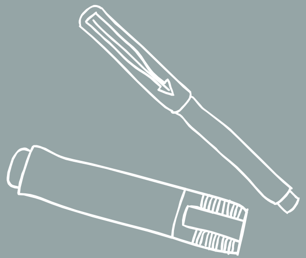
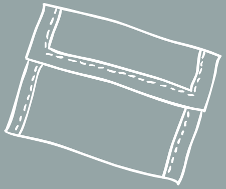
Build Your Machine Learning Model On EDGE with React Native

By: Rashmi Nagpal

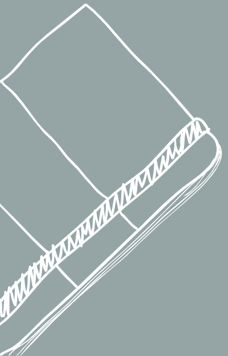
Agenda

- Motivation
- Overview of Machine Learning
- Demo
- Applications
- Challenges
- Key takeaways
- Resources

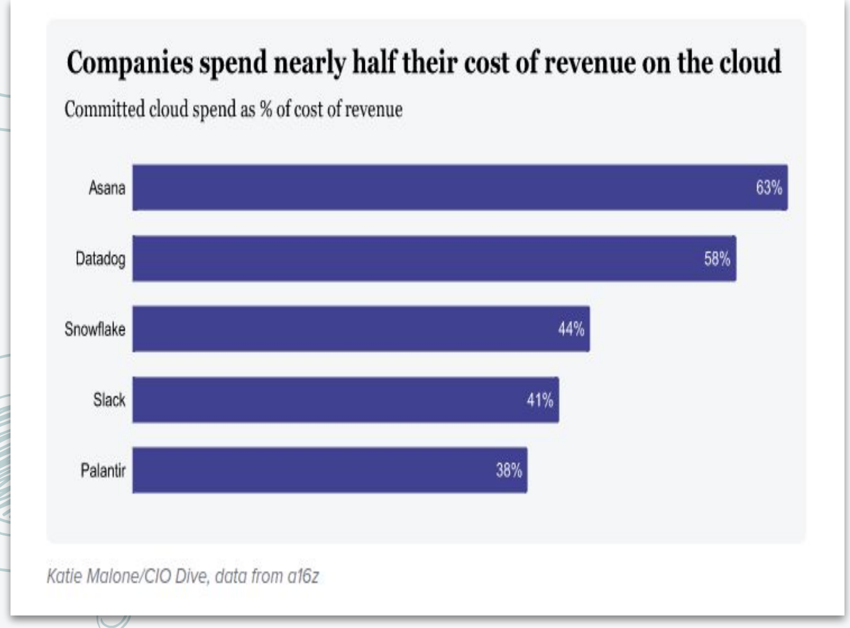




Motivation



Data from A16Z 2022

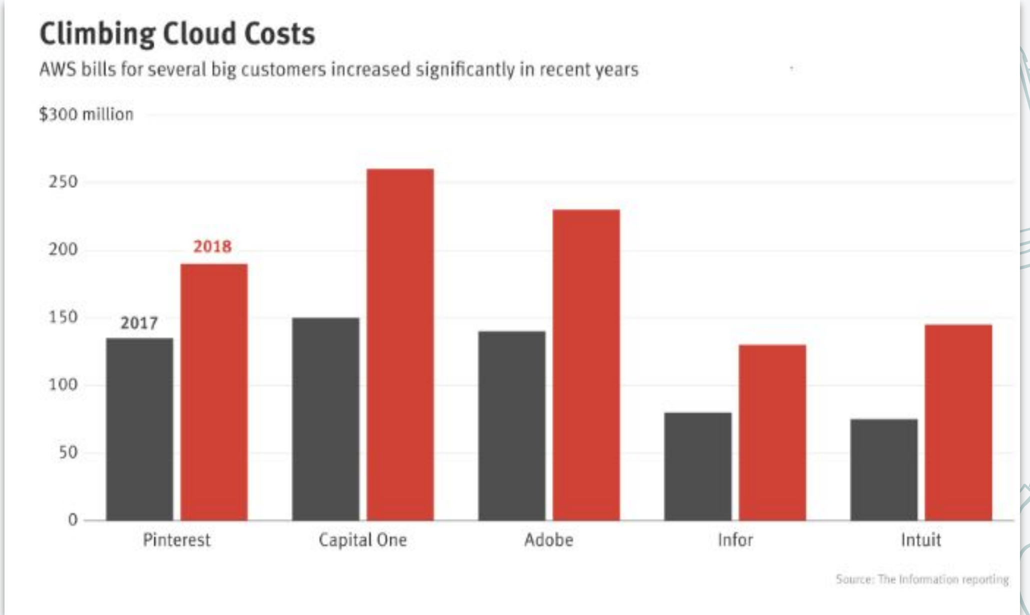


Companies spend nearly half their cost of revenue on the cloud

Committed cloud spend as % of cost of revenue

Katie Malone/CIO Dive, data from a16z

Data from TheInformation.com



Climbing Cloud Costs

AWS bills for several big customers increased significantly in recent years

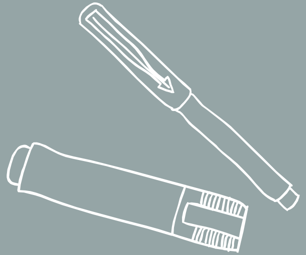
Source: The Information reporting

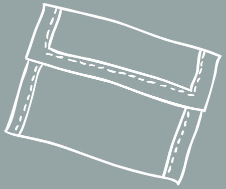
