TensorFlow

MACHINE LEARNING IN JS

2018

Stojan Gancev

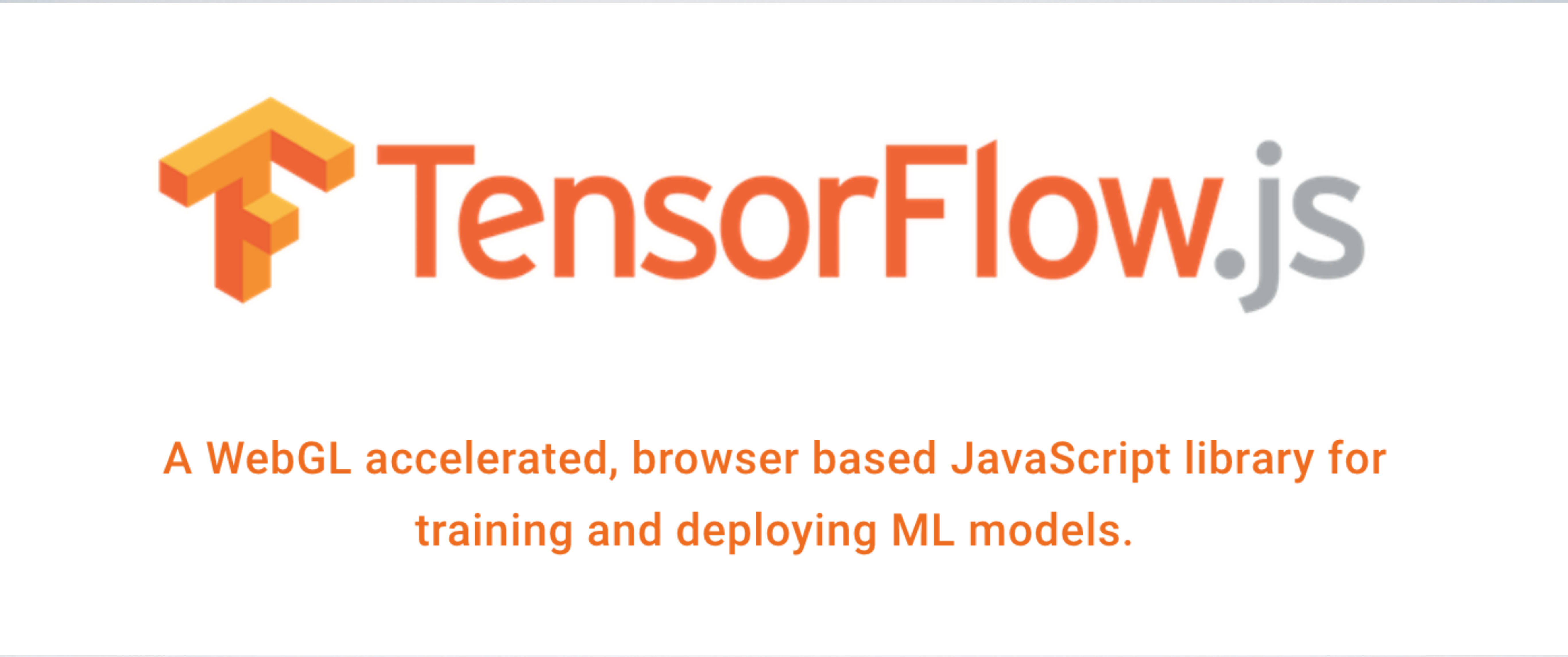
AGENDA

- How it all started.
- What are tools available today.
- How we build a Machine learning model
- Performance overview
- Pros and const
- How to apply it example
- Beer & Questions

Machine learning is for Python and R …
How about Javascript?

