HUGE

HoliMDE.

Image recognition in the browser using Tensorflow.js

Aug 13, 2019

Agenda.

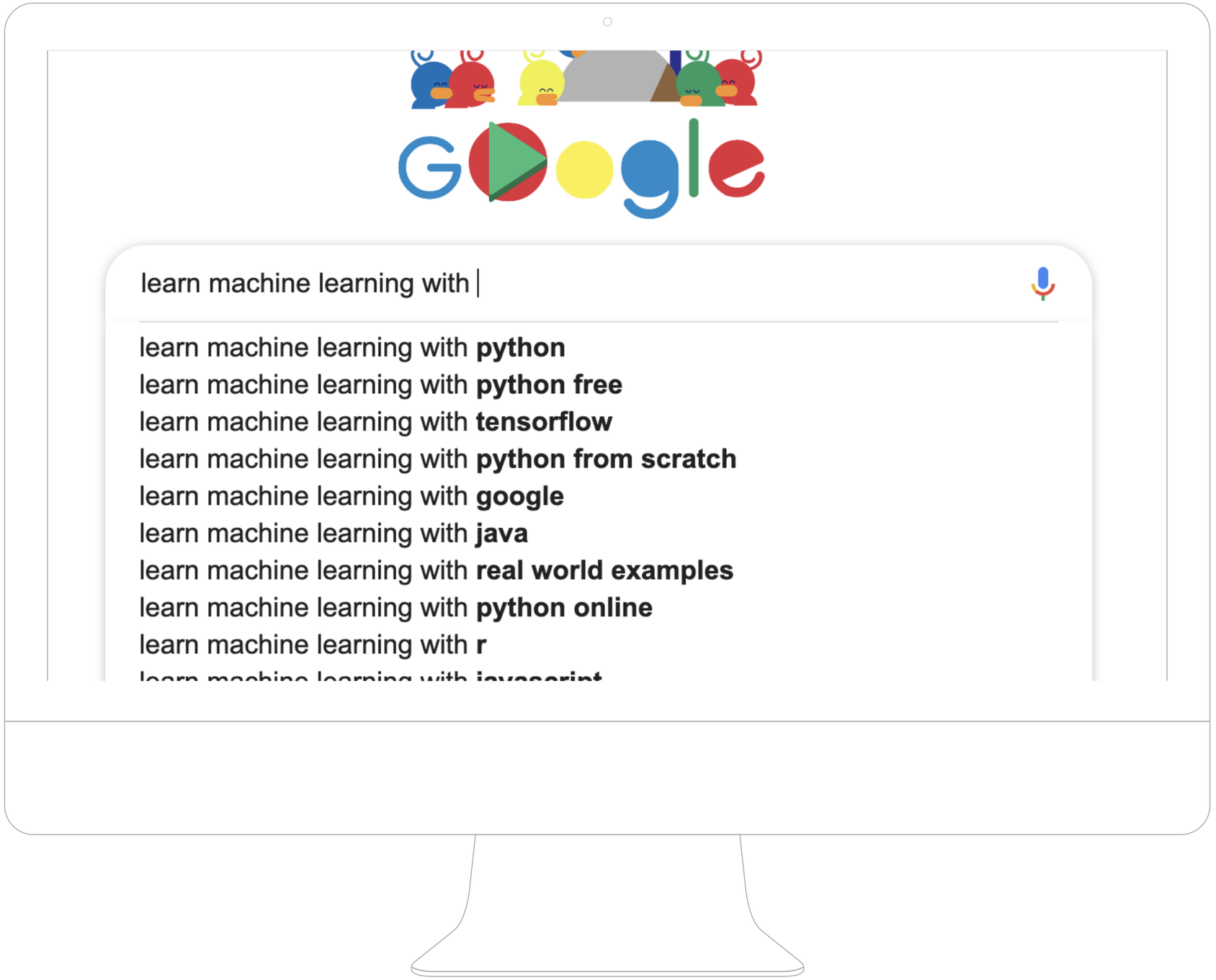
1. Motivation.
2. Neural Networks.
3. Convolutional Neural Networks.
4. Tensorflow.
5. Tensorflow.js.
6. Examples:
   - From scratch
   - Pre-trained models.
   - API

1

Motivation

Motivation

Social Networks

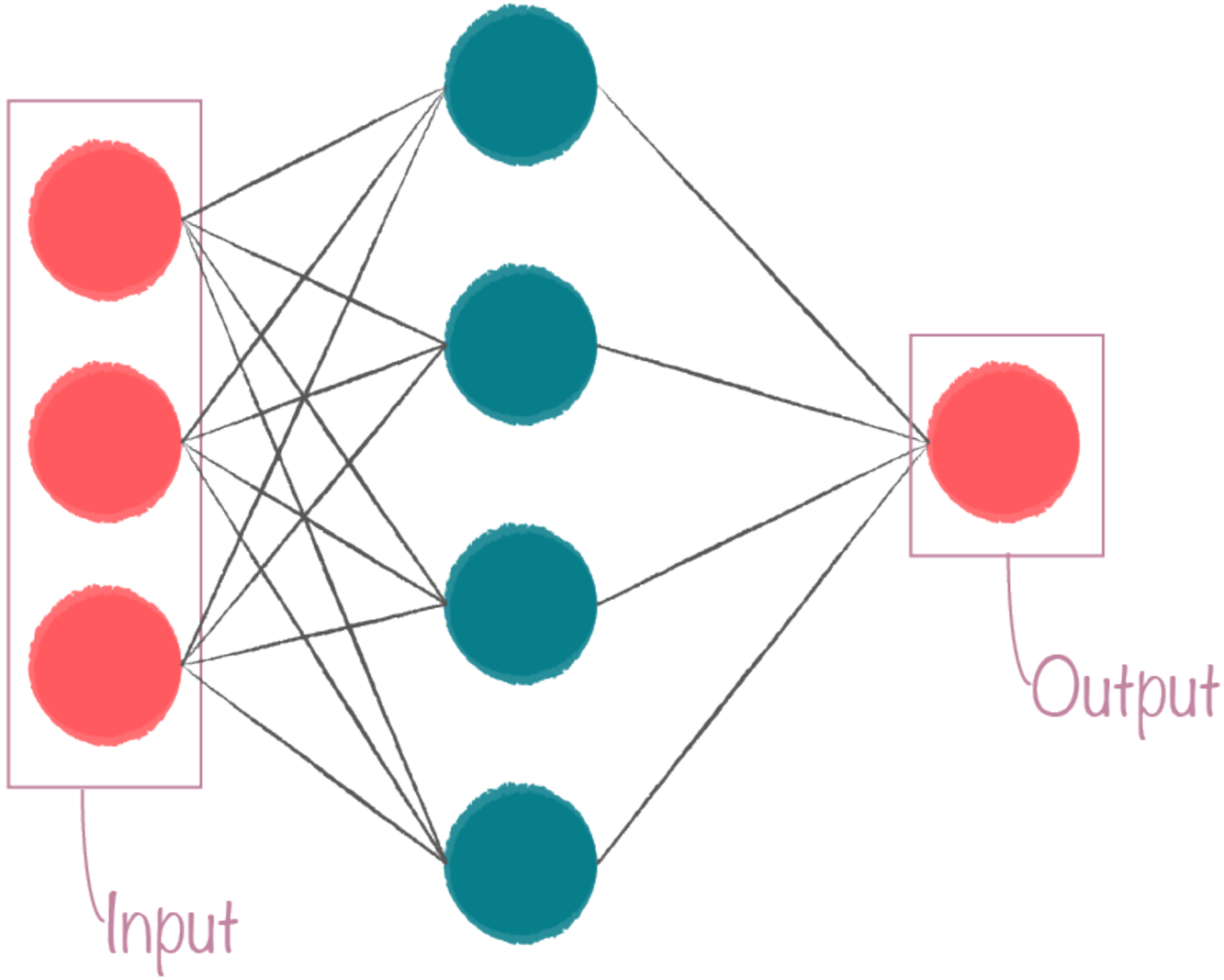

Google
learn machine learning with
learn machine learning with python
learn machine learning with python free
learn machine learning with tensorflow
learn machine learning with python from scratch
learn machine learning with google
learn machine learning with java
learn machine learning with real world examples
learn machine learning with python online
learn machine learning with r

2

Neural Networks.

