In Browser Machine Learning using TensorFlow.js

Janardan Revuru
JavaScript Evangelist

October 17th, 2019

Audience profile

JavaScript Programmers

Mathematicians who like to do programming

Python Programmers

Interested to start AI/ML Journey

Curious mind

This session is not

... deep dive into JavaScript ES6+ features (*async, await, promises*)

... to cover Tensorflow (Python modules)

... to explain machine learning models

... about training an model

By the end of this session ...

You will

... learn how to use Tensorflow.js in your applications

... learn an application face detection ML code

... learn how to import a pre-trained model

*Focuses mainly on **Tensorflow.js** and a case for **In-Browser ML***

Myself

Web developer since 1996

Speaker and Advisor to tech conferences

Author of “JavaScript - The New parts” in *Open Source For You* magazine

Member of *Microsoft Web Tech Advisory Group*

JavaScript Meetup

Started in Dec 2015 after this conference

Meetups (6000+) - 32 Meetups

YouTube Webinars (750+) - 10 Webinars

WhatsApp Group (256) - Super active

Slack (1000+) - moderate activity

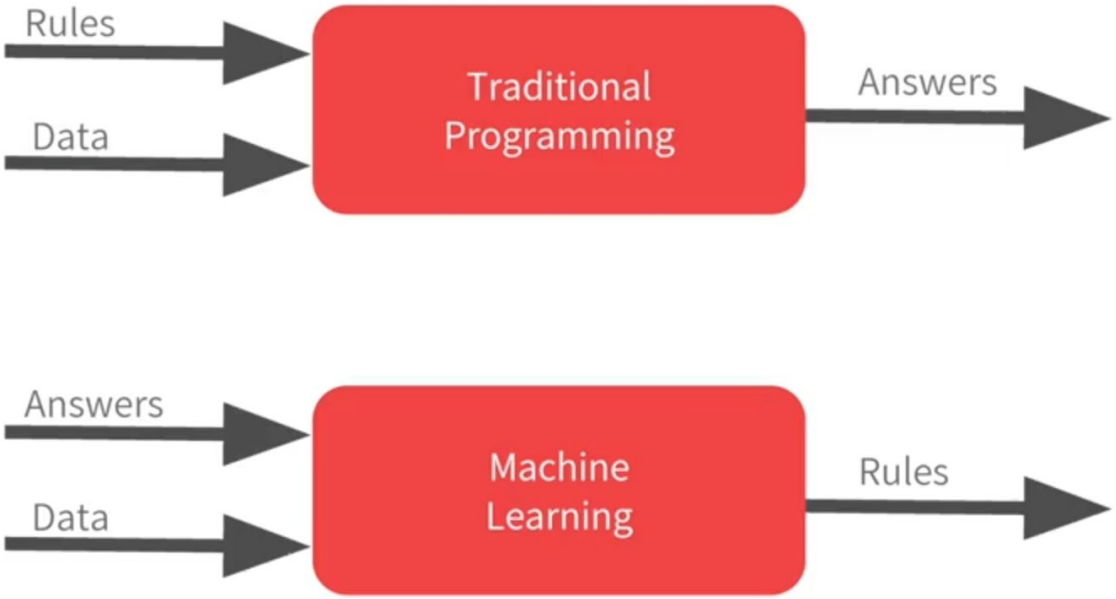
Mission: Equip 50,000 programmers with AI/ML by 2020

https://www.meetup.com/javascriptmeetup/

2 min intro to Machine Learning

How is it different?

Machine Learning



Traditional Programming vs Machine Learning

Activity Recognition

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

```
// Oh crap
```

Equation

x	2	5	10	15	20
y	5	11	21	31	41

$$y = mx + 1$$

3 min intro

TensorFlow

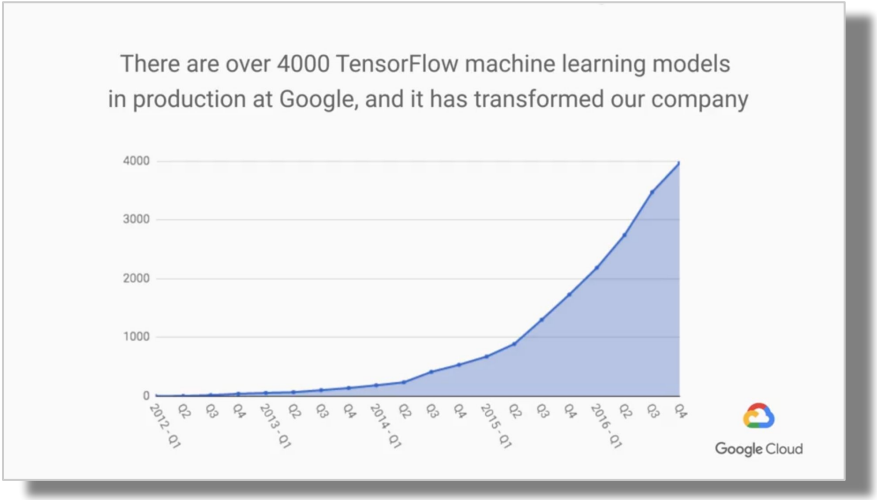
TensorFlow

Core library to help develop and train Machine Learning models

Google

open source software

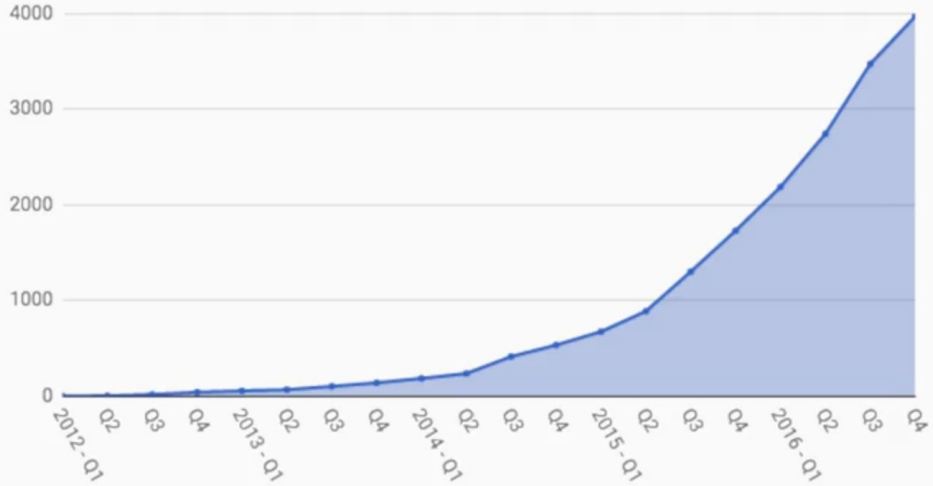
end-to-end platform
from edge devices, browsers to on-prem
production environments and cloud



airbnb Coca-Cola DeepMind GE Healthcare Google intel NERSC

Google uses in apps (201x) -> 4000 TF networks

There are over 4000 TensorFlow machine learning models
in production at Google, and it has transformed our company



Google Cloud

History of TensorFlow

2011: Developed at Google; Built using C++

2015: Open sourced in November under Apache License 2.0

Language bindings with Python, R, Java, Swift (Beta)

2017 Deeplearn.js

2018 Deeplearn.js -> Tensorflow team

Variants of Tensorflow

Core library

Tensorflow for JavaScript (Browser, Node.js)

Tensorflow Lite

Tensorflow Extended

A WebGL accelerated, browser based JavaScript library for training and deploying ML models.

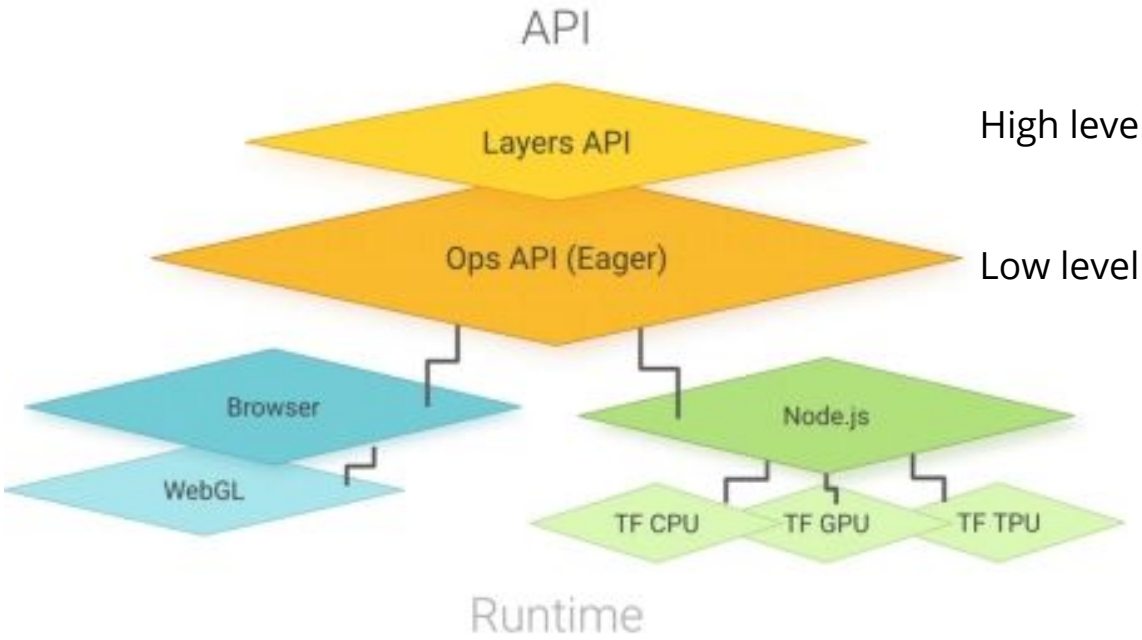
Why TensorFlow.js (JavaScript)

- No installation or setup

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"> </script>
```

- Visualization and Interaction is easy
- Just one language for entire software stack
- Model training in Node.js or Python and using in browser
- Use power of WebGL (*chrome://flags*)
- Increasing number of libraries and high-level APIs
- Use sensors of edge devices

Tensorflow.js Architecture



High level APIs equivalent of keras

Low level tensor arithmetic operations

Tensorflow.js models

Data Type	Models	Purpose
Images	MobileNet PoseNet BodyPix	Classify images with labels Real-time human pose estimation Real-time person and body part segmentation
Audio	Speech commands	Classify 1 second audio snippets
Text	Universal Sentence Encoder	Encode text for further NLP processing such as sentiment classification and textual comparison
General utilities	KNN Classifier	Create classifier using K-Nearest Neighbour Algorithm

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

What is a Tensor?

4 Steps to first ML program

```
// Step 1: Setup variables
const tf = require("@tensorflow/tfjs-node");
const m = tf.variable(tf.tensor(10.0));
// Step 2: Build a model
function f(x) {
   return tf.mul(m, x); // y = m * x
}
// Step 3: Train the model to learn good values for the coefficients
function loss() {
   return f(5).sub(5).square(); // mean square error
}
// Step 4: Training loop to run optimizer to minimize loss
const optimizer = tf.train.sgd(0.01);
for(let i=0; i<20; i++) {
   optimizer.minimize(loss);
   console.log(m.dataSync(), f(5).dataSync());
}
```

Optimizer function decides how much change is required for each parameter in the model, given current model prediction

Loss function minimize the deviation

Emotion Detection using tensorflow.js and face-api.js

(Deep neural networks)

Exporting model from Python and importing in Tensorflow.js

Step 1: Convert an existing Keras model to TF.js Layers format.

A command line tool

```
tensorflowjs_converter
```

Step 2: Load the model into TensorFlow.js

```
// JavaScript

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

const model = await tf.loadLayersModel('https://foo.bar/tfjs_artifacts/model.json');
```

Face detection and Emotion detection

index.html

Heading

Links to images

Click handlers to load images

call main() in index.js

index.js

canvas to load image

placeholder to display emotion text

main()

FaceDetector()

EmotionDetector()

detect_faces()

face_detection.js

faceapi.load..Model()

show_emotions()

Emotion_detection.js

Image shaping

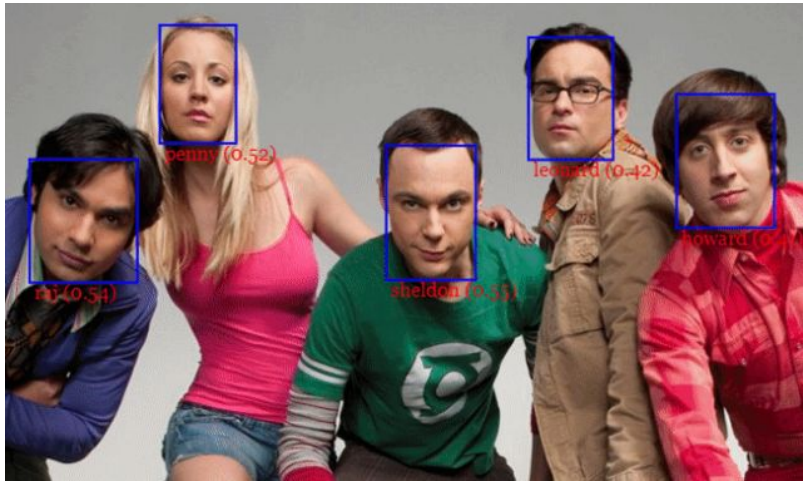
Image sizing

vector arithmetic

load()

classify()

Face-api.js



Implemented on top of tensorflow.js core API

Implements a series of Convolutional Neural Networks (CNN)

Implements SSD Mobilenet V1 (5.4MB), Tiny Face Detector (190KB), experimental MTCNN (2MB)

API Documentation: https://justadudewhohacks.github.io/face-api.js/docs/index.html

Demo: https://hpssjellis.github.io/face-api.js-for-beginners/

Demo

Emotion detection

References

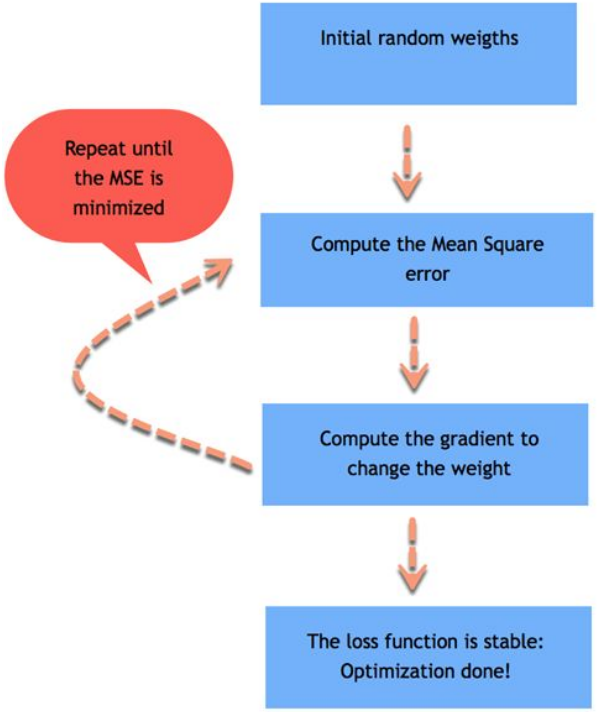
Concepts

Tensor Arithmetic

Optimizer function

Loss function

Stochastic Gradient Descent



https://www.guru99.com/linear-regression-tensorflow.html