nodes one on top of each other in parallel, the numbers become multi-dimensional (a tensor). These flow through the neural network creating the output prediction ("class labels"). The taxicab network uses ReLu activation layers and is a Regression -- if you click on the image instead of on the link, you'll get taken to those settings.



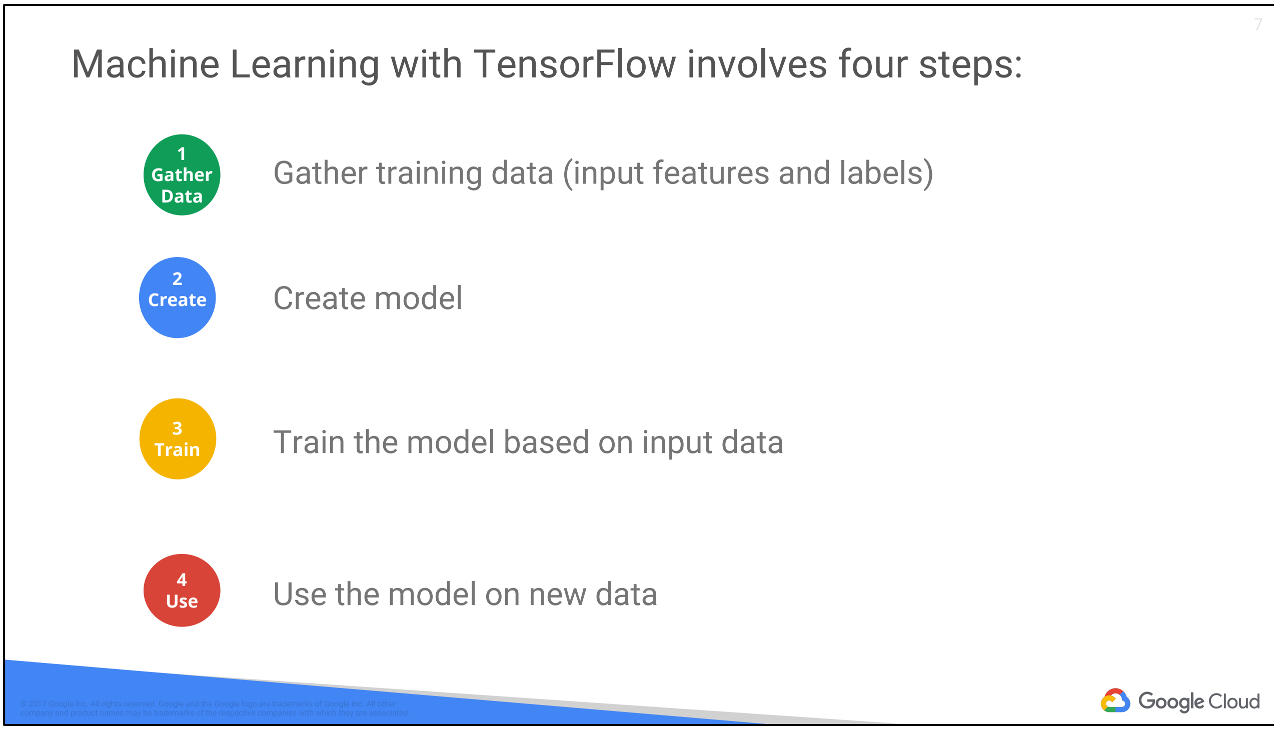
Notes:

Training the model:

- Compute cost (true value - predicted value)
- Update model weights to minimize cost

Prediction model does not need true values, just the predictor variables …

[Neural network image](#) by Dake, Mysid [[CC BY 1.0](#)], via Wikimedia Commons



Notes:

You did #1 in a previous lab. Let's look at the other steps on the following slides.

Gather training data and select input features

1
Gather Data

Input features

	daynumber	dayofweek	mintemp	maxtemp	rain	numtrips
104	77	4	28.9	37.9	0.01	51635
9	356	2	32.0	43.0	0.00	46781
114	67	1	35.1	48.0	0.00	57377
11	354	7	30.0	37.9	0.00	74101
316	49	3	19.0	39.9	0.05	28463

discard

target

Google Cloud

Notes:

If you have the social security number in your data, and you train, you could would get a model that is tied to specific individuals and won't generalize. The reason for numeric is that we are going to be adding, multiplying, and so on, these numbers in the neural network. The day of the week, for example, is categorical, not numeric. Sunday=1 is a numeric code, not a number. For example: Sunday * Tuesday makes no sense.