WHAT IS EDGE COMPUTING?

- Edge computing refers to distributed computing architecture where computing resources are placed closer to the edge devices thus providing faster response time and reduced data transfer cost.

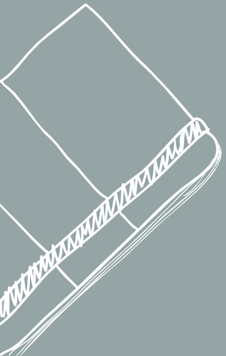
Why Edge Computing?

- Cost Reduction
- Lower Network Latency
- Increased Security and Privacy
- Increased Reliability



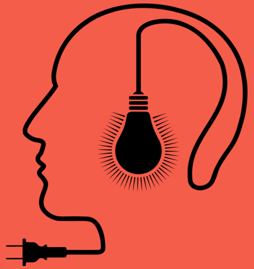


OVERVIEW



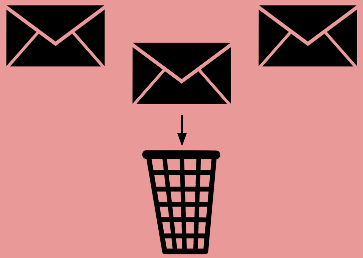
ARTIFICIAL INTELLIGENCE

Any technique that enables computers to mimic human behavior



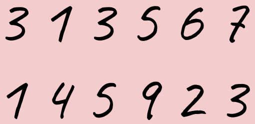
MACHINE LEARNING

Ability to learn without explicitly being programmed

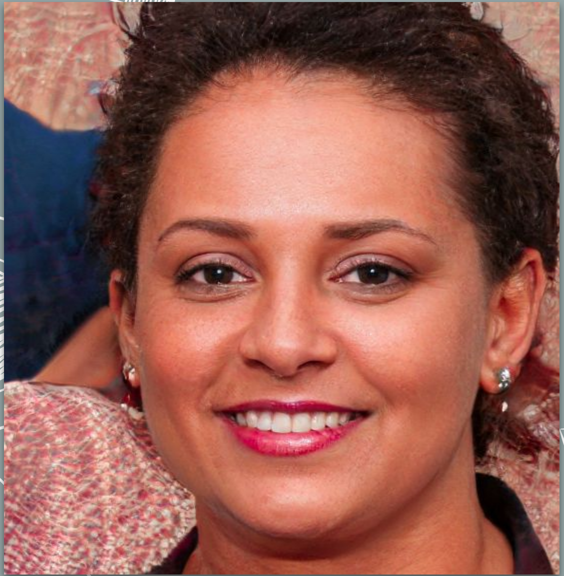


DEEP LEARNING

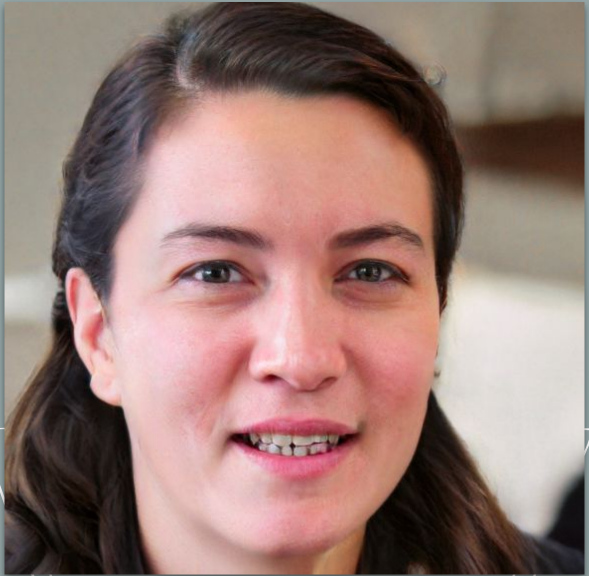
Extract patterns from data using neural networks