IN BROWSER ML

- No drivers
- sensors: microphone , camera
- interactive
- data can stay on the client. GDPR



TensorFlow.js

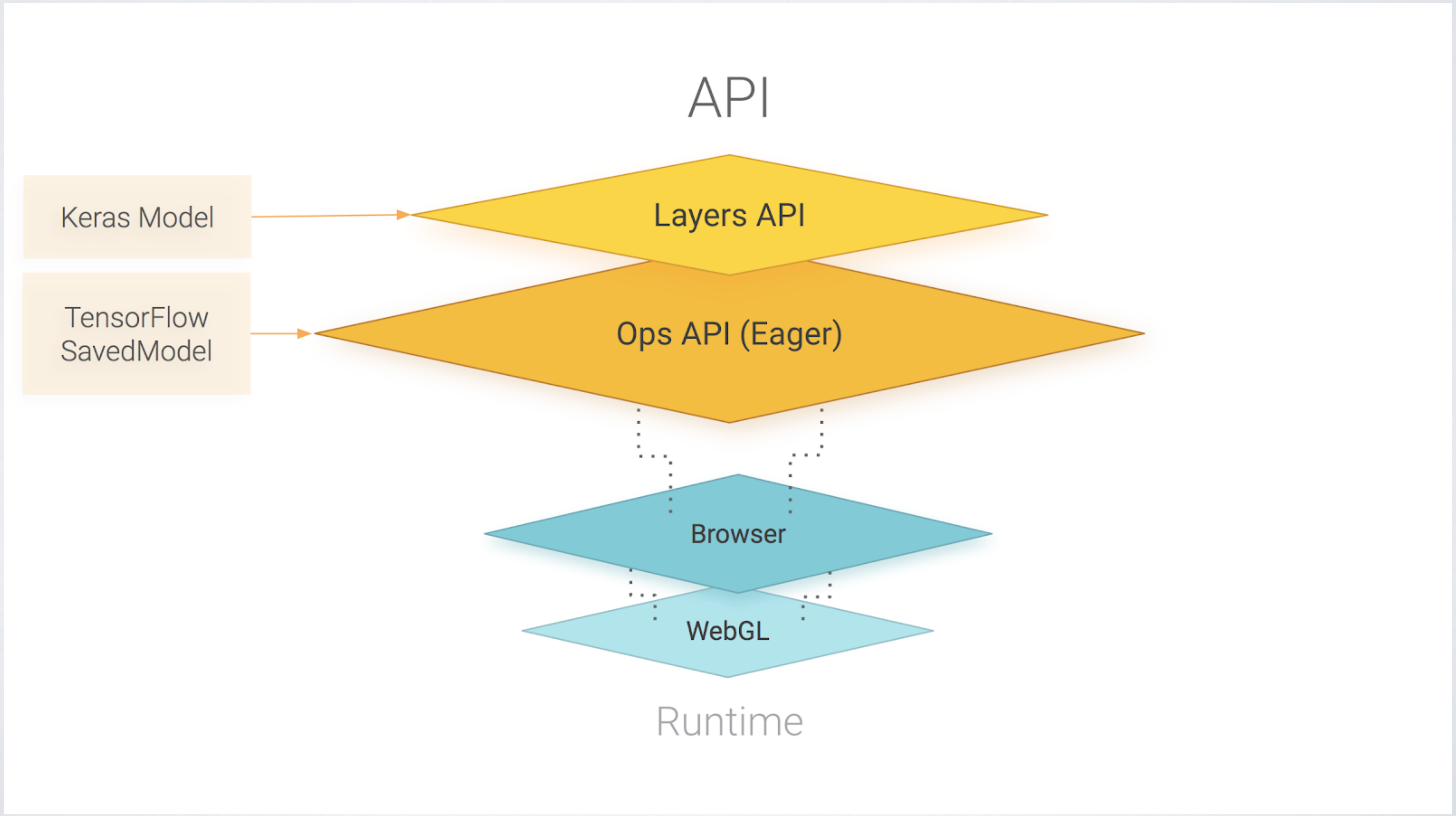
A WebGL accelerated, browser based JavaScript library for training and deploying ML models.