All input features need to be numeric

1
Gather Data

Use as-is

	mintemp	maxtemp	rain	day_1	day_2	day_3	day_4	day_5	day_6	day_7
104	28.9	37.9	0.01	0	0	0	1	0	0	0
9	32.0	43.0	0.00	0	1	0	0	0	0	0
114	35.1	48.0	0.00	1	0	0	0	0	0	0
11	30.0	37.9	0.00	0	0	0	0	0	0	1
316	19.0	39.9	0.05	0	0	1	0	0	0	0

One-hot encoding

Google Cloud

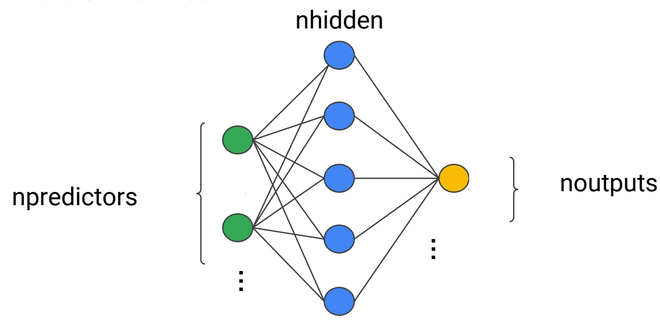
Notes:

Predictor columns need to be numeric (not categorical data or “codes”)

The reason for numeric is that we are going to be adding, multiplying, and so on, these numbers in the neural network. The day of the week, for example, is categorical, not numeric. Sunday=1 is a numeric code, not a number. For example: Sunday * Tuesday makes no sense.

Create a neural network model, defining the number of feature columns and hidden units

2
Create



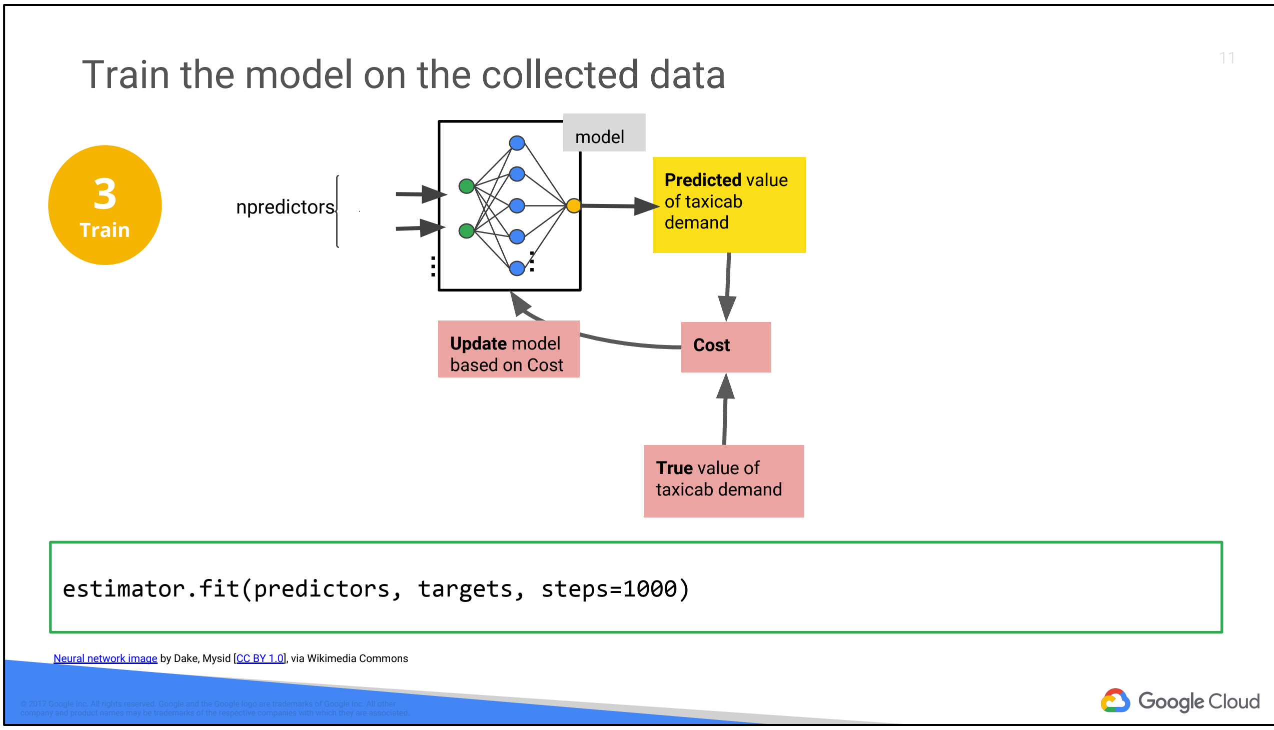

```
estimator = DNNRegressor(hidden_units=[5], feature_columns=[...])
```