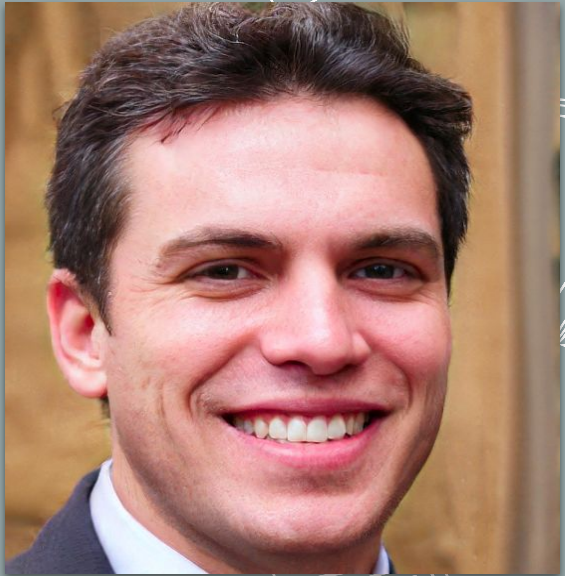
A



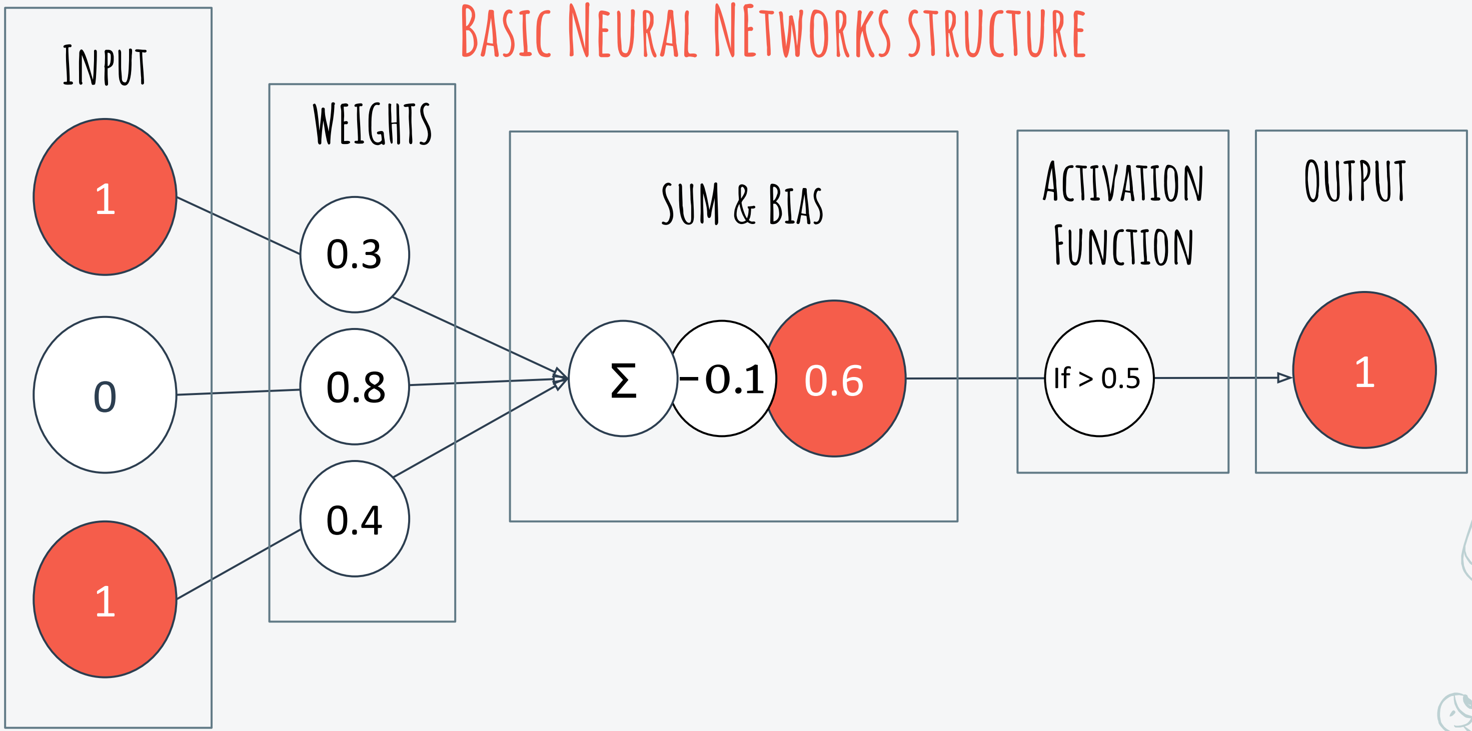
B



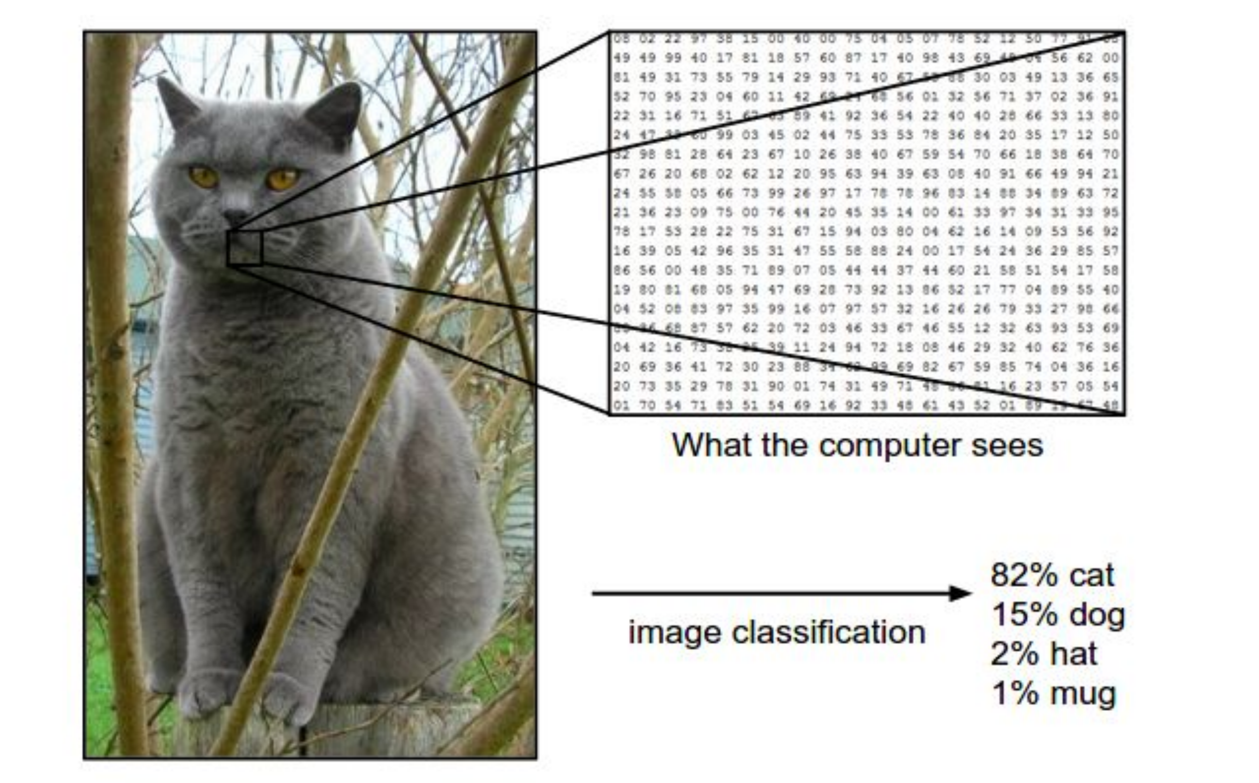
C



Basic Neural NEtworks structure



Example - Image Recognition



What the computer sees

image classification

82% cat
15% dog
2% hat
1% mug

Img Src: KDnuggets Blog

