



Tensorflow.js ile İstemci Bazlı (Client Side) Derin Öğrenme Uygulamaları



YAVUZ KÖMEÇOĞLU
KODİKS



www.yavuzkomecoglu.com

Tarayıcı Üzerinde Makine Öğrenmesi

- Driver yok / yükleme yok

Tarayıcı Üzerinde Makine Öğrenmesi

- Driver yok / yükleme yok
- İnteraktif

Tarayıcı Üzerinde Makine Öğrenmesi

- Driver yok / yükleme yok
- İnteraktif
- Sensörler



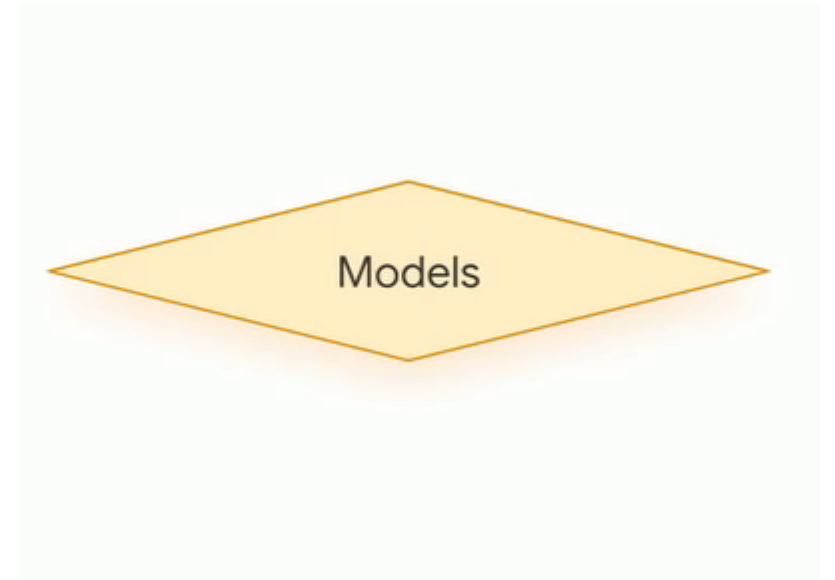
Tarayıcı Üzerinde Makine Öğrenmesi

- Driver yok / yükleme yok
- İnteraktif
- Sensörler
- Mahremiyet (Veri client'da kalır)