Neural network image by Dake, Mysid [CC BY 1.0], via Wikimedia Commons

Google Cloud

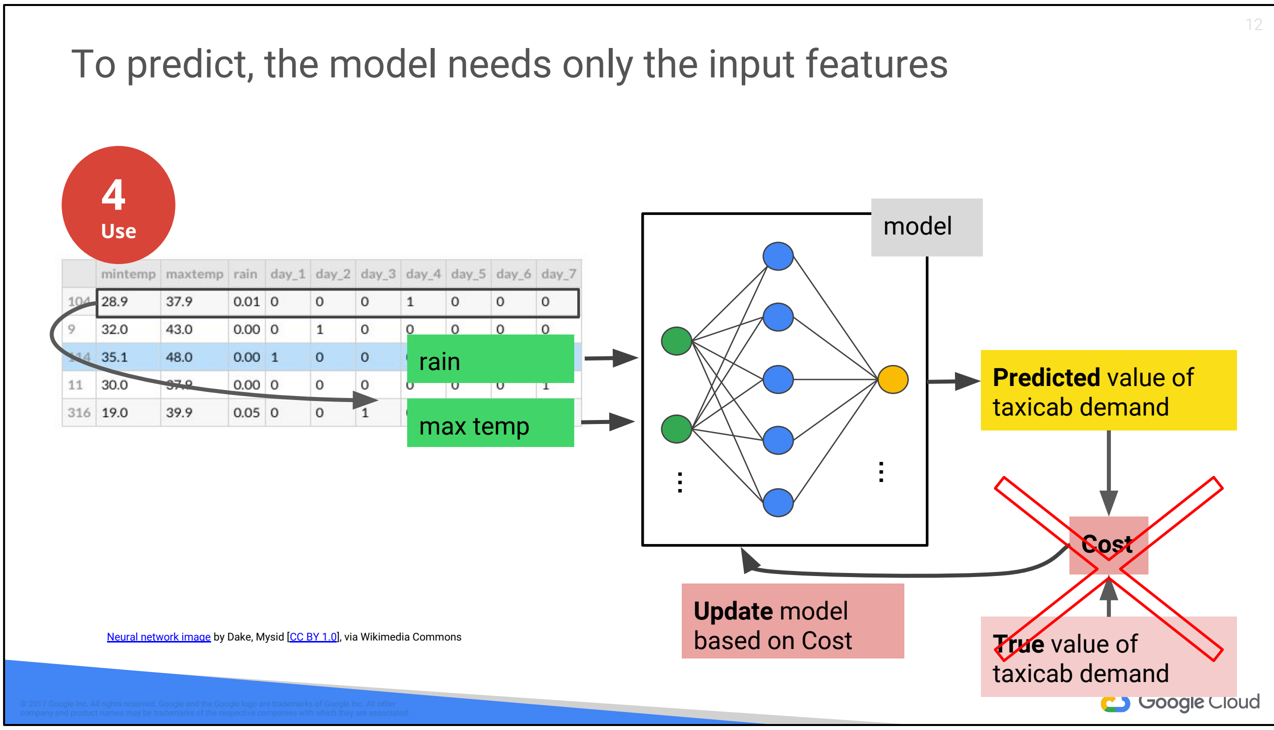
Notes:
Neural network image by Dake, Mysid [CC BY 1.0], via Wikimedia Commons

At this points, the weights are all random variables. We need to compute the weights



The code to train the model:

estimator.fit(predictors, targets, steps=1000)



Notes:

Neural network image by Dake, Mysid [CC BY 1.0], via Wikimedia Commons

Use the model to predict

4
Use

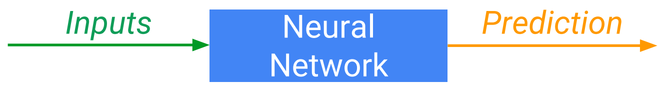
```
input = pd.DataFrame.from_dict(data =
                               {'dayofweek' : [4, 5, 6],
                                'mintemp' : [60, 15, 60],
                                'maxtemp' : [80, 80, 65],
                                'rain' : [0, 0.8, 0]})


# read trained model from /tmp/trained_model
estimator = DNNRegressor(model_dir='/tmp/trained_model',
                                          hidden_units=[5])

pred = estimator.predict(input.values)
print pred
```

Lab 2, Part 2: Carry out ML with TensorFlow

Lab 2, Part 2: Carry out ML with TensorFlow



In this lab, you build a neural network to predict taxicab demand on a day-by-day basis using TensorFlow.

Google Cloud

Notes:

Image from https://pixabay.com/en/cab-yellow-cab-cars-street-taxi-453028/ (cc0).

Agenda

Machine learning with TensorFlow + Lab

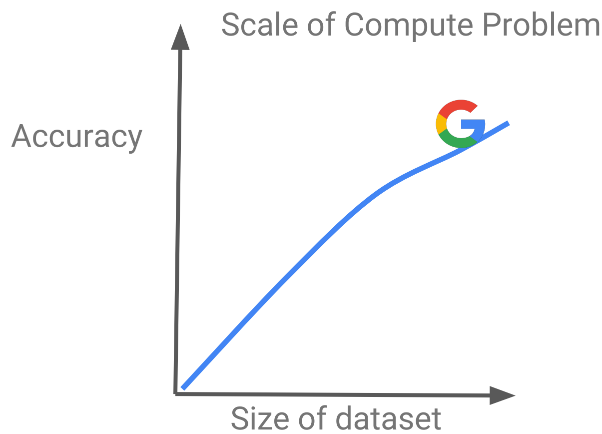
Pre-built machine learning models + Lab

Google Cloud

Notes:

In this chapter, we look at building custom machine learning models using TensorFlow.
In some cases though, you can take advantage of pre-built ML models, and simply apply them.