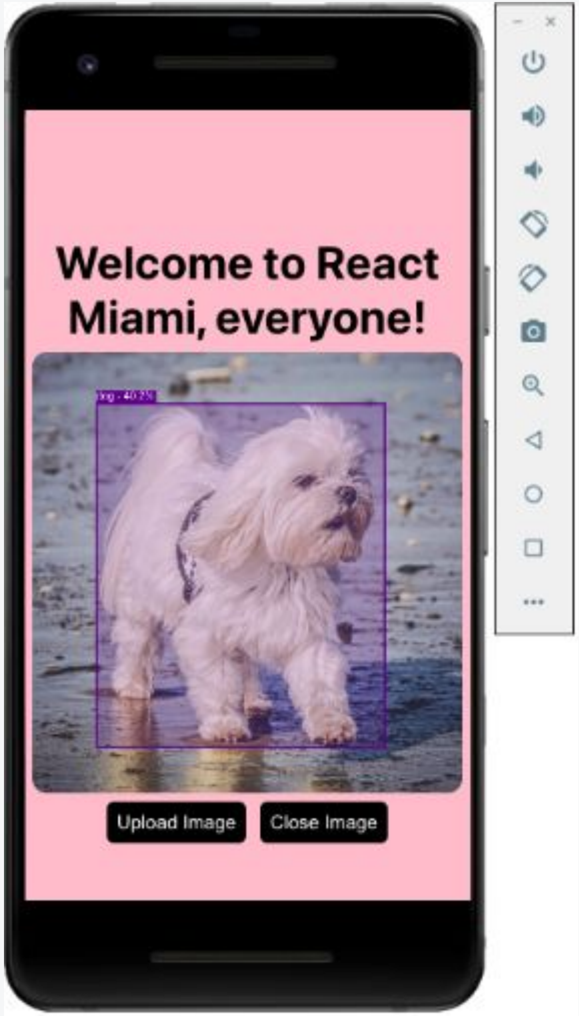
CRUCIAL QUESTION!

Can we develop fast machine learning models in browser/on-device?

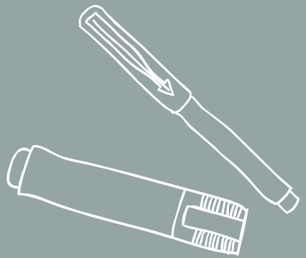
Definitely, Yes!

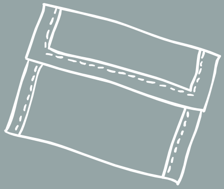
- Adopting “quantization” techniques
