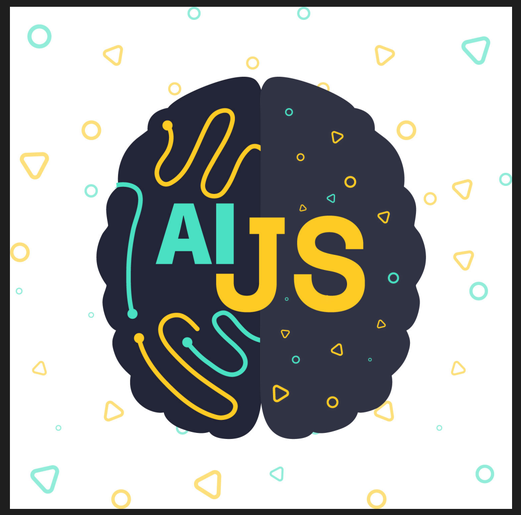
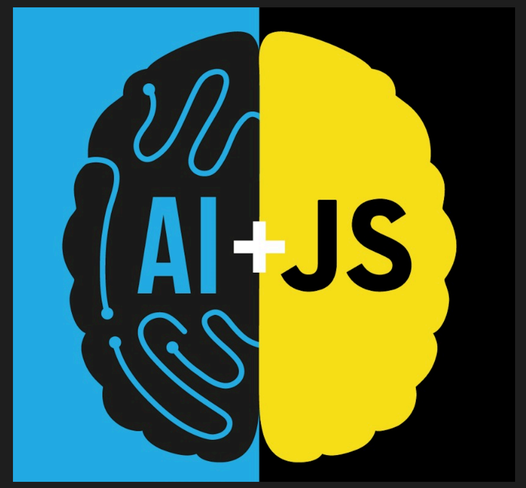
@EleanorHaproff

@MarvelApp



aijs.rocks



AI JavaScript London

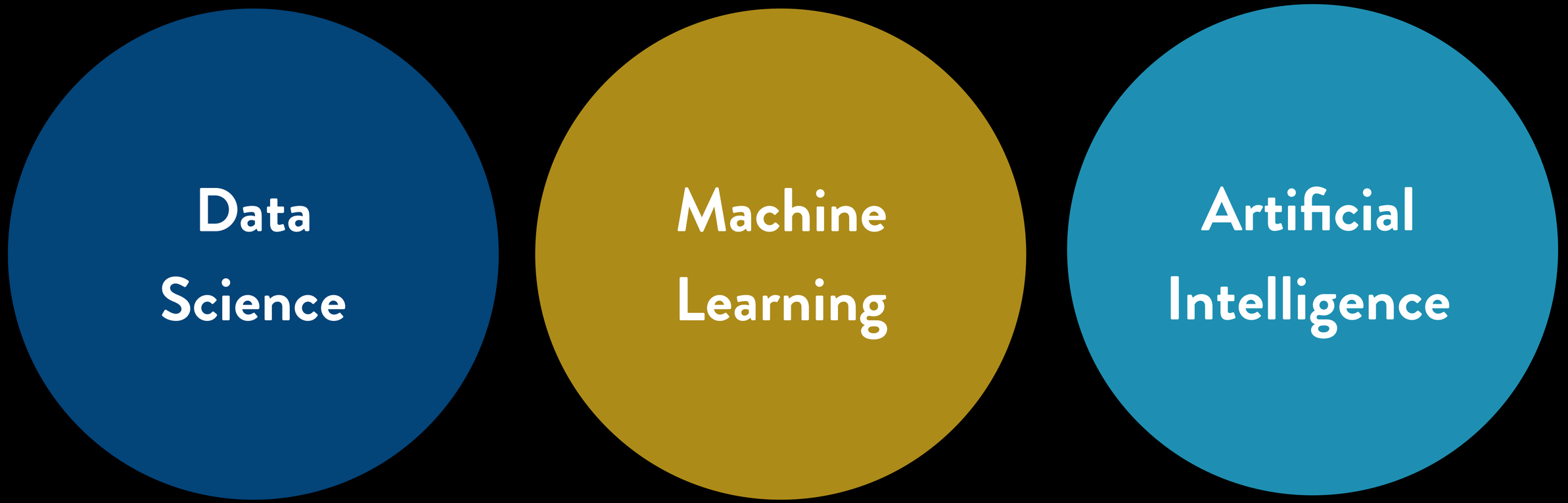
@aijavascript

Machine Learning & JavaScript

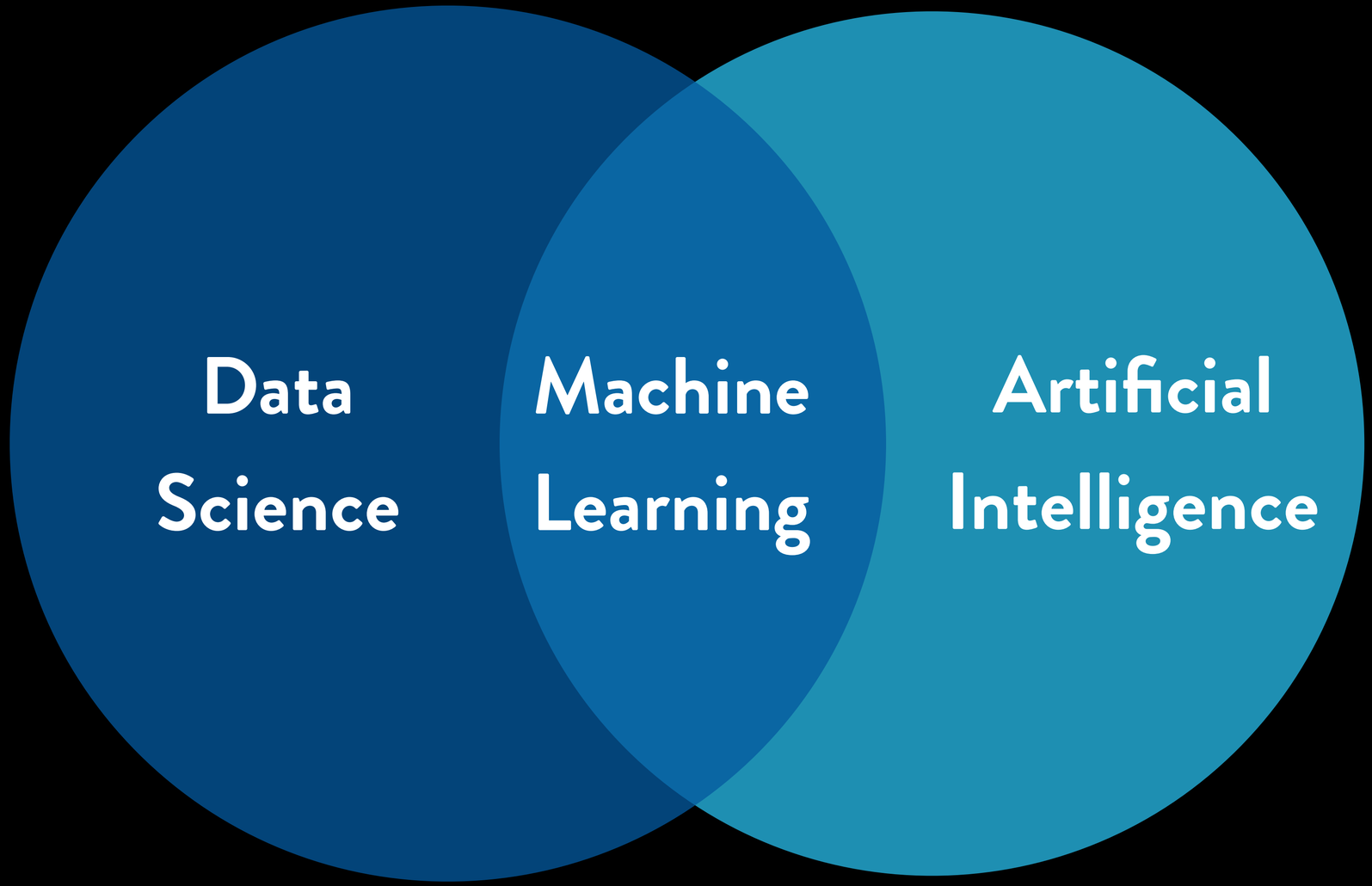
What is AI, Machine Learning & Data Science?

Solving complex problems using data

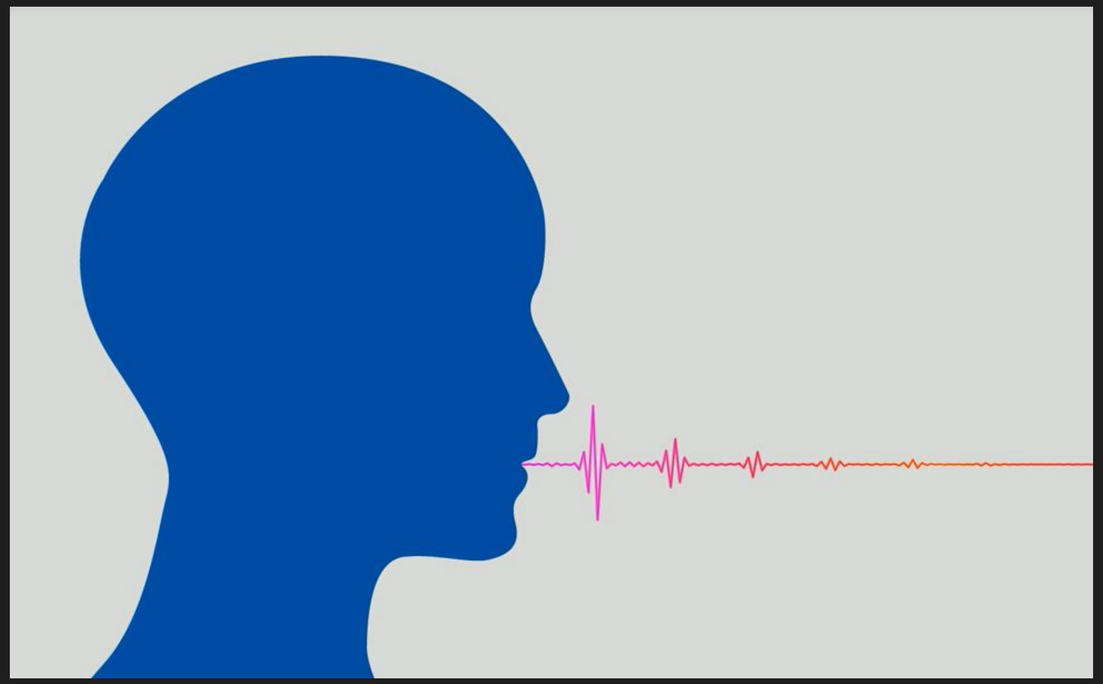
Simulation of a human brain function by machines