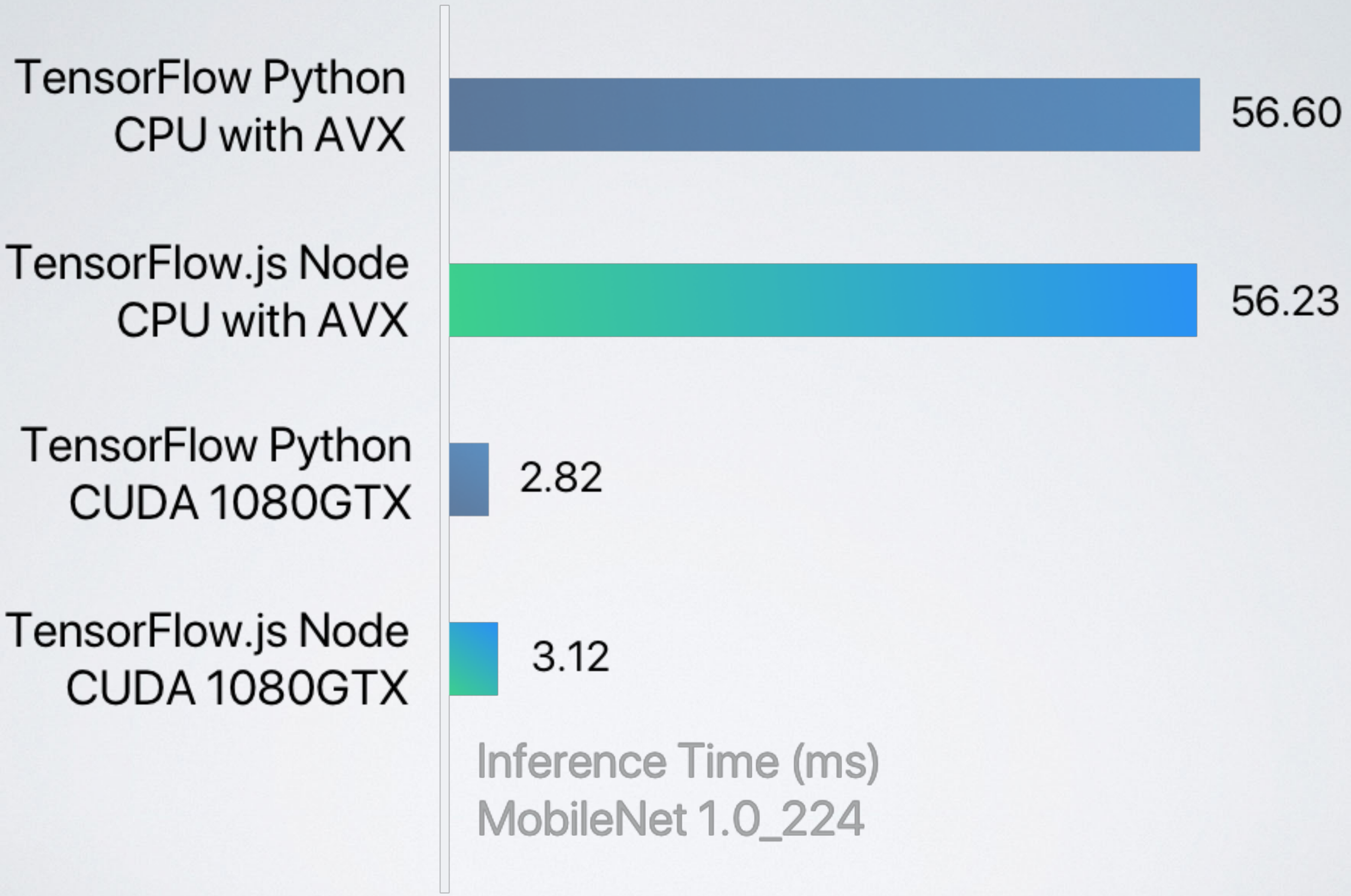
The beginnings are

https://playground.tensorflow.org

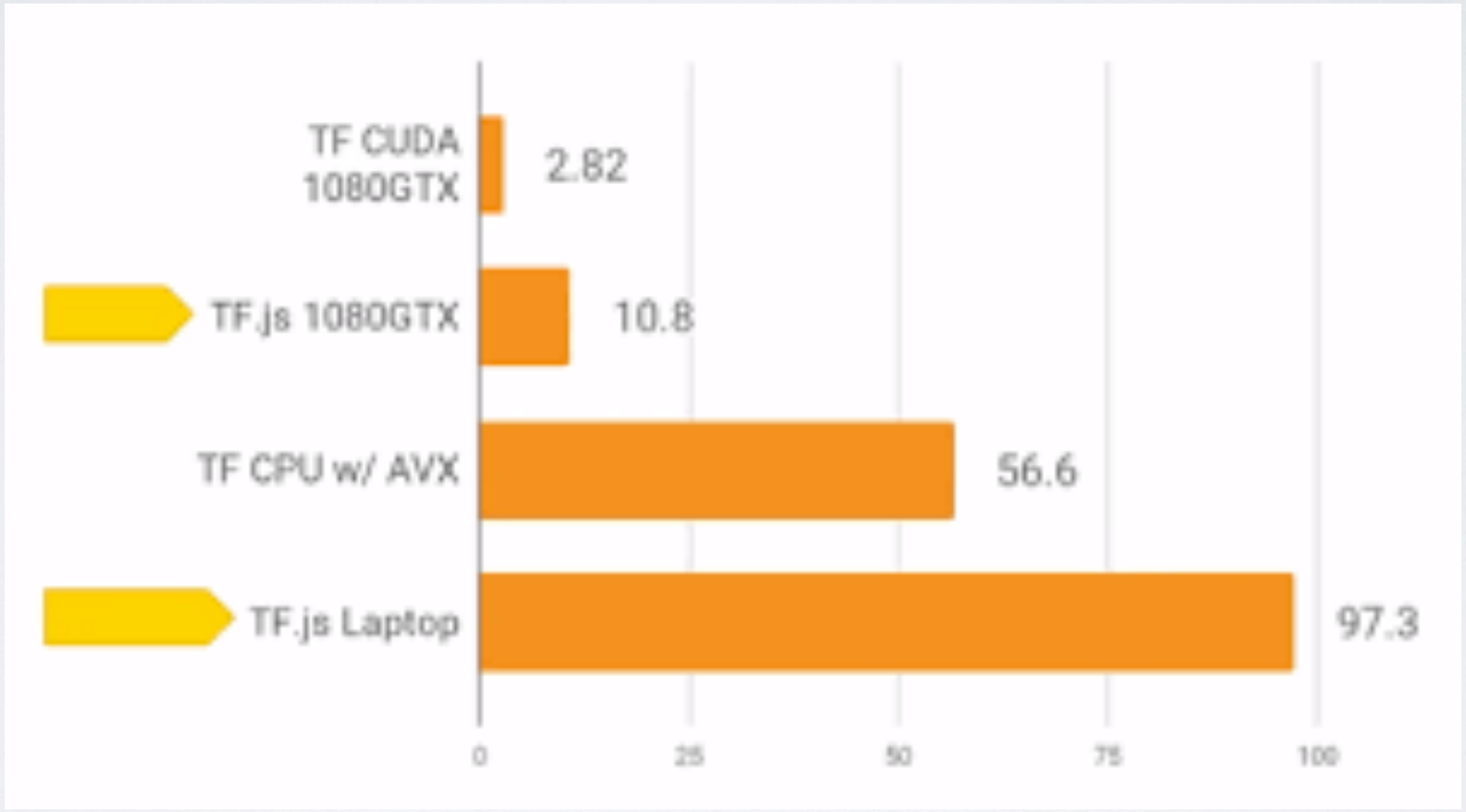
WITH TFJS YOU CAN

- Train model directly in the browser
- Use pertained models and run the in the browser




TensorFlow Python CPU with AVX
56.60
TensorFlow.js Node CPU with AVX
56.23
TensorFlow Python CUDA 1080GTX
2.82