The accuracy of a ML problem is driven largely by the size and quality of the dataset; this is why ML requires massive compute



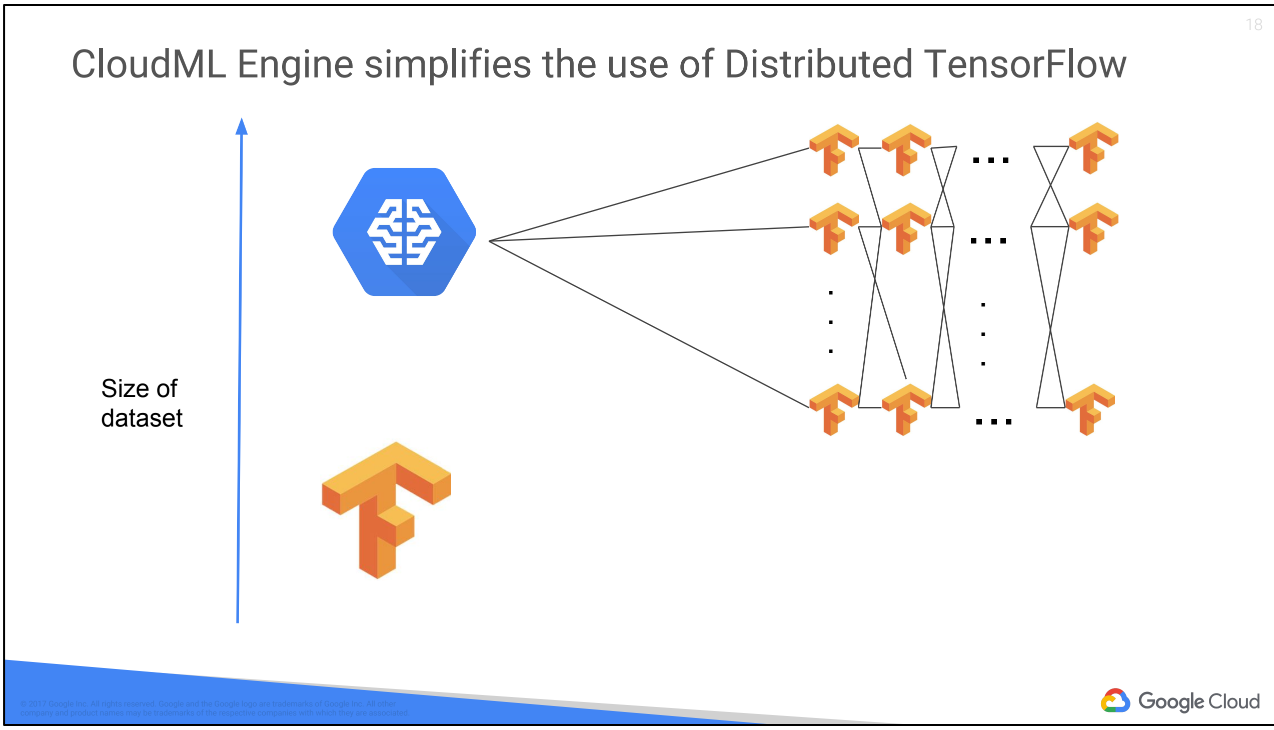
https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html

Google Cloud

Notes:

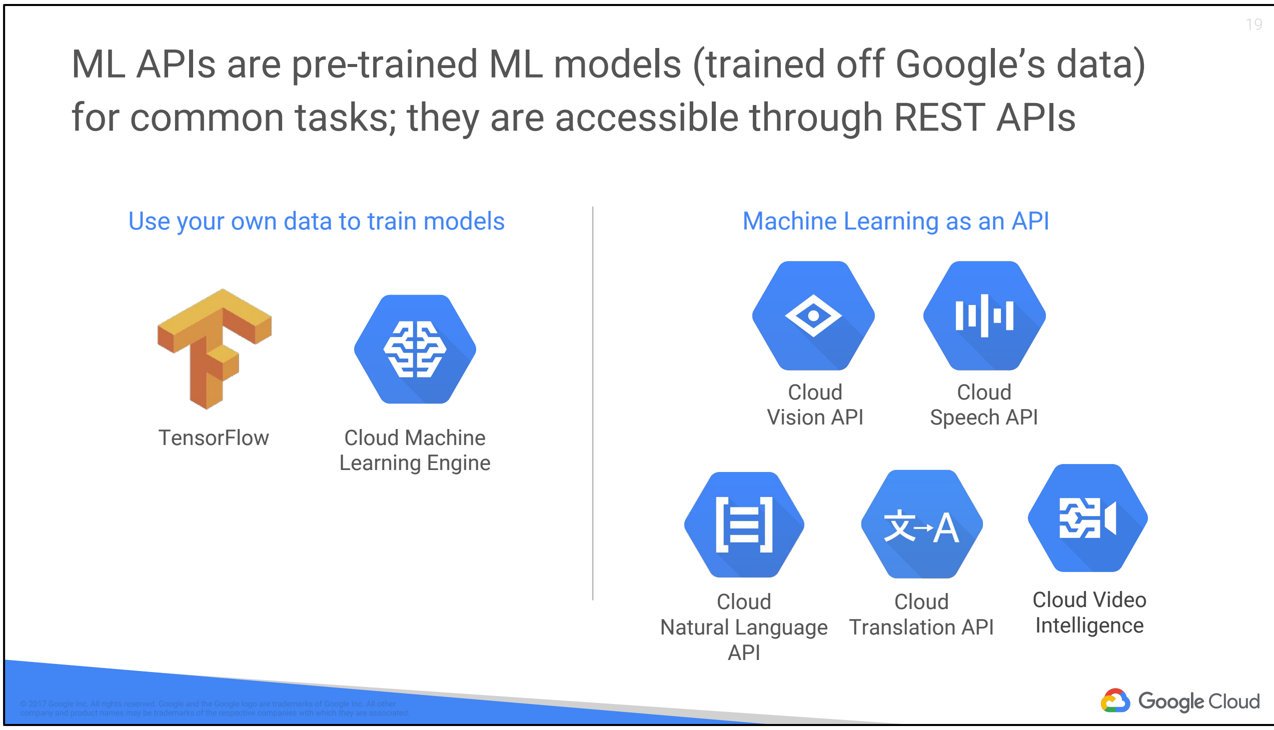
When you look at vision, translate, etc. you may ask why Google? Because we have the large scale distributed system, the extremely large datasets and the ML expertise to get the most accurate solutions to these problems.

Because Cloud Machine Learning is built on open-sourced TensorFlow, you aren't tied to Google Cloud Platform.

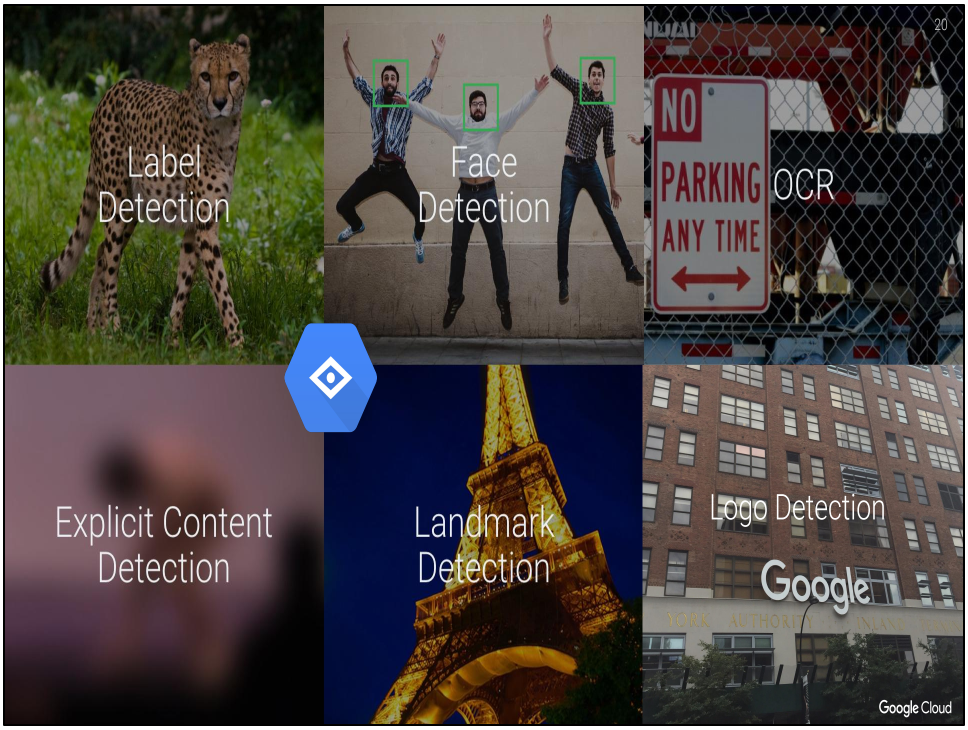


Notes:

As the size of dataset increases, we can get better performance by building deeper and larger neural networks. However, training time and computation needs will also dramatically increase. If you use a single GPU, you have to wait months for training to complete. So you need TensorFlow Distributed, and that needs to run on the cloud. Google has been building infrastructure to do this, and with Cloud Machine Learning, you get to use deep networks, TensorFlow Processing Units (TPUs) in a distributed, no-ops network.