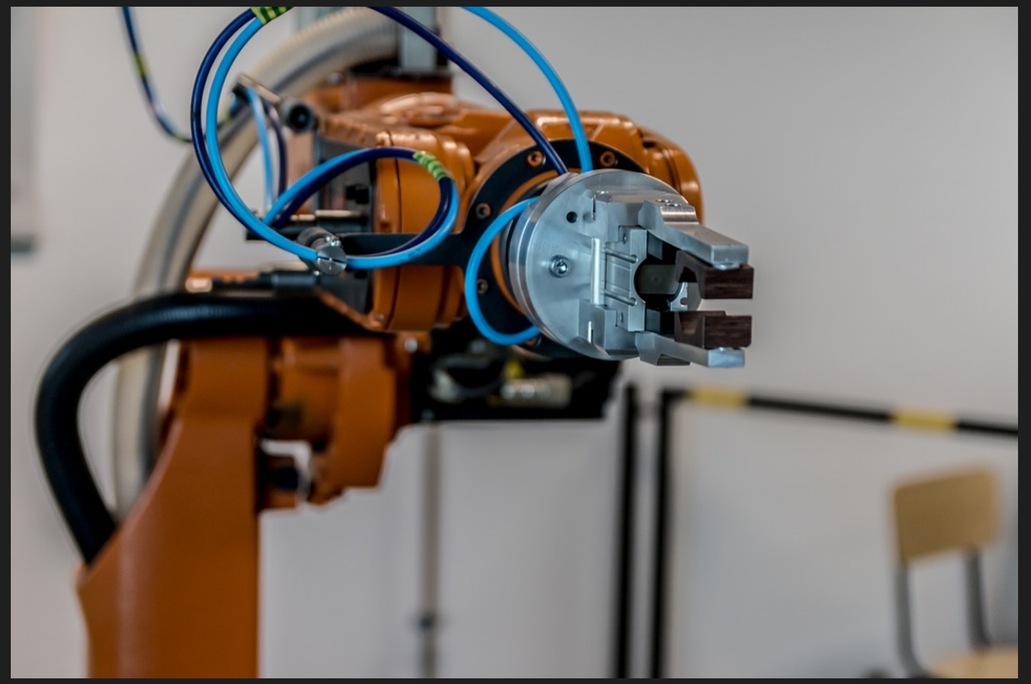


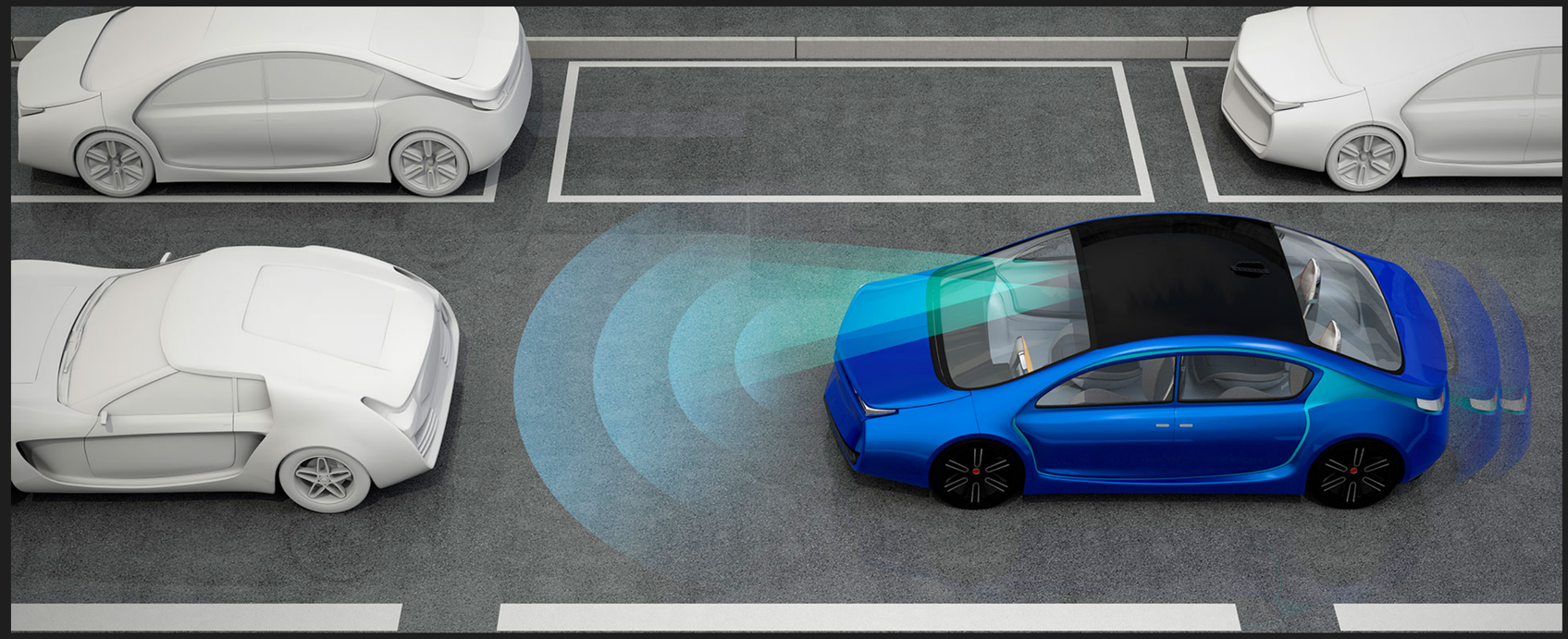
Learning from data


Data
Science
Machine
Learning
Artificial
Intelligence

Artificial Neural Networks

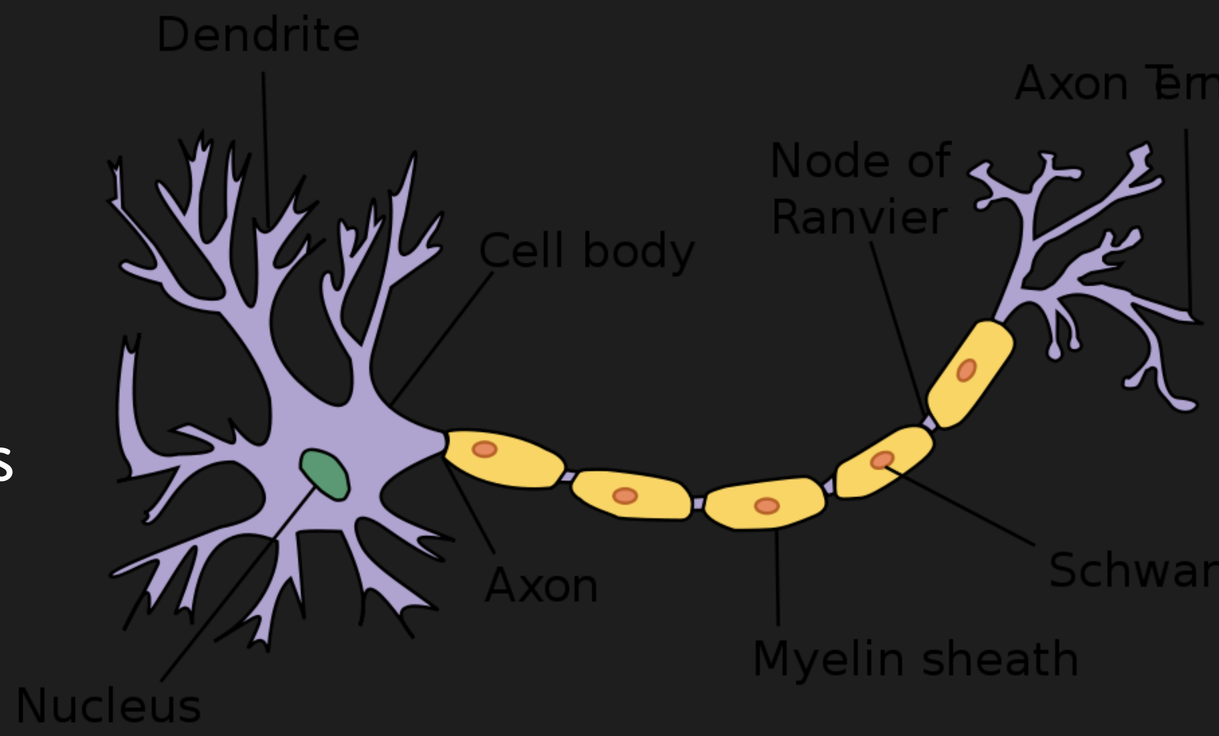




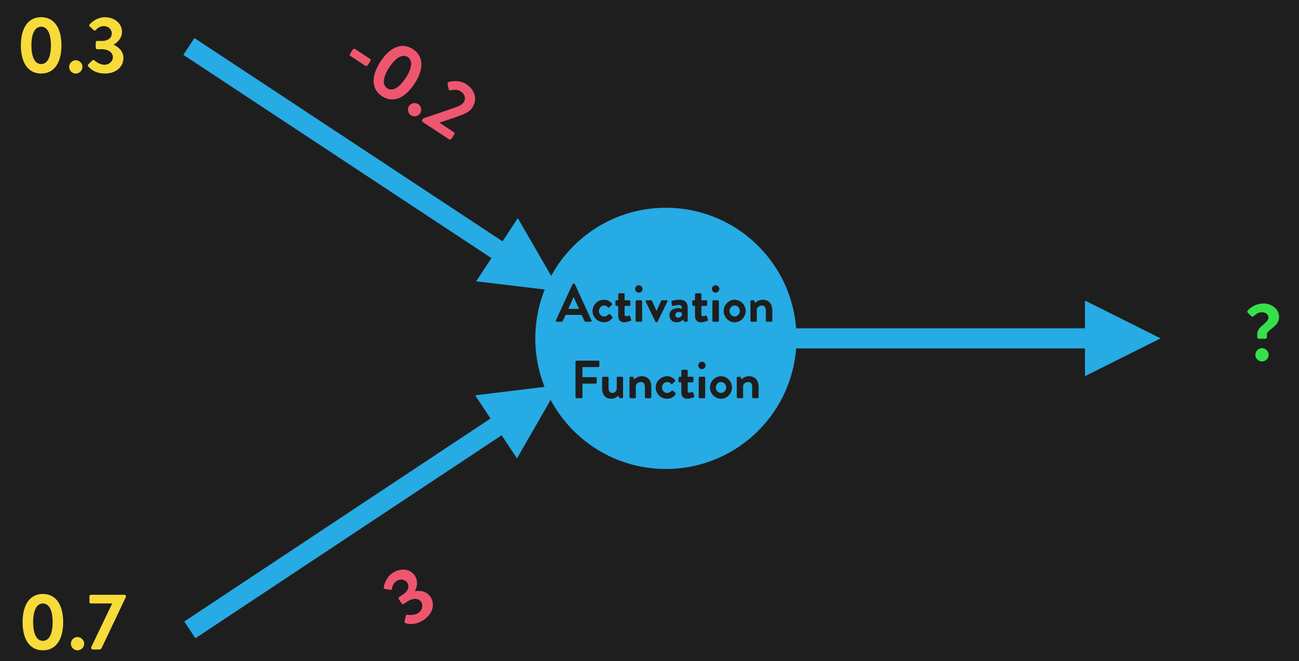


Axon

Biological Neuron



Artificial Neuron