TensorFlow.js Node CUDA 1080GTX
3.12
Inference Time (ms)
MobileNet 1.0_224

PERFORMANSE



THE FLOW FOR BUILDING A MODEL

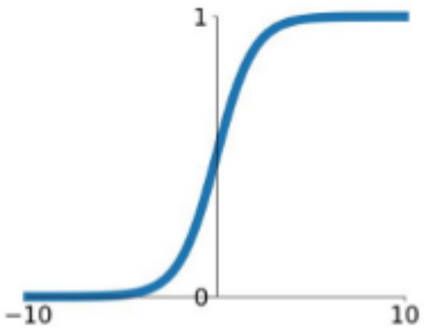
1. Define Network
2. Compile Network
3. Fit Network
4. Evaluate Network
5. Make Predictions

SHORTCUTS

Activation Functions

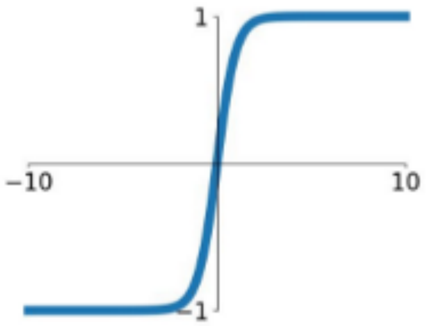
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



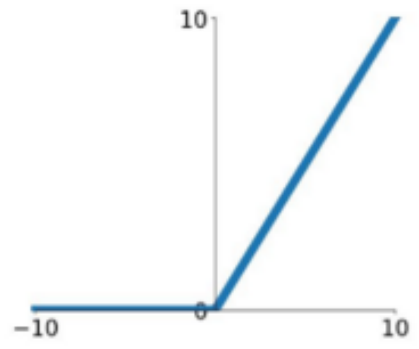
tanh

$\tanh(x)$



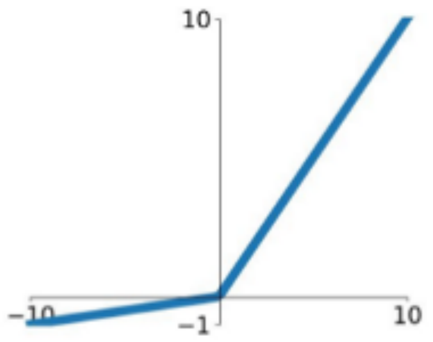
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

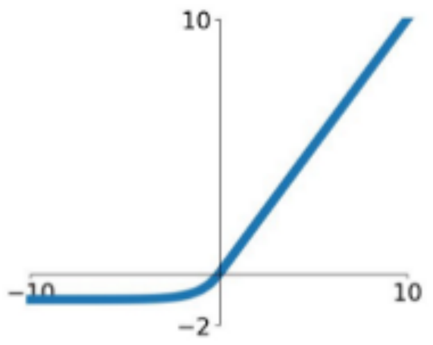


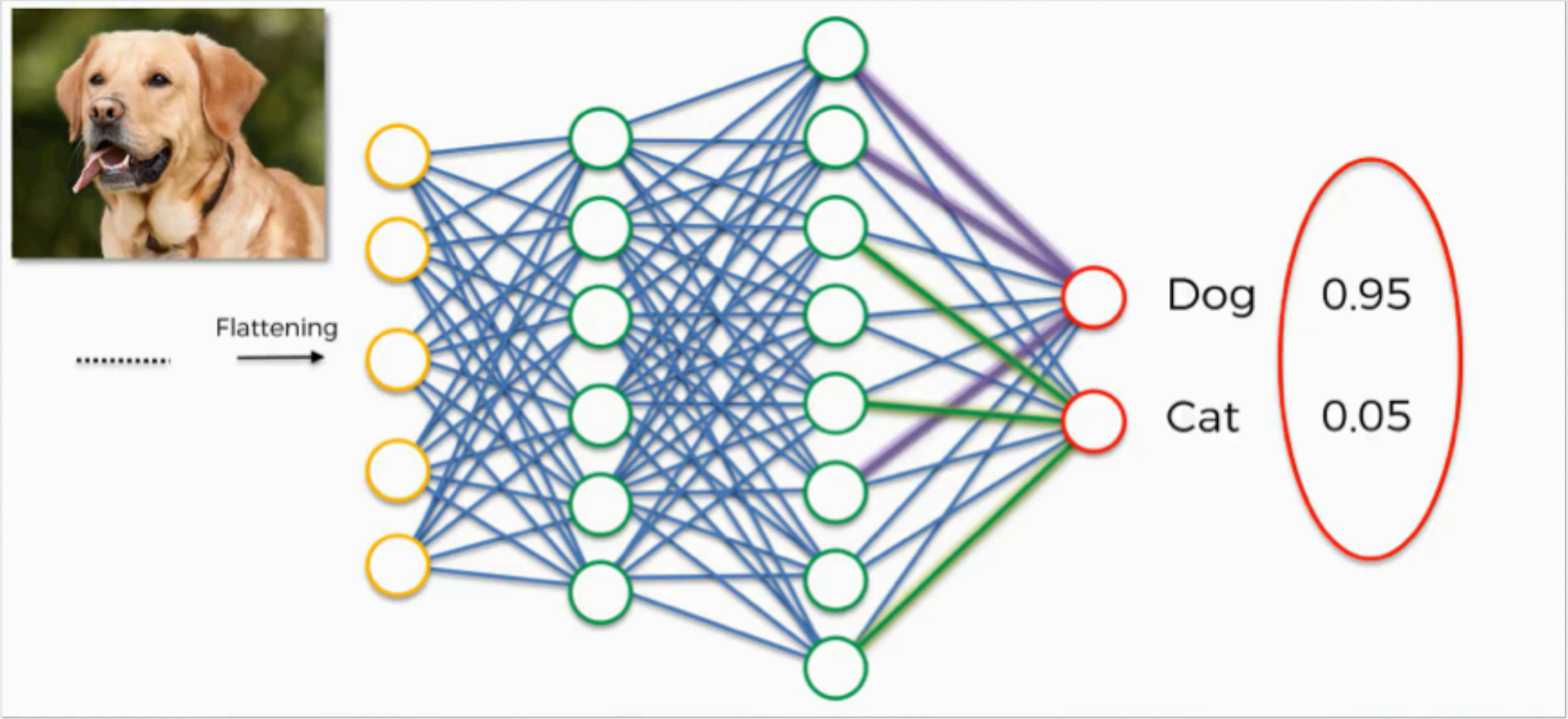
Maxout

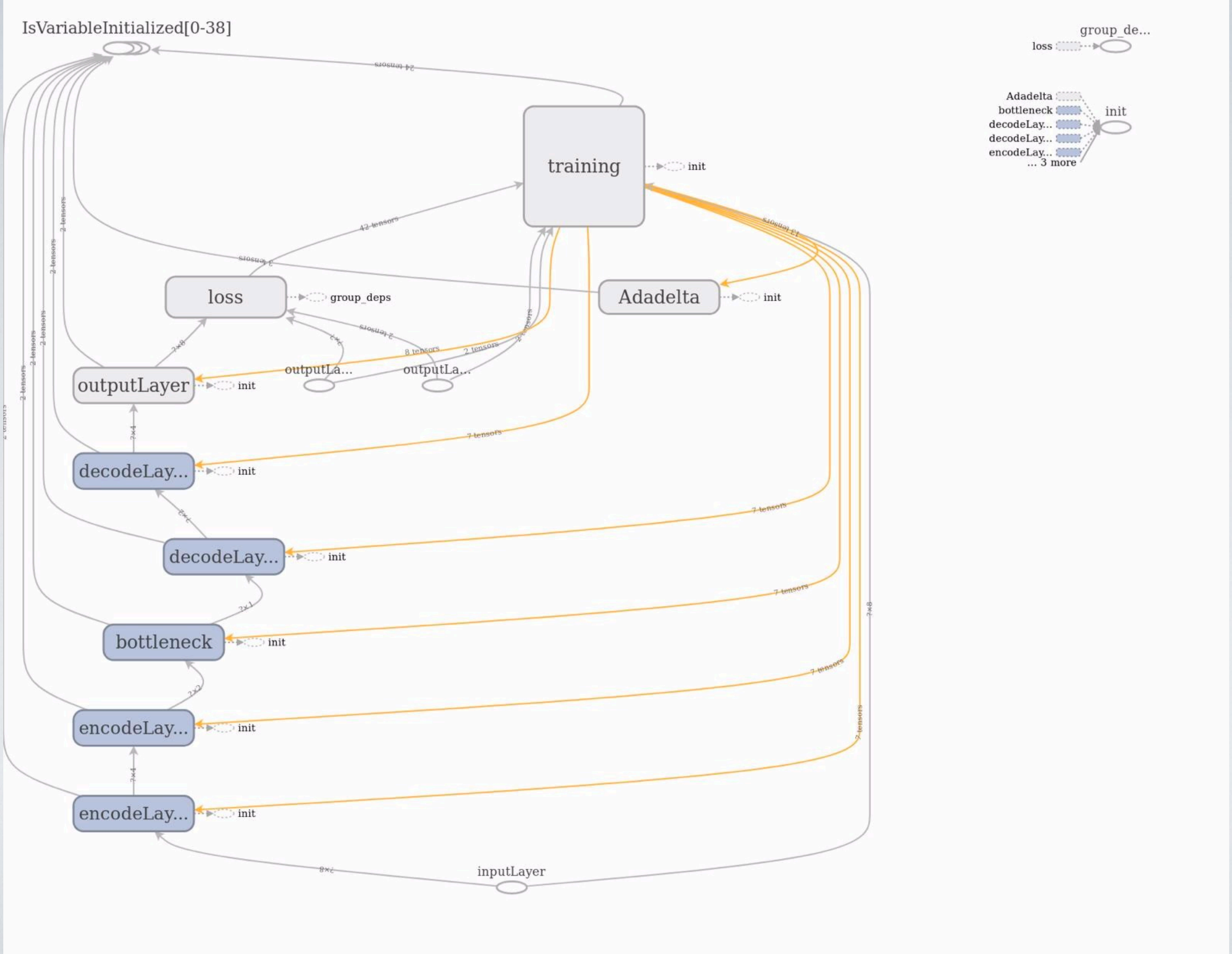
$\max(w_1^T x + b_1, w_2^T x + b_2)$

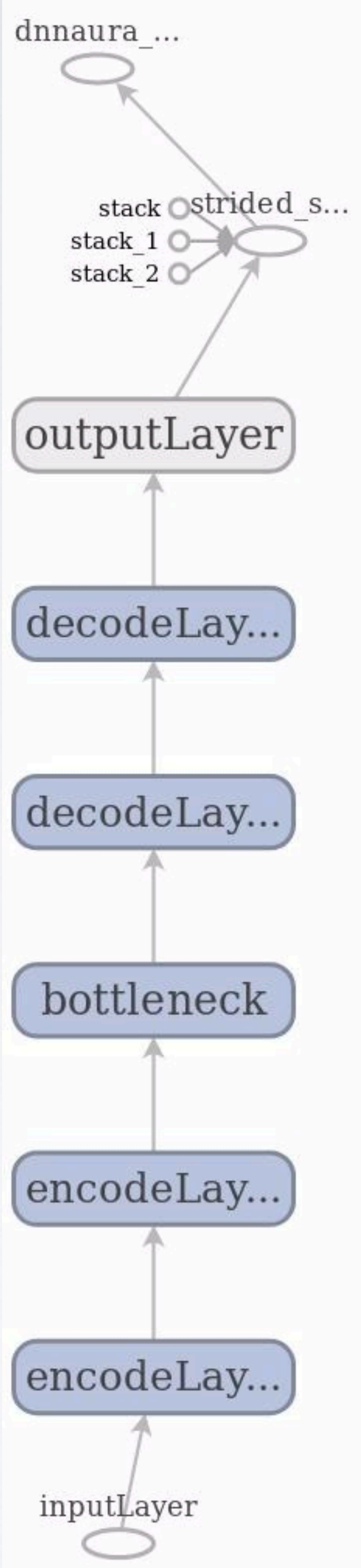
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