Not everything needs to be a custom model. You can piggy back off Google for common tasks



Let me walk you through what Vision API can really do:

- **Label Detection:** The API can detect broad sets of categories within an image, ranging from modes of transportation to animals.
- **Face Detection:** The API can detect multiple faces within an image, along with the associated key facial attributes like emotional state or wearing headwear.
- **OCR:** The API can detect and extract text within an image, with support for a broad range of languages.
- **Explicit Content Detection:** We can detect explicit content like adult content or violent content within an image.
- **Landmark Detection:** The APi can detect popular natural and manmade structures within an image.
- **Logo Detection:** We can detect popular product logos within an image.

Face detection

```
"faceAnnotations" : [
  {
    "headwearLikelihood" : "VERY_UNLIKELY",
    "surpriseLikelihood" : "VERY_UNLIKELY",
    rollAngle" : -4.6490049,
    "angerLikelihood" : "VERY_UNLIKELY",
    "landmarks" : [
      {
        "type" : "LEFT_EYE",
        "position" : {
          "x" : 691.97974,
          "y" : 373.11096,
          "z" : 0.000037421443
        }
      }
```

```
"detectionConfidence" : 0.93568963,
"joyLikelihood" : "VERY_LIKELY",
"panAngle" : 4.150538,
"sorrowLikelihood" : "VERY_UNLIKELY",
"tiltAngle" : -19.377356,
"underExposedLikelihood" : "VERY_UNLIKELY",
"blurredLikelihood" : "VERY_UNLIKELY"
```


Google Cloud

Here's an example of what the JSON response looks like for face detection - it's a picture [Googler] Sara took with 2 other teammates on a trip to Jordan.

It returns an object for each face found in an image

Web annotations


```
{
  "entityId": "/m/0gff2yr",
  "score": 5.92256,
  "description": "ArtScience Museum"
}
```

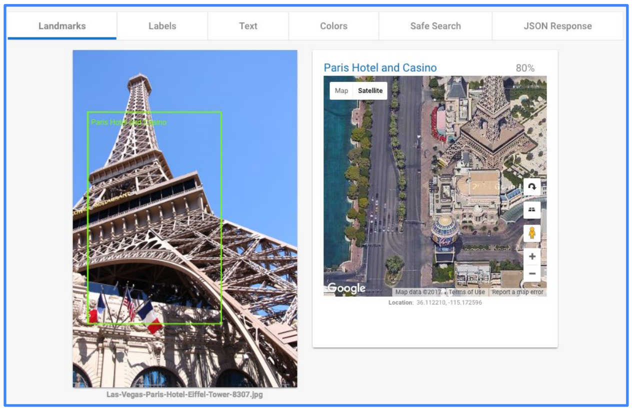
```
{
  "entityId": "/m/016ms7",
  "score": 1.44038,
  "description": "Ford Anglia"
}
```

```
{
  "entityId": "/m/0h898pd",
  "score": 7.4162,
  "description": "Harry Potter (Literary Series)"
}
```


Google Cloud

Image source:
https://www.google.com/search?site=imghp&tbm=isch&q=star%20wars%20tshirt&tbs=sur:fmc#tbs=sur:fmc&tbm=isch&q=harry+potter+tour+flying+car&*&imgrc=fhR-slSfplfxvM:

Try it in the browser with your own images



cloud.google.com/vision

Google Cloud

The Translation API supports 100+ languages

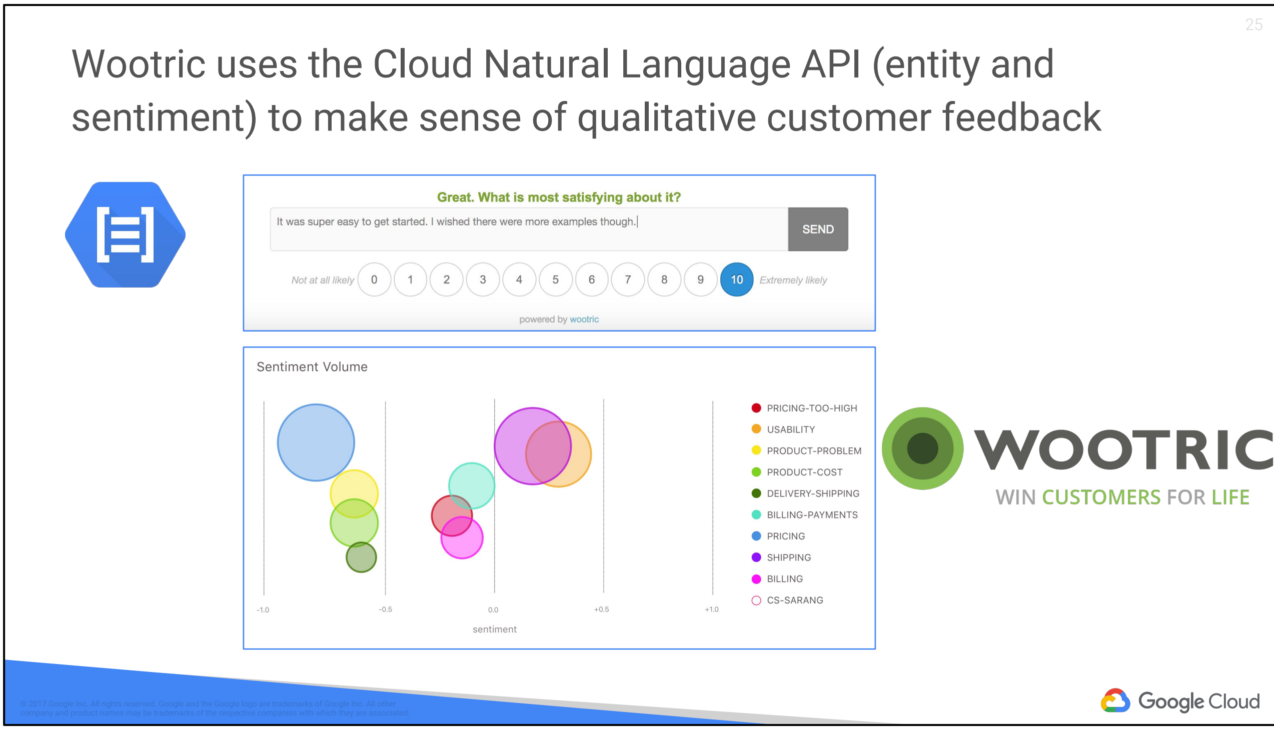
https://cloud.google.com/translate/

Google Cloud

Notes:

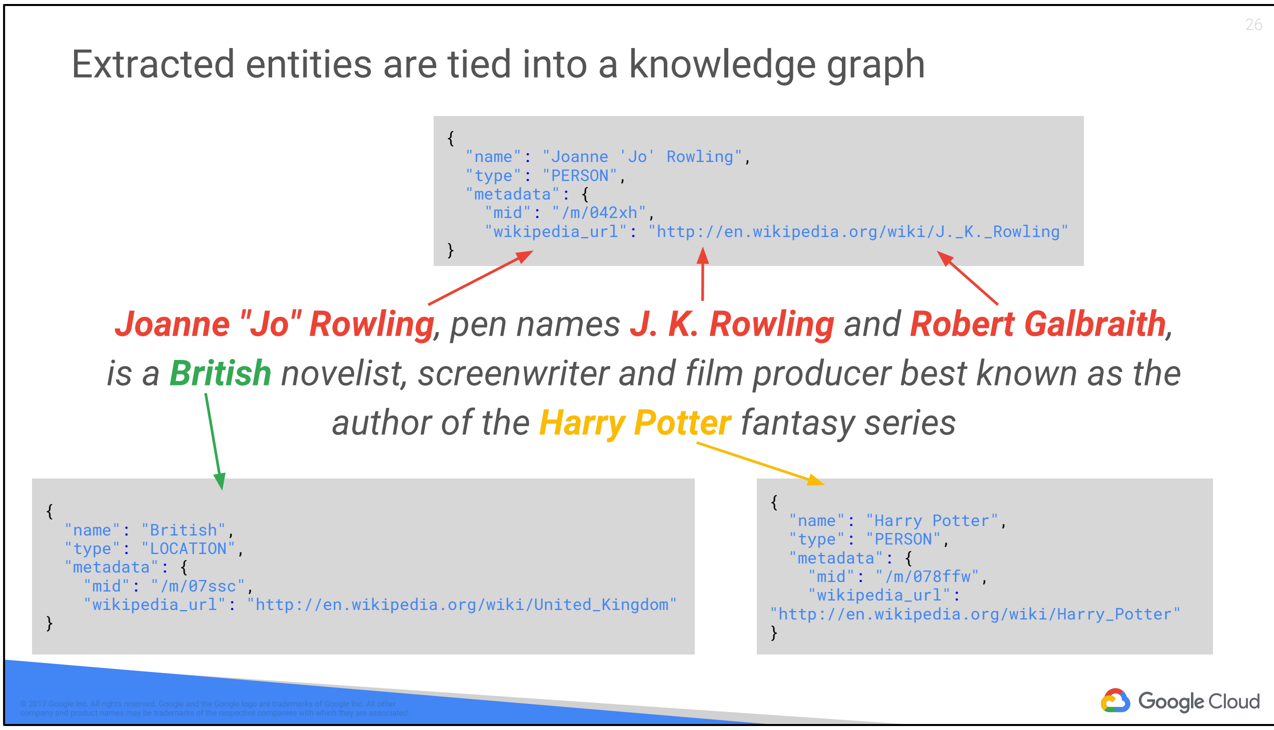
The Translate API [they've all seen it on their phones, so no need to demo] underlies the product shown above. Simply place your camera over a sign, and it gets auto-translated for you. This is a combination of Vision API (to do optical character recognition) and Translate API (to do actual translation). Vision API supports 90+ languages, can detect the language of source text, and is highly scalable.

You can try this one too on the web. Click on image to check it out ...