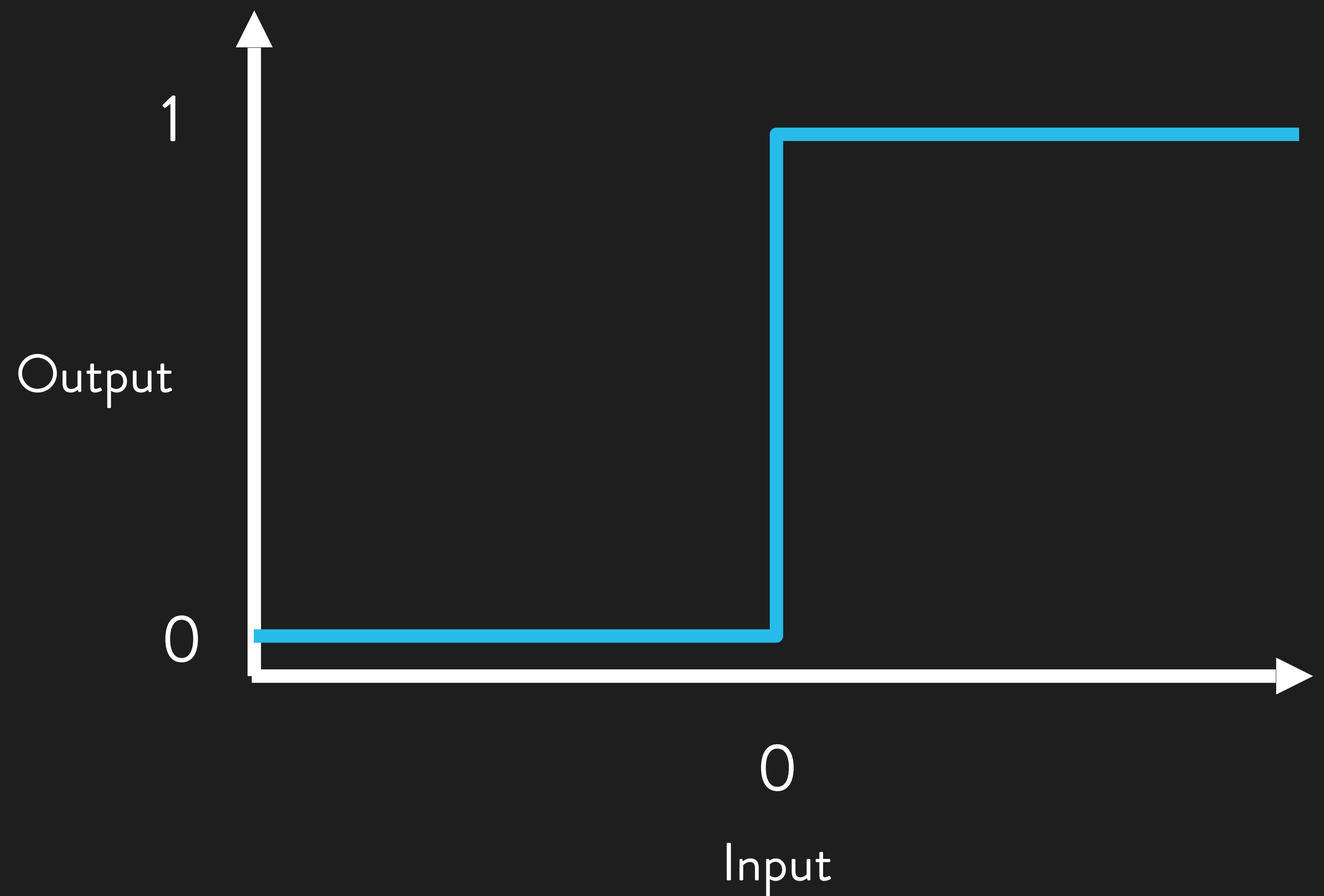
Artificial Neuron

0.3 x -0.2 = -0.6

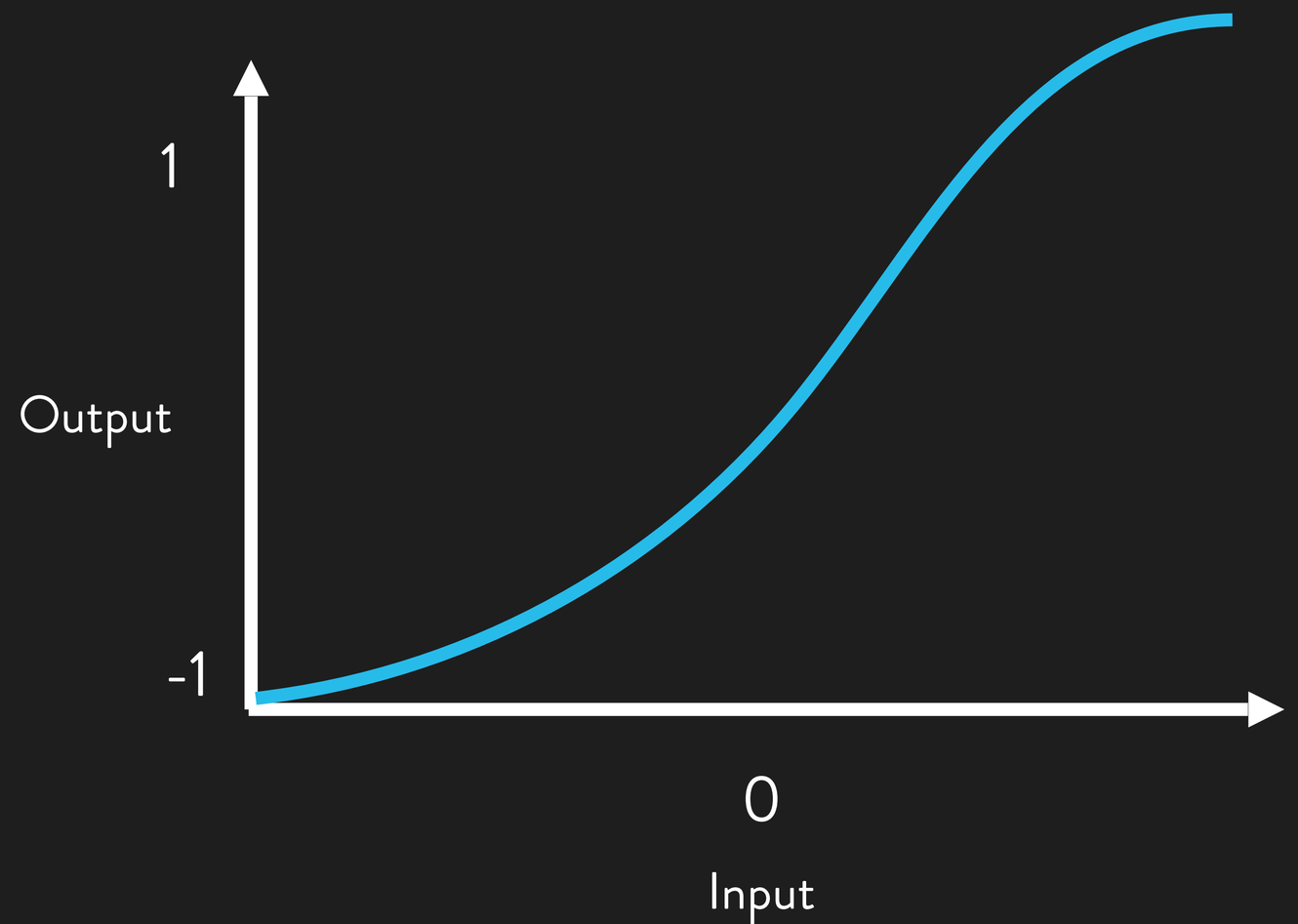
0.7 x 3 = 2.1

} 2.7 -> activation(...) -> 

Activation functions

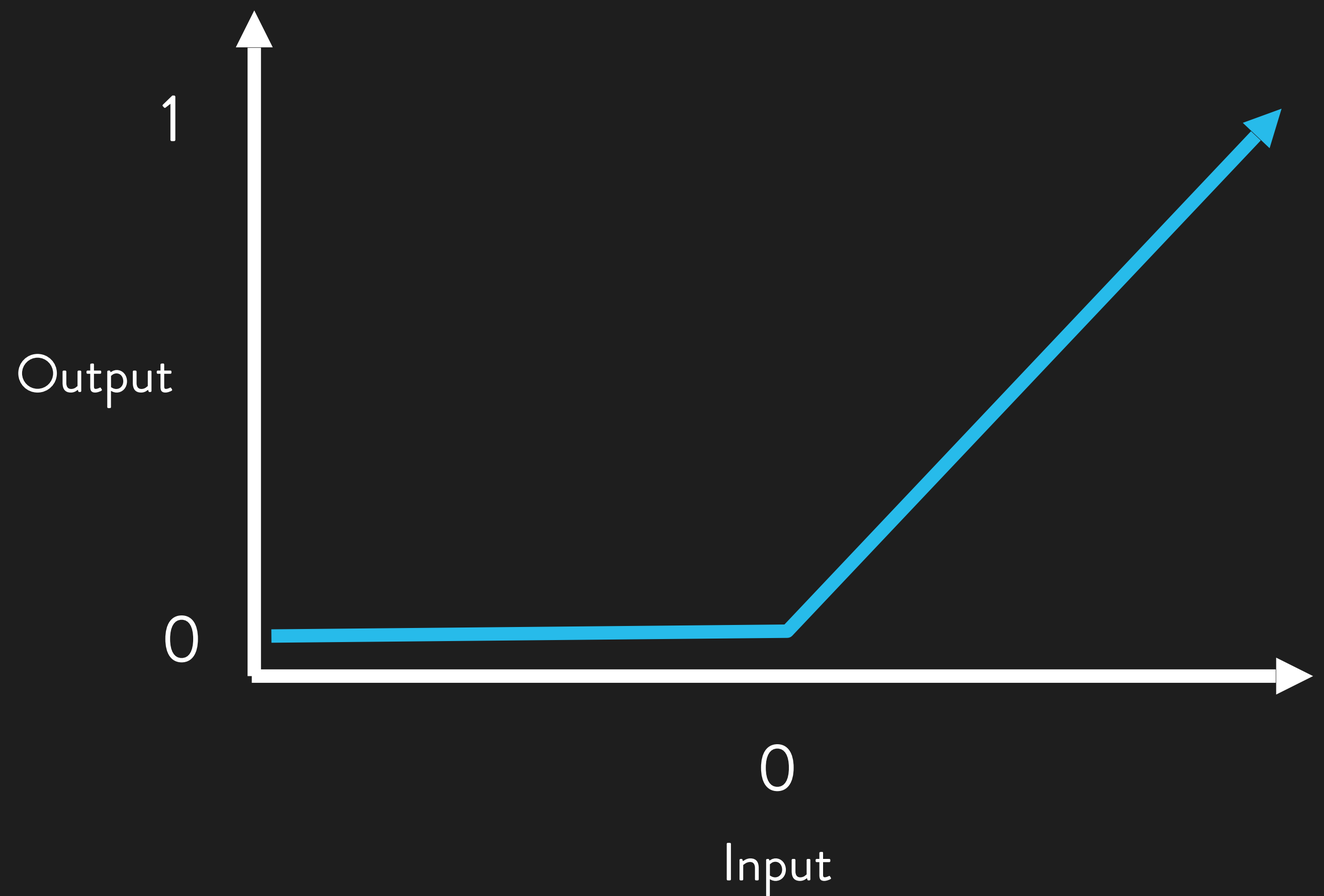


Activation functions



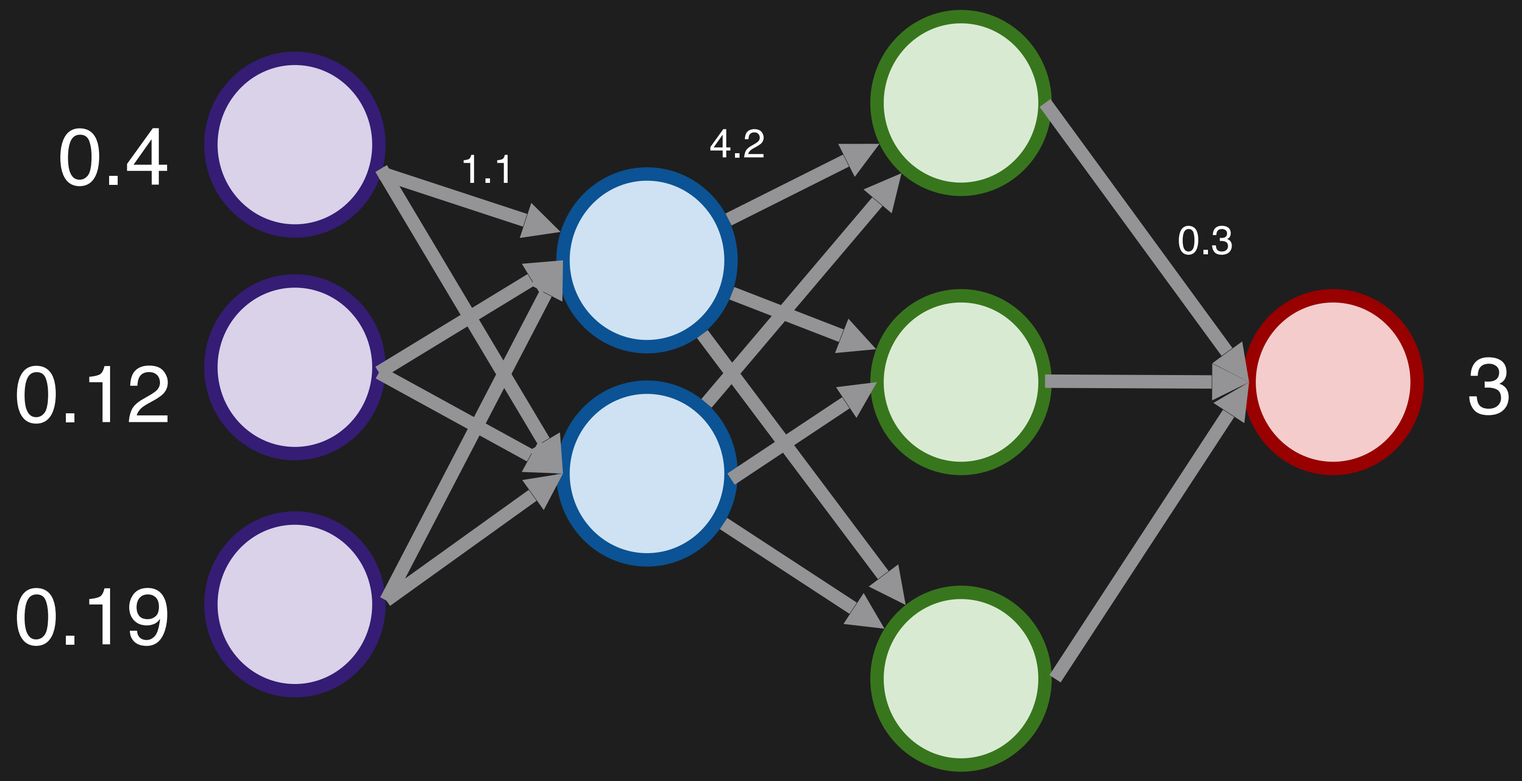
Sigmoid

Activation functions

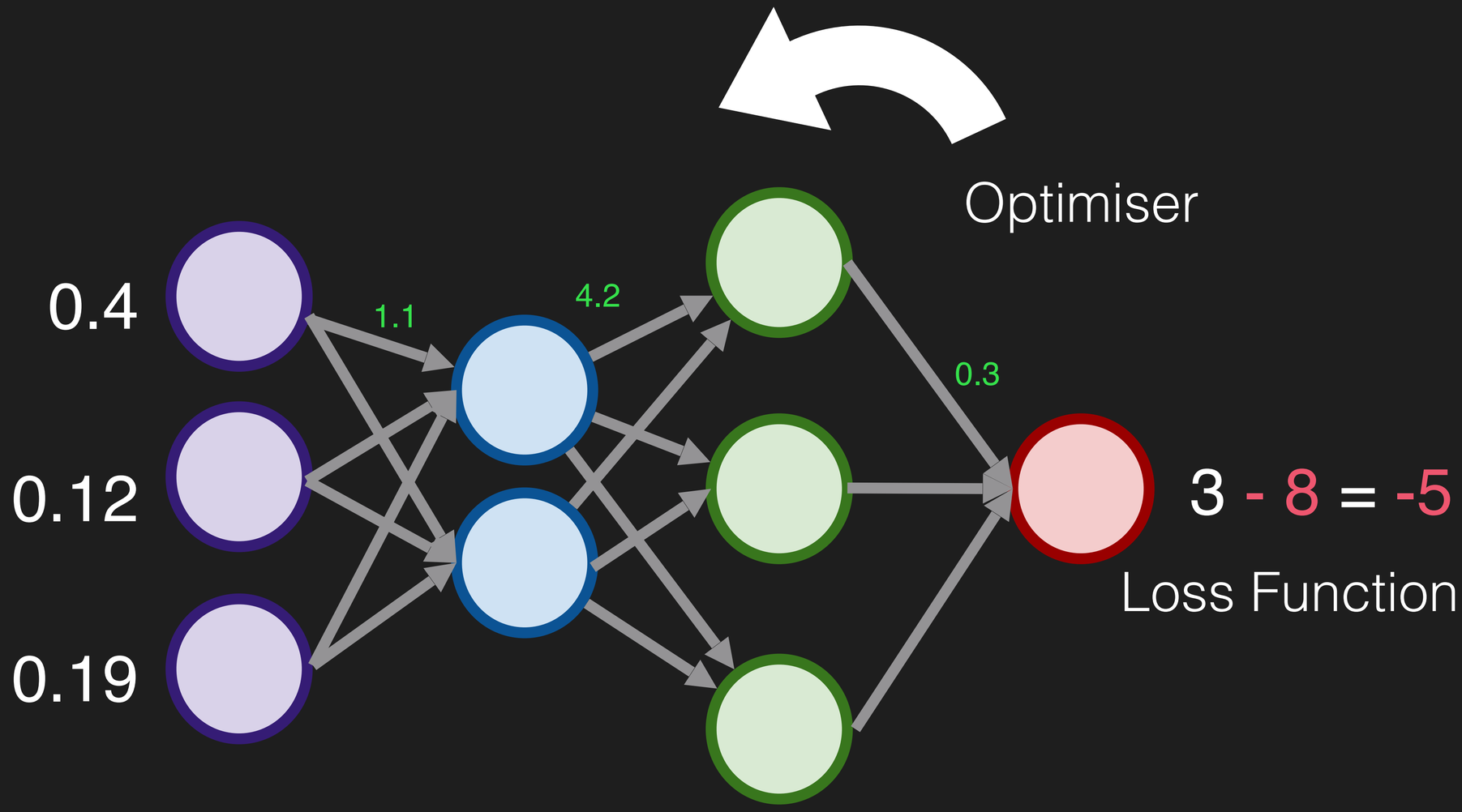