Neural Networks.

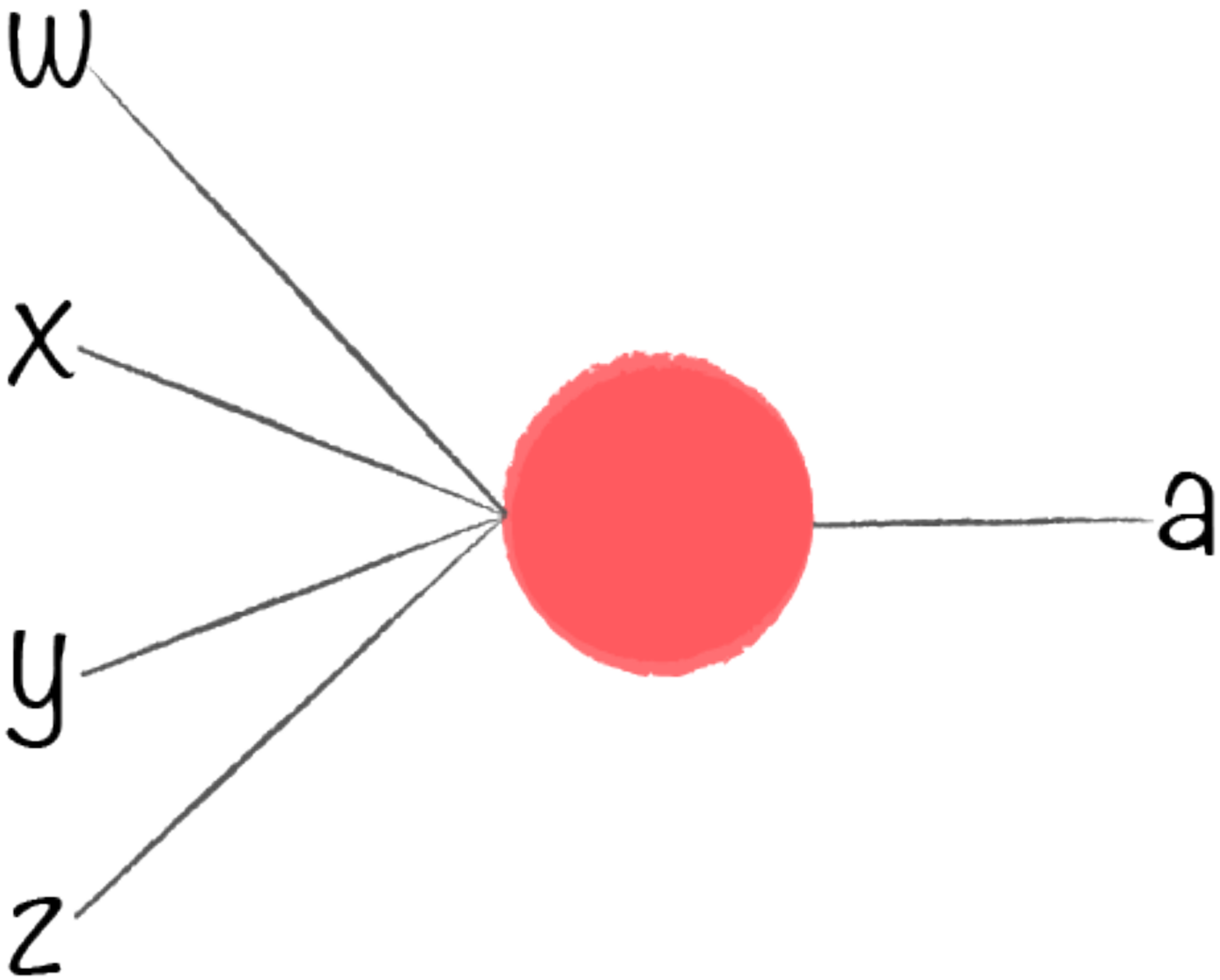
Computational models that emulate the behavior of the human brain.



Neuron

Building blocks.

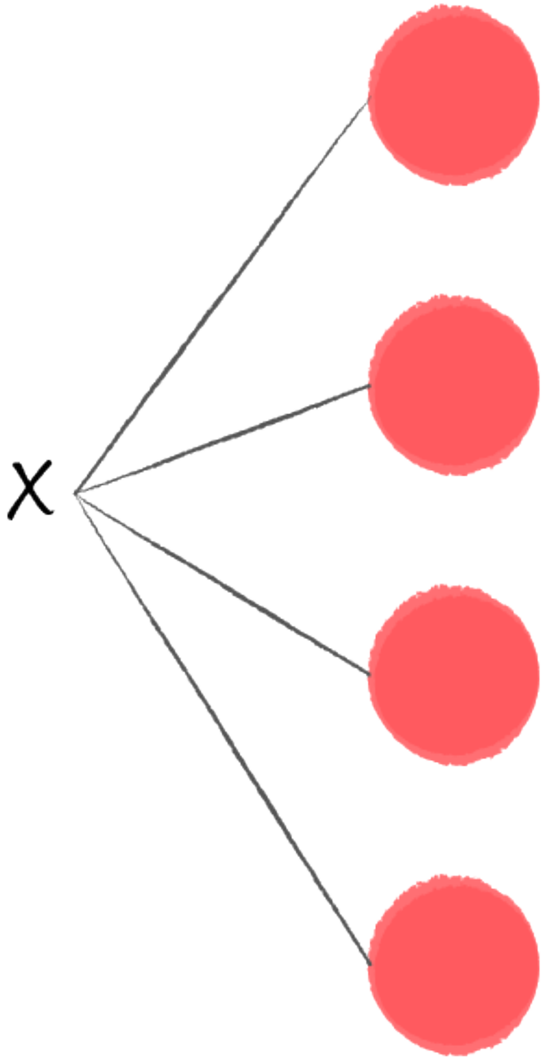
Takes inputs, does some math with them, and produces one output.



Layer

A bunch of neurons connected.