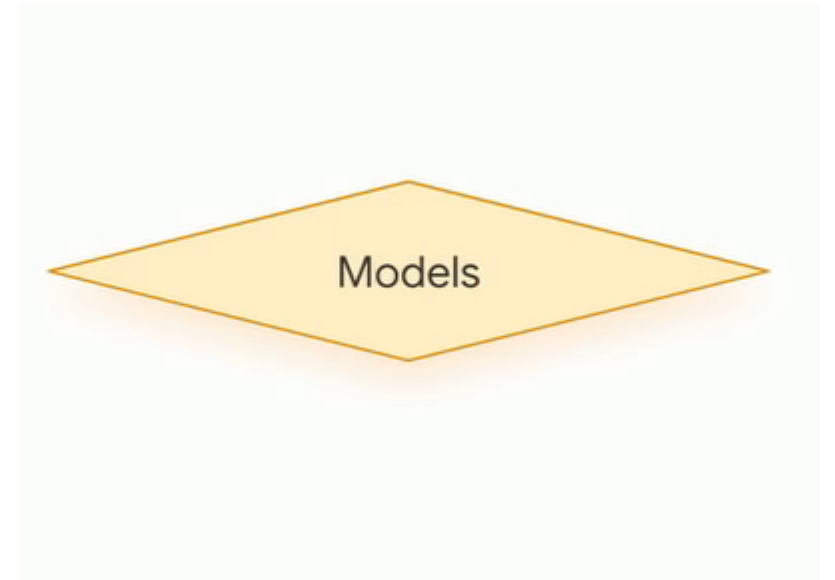
TensorFlow.js kullanarak ne yapılabilir

- Model direk browser üzerinde yazılabilir



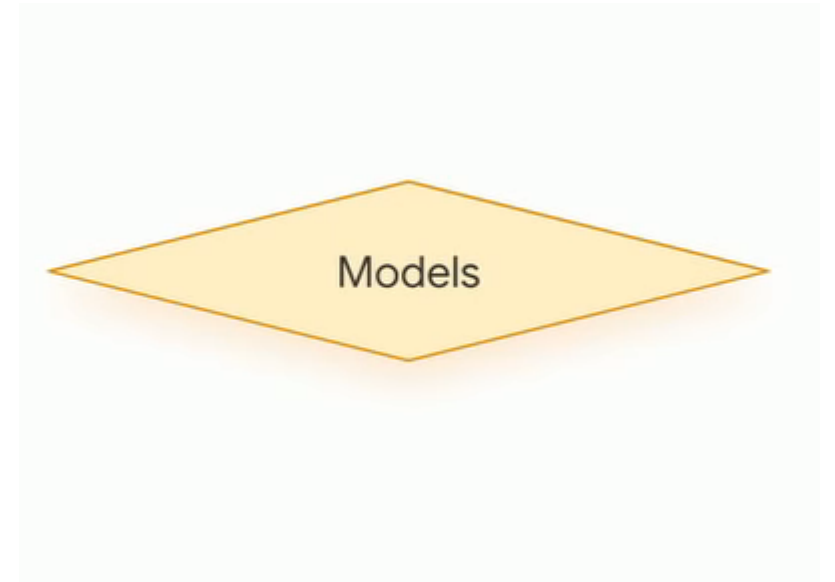
TensorFlow.js kullanarak ne yapılabilir

- Model direk browser üzerinde yazılabilir



TensorFlow.js kullanarak ne yapılabilir

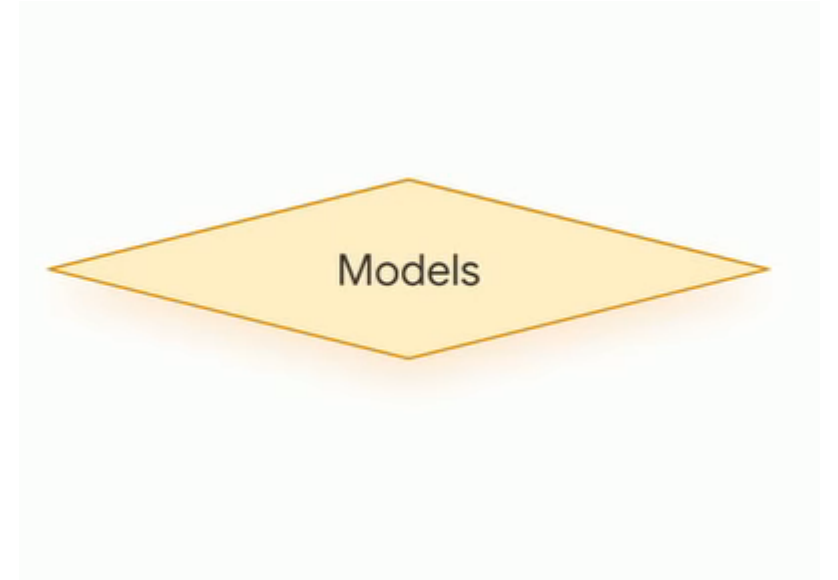
- Model direk browser üzerinde yazılabilir
- Önceden eğitilmiş (pre-trained) modeller kullanılabilir



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

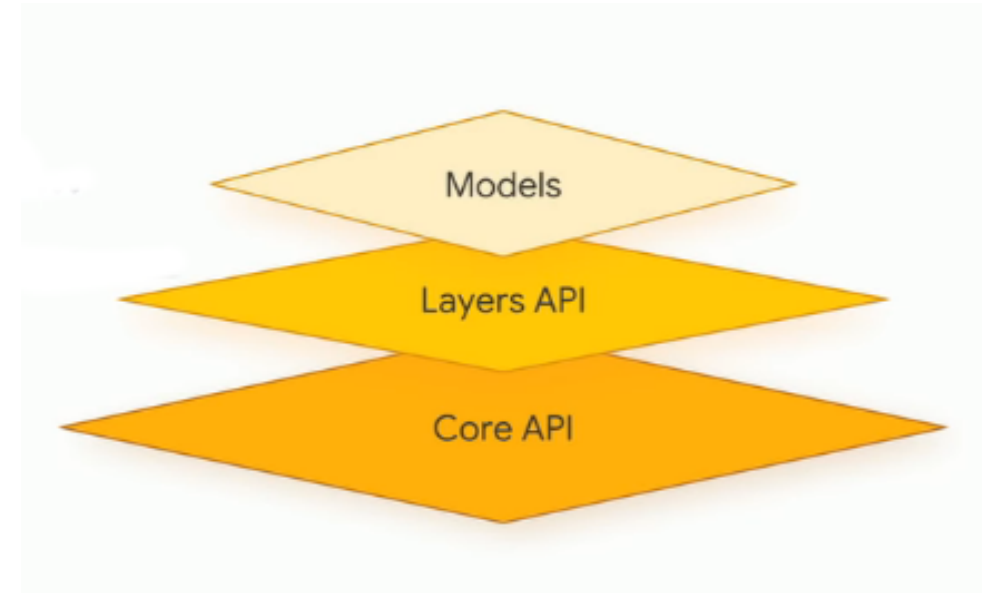
TensorFlow.js kullanarak ne yapılabilir

- Model direk browser üzerinde yazılabilir
- Önceden eğitilmiş (pre-trained) modeller kullanılabilir
- Mevcut Python modelleri dönüştürülüp kullanılabilir



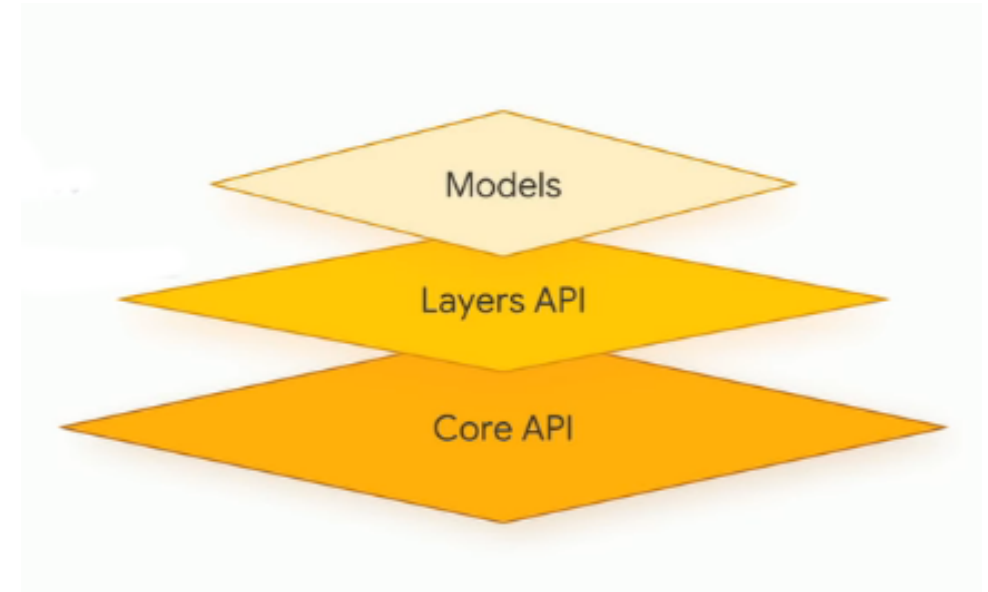
TensorFlow.js kullanarak ne yapılabilir

- Model direk browser üzerinde yazılabilir
- Önceden eğitilmiş (pre-trained) modeller kullanılabilir
- Mevcut Python modelleri dönüştürülüp kullanılabilir
- Yüklenen model yeniden eğitilebilir



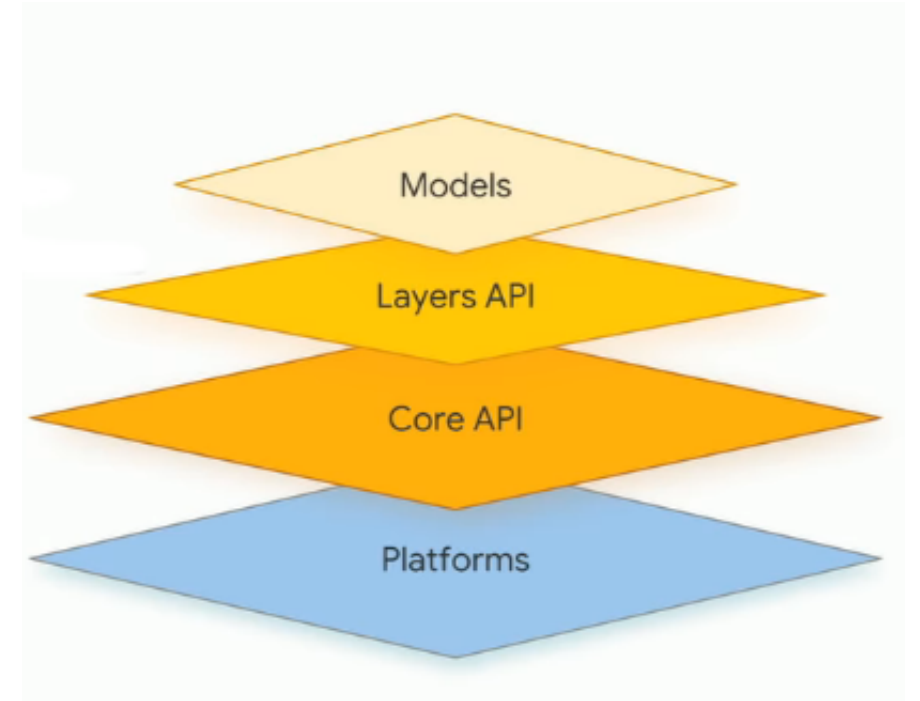
TensorFlow.js kullanarak ne yapılabilir

- Model direk browser üzerinde yazılabilir
- Önceden eğitilmiş (pre-trained) modeller kullanılabilir
- Mevcut Python modelleri dönüştürülüp kullanılabilir
- Yüklenen model yeniden eğitilebilir
- Tarayıcı üzerinde eğitim (training) yapılabilir

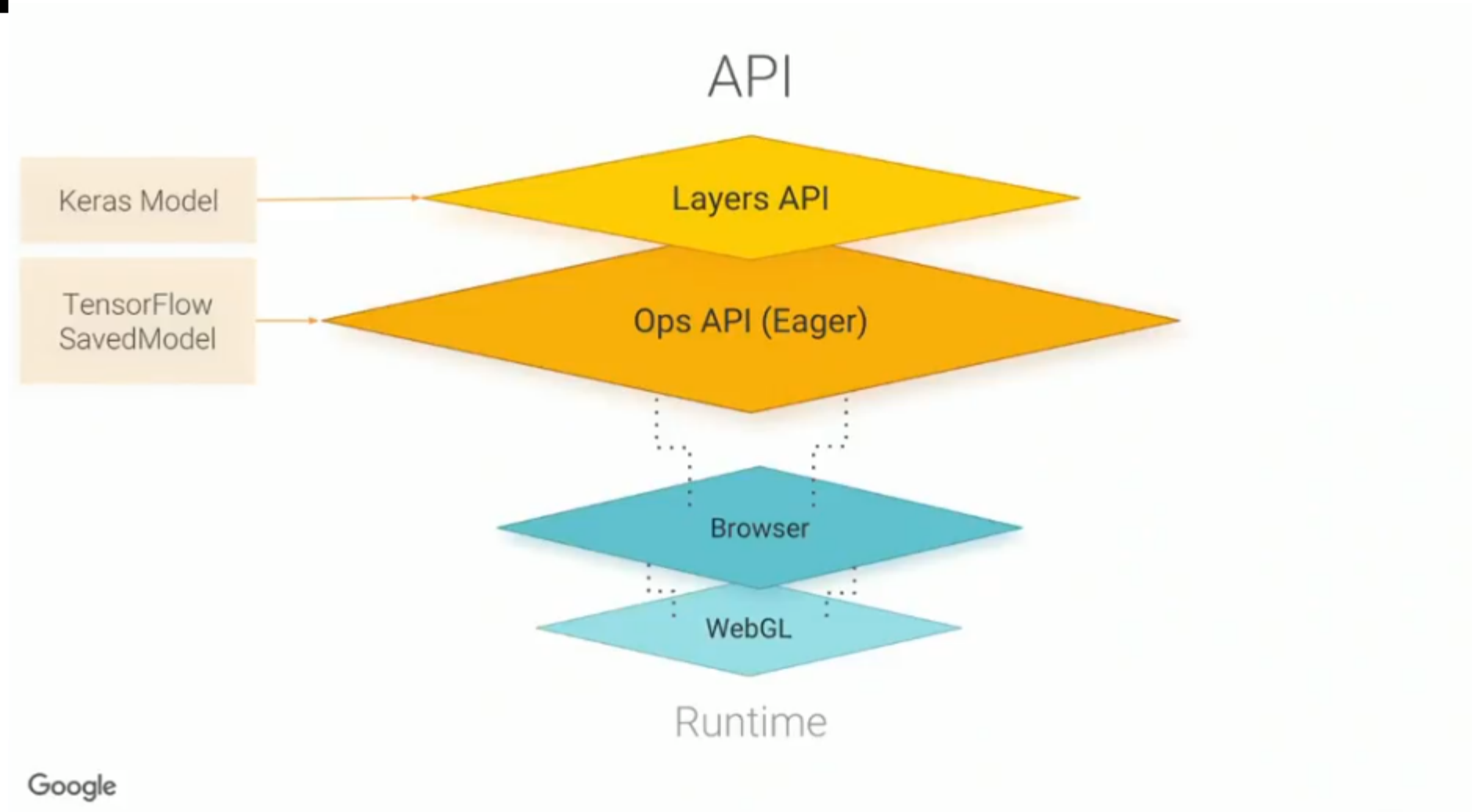


TensorFlow.js kullanarak ne yapılabilir

- Model direk browser üzerinde yazılabilir
- Önceden eğitilmiş (pre-trained) modeller kullanılabilir
- Mevcut Python modelleri dönüştürülüp kullanılabilir
- Yüklenen model yeniden eğitilebilir
- Tarayıcı üzerinde eğitim (training) yapılabilir
- Node.js ile bir çok platforma dağıtılabilir



Mimari





Epoch
000,725

Learning rate

0.001

Activation

ReLU

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



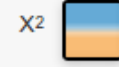
Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?



2 HIDDEN LAYERS



4 neurons



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



2 neurons

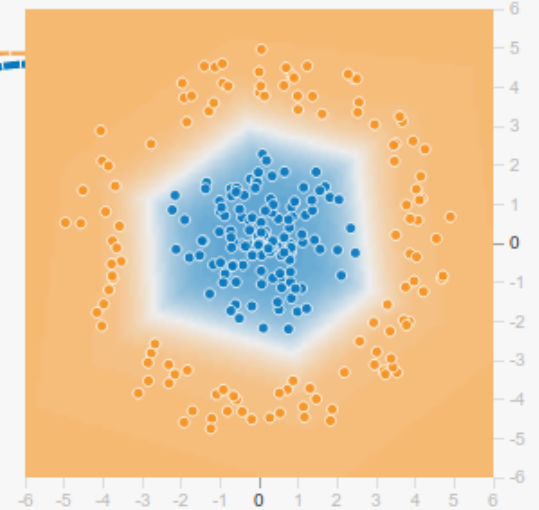


The outputs are mixed with varying weights, shown by the thickness of the lines.

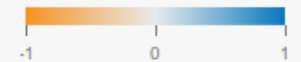
OUTPUT

Test loss 0.031

Training loss 0.020



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

playground.tensorflow.org

Mevcut Eğitilmiş Modeli Yükleme

Python



Save

```
builder =  
saved_model.builder.SavedModelBuilder(export_dir)  
with tf.Session(graph=tf.Graph()) as sess:  
    ...  
    builder.add_meta_graph_and_variables(sess)  
builder.save()
```

TensorFlow SavedModel

```
model = Sequential()  
model.add(Dense(2, input_dim=3, name='dense_1'))  
model.add(Dense(3, name='dense_2'))  
...  
model.save_weights('model.hdf5')
```

Keras Model

Mevcut Eğitilmiş Modeli Yükleme

Python

Save

Convert

```
$ pip install tensorflowjs
$ tensorflowjs_converter \
  --input_format tf_saved_model \
  --output_node_names 'Predictions/Reshape_1' \
  path/to/tf_saved_model \
  path/to/tfjs_model
```

TensorFlow SavedModel

```
$ pip install tensorflowjs
$ tensorflowjs_converter \
  --input_format keras \
  path/to/input/model.h5 \
  path/to/output/tfjs_model
```

Keras Model

Mevcut Eğitilmiş Modeli Yükleme

Python

TensorFlow.js

Save

Convert

Load

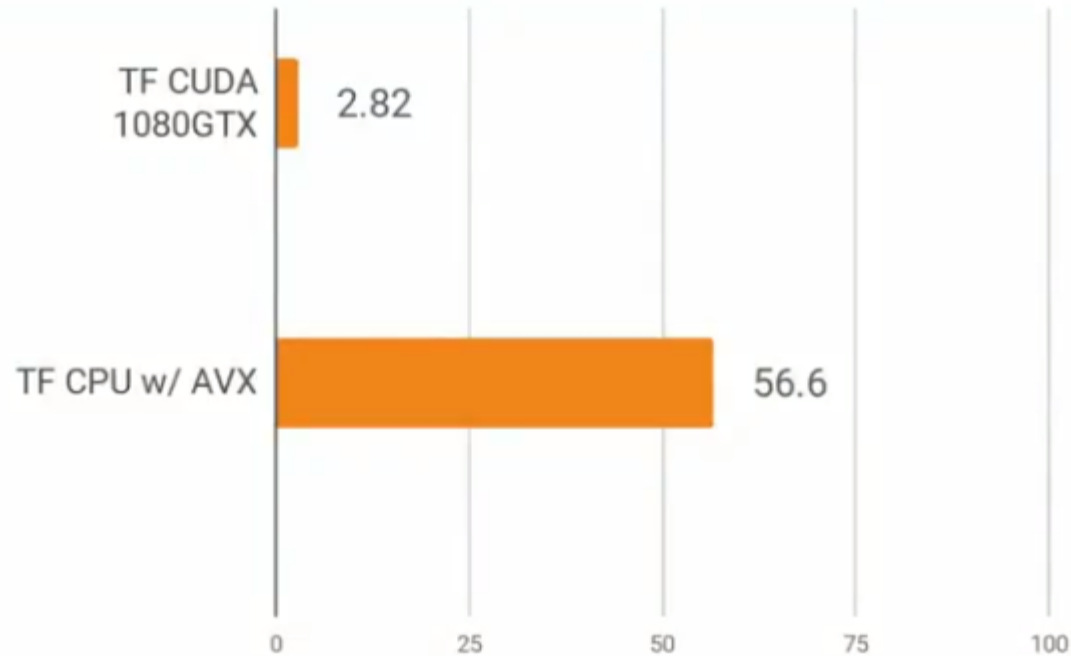
```
const model =  
    await tf.loadFrozenModel(modelUrl, weightUrl);  
const feed = {'input_placeholder': input};  
const prediction = model.execute(feed);
```

