



# Google I/O



Dennis Ruffer

May 8-10, 2018

# Google Keynote

Learn about the latest product and platform innovations at Google in a Keynote led by Sundar Pichai.

# Developer Keynote

Learn about the latest updates to our developer products and platforms at Google in a Keynote led by Jason Titus.

## Android vitals: debug app performance and reap rewards

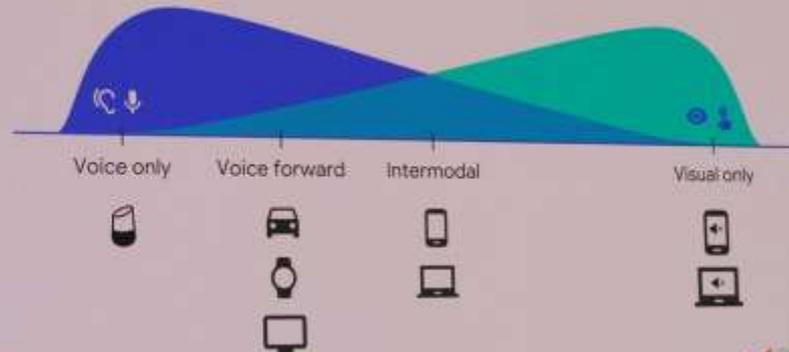
Learn about Android vitals and improved tools in the Play Console and Android Studio to help improve your app quality. This session will explain how app performance impacts key business metrics, such as ratings and engagement, and covers best practices to help fix performance issues in your app.

<https://developer.android.com/topic/performance/vitals/>

# Best practices for testing your Actions

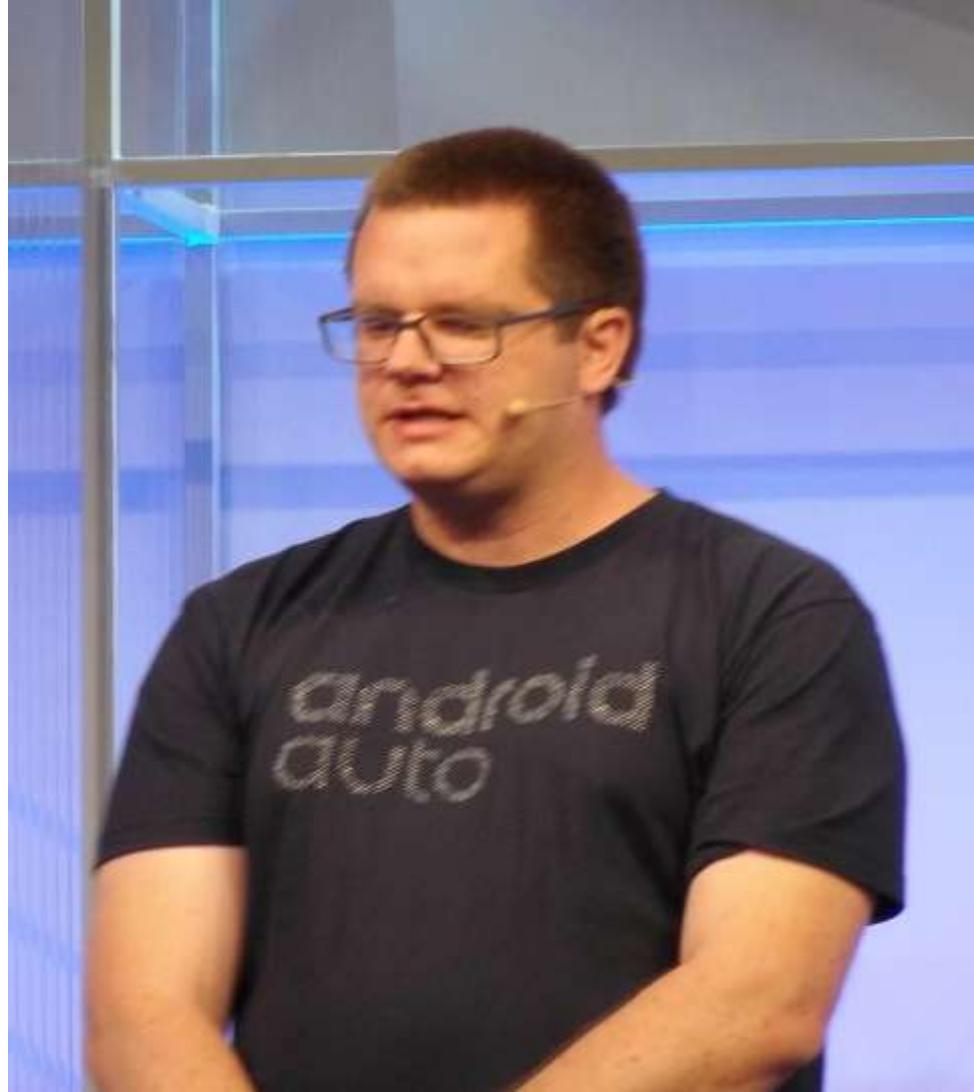
Robust testing is essential for developing high-quality software and creating user satisfaction. This session will dive deep into developing end-to-end tests for your Actions, and cover the tools that are available to make the process easier. It will also share best practices on a variety of topics, like how to handle unexpected user queries.

## Multimodal Spectrum

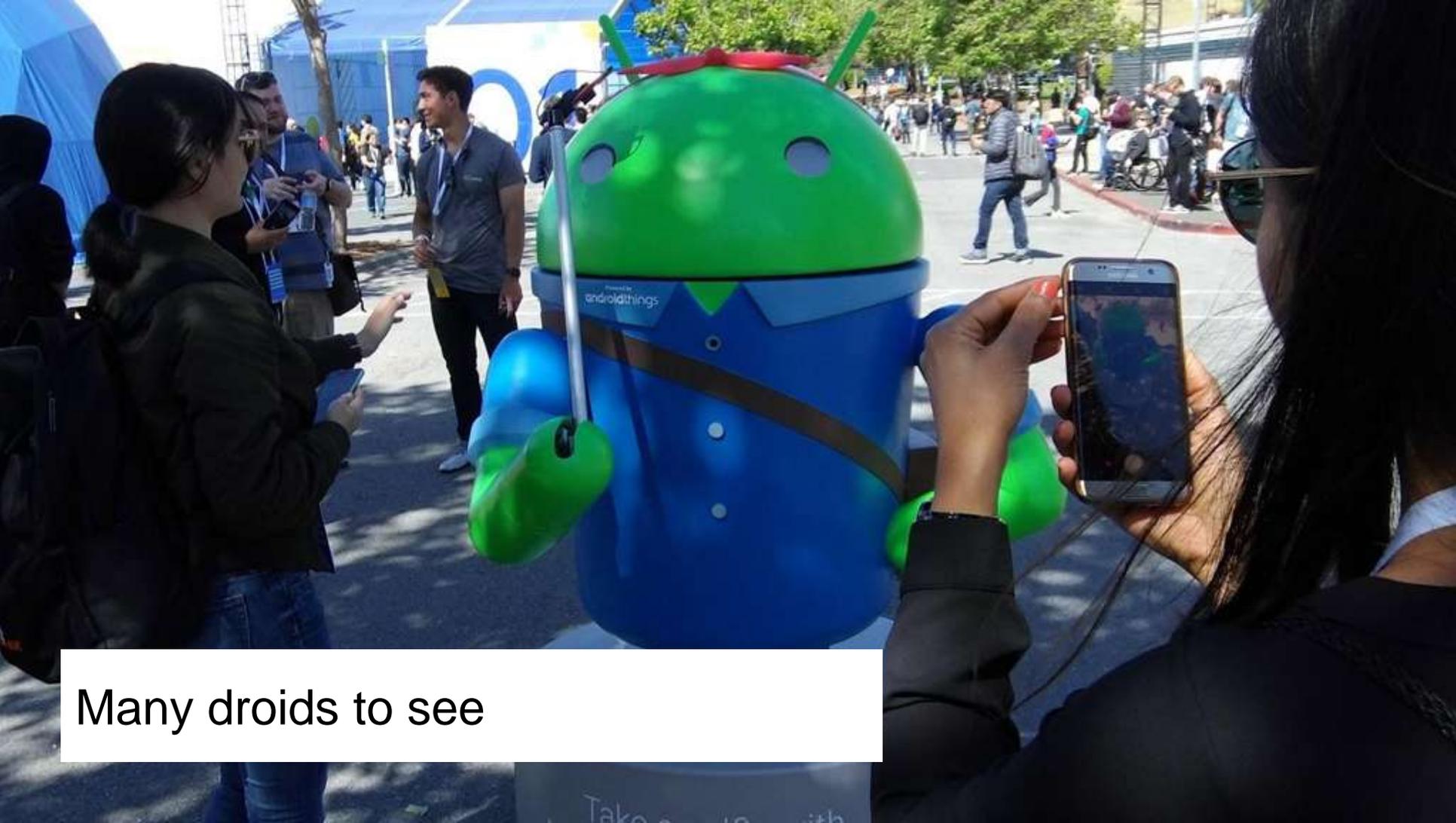


## What's new in automotive

This session will cover recent developments in automotive, discuss why automotive is a crucial area for developers, and show an exciting preview of what's ahead. Learn what the industry looks like, what automotive partners are doing in the space, and how this impacts Android developers.







Many droids to see

## Project Nitrogen

A single entry point for tests



#io.o

## Android Test Resources

### Codelabs

[code-labs.io/codelabs/android-testing](https://code-labs.io/codelabs/android-testing)

[code-labs.io/codelabs/bazel-android-intro](https://code-labs.io/codelabs/bazel-android-intro) *new!*

### Testing samples

[github.com/googlesamples/android-testing](https://github.com/googlesamples/android-testing)

### Android Testing Documentation

[developer.android.com/testing](https://developer.android.com/testing)



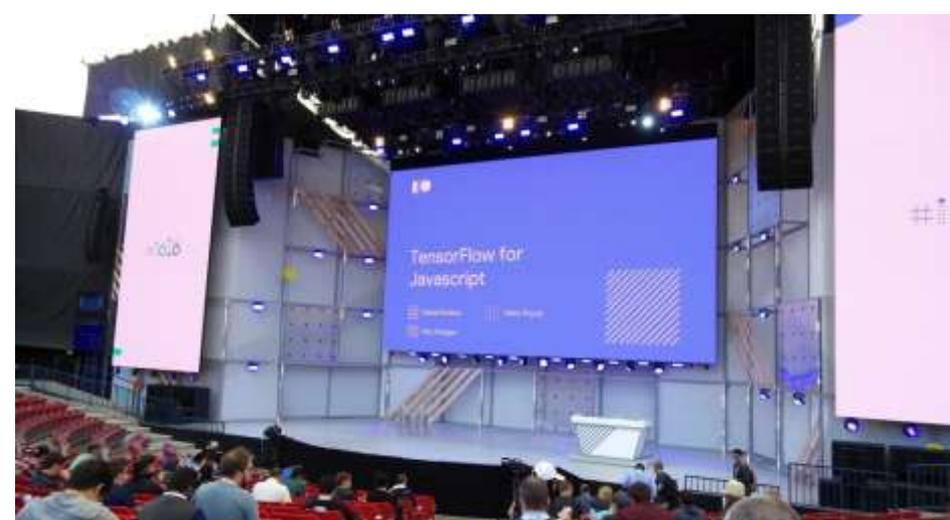
#io.o

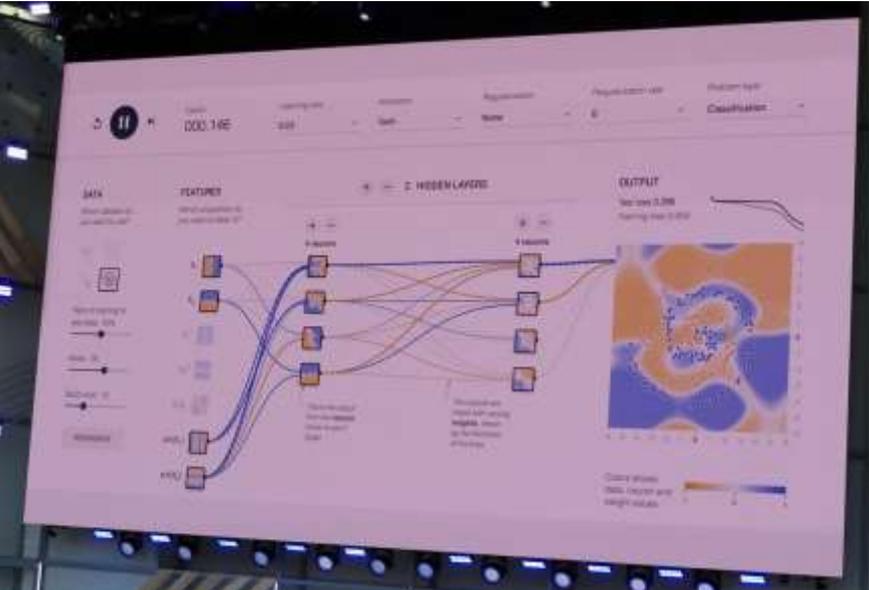
# Frictionless Android testing: write once, run everywhere

There are many testing tools available for Android, and selecting the right tool can be confusing. This session will showcase the Android Testing Support Library (ATSL) — a new set of testing APIs that allow developers to write tests of all sizes across different execution environments. These new APIs will make testing easy, reducing the cognitive load for developers and keeping them in the zone while rapidly iterating.

# TensorFlow for JavaScript

TensorFlow has been extended to simplify model training and deployment using the JavaScript language. This session will offer a detailed description of how to use JavaScript to train and deploy your models.





# deeplearn.js

- Released August 2017
- GPU-accelerated via WebGL
- Allows inference and training entirely in the browser

## In-browser ML

- Full browser / PWA capable
- Intuitive
- Secure



```
import * as tf from '@tensorflow/tfjs';

const a = tf.tensor(0.1).variable();
const b = tf.tensor(0.1).variable();
const c = tf.tensor(0.1).variable();

// f(x) = ax^2 + bx + c
const f = x => tf.add(tf.add(tf.mul(a, tf.square(x)), tf.mul(b, x)), c);

// ...
```

```
import * as tf from '@tensorflow/tfjs';

const a = tf.tensor(0.1).variable();
const b = tf.tensor(0.1).variable();
const c = tf.tensor(0.1).variable();

// f(x) = ax^2 + bx + c
const f = x => tf.add(tf.add(tf.mul(a, tf.square(x)), tf.mul(b, x)), c);
const F = a => a.mul(x.square()).add(b.mul(x)).add(c);

// ...
```

```
// ...
```

```
// Mean-squared error
const loss = (preds, label) => preds.sub(label).square().mean();
```

```
const optimizer = tf.train.sgd(learningRate);
```

```
for (let i = 0; i < EPOCHS; i++) {
  optimizer.minimize(() => loss(f(data.xs), data.ys));
}
```

```
import os, tf from '@tensorflow/tfjs'

const model = tf.sequential({
  layers: [
    tf.layers.dense({units: 100, kernelSize: [kernel, vocabSize]}),
    tf.layers.dense({units: digits + 1})
  ]
});
```

```
model.compile({
  loss: 'categoricalCrossentropy',
  optimizer: 'adam',
  metrics: ['accuracy']
});
```

```
const history = await model.fit({x: data.xs, y: data.ys, batchSize});
```

```
import os, tf from '@tensorflow/tfjs'

const model = tf.sequential({
  layers: [
    tf.layers.dense({units: hiddenSize, kernelSize: [kernel, vocabSize]}),
    tf.layers.dense({units: digits + 1})
  ]
});

tf.layers.dense({units: hiddenSize, returnSequences: true}),
tf.layers.timeDistributed({layer: dense({units: vocabSize})}),
tf.layers.activation({activation: 'softmax'}
});
```



# EMOJI SCAVENGER HUNT

Locate the emoji we show you in the real world with your phone's camera. A neural network will try to guess what it's seeing.  
Make sure your sound is on.

 **LET'S PLAY**

Emoji Scavenger Hunt is best experienced on phones.

Python

Save

```

builder = tf.nn.builders.Builder(report=True)
save_model(builder.get_session(), report=True)

builder.add_ops(graph_and_variables_ops)
builder.save()

```

```

model = keras.models.Sequential([
    keras.layers.Dense(10, name='dense_1'),
    keras.layers.Dense(10, name='dense_2')])

model.save_weights('model.h5')

```

TensorFlow SavedModel      Keras Model

Python

Save

Convert

TensorFlow.js

Load

```

const model =
  await tf.loadFrozenModel(modelUrl, weightUrl);
const feed = {'input_placeholder': input};
const prediction = model.execute(feed);

```

TensorFlow SavedModel

```

const model = await tf.loadModel(
  'http://foo_bar/model.json');
const prediction = model.predict(input);

```

Keras Model

Python

Save

Convert

```

tf.nn.builders.Builder
tf.nn.builders.Builder
tf.nn.builders.Builder
tf.nn.builders.Builder
tf.nn.builders.Builder

```

```

pip install tensorflowjs
tensorflowjs_converter --input_format keras --output_dir=./ --input_model=./ --output_model=./

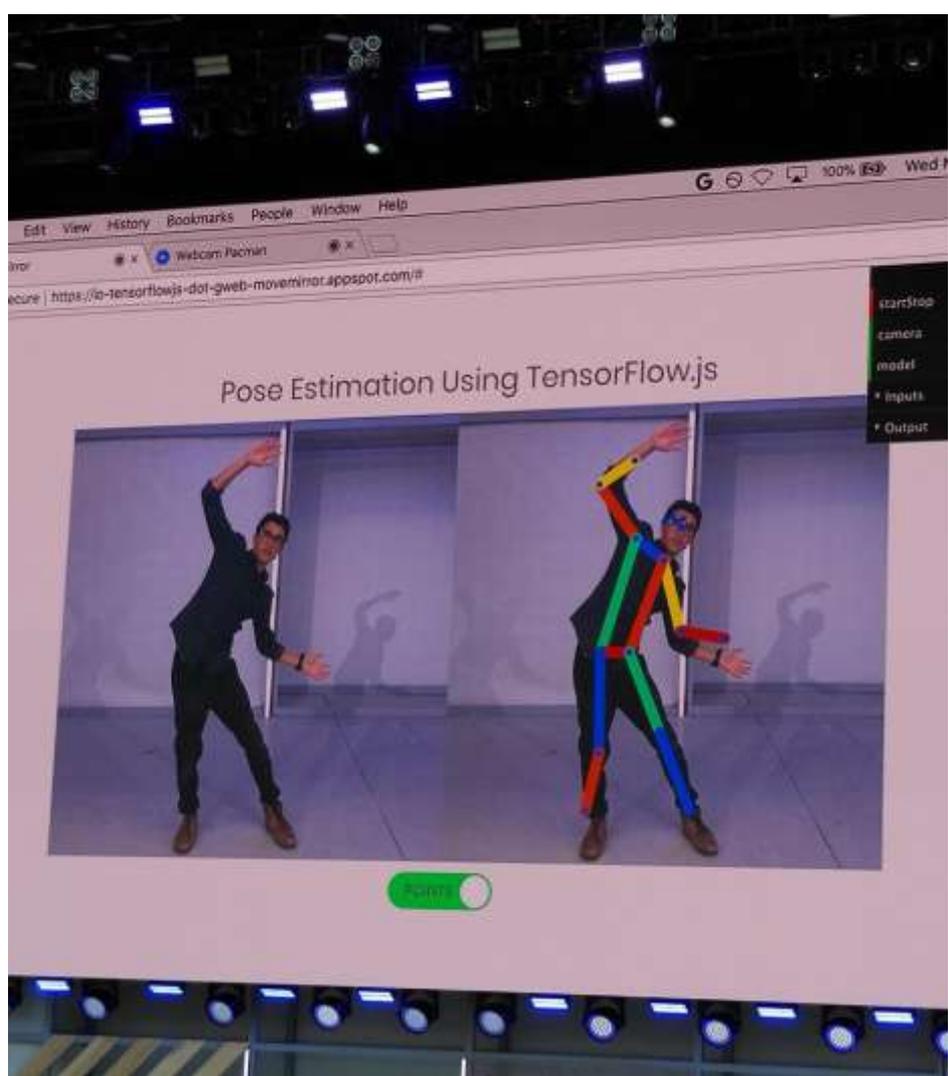
```

TensorFlow SavedModel      Keras Model



# Model Conversion

- Graph optimization
- Weights are optimized for browser caching
- Support for 90+ TensorFlow ops
- Support for 32+ most commonly used TF / Keras Layers
  - Inference, Training, and Evaluation



The screenshot shows a web browser window with the address bar displaying `https://a-tensorflowjs-dot-gweb-moveminor.appspot.com/#`. The page title is "Pose Estimation Using TensorFlow.js". The main content area features two side-by-side images of a man in a dark suit performing a pose. The image on the right has a colorful skeleton overlay on the man's body, representing the pose estimation results. A green "Start" button is located below the images. On the right side of the browser window, there is a vertical control panel with the following options: "start/stop", "camera", "model", "Inputs", and "Output". The "Inputs" and "Output" options are currently selected, indicated by small white circles next to them.



## Fastball vs Curveball



## Node.js Pitch Server Overview

ML/DL on Device

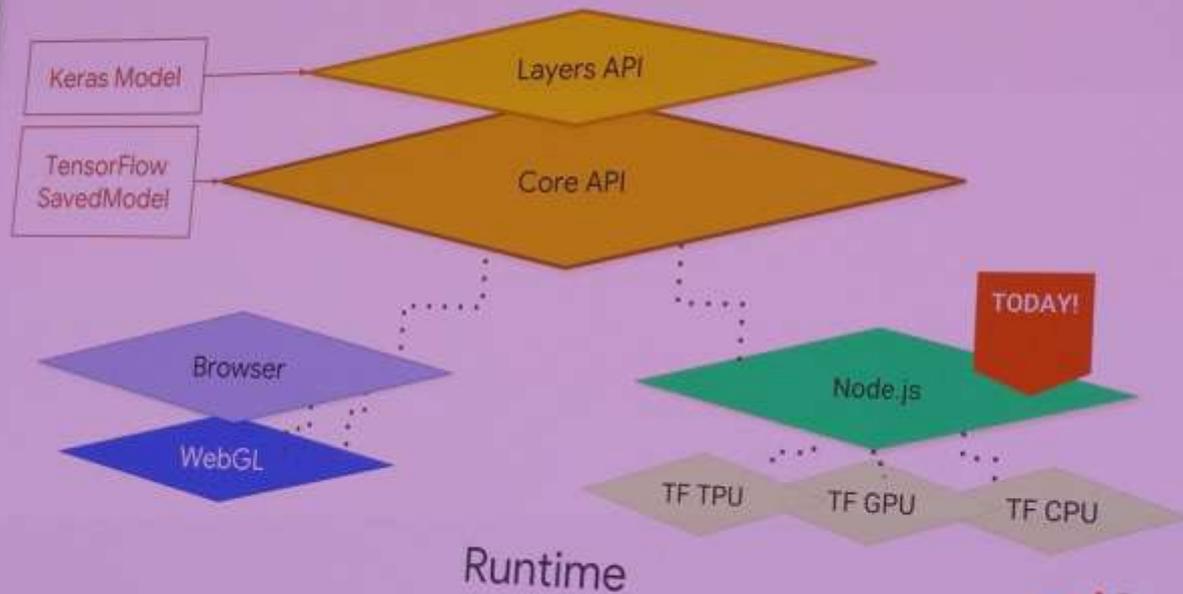
Node.js Server

On-device experience



Data used with permission from Major League Baseball Advanced Media

## API





# Thank you



Daniel Smilkov

Google

 @dsmilkov



Nikhil Thorat

Google

 @nsthorat



Nick Kreeger

Google

 @nkreeger

Helpful resources

[js.tensorflow.org](https://js.tensorflow.org)

[github.com/tensorflow/tfjs](https://github.com/tensorflow/tfjs)

Community mailing list: [goo.gl/drqpT5](https://goo.gl/drqpT5)

Feedback: [google.com/io/schedule](https://google.com/io/schedule)

# Build the future of the web with WebAssembly and more

This talk will cover how to use the most advanced modern web technologies to build experiences that were never possible on the web before.

WebAssembly is enabling the browsers to expose lower-level primitives that can be built on by developers to create performance demanding functionality, like real time media processing, without having to wait for it to be standardized and implemented. See some of the amazing experiences that have already been built and learn how to apply them to today.



Building the Future  
Web with  
WebAssembly

Thomas Nattestad

# WA

WebAssembly is

- A new low-level binary format for the web
- Compiled from other languages
- Offer maximized performance



# emscripten

emscripten.org

## World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents. Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists, Privacy, W3 news, Frequently Asked Questions.

### What's out there?

Pointers to the world's online information, subjects, W3 servers, etc.

### Help

on the browser you are using

### Software Products

A list of W3 project components and their current state. (e.g. Line Mode, X11 Viola, NeXTStep, Servers, Tools, Mail robot, Library)

### Technical

Details of protocols, formats, program internals etc

### Bibliography

Paper documentation on W3 and references.

### People

A list of some people involved in the project.

### History

A summary of the history of the project.

### How can I help?

If you would like to support the web..

### Getting code

Getting the code by anonymous FTP, etc.

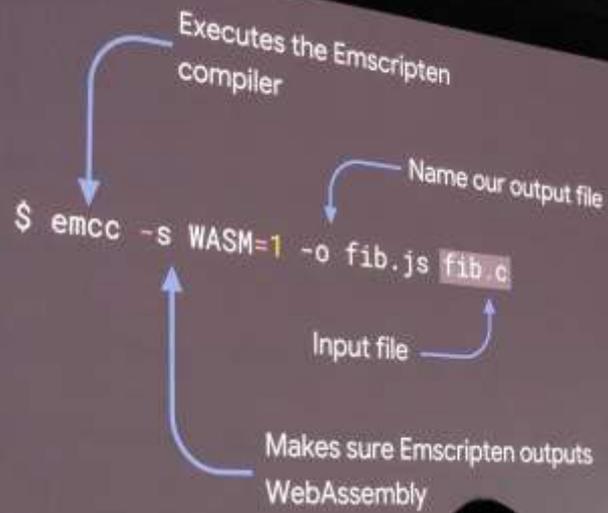
fib.c:

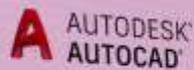
```
#include <emscripten.h>

EMSCRIPTEN_KEEPALIVE
int fib(int n) {
  int i, t, a = 0, b = 1;
  for (i = 0; i < n; i++) {
    t = a + b;
    a = b;
    b = t;
  }
  return b;
}
```

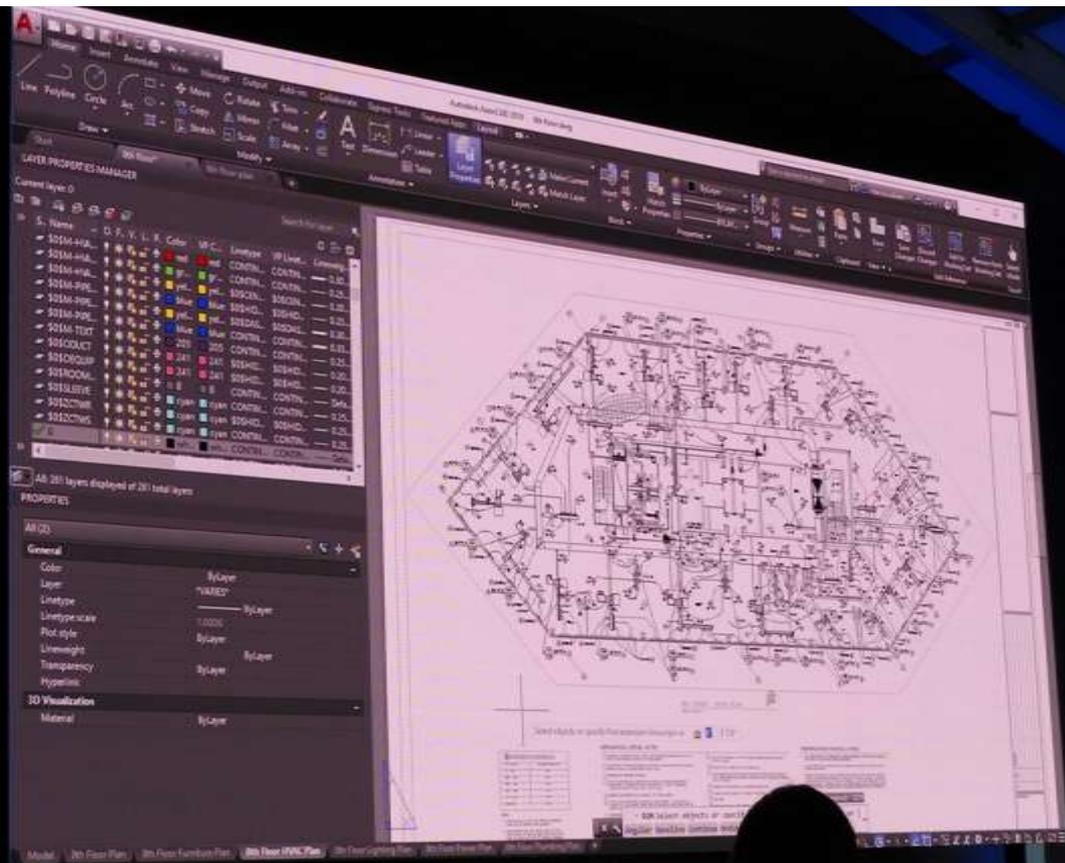
```
<script src="fib.js"></script>
<script>
  Module.onRuntimeInitialized = _ => {
    const fib = Module.cwrap('fib', 'number', ['number']);
    console.log(fib(12));
    // → 233
  };
</script>
```

[bit.ly/emscripten-c-to-wasm](https://bit.ly/emscripten-c-to-wasm)

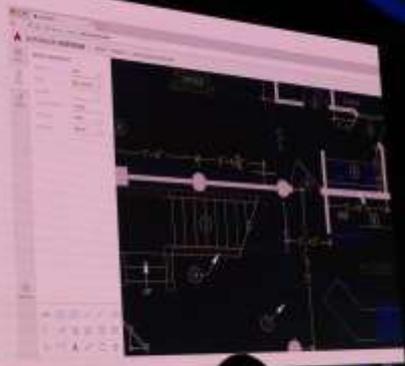




Marcus O'Brien  
Senior Product Line Manager



March 22, 2018  
AutoCAD web app launched





# Thank you



Thomas Nattestad

Google



@ThomasTheDane



@ThomasTheDane



@v8js



# Autonomous and customized pre-launch testing in the Google Play Console

Introduction



# Autonomous and customized pre-launch testing in the Google Play Console

Learn how to use the intelligent, autonomous app crawler to test your APK before releasing it to production. Run QA, set up test channels, and let the crawler test your app or game for you on a farm of real and virtual devices. Tests can be autonomous or customized: learn about both, and about new features in pre-launch report.

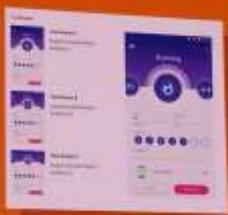
## Why Pre-Launch Report?

Although our team does a LOT of internal testing, there are always issues that manage to go undetected. We rely on Pre-Launch Report to fill in those gaps.

— Zeddy

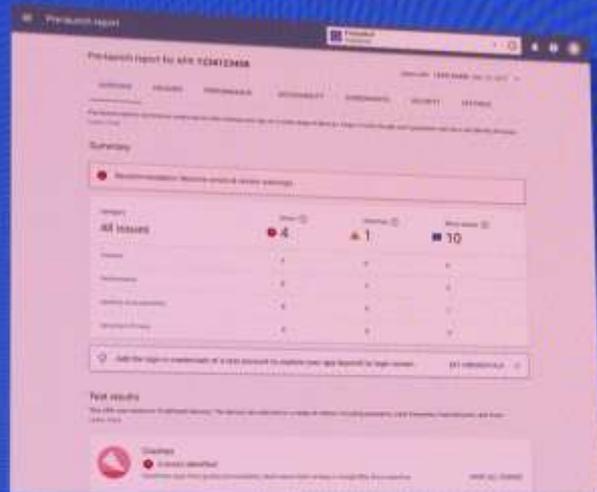
86% of Developers consider Pre-Launch Report to be important when deciding whether to release

— Surveyed Users



Performance and screenshots

#iolo

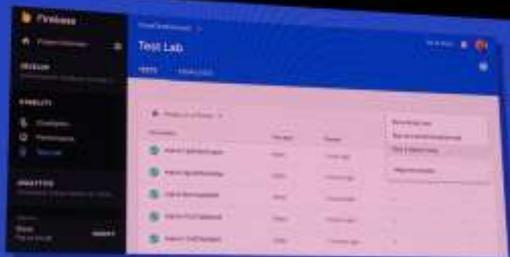


"I don't go to production if there's something wrong in Pre-Launch Report"

- Mr Rocco

Launch recommendation

#iolo



Firebase Test Lab

#iolo

# More Information

Test Lab (iOS Signup)

[https://console.firebase.google.com/project/\\_/testlab](https://console.firebase.google.com/project/_/testlab)

AirTest ITE:

<http://airtest.netease.com/>

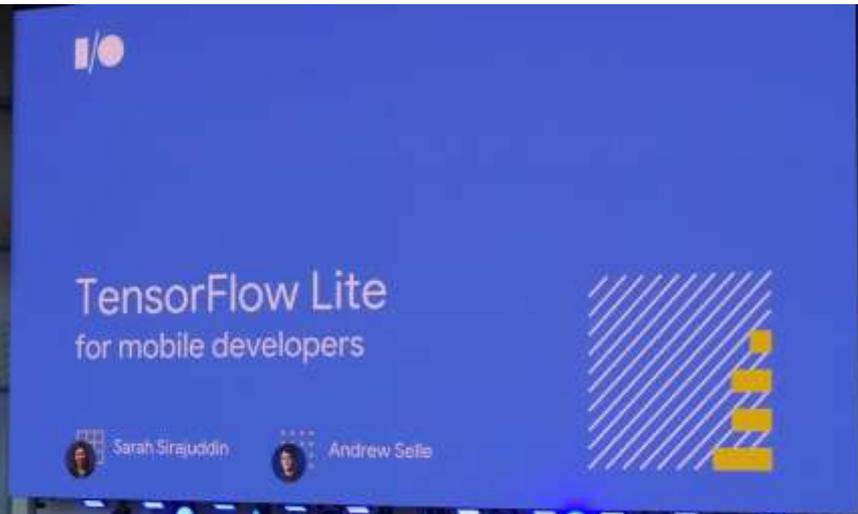
Slack Channel:

<https://firebase-community.slack.com/#test-lab>



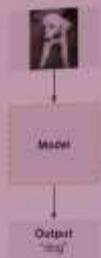
# TensorFlow Lite for mobile developers

TensorFlow Lite enables developers to deploy custom machine learning models to mobile devices. This technical session will describe in detail how to take a trained TensorFlow model, and use it in a mobile app through TensorFlow Lite.



## What is machine learning?

Build mathematical functions using data  
Functions are known as models  
Models perform prediction (a.k.a. inference)



## On-device ML is hard

Tight memory constraints  
Low energy usage to preserve batteries  
Little compute power



## Why on-device ML?

- 1 Lower latency, no server calls
- 2 Works offline
- 3 Data stays on device
- 4 Power efficient
- 5 All sensor data accessible on-device



## TensorFlow Lite Design



#tflite

## TensorFlow Lite Size

Core Interpreter: **~75 KB** (vs 1.1 MB for TensorFlow)

Core Interpreter + all supported ops: **~400 KB**

*How?*

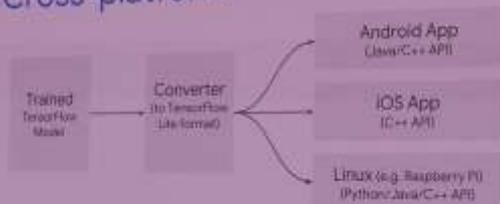
Compact interpreter and FlatBuffer parsing

Tight dependencies

Selective registration

#tflite

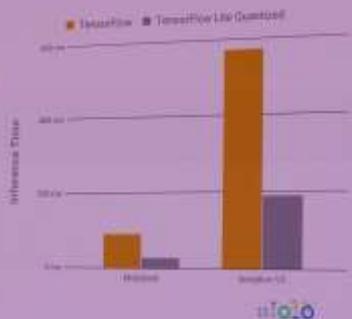
## Cross-platform



#tflite

## Performance TensorFlow Lite

3x Inference Speedup on  
MobileNet and InceptionV3



# ML Kit

ML Kit Custom APIs powered by TensorFlow Lite



## Op and model support

~50 commonly used operations

Extensible design allows using '**custom defined**' ops

Currently **limited to inference ops**

Supported models: MobileNet, InceptionV3, ResNet50, SqueezeNet, DenseNet,  
InceptionV4, SmartReply and others

**Quantized** versions of MobileNet, InceptionV3



## Using TensorFlow Lite



#tflite

## Converting to TensorFlow Lite Format

```
#!/usr/bin/env python
import sys
from tf.contrib.lite import convert_savedmodel
convert_savedmodel.convert(
    saved_model_dir="/tmp/awesome_model",
    output_tflite="/tmp/awesome_model.tflite")
```

Could be trained or downloaded.

## Conversion strategies

Use a frozen graphdef or SavedModel and avoid unsupported operators.  
Write custom operators for any missing functionality.

#tflite



## TensorFlow Lite Roadmap

More operations  
Improved tools, documentation and ease-of-use  
On-device training  
... and lots more

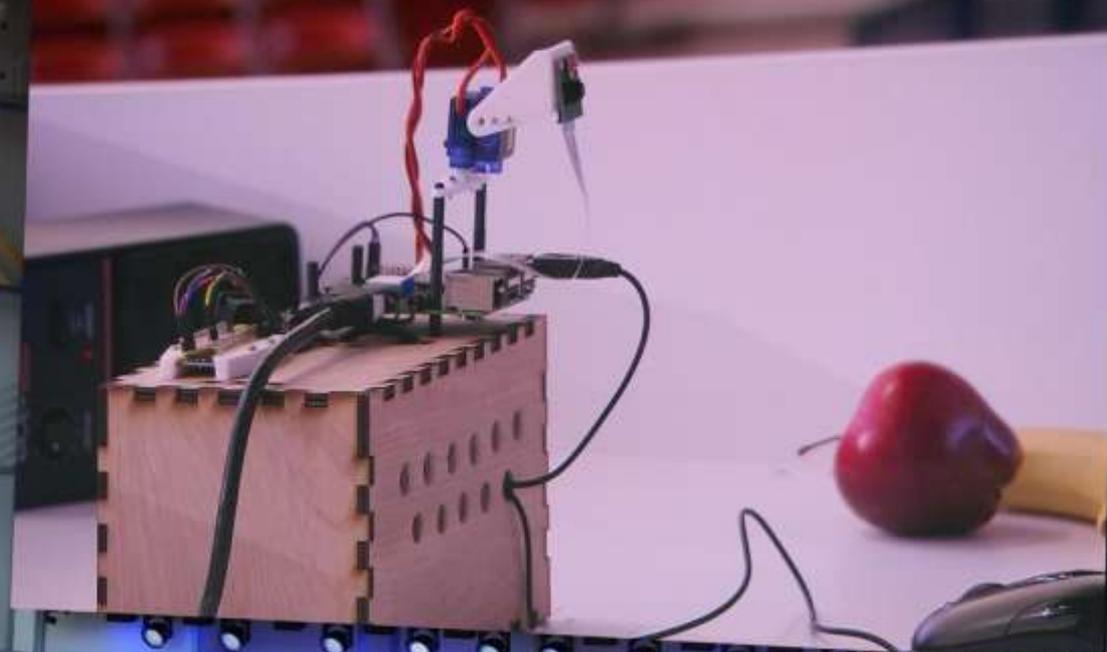
#tolo

## When should I use TensorFlow Lite?



TensorFlow Lite will be the standard solution for running ML models on device.  
TensorFlow Lite currently supports a subset of TensorFlow ops.  
Our recommendation: Use TensorFlow Lite if it has support for your model.  
Let us know about any missing functionality you need.

#tolo





# TensorFlow Lite

## Docs

[tensorflow.org/mobile/tflite/](https://tensorflow.org/mobile/tflite/)

## Code

[github.com/tensorflow/tensorflow](https://github.com/tensorflow/tensorflow)

## Discussion

[tflite@tensorflow.org](mailto:tflite@tensorflow.org)



# Tensorflow and deep reinforcement learning, without a PhD

Martin Gömer  
@martin\_gomer

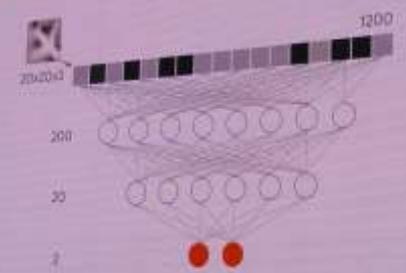
## TensorFlow and deep reinforcement learning, without a PhD

On the forefront of deep learning research is a technique called reinforcement learning, which bridges the gap between academic deep learning problems and ways in which learning occurs in nature in weakly supervised environments. This technique is heavily used when researching areas like learning how to walk, chase prey, navigate complex environments, and even play Go. This session will teach a neural network to play the video game Pong from just the pixels on the screen. No rules, no strategy coaching, and no PhD required.

>TensorFlow and \  
deep reinforcement learning  
without a PhD\_

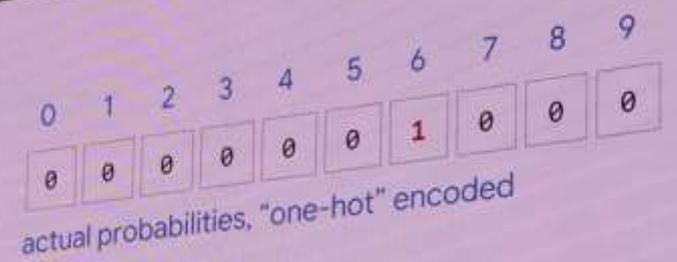


### Dense neural network



#iolo

### Success?



actual probabilities, "one-hot" encoded

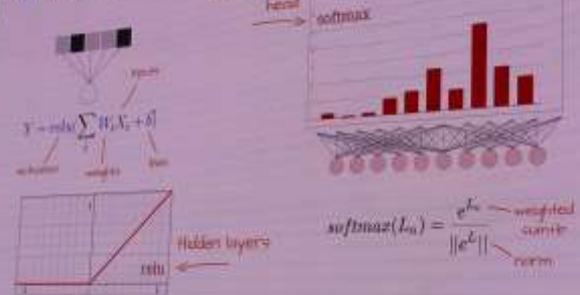
Cross entropy:  $-\sum Y'_i \cdot \log(Y_i)$

computed probabilities



this is a "6"

### Activation functions



#iolo



We want to hear from you

Please provide feedback on this session by signing in on [google.com/io/schedule](https://google.com/io/schedule).

Thank you



Martin Görner

Google



@martin\_gorner



Yu-han Liu

Google

The code:

[github.com/GoogleCloudPlatform/  
tensorflow-without-a-phd](https://github.com/GoogleCloudPlatform/tensorflow-without-a-phd)

# Android Studio Profilers



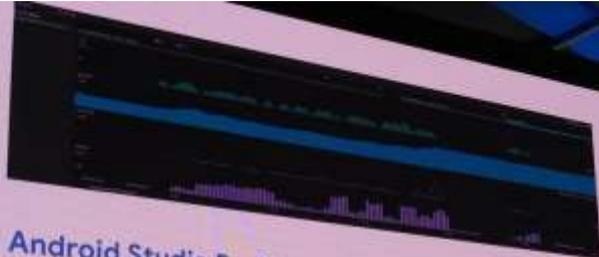
Esteban de la Canal



## Improve app performance with Android Studio Profilers

This talk will demonstrate how to diagnose and troubleshoot performance problems with your app using Android Studio Profilers. It will cover examples of how to use the CPU, memory, network profilers, and highlight what's new.





**Android Studio Profilers** are a suite of tools built into your development environment to help you troubleshoot and optimize the performance of your app.



## Network Profiler

Troubleshoot your app's usage of network resources.

Inspect request details, response details, timing information, and call stacks.

## CPU Profiler

Troubleshoot your app's CPU usage.

Supports method tracing and system tracing.

For method tracing, visualize your Java or C++ code.

## Memory Profiler

Troubleshoot your app's memory usage and find potential leaks.

Take heap dumps and inspect details of memory allocation.

## Energy Profiler

Troubleshoot your app's energy usage.

Visualize a breakdown of your app's estimated energy usage of system components.

Inspect background events that may contribute to battery drain.

# Thank you



Esteban de la Canal

Android Studio

Helpful resources

[developers.google.com](https://developers.google.com)

[google.com/io](https://google.com/io)



@estebandlc

## Breakthroughs in Machine Learning



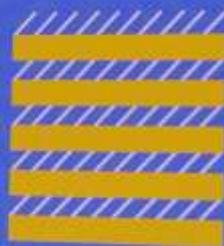
Laurence  
Moroney  
Douglas Eck



Debbie Bard



Vincent  
Vanhoucke

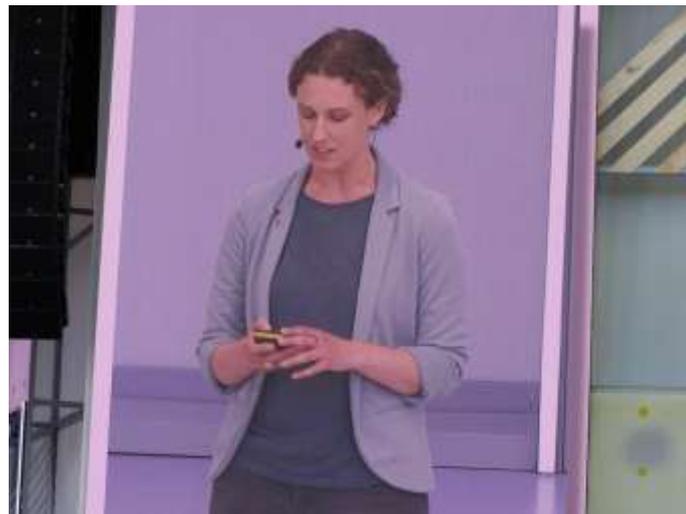
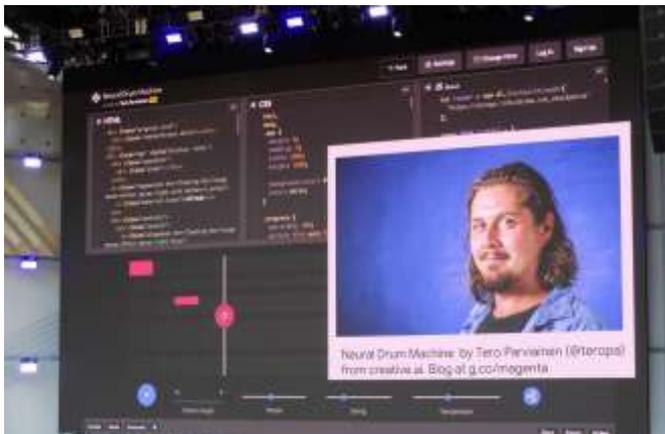


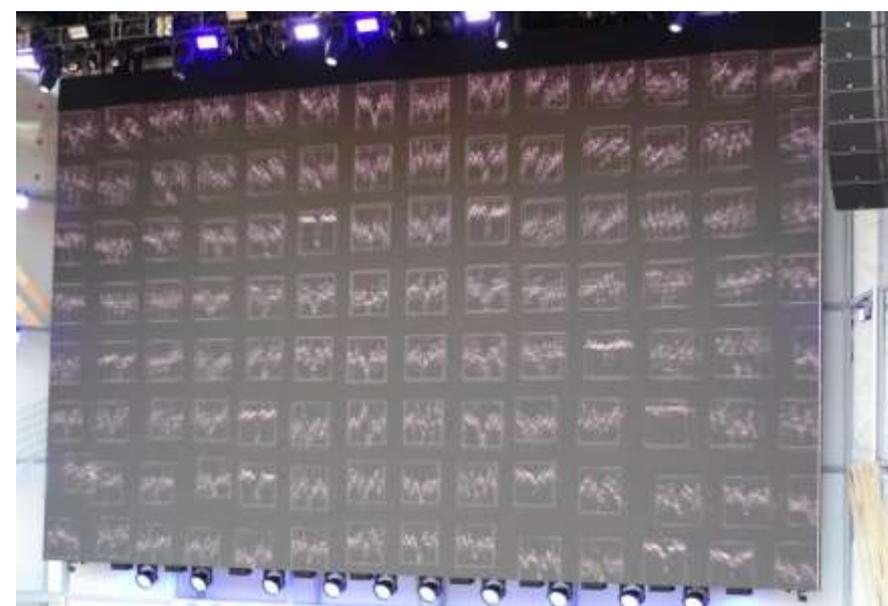
There's a revolution in  
machine learning

#olo

# Advances in machine learning and TensorFlow

Artificial intelligence affects more than just computer science. Join this session to hear a collection of short presentations from top machine learning researchers: the TensorFlow engineers working on robotics, and the Magenta team exploring the border between machine learning and art.





### Sketch RNN

A neural network that learns to draw

David Ha, Google Brain (Dartmouth)

### Visual Imitation

### 3D volumes: using TensorFlow to estimate the parameters that describe our universe

Mehryar Afshar, Mersiyeh et al. in prep.

# Takeaways

- 1) Cosmology has some cool deep learning problems!
- 2) Scientific data is different from natural image data; often has well-understood associated statistics that can help quantify the accuracy of deep learning methods
- 3) MPI allreduce is the optimal strategy for scaling up tensorflow to multiple nodes!





# Distributed TensorFlow training

To efficiently train machine learning models, you will often need to scale your training to multiple GPUs, or even multiple machines. TensorFlow now offers rich functionality to achieve this with just a few lines of code. Join this session to learn how to set this up.

## Distributed TensorFlow Training



Anjali Sridhar

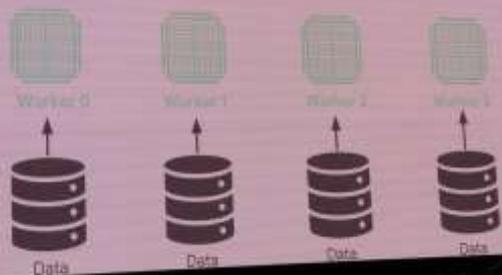


Priya Gupta

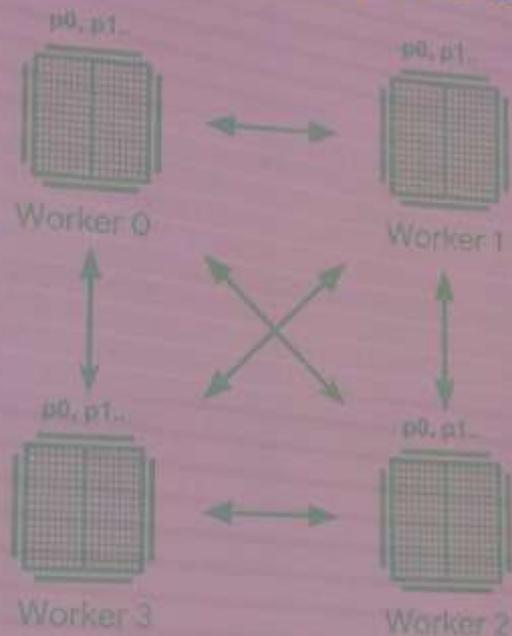


## Data parallelism

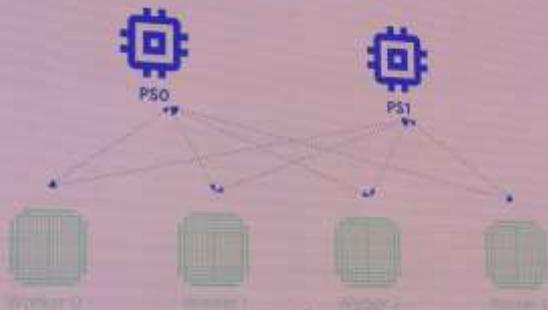
Update model parameters



## Sync Allreduce Architecture



## Async Parameter Server



## Multi machine Distributed Training

Use the Estimator's `train_and_evaluate` API.

It uses the async Parameter Server approach.

[https://www.tensorflow.org/api\\_docs/python/tf/estimator/train\\_and\\_evaluate](https://www.tensorflow.org/api_docs/python/tf/estimator/train_and_evaluate)

## TensorFlow Resources

### Distribution Strategy API

<https://tinyurl.com/tf-distribute>

### ResNet50 Model Garden example with MirroredStrategy API

<https://github.com/tensorflow/models/tree/master/official/resnet>

### Input Pipeline Performance

[https://www.tensorflow.org/performance/datasets\\_performance](https://www.tensorflow.org/performance/datasets_performance)

### Commands to set up a GCE instance

<https://tinyurl.com/tf-demo-setup>



Thank you



Anjali Soodher

Google



Priya Gupta

Google