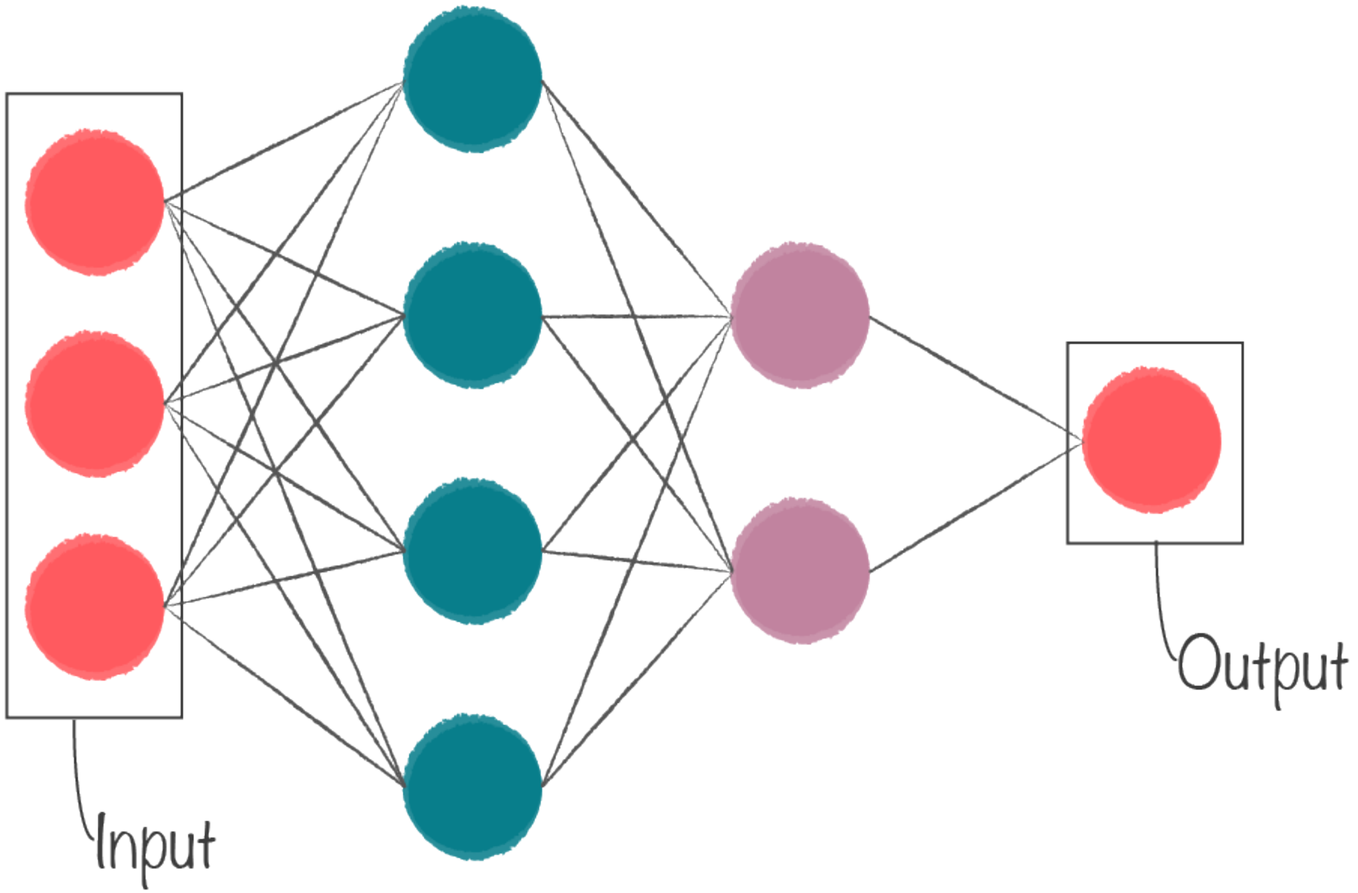
Can recognize simple patterns.



Network

A bunch layers connected.

Can recognize complex patterns.



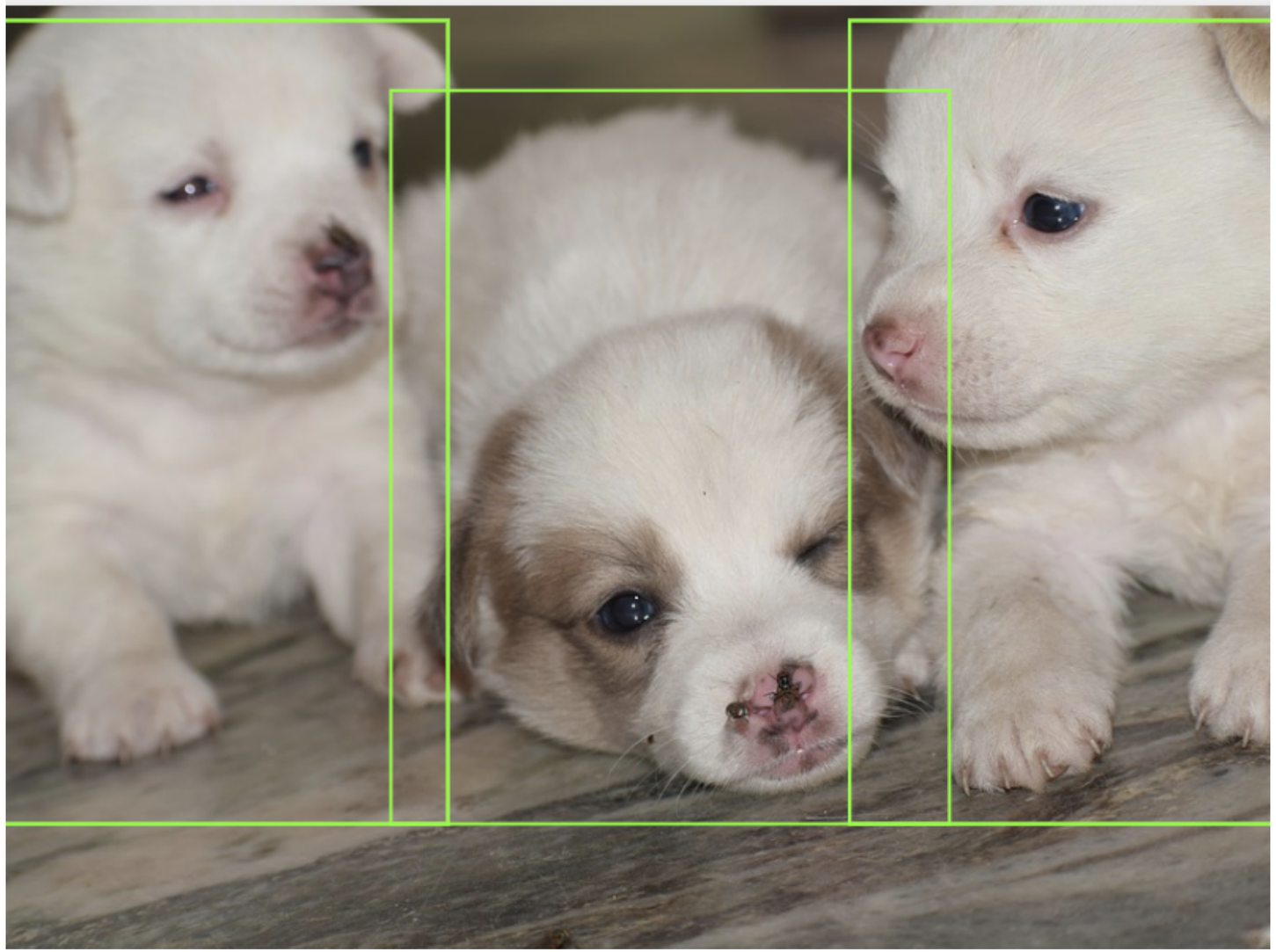
3

Convolutional Neural Networks

Convolutional Neural Networks.

They are used for image detection and recognition.

The CNN assume that the inputs are images, in this way, they can optimize the calculations.

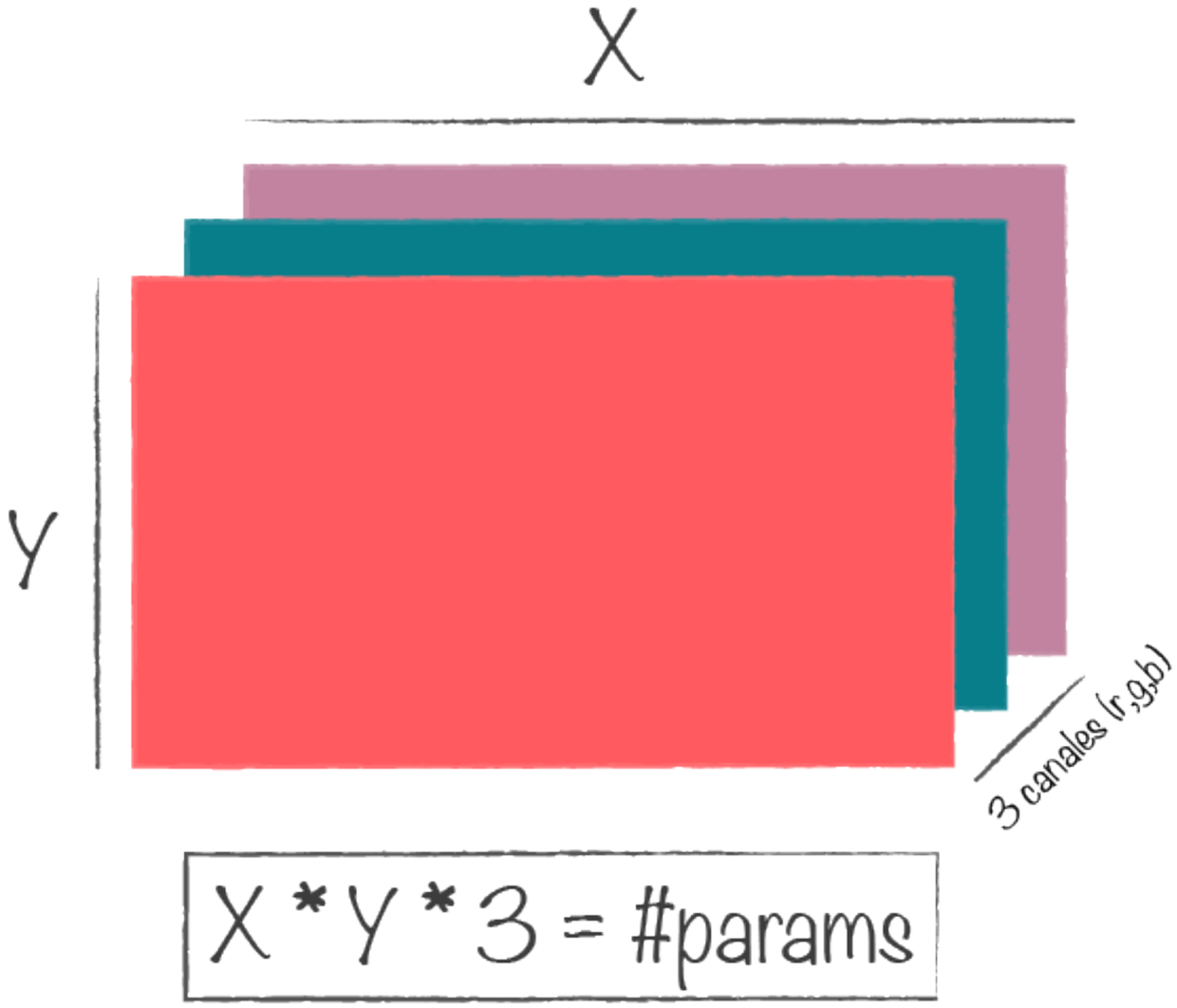


Why is this necessary?

Image = Pixel matrix

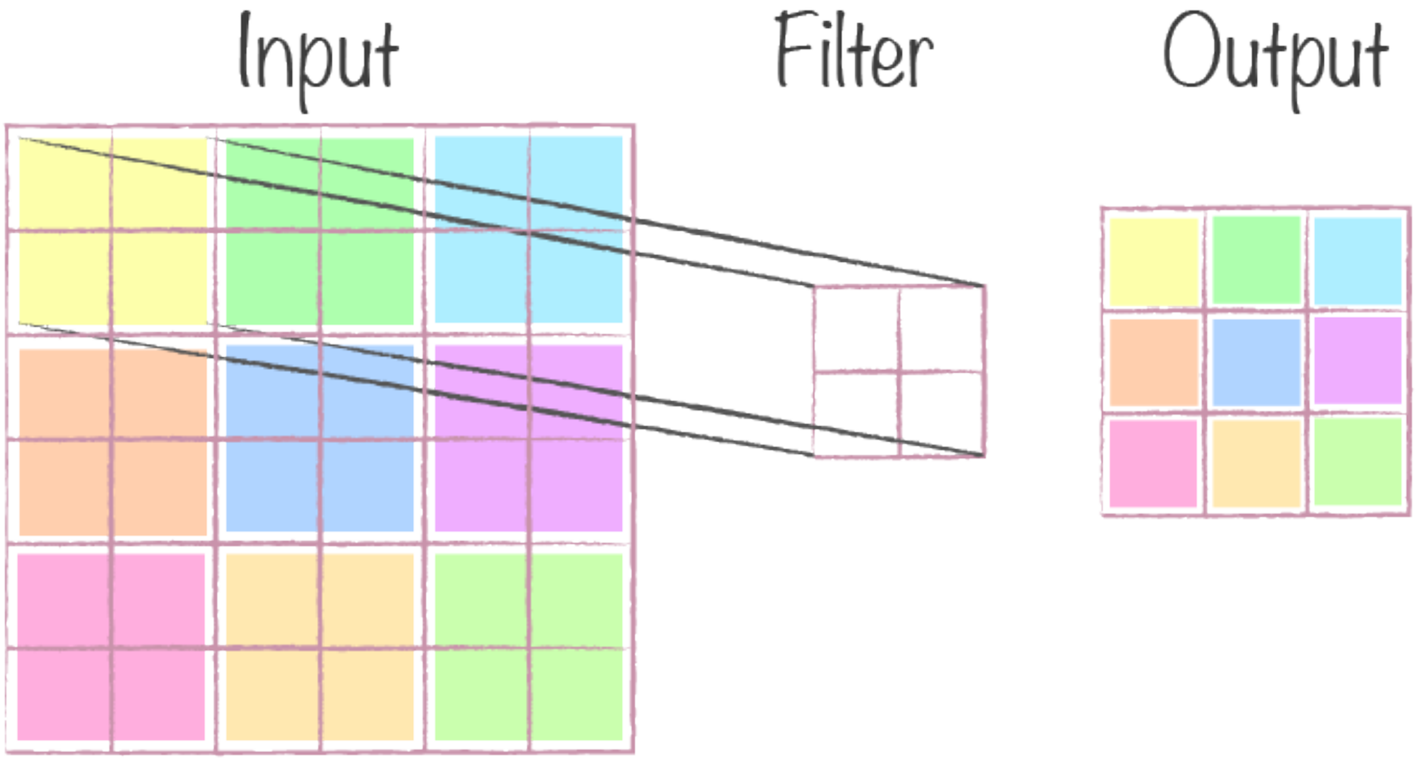
32*32*3 = 3072

2048*1536*3 = 9.437.814



Layers

Convolution

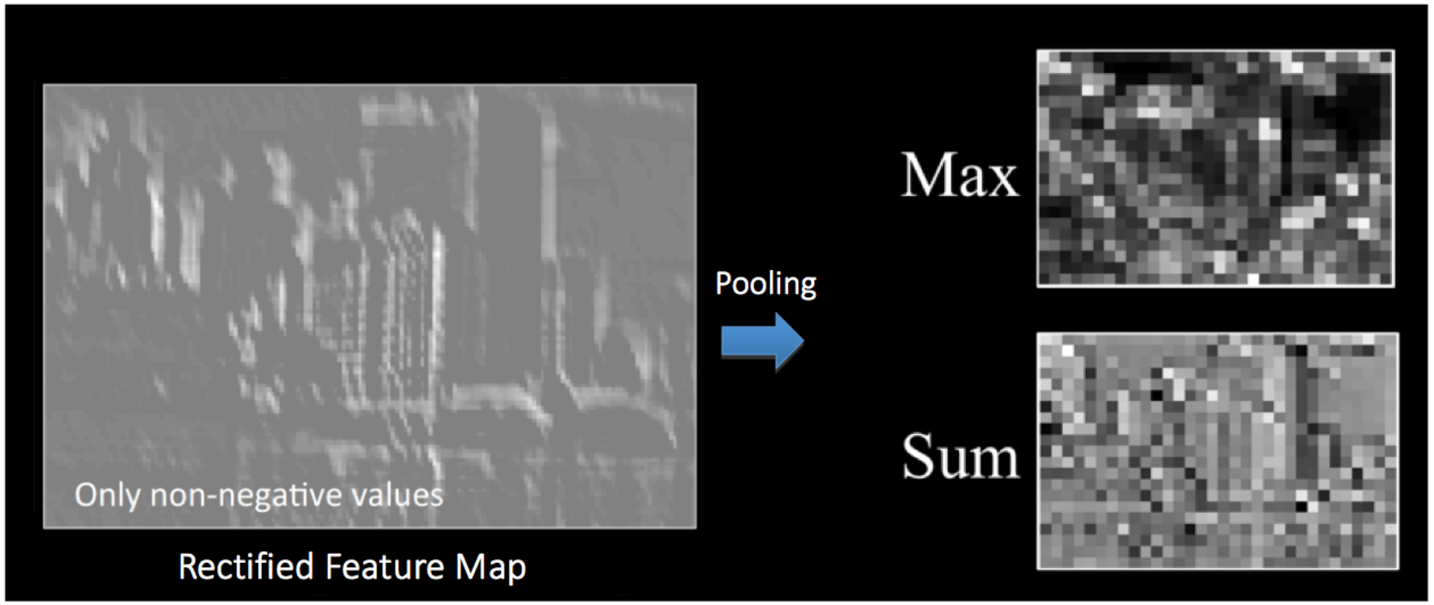


Convolution

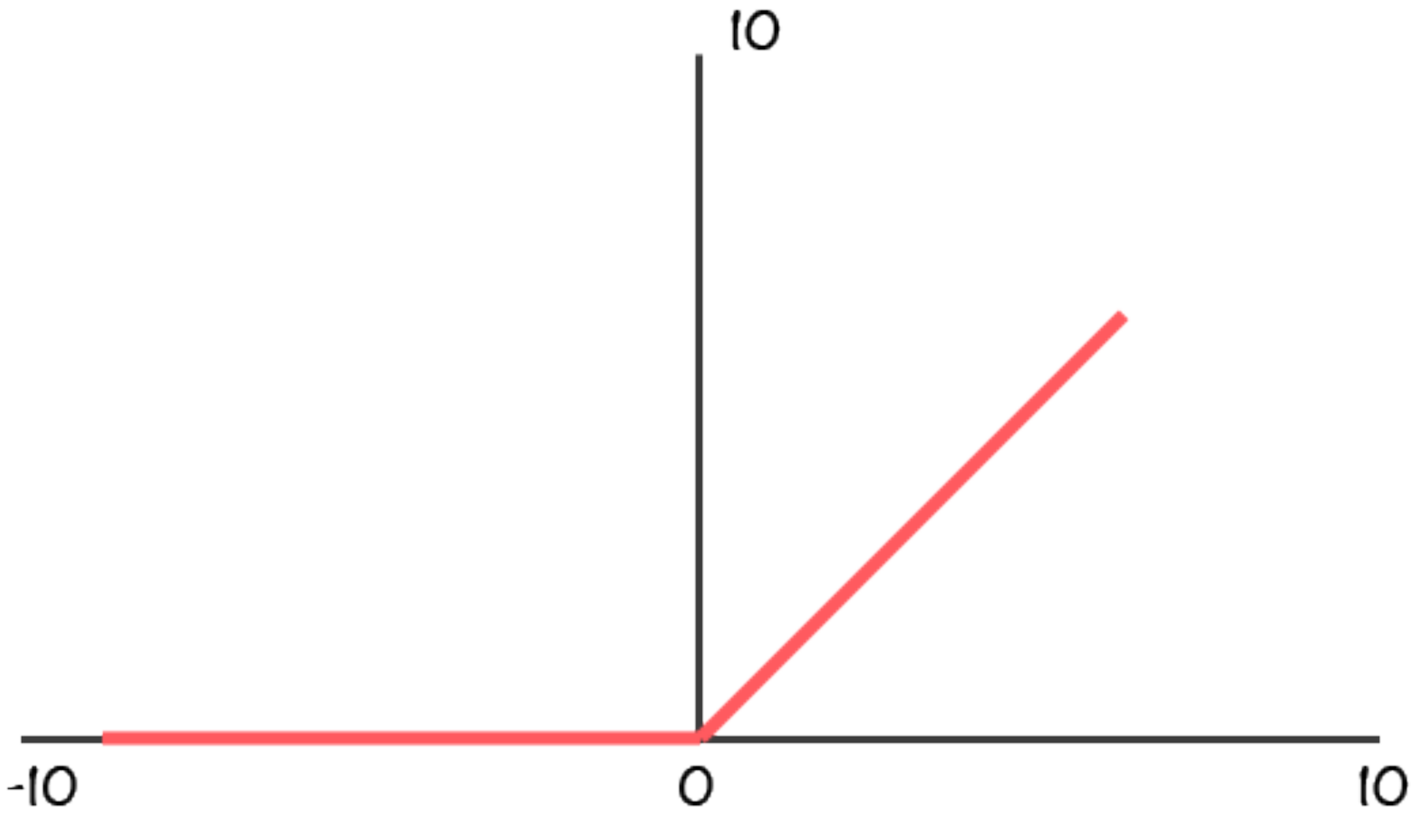
Pooling

Convolutional Neural Networks - Layers.

Pooling



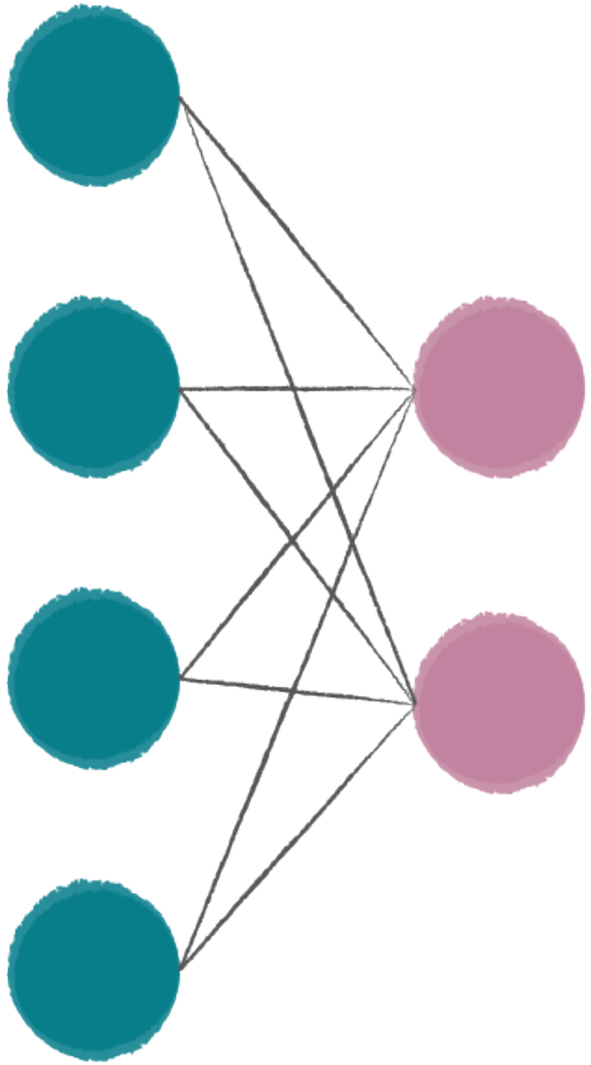
Activation



Activation

Convolutional Neural Networks - Layers.

Fully connected



Architectures

ImageNet

- ~14 million of labeled images.
- 20 thousand classes.
- ImageNet Challenge on Kaggle

Architectures

- AlexNet (2012)
- Inception - GoogLeNet (2014)
- VGG (2014)
- ResNet (2015)

4

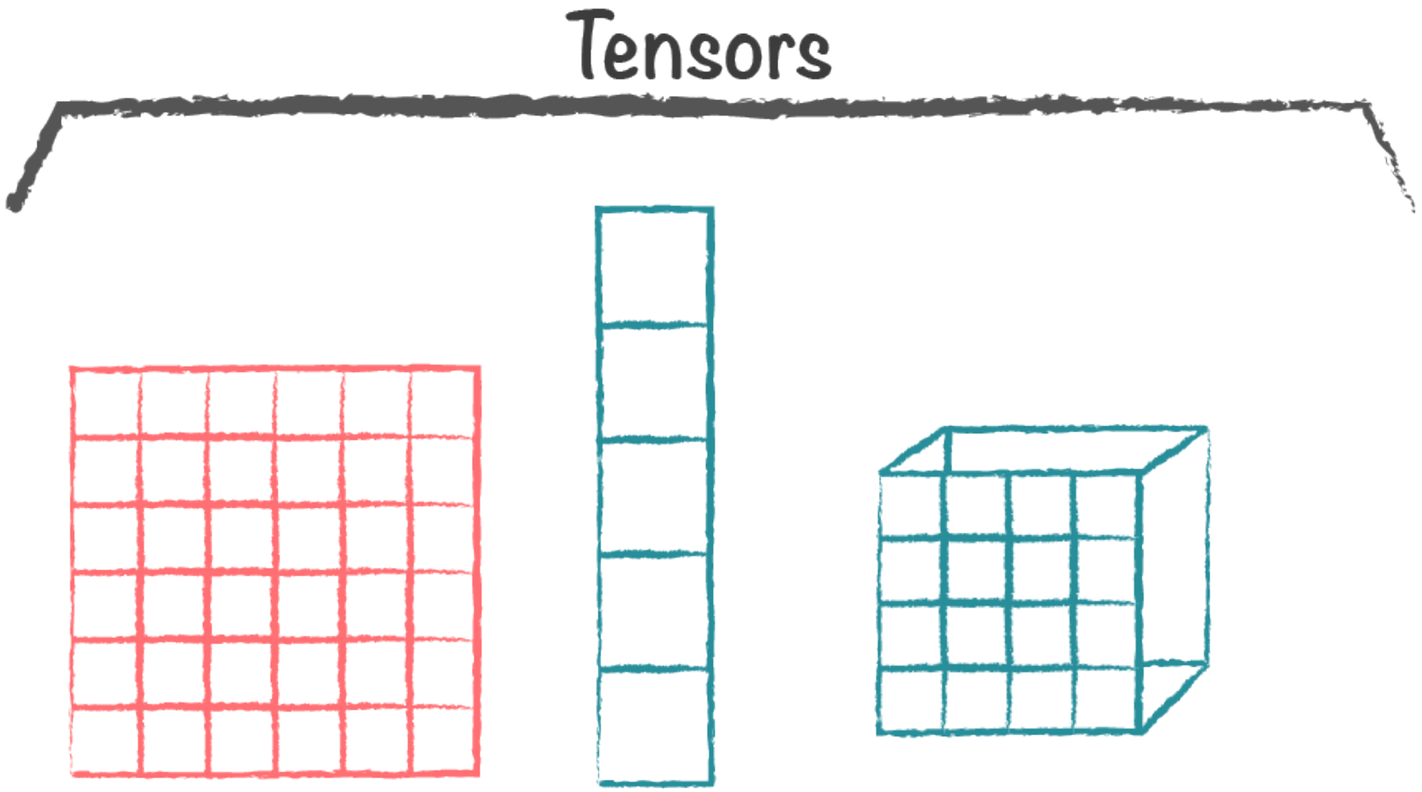
TensorFlow

TensorFlow.

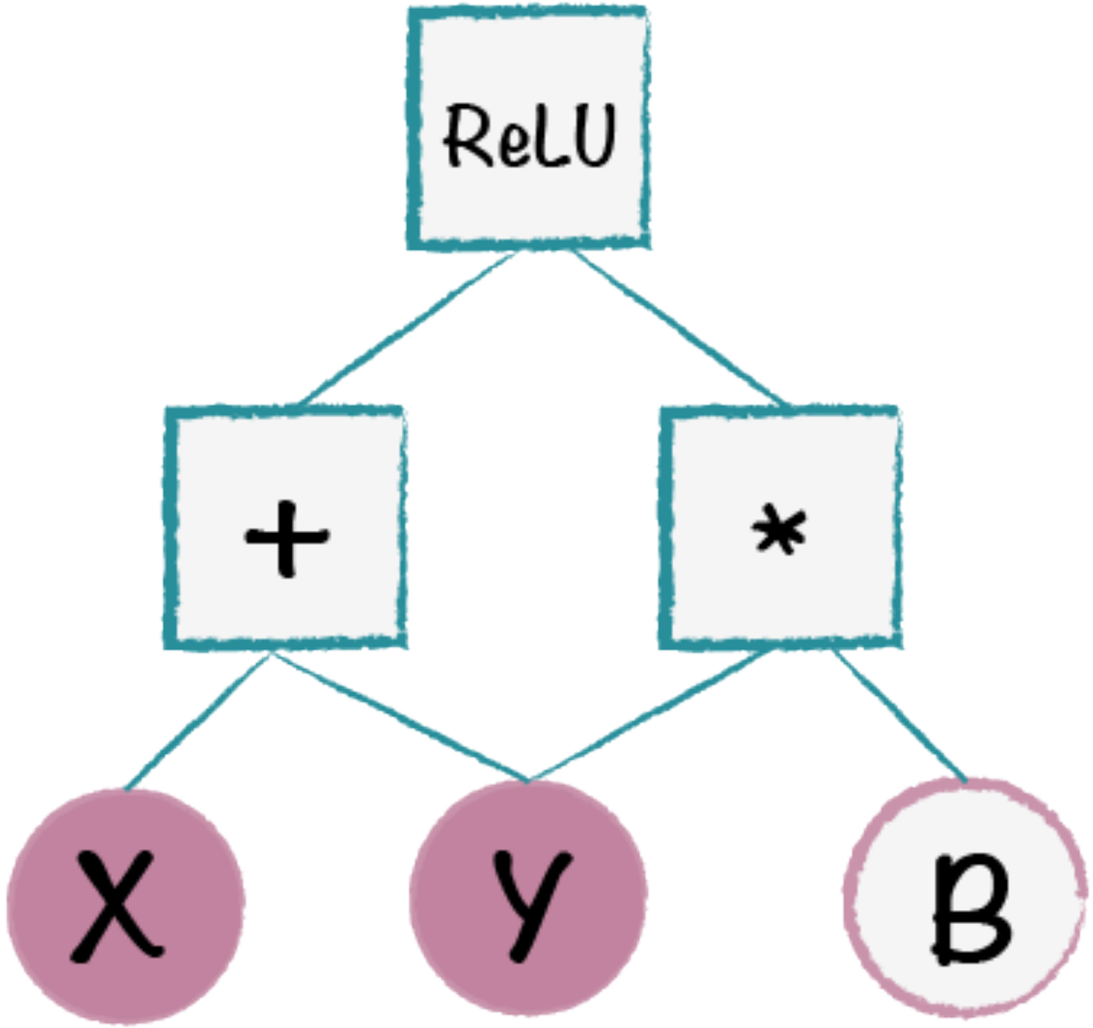
An open source machine learning library for research and production.

Tensor

Mathematical object more general than a vector or matrix.

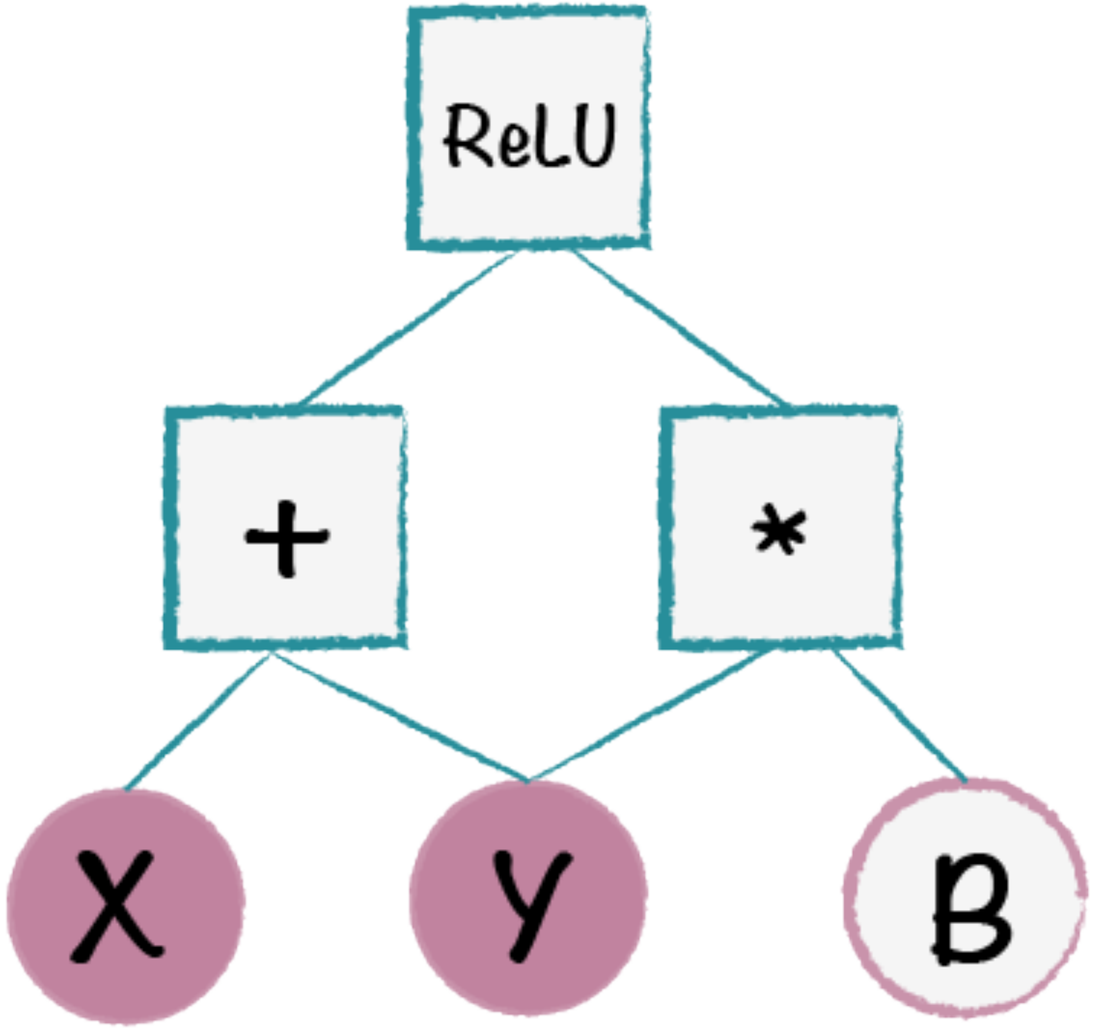


Data Flow Graph



Data Flow Graph

- Parallelism
- Distributed execution
- Compilation
- Portability

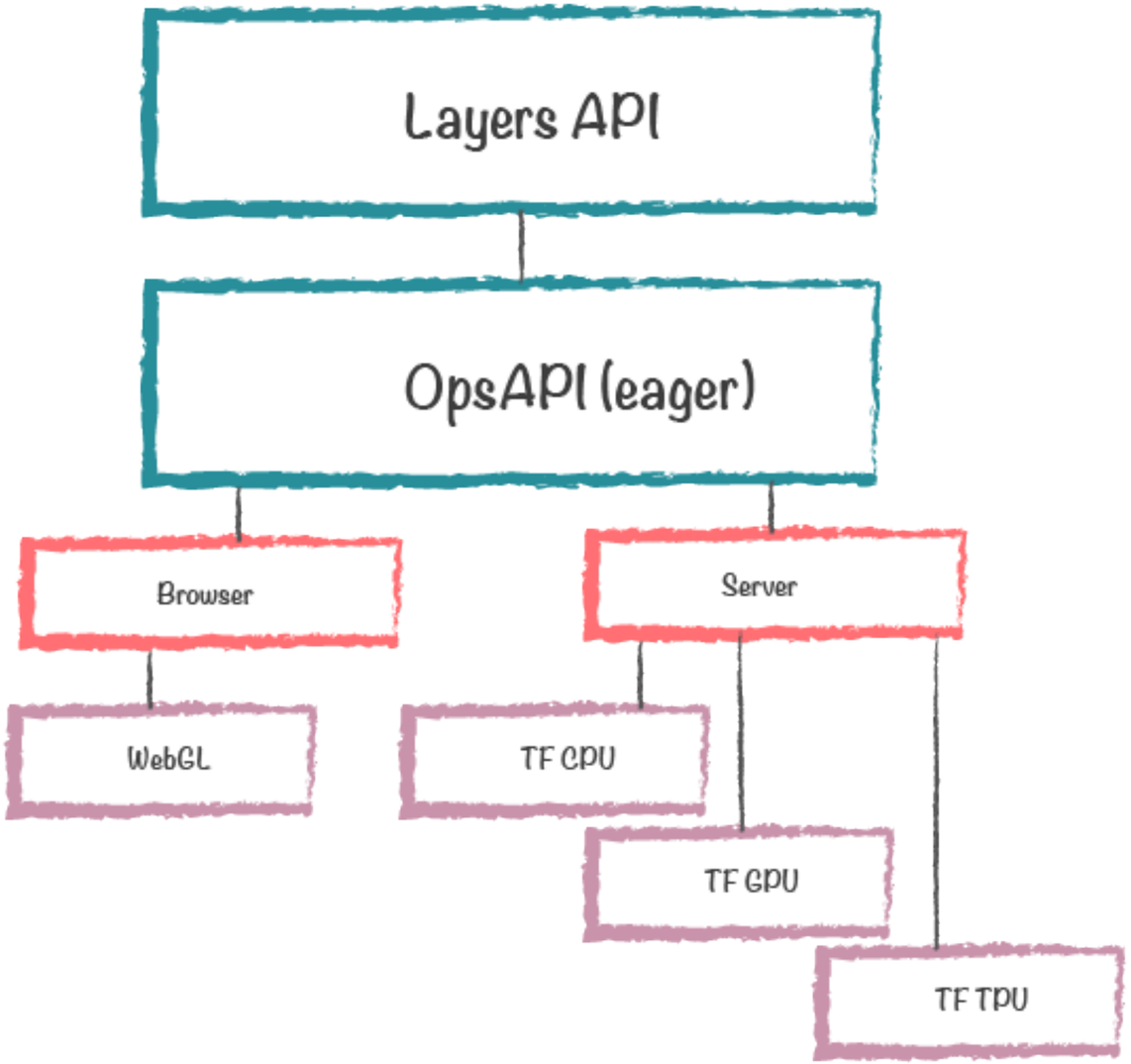


5

TensorFlow.js

Tensorflow.js

Tensorflow.js



Required eyebrow.

WebGL.

is a JavaScript API for rendering interactive 2D and 3D graphics

Advantages

- Data privacy.
- Low latency.

Examples

CNN From Scratch Iris Dataset

CNN from scratch - Iris Dataset.

Setosa

Versicolor

Virginica

CNN from scratch - Iris Dataset.

```
import * as tf from "@tensorflow/tfjs"
import "@tensorflow/tfjs-node"
import iris from "./iris.json"
import irisTesting from "./iris-testing.json"
```

CNN from scratch - Iris Dataset.

```
const trainingData = tf.tensor2d(iris.map(item => [
  item.sepal_length, item.sepal_width, item.petal_length, item.petal_width,
]))

const testingData = tf.tensor2d(irisTesting.map(item => [
  item.sepal_length, item.sepal_width, item.petal_length, item.petal_width,
]))

const outputData = tf.tensor2d(iris.map(item => [
  item.species === "setosa" ? 1 : 0,
  item.species === "virginica" ? 1 : 0,
  item.species === "versicolor" ? 1 : 0,
]))
```

CNN from scratch - Iris Dataset.

```
const trainingData = tf.tensor2d(iris.map(item => [
  item.sepal_length, item.sepal_width, item.petal_length, item.petal_width,
]))

const testingData = tf.tensor2d(irisTesting.map(item => [
  item.sepal_length, item.sepal_width, item.petal_length, item.petal_width,
]))

const outputData = tf.tensor2d(iris.map(item => [
  item.species === "setosa" ? 1 : 0,
  item.species === "virginica" ? 1 : 0,
  item.species === "versicolor" ? 1 : 0,
]))
```

CNN from scratch - Iris Dataset.

```
model.fit(trainingData, outputData, {epochs: 100})
  .then((history) => {
    model.predict(testingData).print()
  })
```

```
Tensor
    [[0.9829643, 0.0000683, 0.0263934], //setosa
     [0.0032604, 0.9411913, 0.0756058], //versicolor
     [0.0151716, 0.0281591, 0.9535264]] //virginica
```

Pre-trained models MobileNet

Pre-trained models

- PoseNet: Human pose estimation.
- BodyPix: Body part segmentation.
- Toxicity: Score the perceived impact of a comment.
- Universal sentence encoder: Natural language processing.
- Coco SSD: Object detection
- Speech commands: Classify 1 second audio snippets.
- KNN Classifier: K-Nearest Neighbors algorithm.
- MobileNet: Classify images with labels from ImageNet.

Pre-trained models.

```
const classifier = knnClassifier.create();

async function app() {
    net = await mobilenet.load();
    await setupWebcam();

    document.getElementById('class-1').addEventListener('click', () => addExample(0));
    document.getElementById('class-2').addEventListener('click', () => addExample(1));
    document.getElementById('class-3').addEventListener('click', () => addExample(2));

    const addExample = classId => {
        const activation = net.infer(webcamElement, 'conv_preds');
        classifier.addExample(activation, classId);
    };
}
```

Pre-trained models.

prediction: B

probability: 0.6666666666666666

Add 1 Add 2 Add 3

Elements Console Sources Network Performance Memory Application Security Audits

<!doctype html>

APIs
FaceAPI

FaceAPI

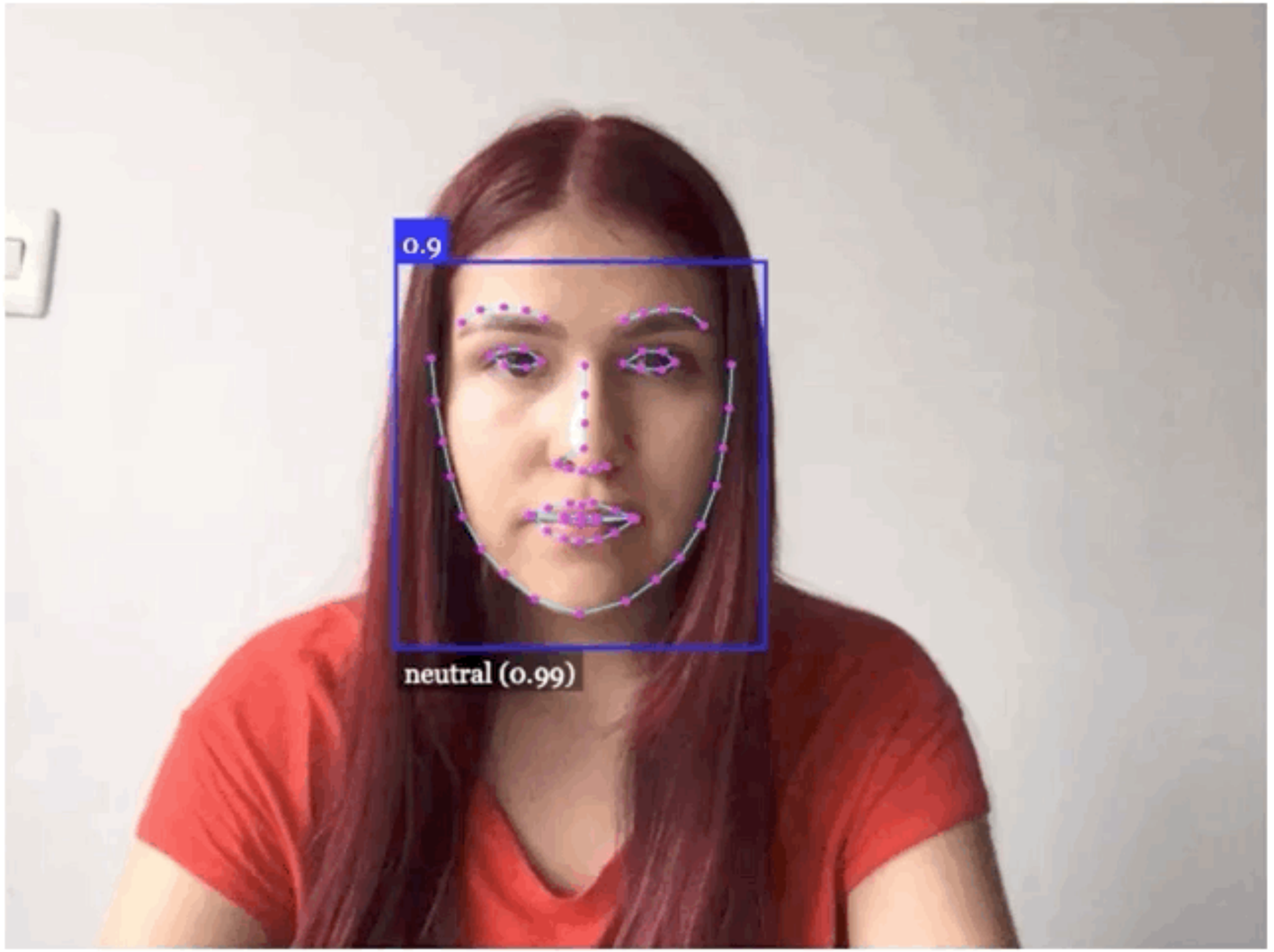
```
<script defer src="face-api.min.js"></script>
```

```
Promise.all([
  faceapi.nets.tinyFaceDetector.loadFromUri(MODEL_URL),
  faceapi.nets.faceLandmark68Net.loadFromUri(MODEL_URL),
  faceapi.nets.faceRecognitionNet.loadFromUri(MODEL_URL),
  faceapi.nets.faceExpressionNet.loadFromUri(MODEL_URL),
]).then(startVideo);
```

FaceAPI

```
function onPlay(el) {
  const displaySize = faceapi.matchDimensions(canvas, videoInput, true);

  setInterval(async () => {
    const detections = await faceapi
                            .detectAllFaces(videoInput,new faceapi.TinyFaceDetectorOptions())
                            .withFaceLandmarks()
                            .withFaceExpressions();

    const resizedDetections = faceapi.resizeResults(detections, displaySize);

    canvas.getContext('2d').clearRect(0, 0, canvas.width, canvas.height);

    faceapi.draw.drawFaceExpressions(canvas, resizedDetections);
    faceapi.draw.drawFaceLandmarks(canvas, detections);
    faceapi.draw.drawDetections(canvas, detections);
  }, 100);
}
```

FaceAPI



FaceAPI

Repo: http://github.com/ManuCastrillonM/cnn-in-the-browser

HUGE

Done.

Image recognition in the browser using Tensorflow.js

Aug 13, 2019