ReLU

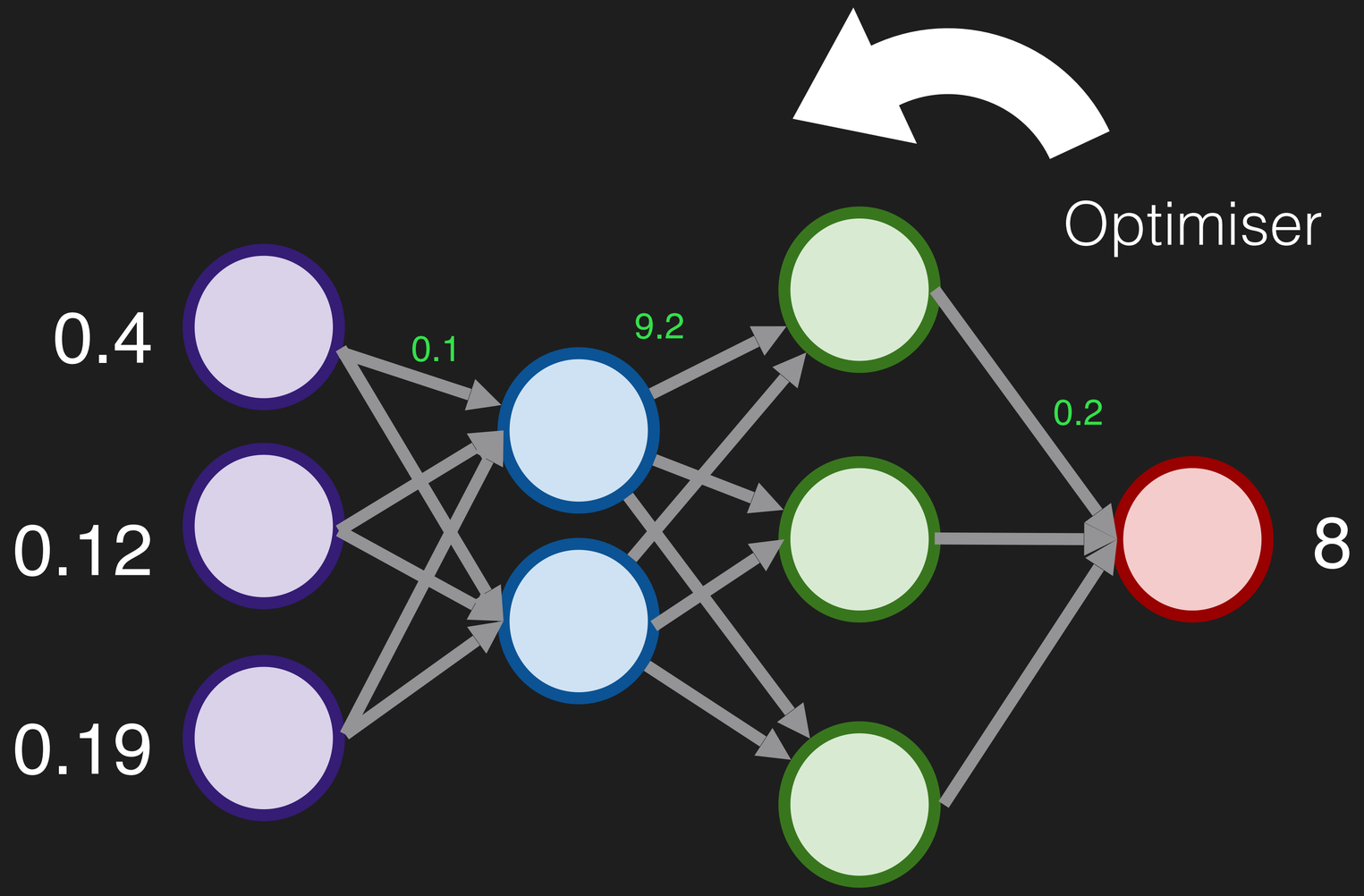
Artificial Neural Networks



Artificial Neural Networks

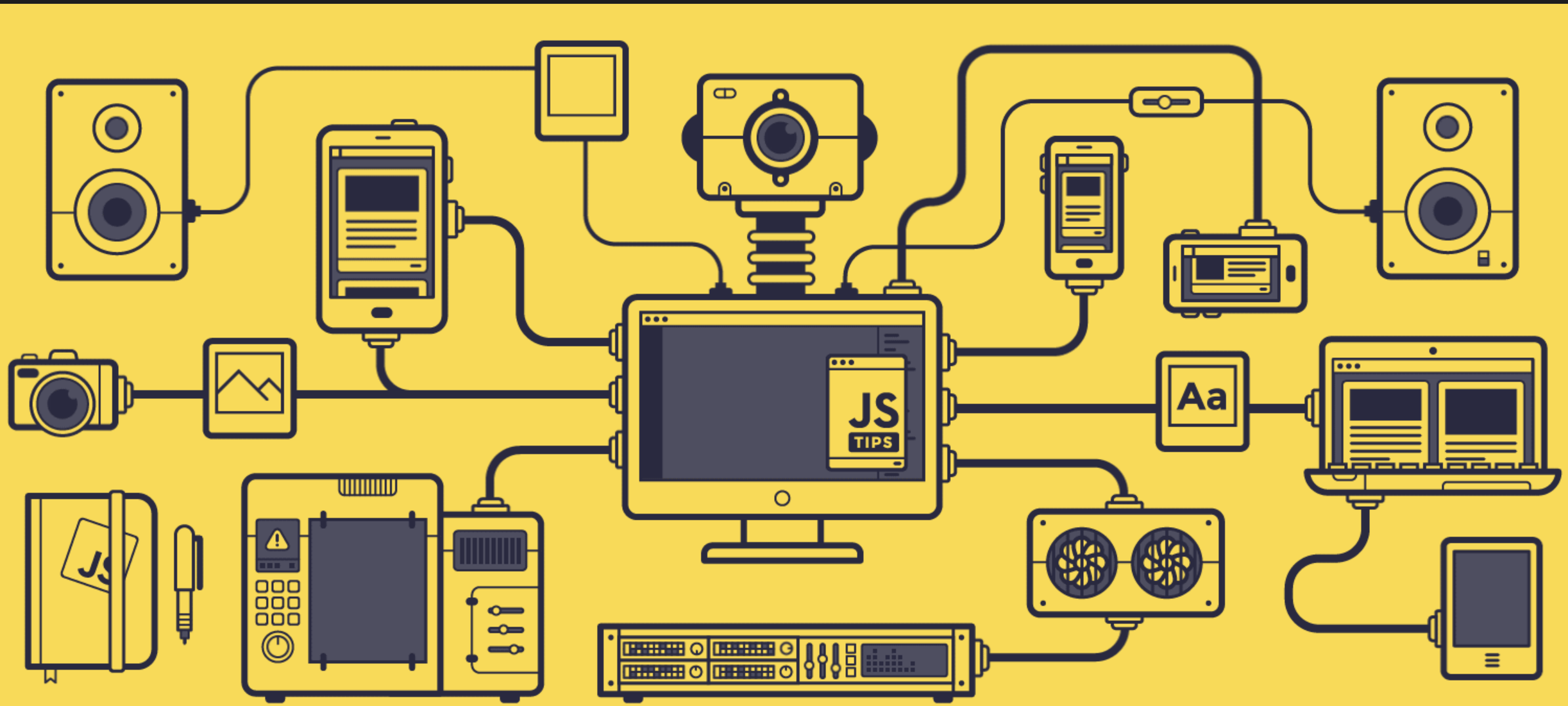


Artificial Neural Networks



Existing Landscape of ML


MACHINE
LEARNING
acer
acer
acer



http://www.jstips.co/en/about/

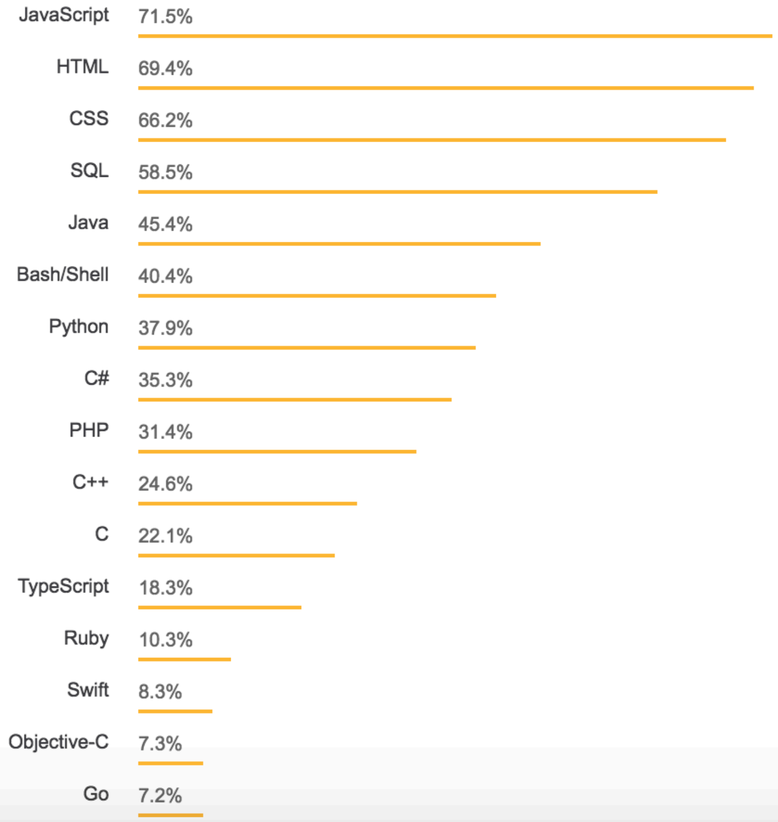
Why JavaScript?