TensorFlow SavedModel

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

Keras Model

Performans: Tarayıcı Üzerinde Eğitim

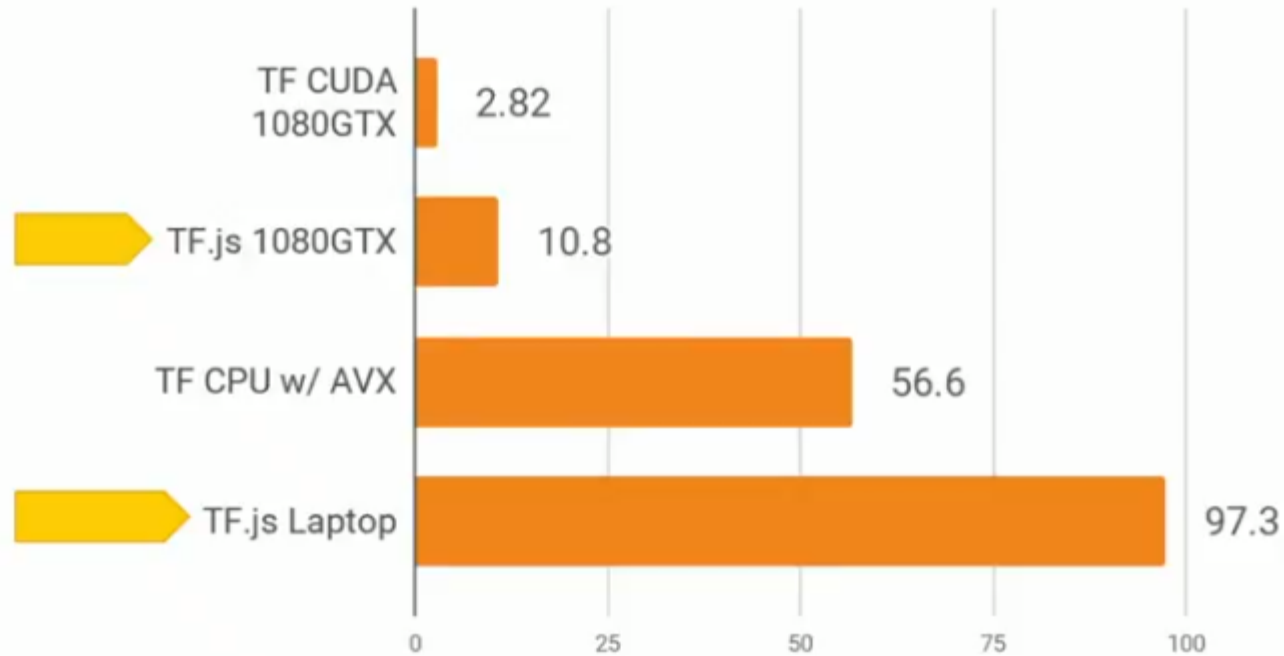


Inference Time (ms) of MobileNet 1.0_224

Average of 200 runs

Kaynak: <https://thekevinscott.com/bostonjs/>

Performans: Tarayıcı Üzerinde Eğitim

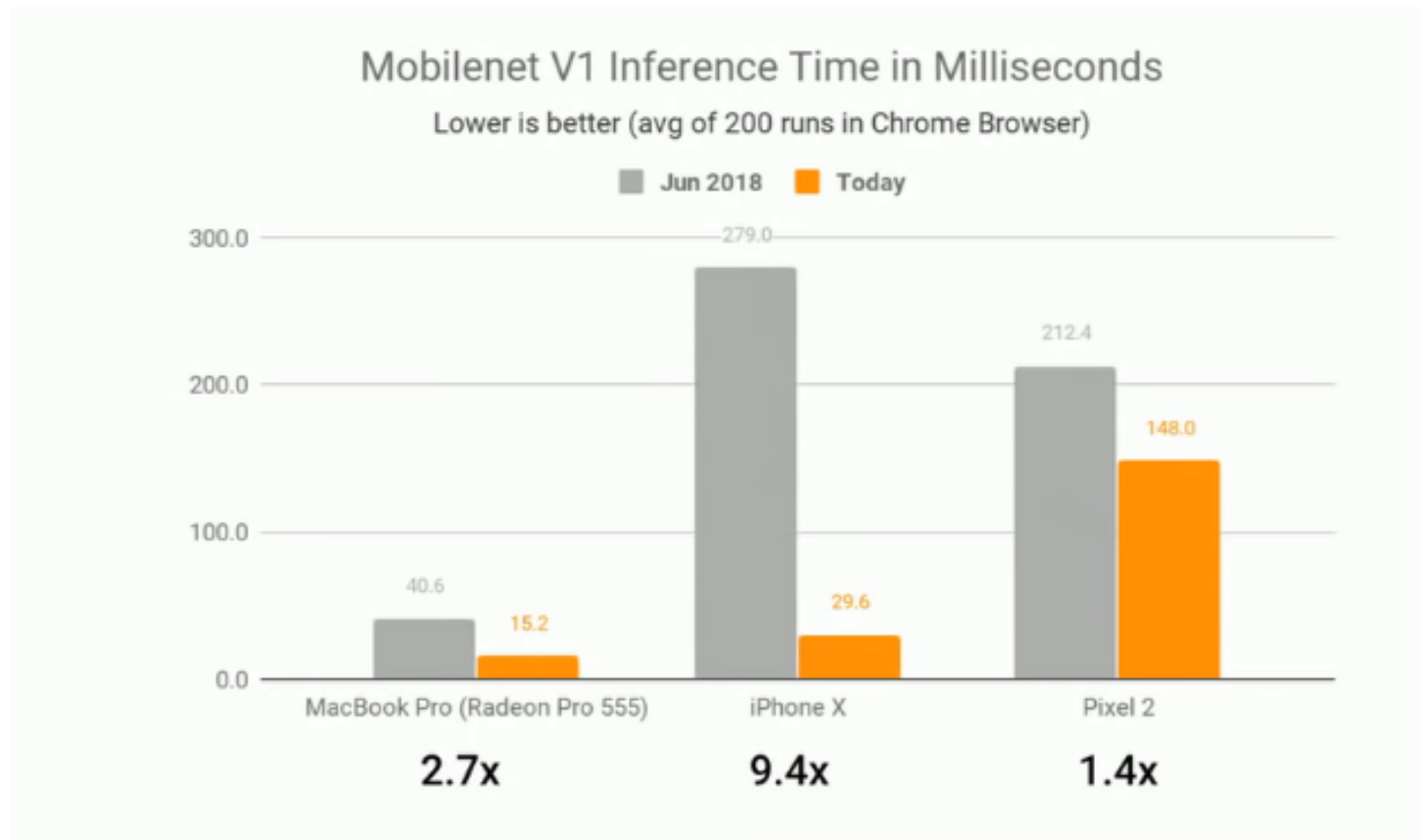


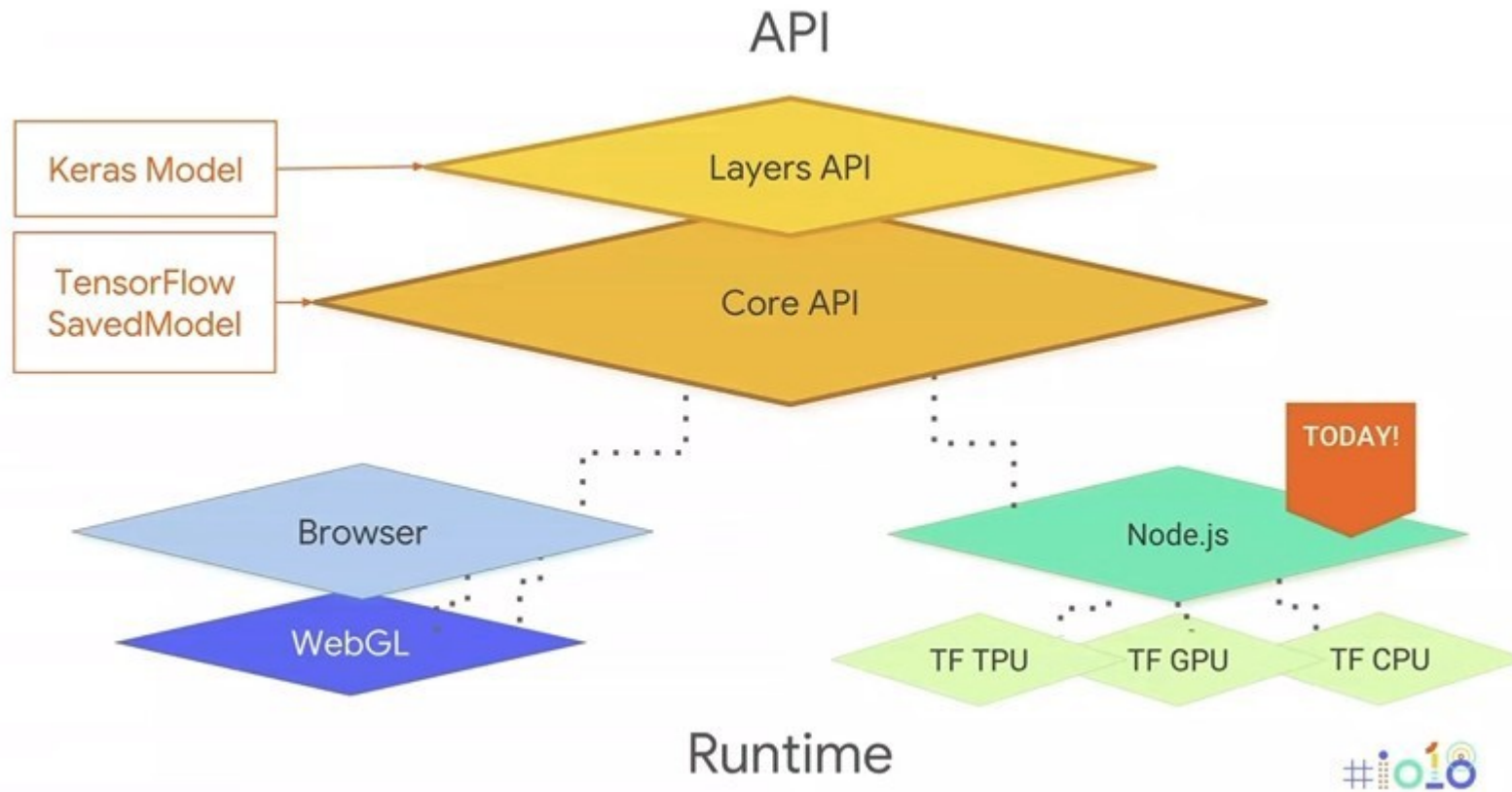
Inference Time (ms) of MobileNet 1.0_224

Average of 200 runs

Kaynak: <https://thekevinscott.com/bostonjs/>

Performans: Mobil Üzerinde





Kaynaklar

- <https://www.tensorflow.org/js>
- <https://ml5js.org/>

Demo

- <https://teachablemachine.withgoogle.com/>
- <https://github.com/shekit/alexa-sign-language-translator>
- <https://www.metacar-project.com/>

Diğer resmi örnekler

- <https://www.tensorflow.org/js/demos/>

TEŞEKKÜRLER

YAVUZ KÖMEÇOĞLU

www.yavuzkomecoglu.com

   / YavuzKomecoglu

komecoglu.yavuz@gmail.com