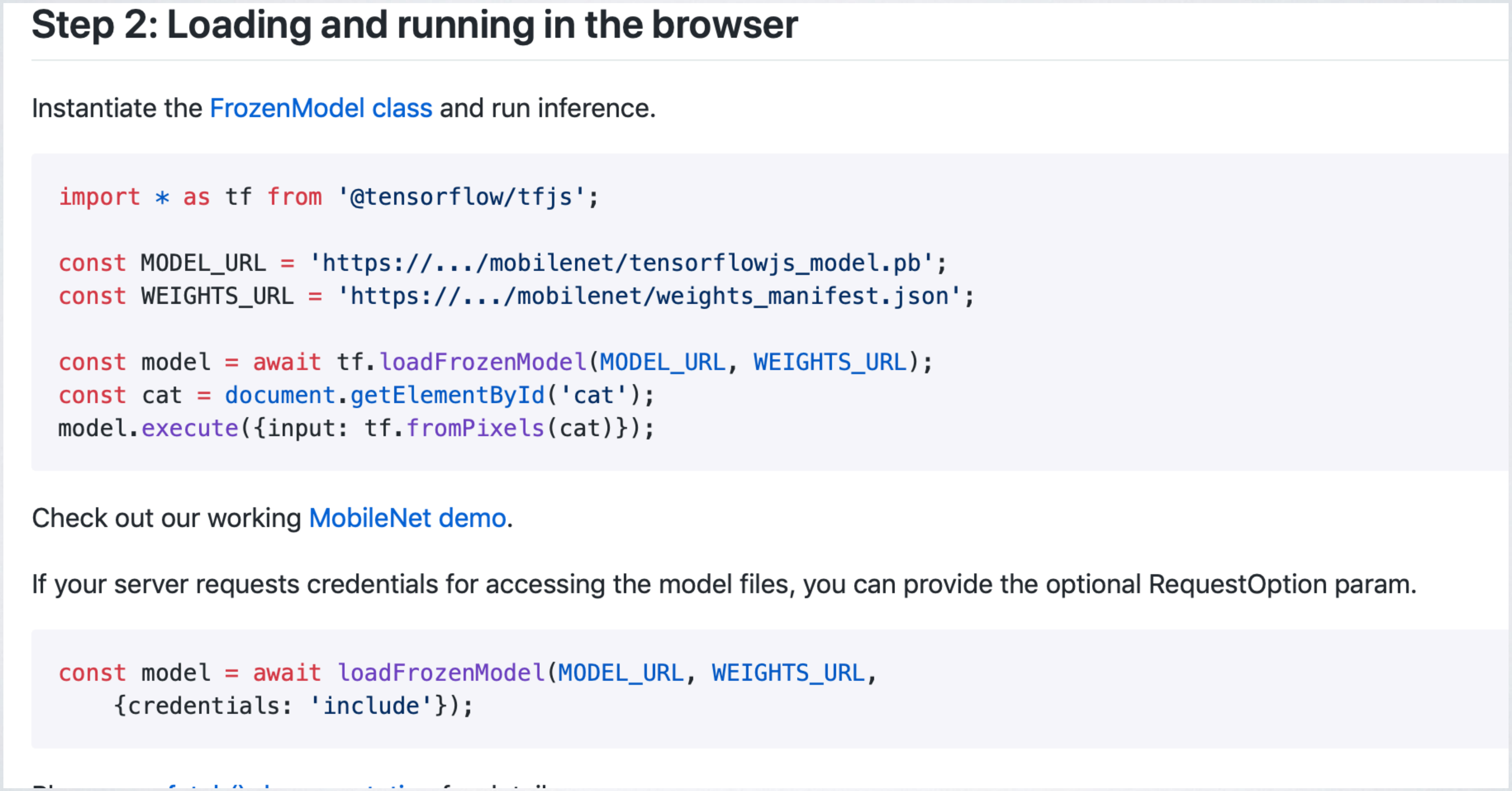

Flattening
Dog
Cat
0.95
0.05


IsVariableInitialized[0-38]
24 tensors
group_de...
loss
Adadelta
bottleneck
decodeLay...
decodeLay...
encodeLay...
... 3 more
init
training
init
42 tensors
13 tensors
3 tensors
loss
group_deps
Adadelta
init
2 tensors
2 tensors
2 tensors
2 tensors
2 tensors
2 tensors
?×2
2 tensors
8 tensors
2 tensors
2 tensors
?×8
outputLa...
outputLa...
outputLayer
init
?×4
7 tensors
decodeLay...
init
?×2
decodeLay...
init
7 tensors
?×1
7 tensors
bottleneck
init
?×2
7 tensors
encodeLay...
init
?×4
7 tensors
?×8
encodeLay...
init
?×8
inputLayer


dnnaura_...
stack
strided_s...
stack_1
stack_2
outputLayer
decodeLay...
decodeLay...
bottleneck
encodeLay...
encodeLay...
inputLayer

CONVERSION

- install tensorflowjs pip3 package.
- Steps https://github.com/tensorflow/tfjs-converter



Step 2: Loading and running in the browser

Instantiate the FrozenModel class and run inference.

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

const MODEL_URL = 'https://.../mobilenet/tensorflowjs_model.pb';
const WEIGHTS_URL = 'https://.../mobilenet/weights_manifest.json';

const model = await tf.loadFrozenModel(MODEL_URL, WEIGHTS_URL);
const cat = document.getElementById('cat');
model.execute({input: tf.fromPixels(cat)});
```

Check out our working MobileNet demo.

If your server requests credentials for accessing the model files, you can provide the optional RequestOption param.

```
const model = await loadFrozenModel(MODEL_URL, WEIGHTS_URL,
    {credentials: 'include'});
```

SAMPLES

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

APPLY

- recommendation, recognition prediction in browser.
- data protection
- server side pre-trained model

QUESTIONS & BEER