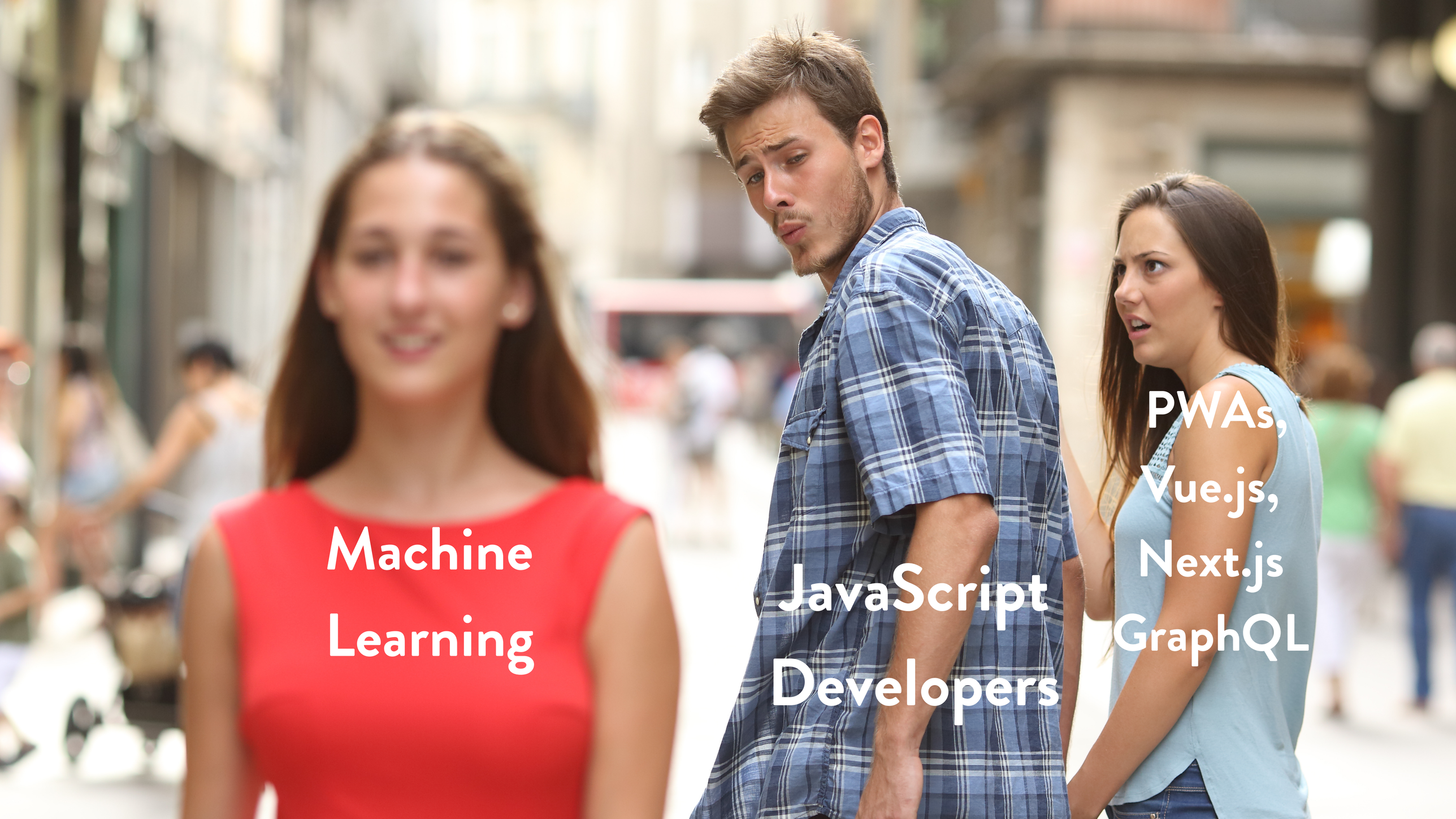
stackoverflow

Most Popular Technologies

Programming, Scripting, and Markup Languages

All Respondents | Professional Developers



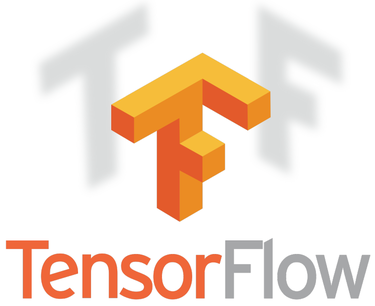

Machine
Learning
JavaScript
Developers
PWAs,
Vue.js,
Next.js
GraphQL


PYTORCH


ANACONDA


scikit
learn


Pandas


TensorFlow


jupyter


python

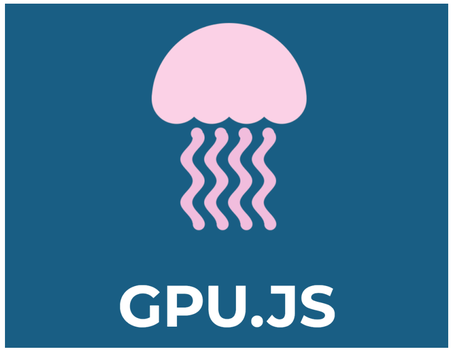

NumPy


K
Keras


theano


spaCy

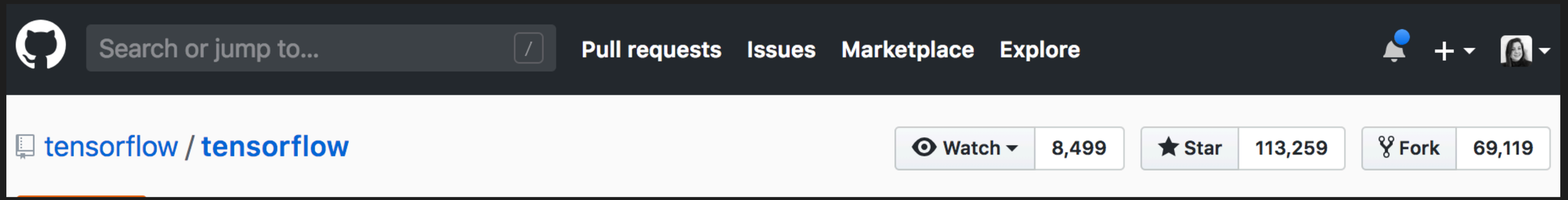
JavaScript and Machine Learning

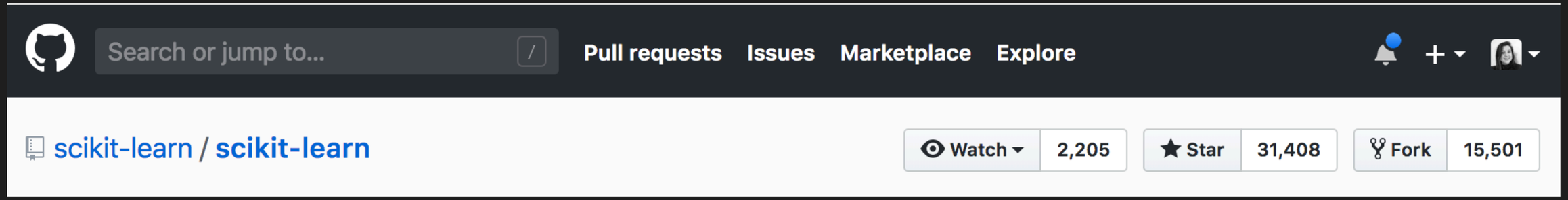


TensorFlow.js

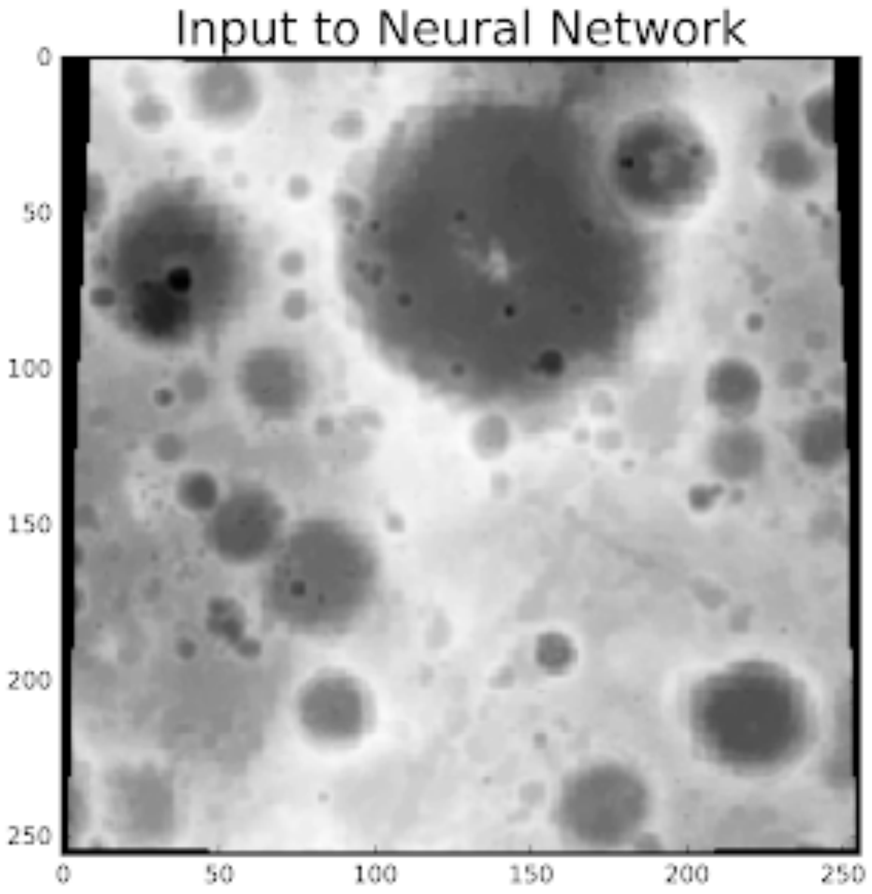
TensorFlow

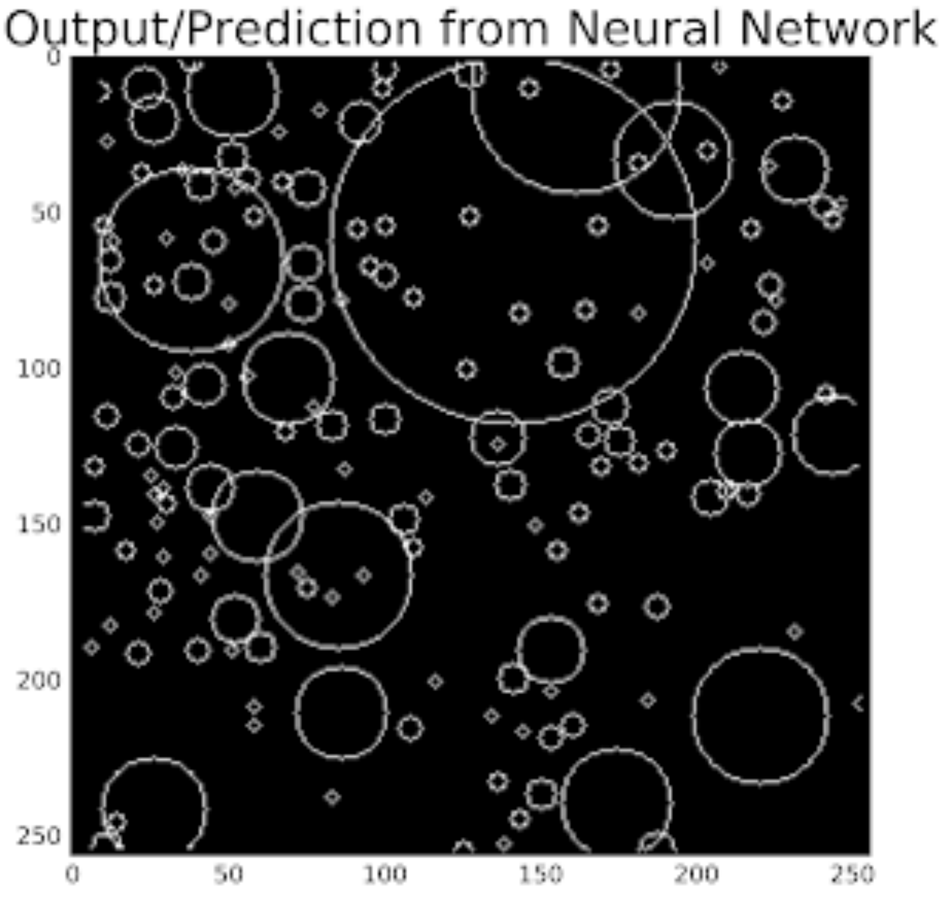
TensorFlow


Search or jump to...
/
Pull requests
Issues
Marketplace
Explore
tensorflow / tensorflow
Watch
8,499
Star
113,259
Fork
69,119


Search or jump to...
/
Pull requests
Issues
Marketplace
Explore
scikit-learn / scikit-learn
Watch
2,205
Star
31,408
Fork
15,501

TensorFlow

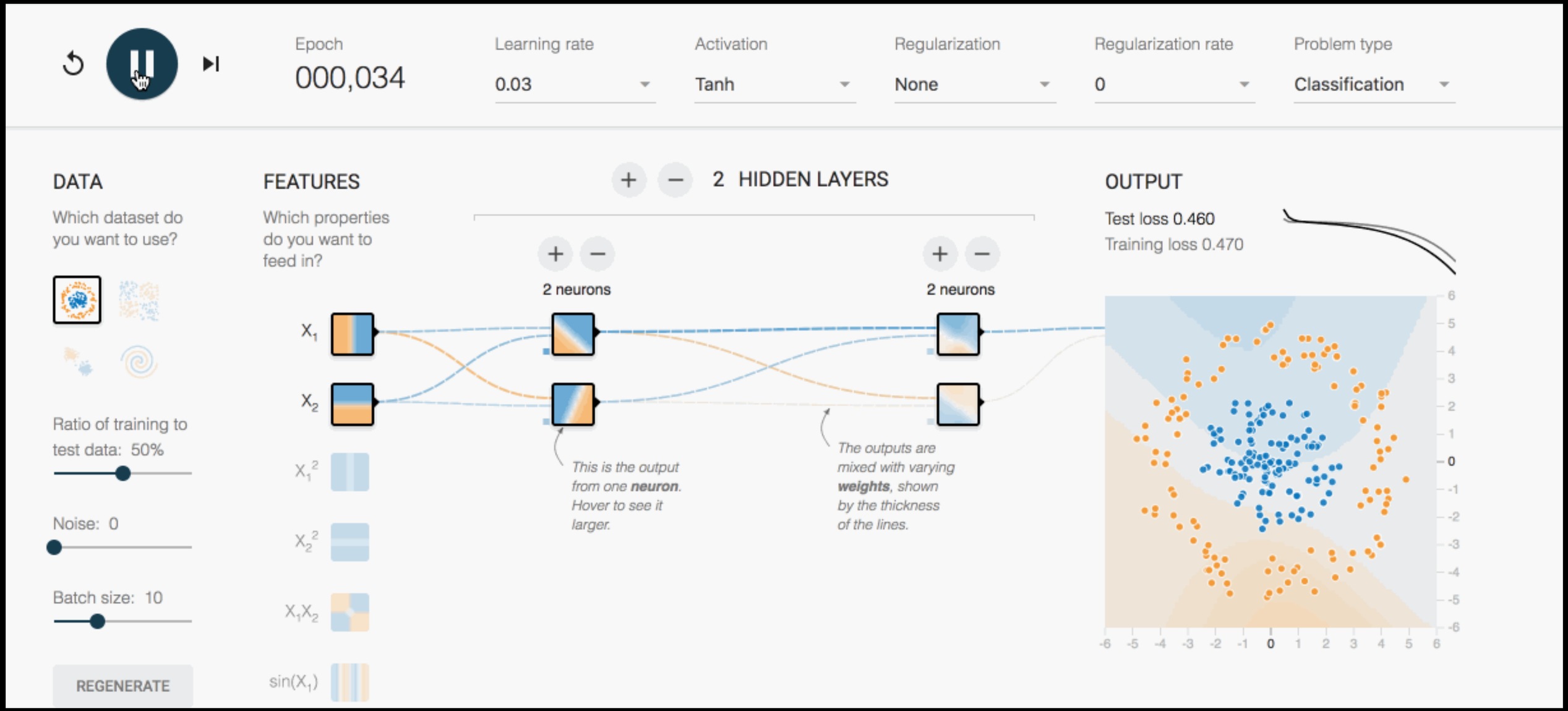

Input to Neural Network
0
50
100
150
200
250
0
50
100
150
200
250


Output/Prediction from Neural Network
0
50
100
150
200
250
0
50
100
150
200
250

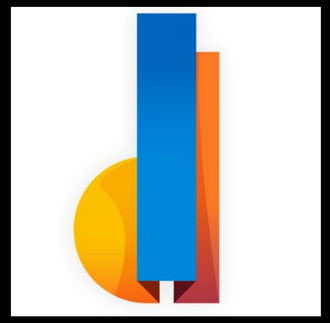
TensorFlow.js

TensorFlow Playground

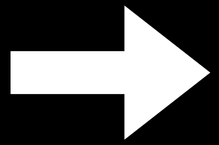
https://playground.tensorflow.org/



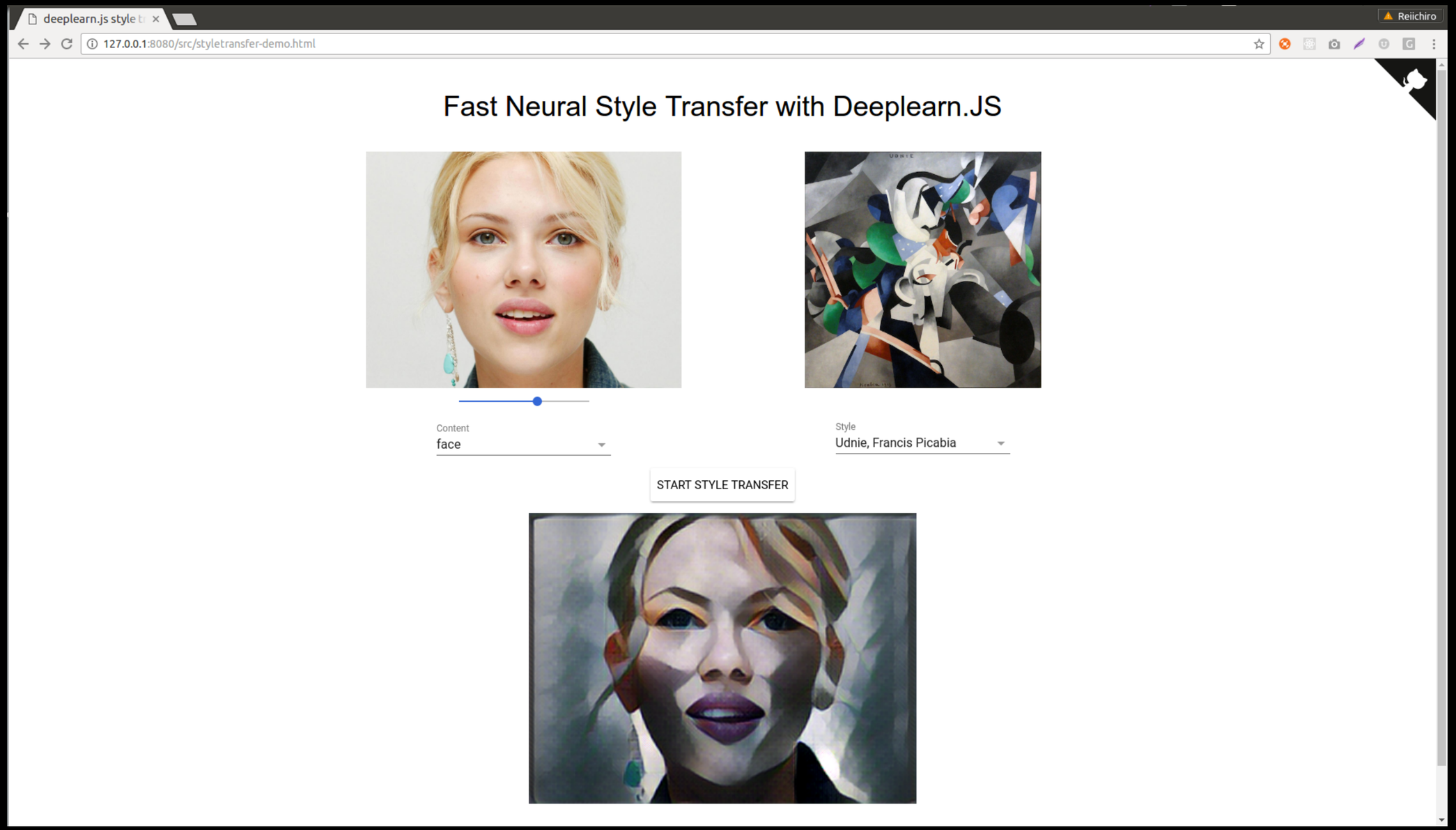
TensorFlow.js



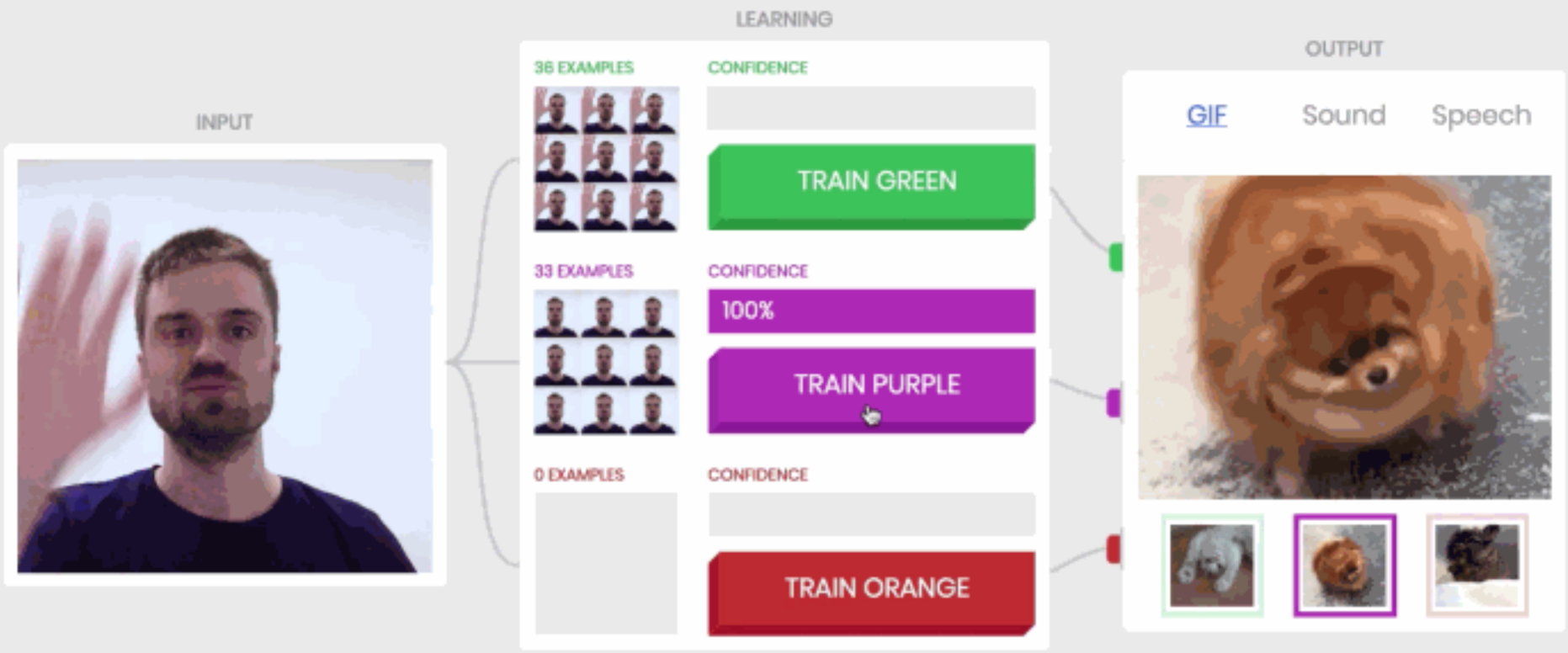
deeplearn.js



deeplearn.js & Style Transfer



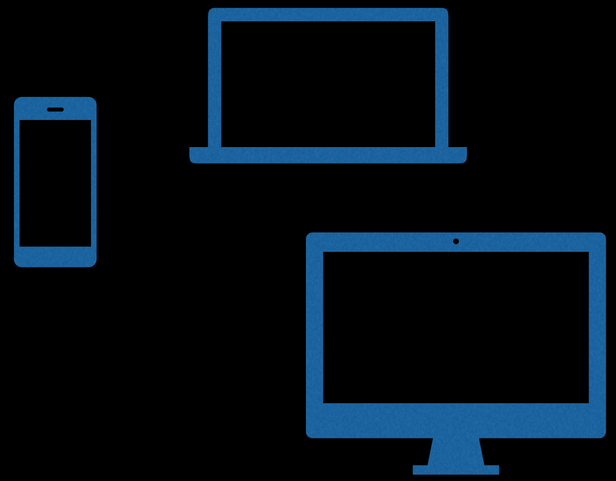
deeplearn.js & Teachable Machine



TensorFlow.js

✗ Drivers

✗ Install



TensorFlow.js

Create and Train Models

Load Pretrained Models

Retrain Existing Models

aijs.rocks

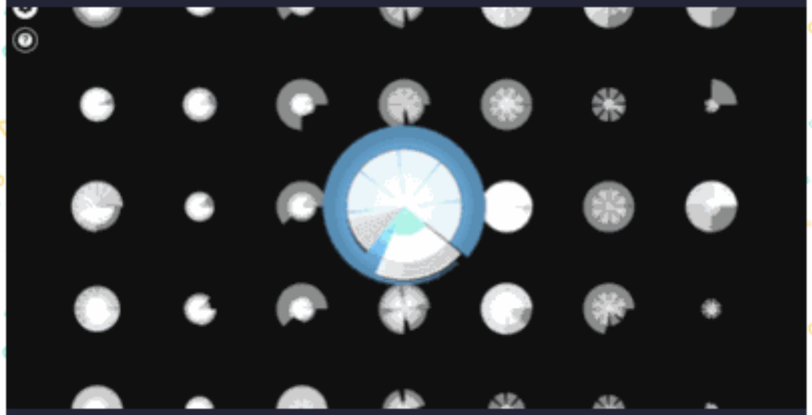
A curated collection of inspirational AI-powered JavaScript apps

An interactive drum machine created with love for Music and AI.

Magenta.js | Tone.js | Magenta DrumRNN

Gogul Ilango — September 02, 2018

Musical Spinners From Latent Space



An AI-powered interactive music experience

TensorFlow.js | Magenta.js | Three.js

Automatically generated emoji captions for your images

Azure | Computer Vision

Alex Saunders — August 30, 2018

Image2Image

Make auto-generated images out of simple line drawings

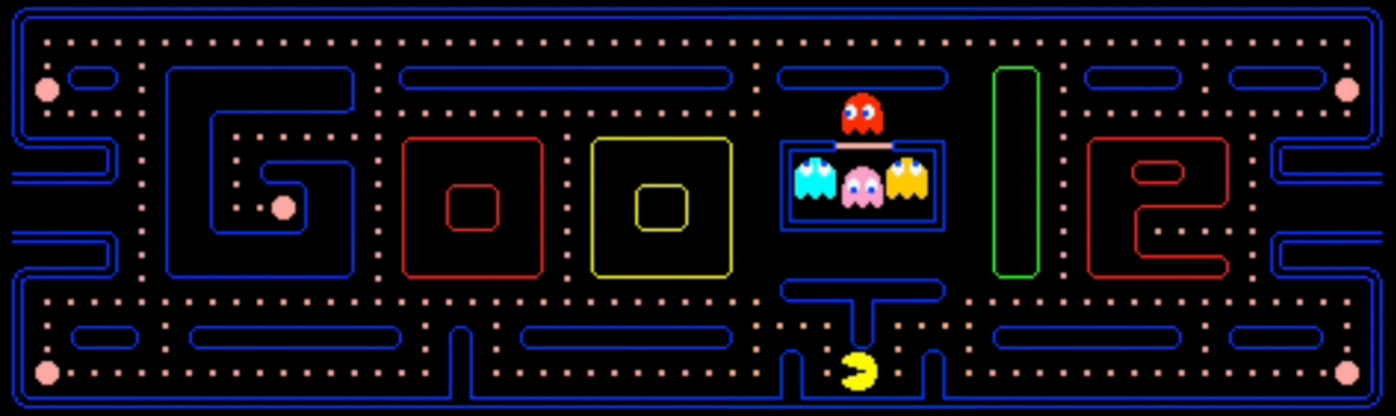
TensorFlow.js | Pix2Pix

About

Inspire

Submit

1UP



TRAIN MODEL

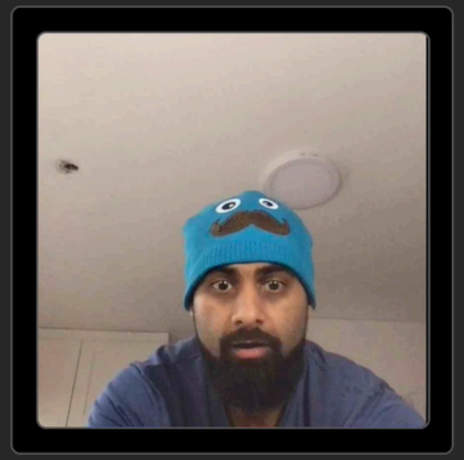
PLAY

Learning rate
0.0001

Batch size
0.4

Epochs
20

Hidden units
100



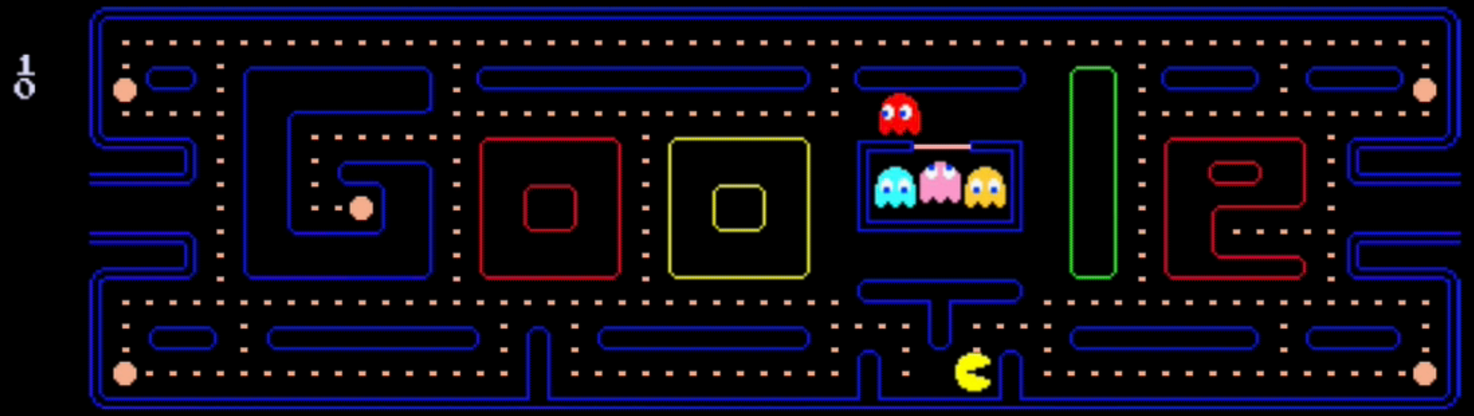
Click to add the current camera view as an example for that control

Add Sample
0 examples

Add Sample
0 examples

Add Sample
0 examples

Add Sample
0 examples



LOSS: 0.00000

PLAY

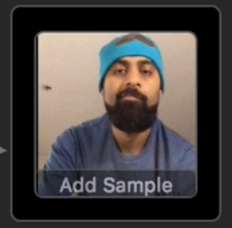
Learning rate
0.0001

Batch size
0.4

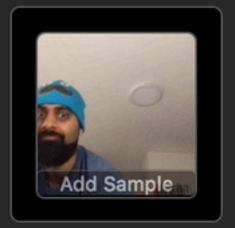
Epochs
20

Hidden units
100

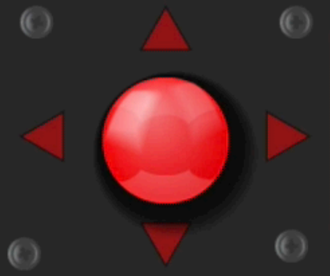
Click to add the current camera view as an example for that control

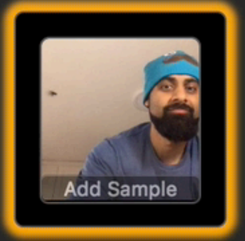


27 examples

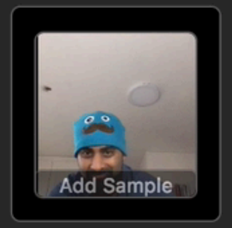


26 examples





20 examples



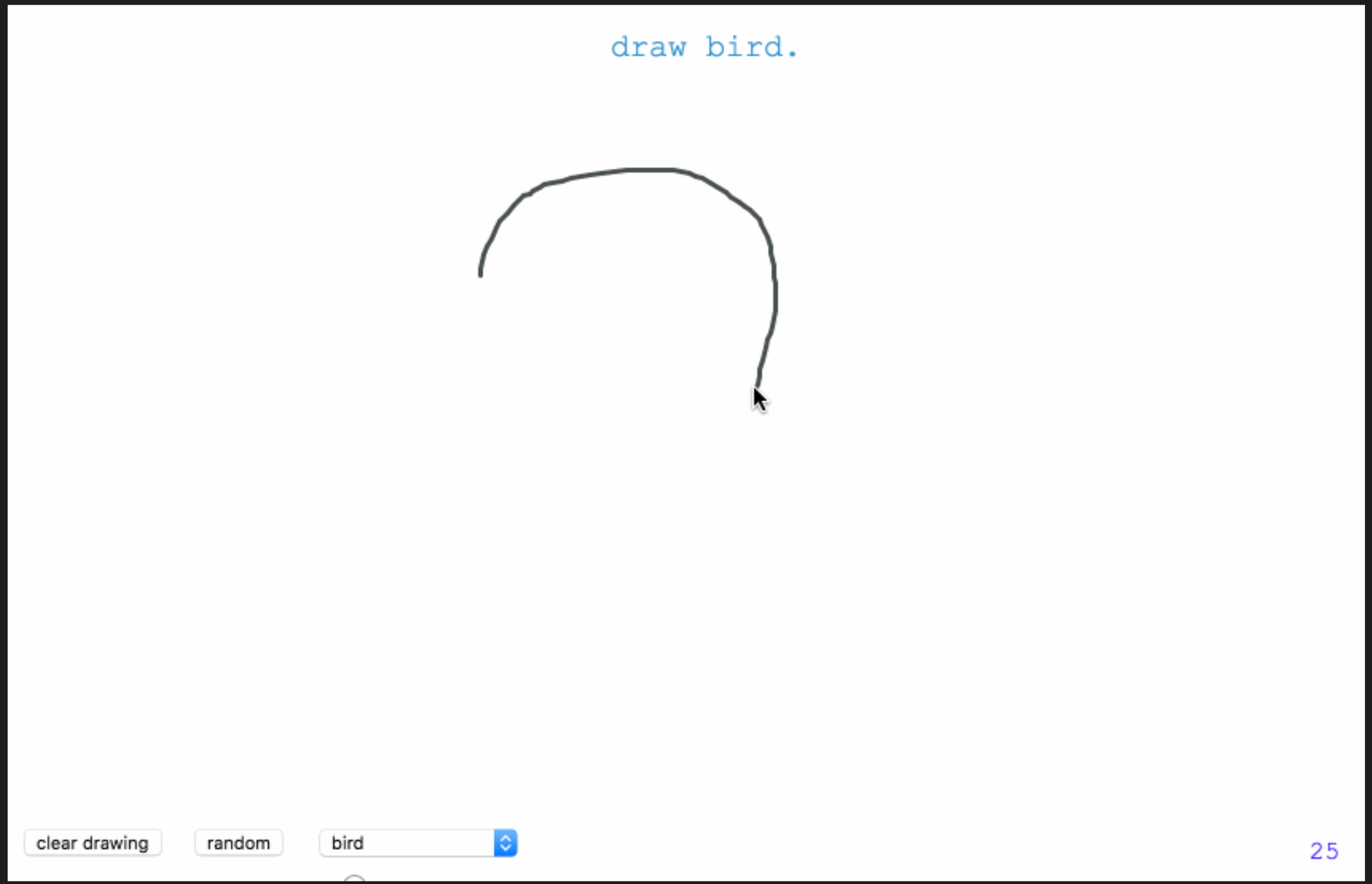
20 examples

MOVE MIRROR

Make a GIF

MOVE MIRROR

Sketch RNN



magenta.js

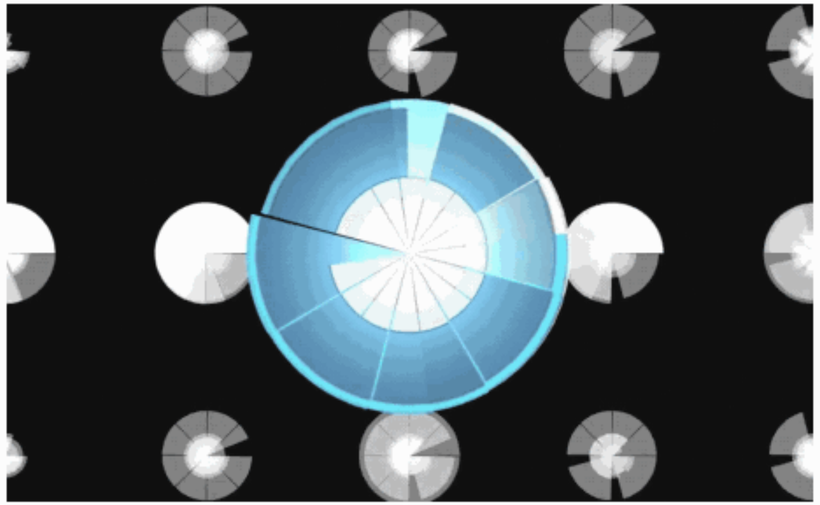
magenta

Home Blog Demos Publications Datasets Discuss GitHub

Featured

The Incredible Musical Spinners From Latent Space [play | blog | code]

Tero Parviainen (, teropa , teropa)



An "AI-powered interactive music experience" that presents a playable 7x7 grid of musical measures. Each row is an interpolation between two endpoints and each column is the "same" measure interpreted in 7 different chords, e.g. C Major. You can choose the key to play in.

TensorFlow.js

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

TensorFlow.js

```
<html>
  <head>
    <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.10.0"> </script>

    <script>
        const x = tf.tensor([1,2,3]);
        tf.print(x);    // Tensor [1, 2, 3]
    </script>
  </head>
</html>
```

TensorFlow.js

Core API 

Layers API 

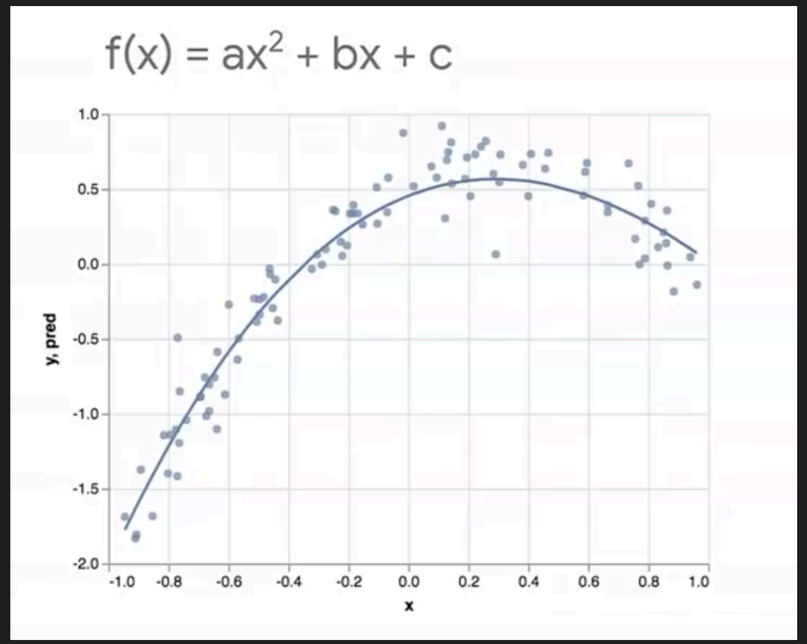
TensorFlow.js

```
const shape = [2, 3]; // 2 rows, 3 columns
const a = tf.tensor([1.0, 2.0, 3.0, 10.0, 20.0, 30.0], shape);
a.print();
// Output: [[1 , 2 , 3 ],
//          [10, 20, 30]]

// The shape can also be inferred:
const b = tf.tensor([[1.0, 2.0, 3.0], [10.0, 20.0, 30.0]]);
b.print();
// Output: [[1 , 2 , 3 ],
//          [10, 20, 30]]
```

Core API

TensorFlow.js

Polynomial Regression




```
// Define function
function predict(input) {
  // y = a * x ^ 2 + b * x + c
  // More on tf.tidy in the next section
  return tf.tidy(() => {
    const x = tf.scalar(input);

    const ax2 = a.mul(x.square());
    const bx = b.mul(x);
    const y = ax2.add(bx).add(c);

    return y;
  });
}

// Define constants: y = 2x^2 + 4x + 8
const a = tf.scalar(2);
const b = tf.scalar(4);
const c = tf.scalar(8);

// Predict output for input of 2
const result = predict(2);
result.print() // Output: 24
```

Layers API

TensorFlow.js

```
const model = tf.sequential();
model.add (
  tf.layers.dense({
    units: 20,
    activation: 'relu',
  }),
});

const optimizer = tf.train.sgd(LEARNING_RATE);
model.compile({optimizer, loss: 'meanSquaredError'});
model.fit({x: data, y: labels});
```

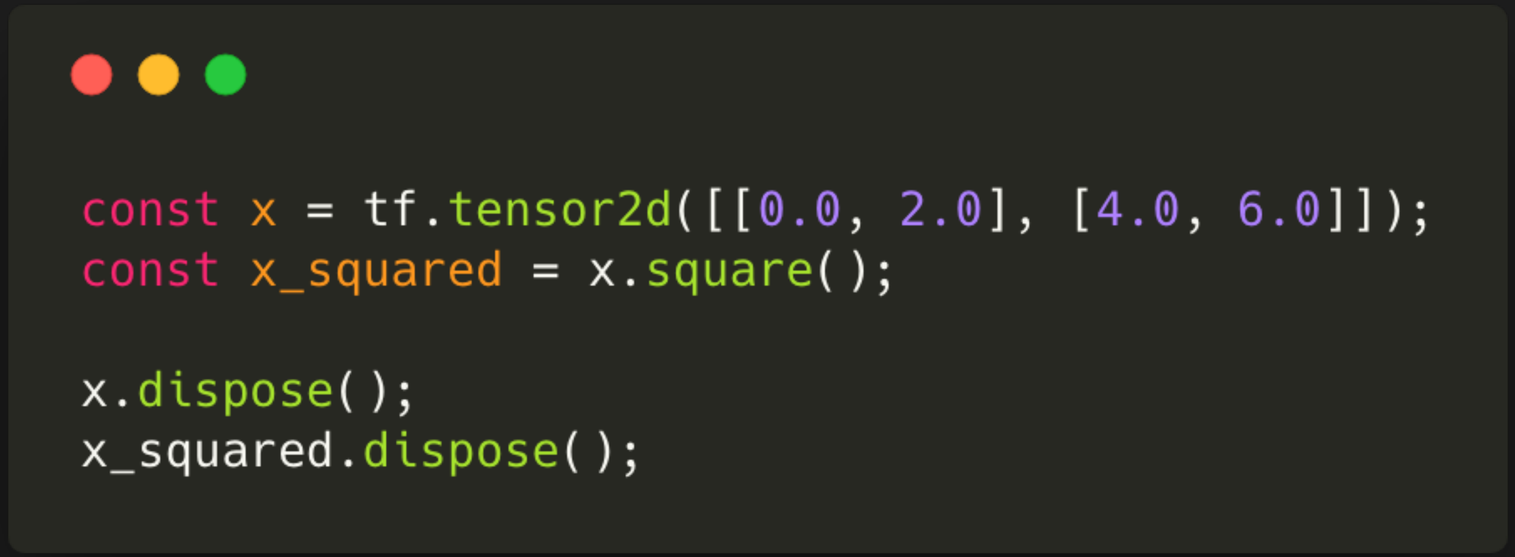
TensorFlow.js

Memory Management

```
// tf.tidy takes a function to tidy up after
const average = tf.tidy(() => {
  // tf.tidy will clean up all the GPU memory used by tensors inside
  // this function, other than the tensor that is returned.
  //
  // Even in a short sequence of operations like the one below, a number
  // of intermediate tensors get created. So it is a good practice to
  // put your math ops in a tidy!
  const y = tf.tensor1d([1.0, 2.0, 3.0, 4.0]);
  const z = tf.ones([4]);

  return y.sub(z).square().mean();
});

average.print() // Output: 3.5
```

TensorFlow.js

Memory Management



```
const x = tf.tensor2d([[0.0, 2.0], [4.0, 6.0]]);
const x_squared = x.square();

x.dispose();
x_squared.dispose();
```

TensorFlow.js

Load pre-trained models

```
const mobilenet = await tf.loadModel(
      'https://storage.googleapis.com/tfjs-models/tfjs/mobilenet_v1_0.25_224/model.json');
```

TensorFlow.js

Transfer Learning

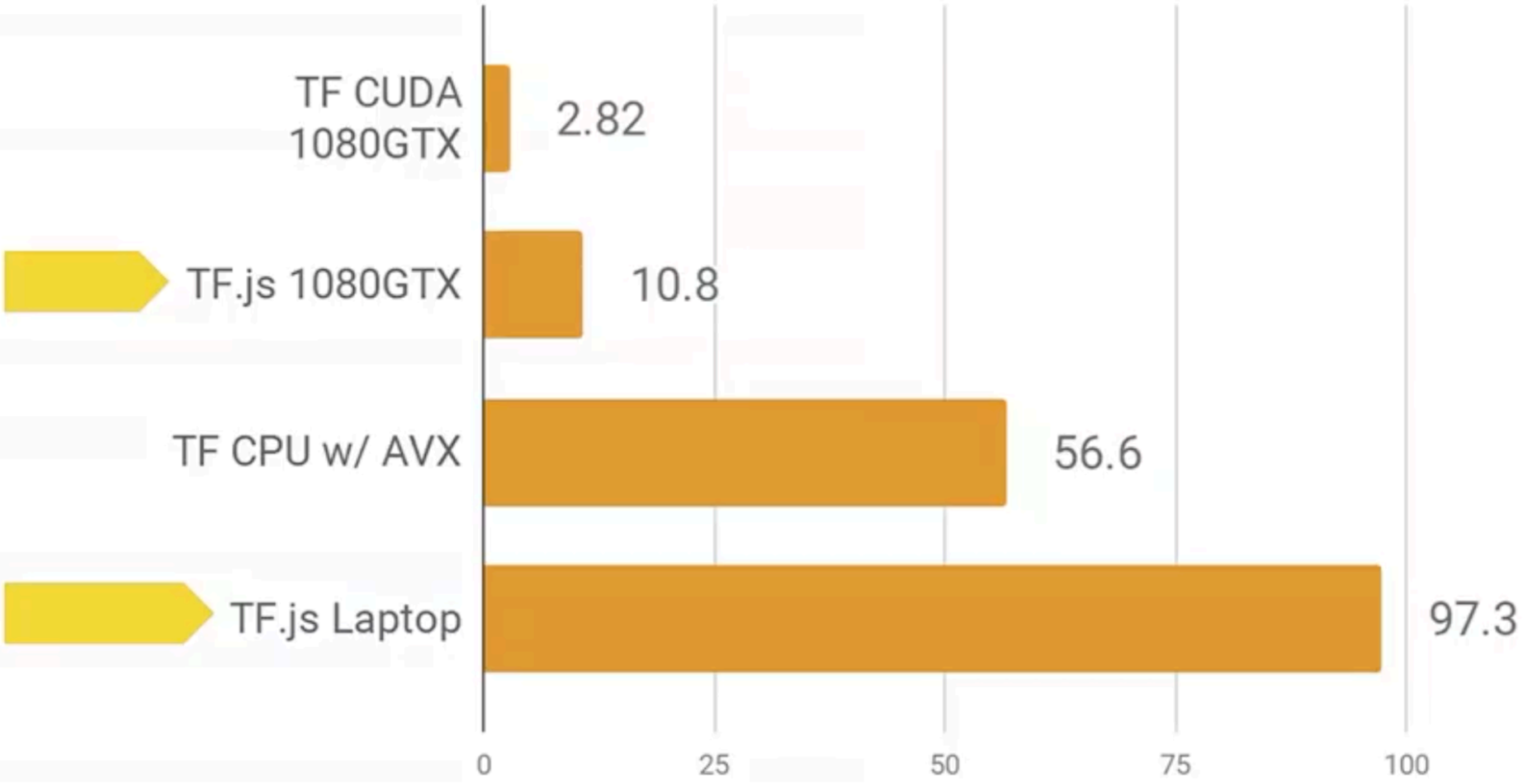
```
async function loadMobilenet() {
  const mobilenet = await tf.loadModel(
      'https://storage.googleapis.com/tfjs-models/tfjs/mobilenet_v1_0.25_224/model.json');

  // Return a model that outputs an internal activation.
  const layer = mobilenet.getLayer('conv_pw_13_relu');
  return tf.model({inputs: model.inputs, outputs: layer.output});
});
```

TensorFlow.js

Create and Train Models

Load Pretrained Models

Retrain Existing Models



Inference Time (ms) of MobileNet 1.0_224

Average of 200 runs

Applications of ML with JavaScript

Getting Started

Getting Started & Resources

js.tensorflow.org

github.com/tensorflow/tfjs

github.com/tensorflow/tfjs-examples

aijs.rocks

ml5.js

The Coding Train

The Future of ML & JavaScript