See:
https://cloud.google.com/blog/big-data/2017/03/analyzing-customer-feedback-using-machine-learning

- **What is Wootric**? Intelligent customer feedback platform to help businesses improve customer service
  - They do this by collecting millions of customer survey responses each week
- **Problem**: With all this feedback flowing in, they need a way to gain insight without having to manually classify it
- **Solution**: automate text and sentiment analysis with the NL API
  - Use syntax and entity analysis to get the subject of the feedback
  - Use sentiment analysis to calibrate feedback??
- **Impact:** automation with NLP saves time and resources for clients who typically have people manually tagging thousands of comments each week
  - Classification allows for targeting routing in realtime



THus Joanne Rowling, J.K.Rowling, etc. are all the same entity and they all point to same wikipedia entry.
Also, entity “British” points to wikipedia entry for United_Kingdom

When you analyze sentiment, you get a score (positive/negative) as well as a magnitude (how intense?)

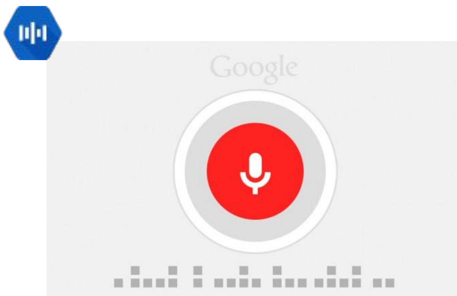
The food was excellent, I would definitely go back!

```
{
  "documentSentiment": {
    "score": 0.8,
    "magnitude": 0.8
  }
}
```


Google Cloud

Magnitude is related the total positivity of a text, so longer text will have higher magnitude (unless it is neutral). Commonly, you'd normalize it by the length of the text

The Cloud Speech API can be used to transcribe audio to text



http://cloud.google.com/speech

Notes:

The Speech API [click on image to launch demo]

Like the Vision API, the Video Intelligence API can identify labels in a video, along with a timestamp

```
{
  "description": "Bird's-eye view",
  "language_code": "en-us",
  "locations": {
    "segment": {
      "start_time_offset": 71905212,
      "end_time_offset": 73740392
    },
    "confidence": 0.96653205
  }
}
```

https://cloud.google.com/video-intelligence/

Google Cloud

What does the JSON from the Video API look like? Here's an example of a video from the White House.

- For each label found in the video, we get a segments object which tells us the times this label appears in the video. Here's an example of the label "Bird's-eye view" - the start and end times are in microseconds
- We also get a confidence value ranging from 0 - 1 → this indicates how confident the API is that it has correctly detected the label

To demo, use one of the mp4 already available, or load your own mp4 video to GCS.

Lab 2, Part 3: Machine Learning APIs

Lab 2, Part 3: Machine Learning APIs

Use several of the Machine Learning APIs (Vision, Translate, Natural Language Processing, Speech) from Python

Notes:

The translation won't be ideal. Translate API works best with full sentences and more context. Street signs are a difficult use case because the text is typically terse and it's not embedded in a longer paragraph.

To find the JSON format expected by each of these APIs, look at
https://developers.google.com/apis-explorer/
Most of the APIs will take either embedded content or a URI to a file on cloud storage
The Vision and Speech in the example shows URI. But you can embed (base64-encoded) image/audio directly into POST as well.

Module Review

Module review

Match the use case on the left with the product on the right

Create, test new machine learning methods
No-ops, custom machine learning applications at scale
Automatically reject inappropriate image content
Build application to monitor Spanish twitter feed
Transcribe customer support calls

1. Vision API
2. TensorFlow
3. Speech API
4. Cloud ML
5. Translation API

Notes:

A. TensorFlow [new methods = Research]
B. Cloud Machine Learning [custom machine learning applications = application of TensorFlow research in your own domain, on your own data]
C. Vision API
D. Translate API
E. Speech API

Module review answer

Match the use case on the left with the product on the right

Create, test new machine learning methods (2)
No-ops, custom machine learning applications at scale (4)
Automatically reject inappropriate image content (1)
Build application to monitor Spanish twitter feed (5)
Transcribe customer support calls (3)

1. Vision API
2. TensorFlow
3. Speech API
4. Cloud ML
5. Translation API

Google Cloud

Notes:

A. TensorFlow [new methods = Research]
B. Cloud Machine Learning [custom machine learning applications = application of TensorFlow research in your own domain, on your own data]
C. Vision API
D. Translation API
E. Speech API

Resources (1 of 2)

Cloud Datastore	https://cloud.google.com/datastore/
Cloud Bigtable	https://cloud.google.com/bigtable/
Google BigQuery	https://cloud.google.com/bigquery/
Cloud Datalab	https://cloud.google.com/datalab/
TensorFlow	https://www.tensorflow.org/

Google Cloud

Notes:

All the products we talked about are well-documented. We didn't cover Cloud Machine Learning in the course because it is in alpha.

Resources (2 of 2)

Cloud Machine Learning	https://cloud.google.com/ml/
Vision API	https://cloud.google.com/vision/
Translation API	https://cloud.google.com/translate/
Speech API	https://cloud.google.com/speech/
Video Intelligence API	https://cloud.google.com/video-intelligence

Notes: