Tensorflow.js

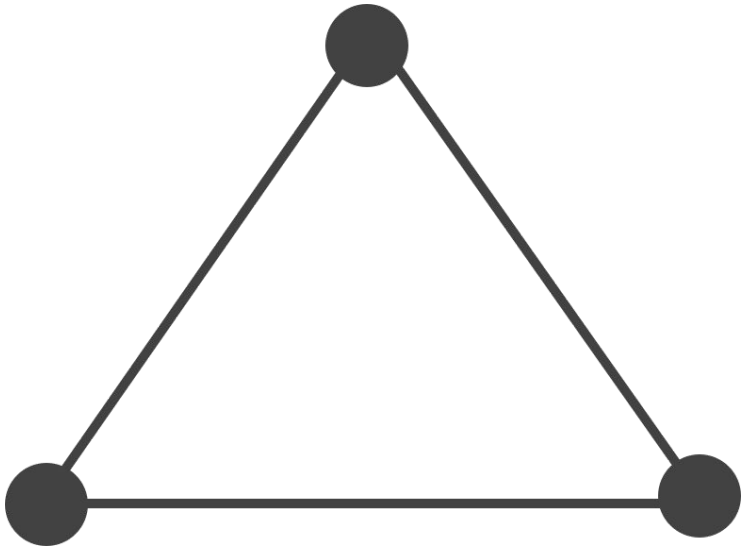
Machine Learning in Javascript

June 6, 2018

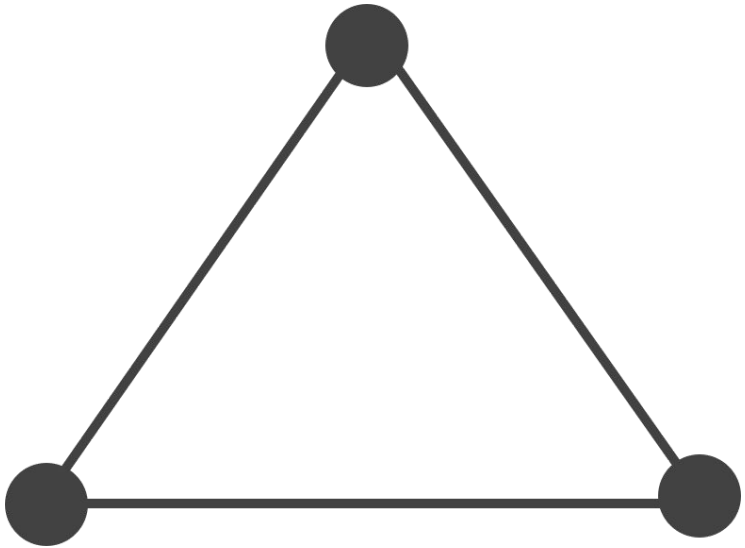
Kevin Scott


Reboot and Select proper Boot device
or Insert Boot Media in selected Boot device and press a key
DELL

Machine Learning

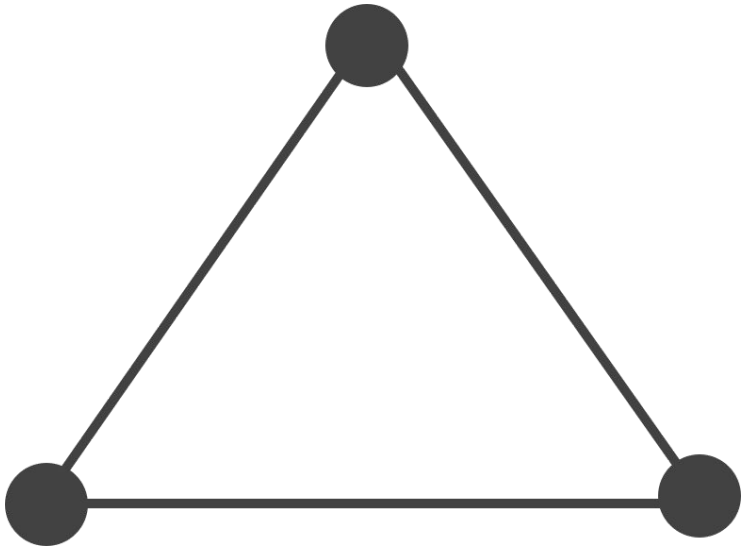


Machine Learning



Mathematics,
Statistics

Machine Learning



Mathematics,
Statistics

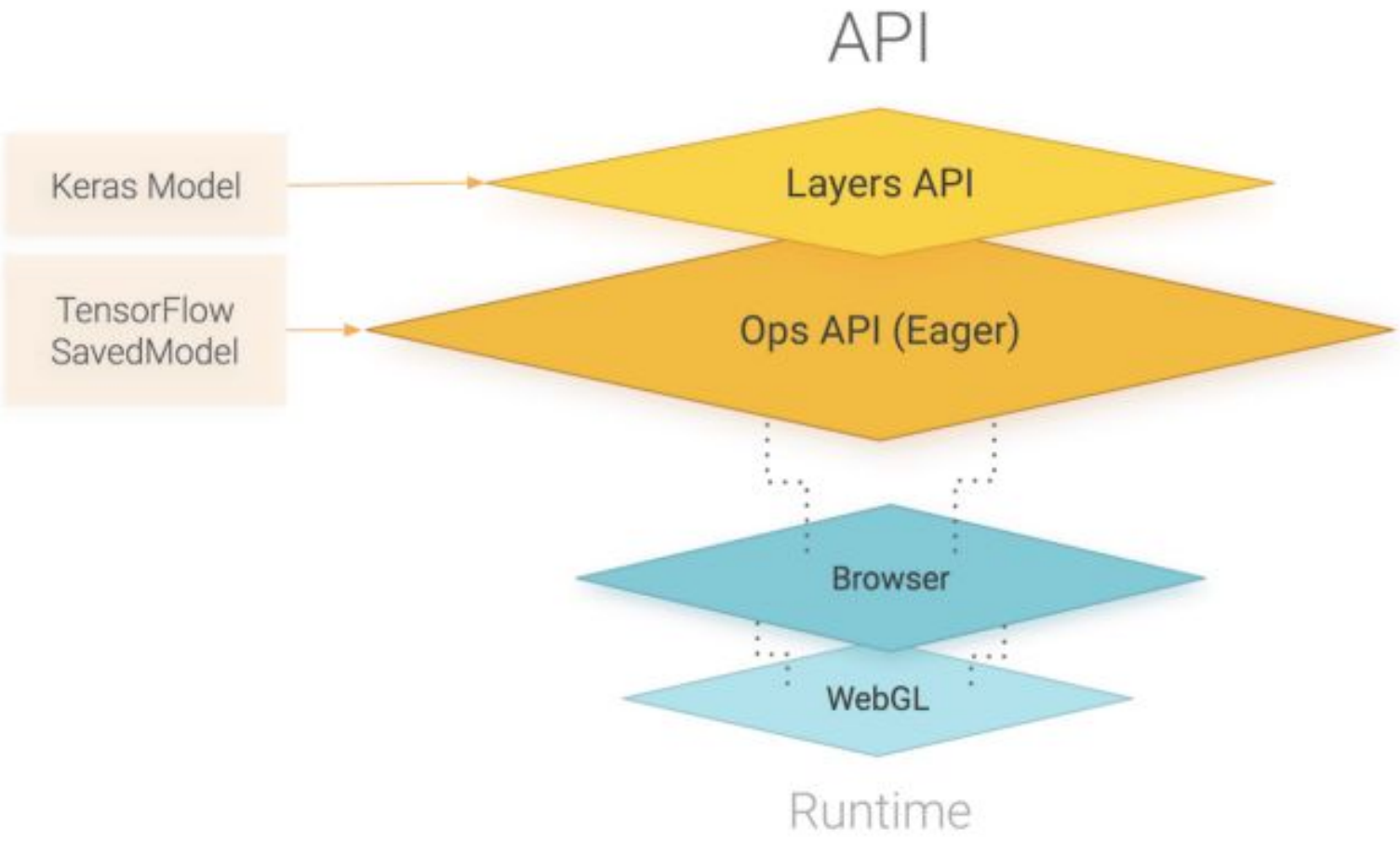
Python, R

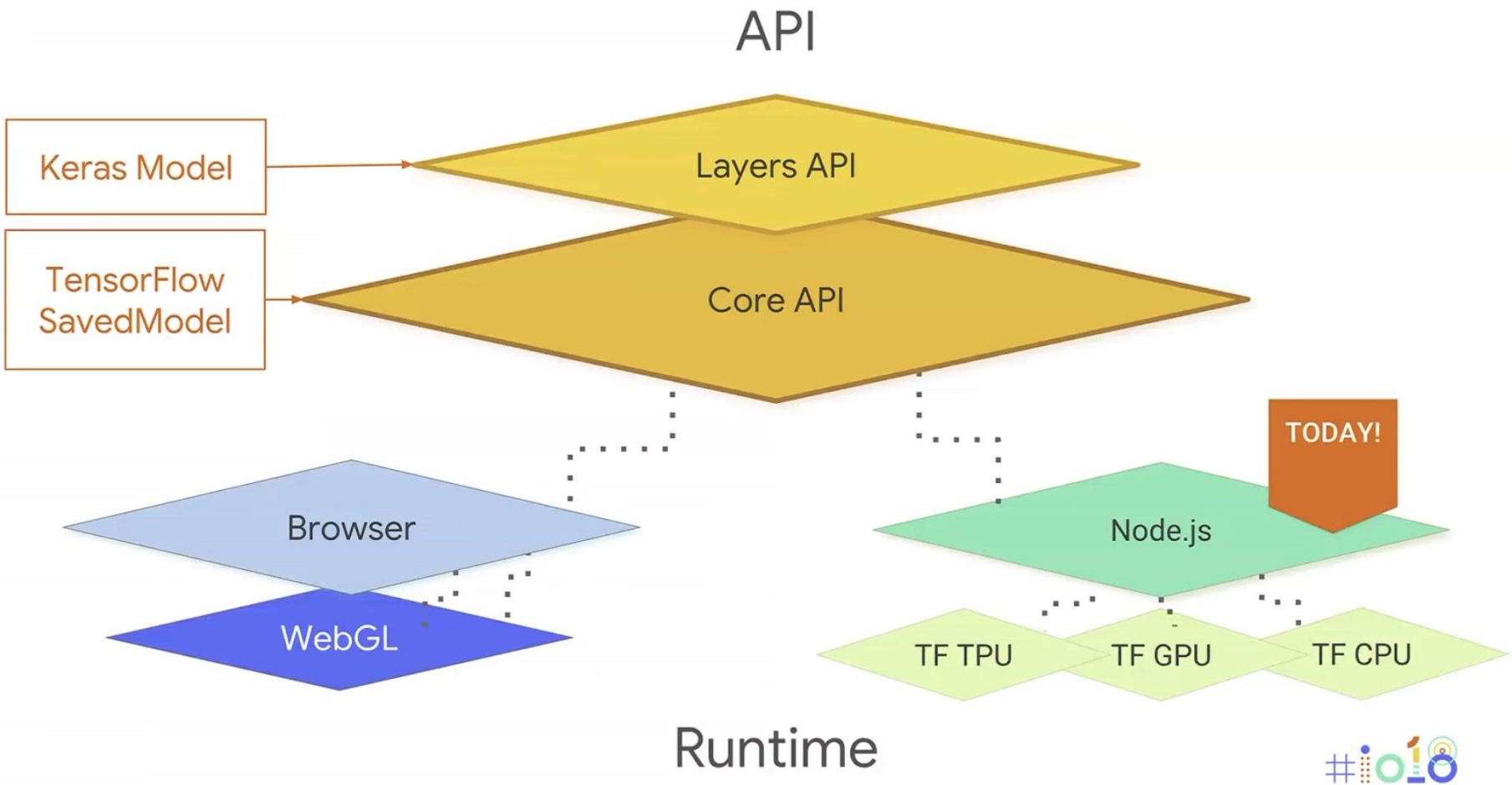

TensorFlow.js


2016

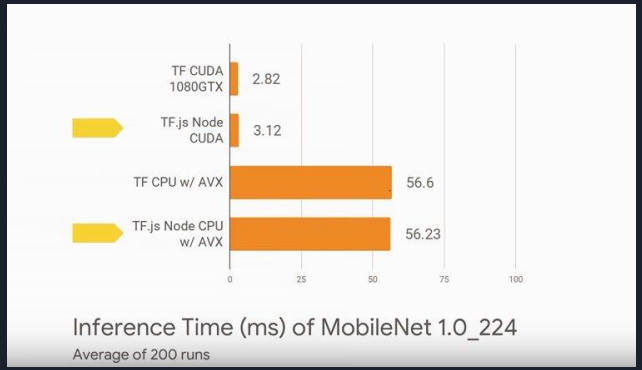
Why run Tensorflow in your browser?

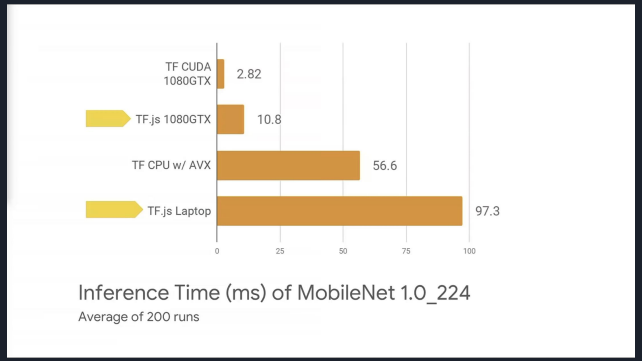
1. Wide distribution
2. Interactive
3. Sensors
4. Privacy


API
Keras Model
TensorFlow SavedModel
Layers API
Ops API (Eager)
Browser
WebGL
Runtime


API
Keras Model
TensorFlow SavedModel
Layers API
Core API
TODAY!
Browser
WebGL
Node.js
TF TPU
TF GPU
TF CPU
Runtime

Node.js Benchmarks





Jeremy Chone

Other recipients: nsth...@google.com, nkats...@gmail.com

Thanks.

Yes, did the test during training and at the end and I am getting a very similar result.

tfjs-node:
training step 0, training accuracy: 0.18 (0.1s)
training step 100, training accuracy: 0.92 (2.1s)
training step 200, training accuracy: 0.94 (2.1s)
training step 300, training accuracy: 1 (2s)
training step 400, training accuracy: 0.92 (2s)
training step 500, training accuracy: 0.98 (2s)
training step 600, training accuracy: 0.94 (2s)
training step 700, training accuracy: 0.98 (2s)
training step 800, training accuracy: 0.92 (2s)
training step 900, training accuracy: 0.98 (2s)

Test set accuracy (batch 1000): 97.00%

python:
step 0, training accuracy 0.14 (0s)
step 100, training accuracy 0.88 (8s)
step 200, training accuracy 0.98 (8s)
step 300, training accuracy 0.94 (8s)
step 400, training accuracy 0.92 (8s)
step 500, training accuracy 0.96 (8s)
step 600, training accuracy 0.96 (8s)
step 700, training accuracy 0.96 (8s)
step 800, training accuracy 1 (8s)
step 900, training accuracy 0.96 (8s)

test accuracy 0.9665

Importing Models

You can import pre-trained models

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

const model = await tf.loadModel('http://foo.bar/tfjs_artifacts/model.json');
// Now the model is ready for inference, evaluation or re-training.
```

Epoch
000,000

Learning rate
0.03

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?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

2 HIDDEN LAYERS

5 neurons

2 neurons

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

*This is the output from one **neuron**. Hover to see it larger.*

OUTPUT

Test loss 0.519

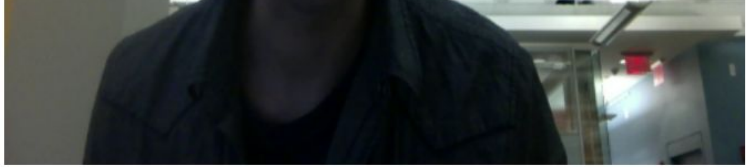
Training loss 0.513

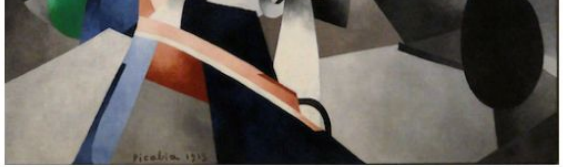
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6

Colors shows data, neuron and weight values.

-1 0

Discretize output





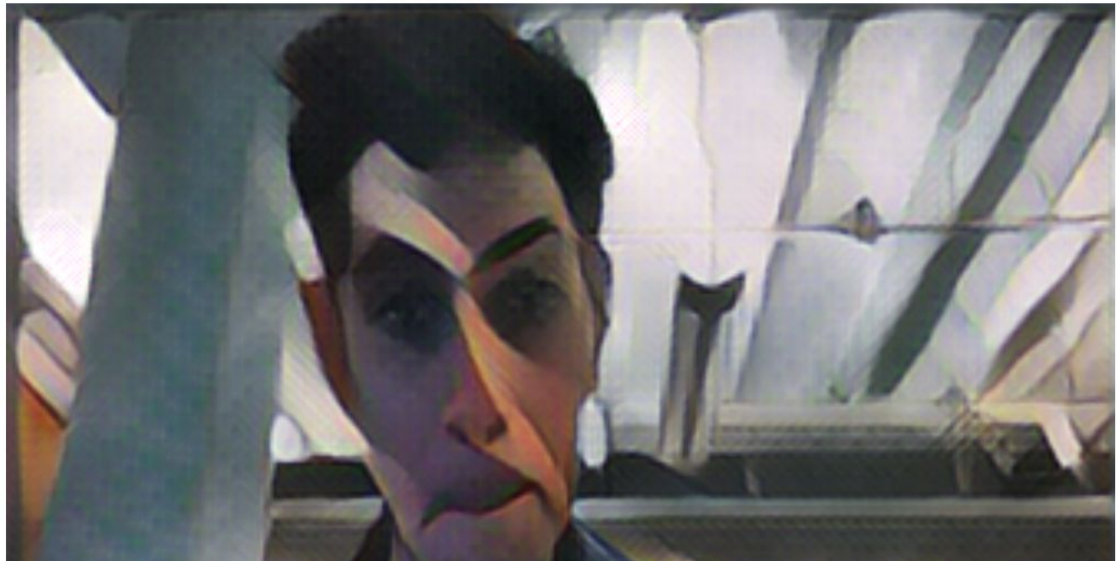
Content

Use webcam

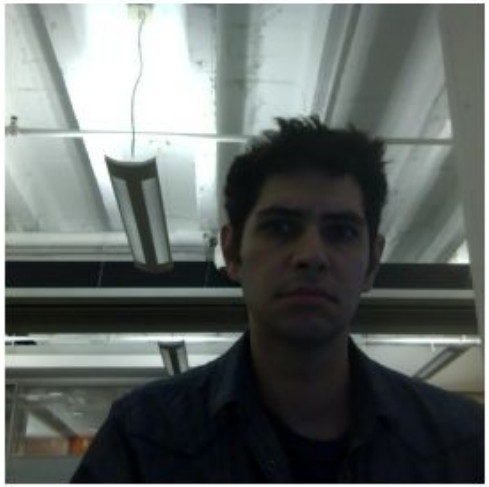
Style

Udnie, Francis Picabia

START STYLE TRANSFER



INPUT



LEARNING

22 EXAMPLES

CONFIDENCE

100%

TRAIN GREEN

21 EXAMPLES

CONFIDENCE

TRAIN PURPLE

32 EXAMPLES

CONFIDENCE

TRAIN ORANGE

OUTPUT

GIF Sound Speech

Thanks!

Google https://js.tensorflow.org

https://github.com/tensorflow/tfjs-examples

https://playground.tensorflow.org

Me https://thekevinscott.com/slides/tfjs

https://twitter.com/thekevinscott

https://thekevinscott.com

Demos Style transfer in browser

Teachable Machine

Pose Estimator