- Using specialized hardware such as GPU’s, TPUs
- In-browser ML frameworks like, tensorflow.js, onnx.js or CoreML

Demo

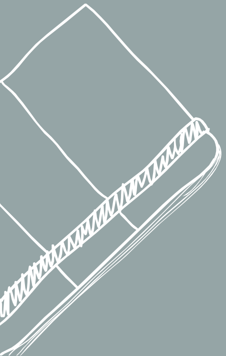


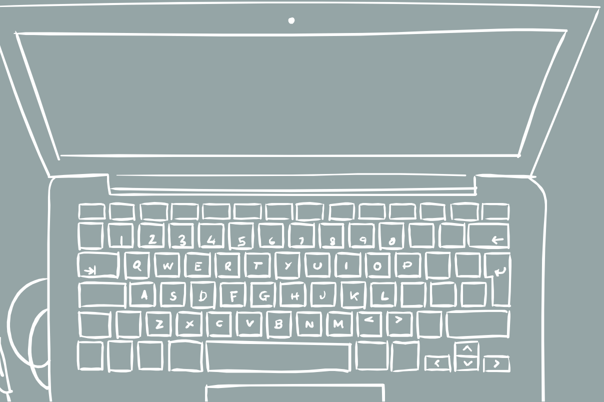
Welcome to React Miami, everyone!



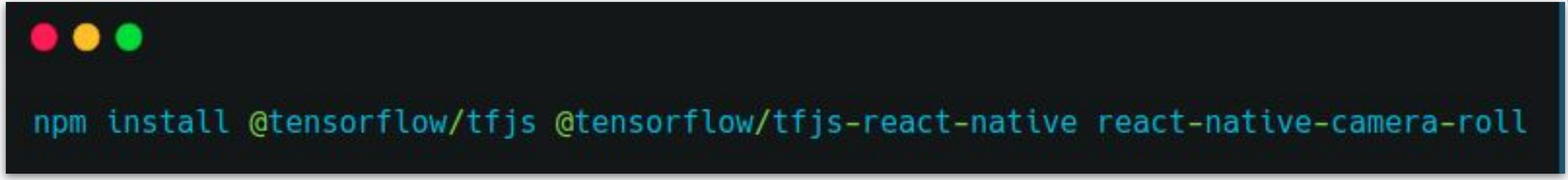


Lets Build your OWn Model



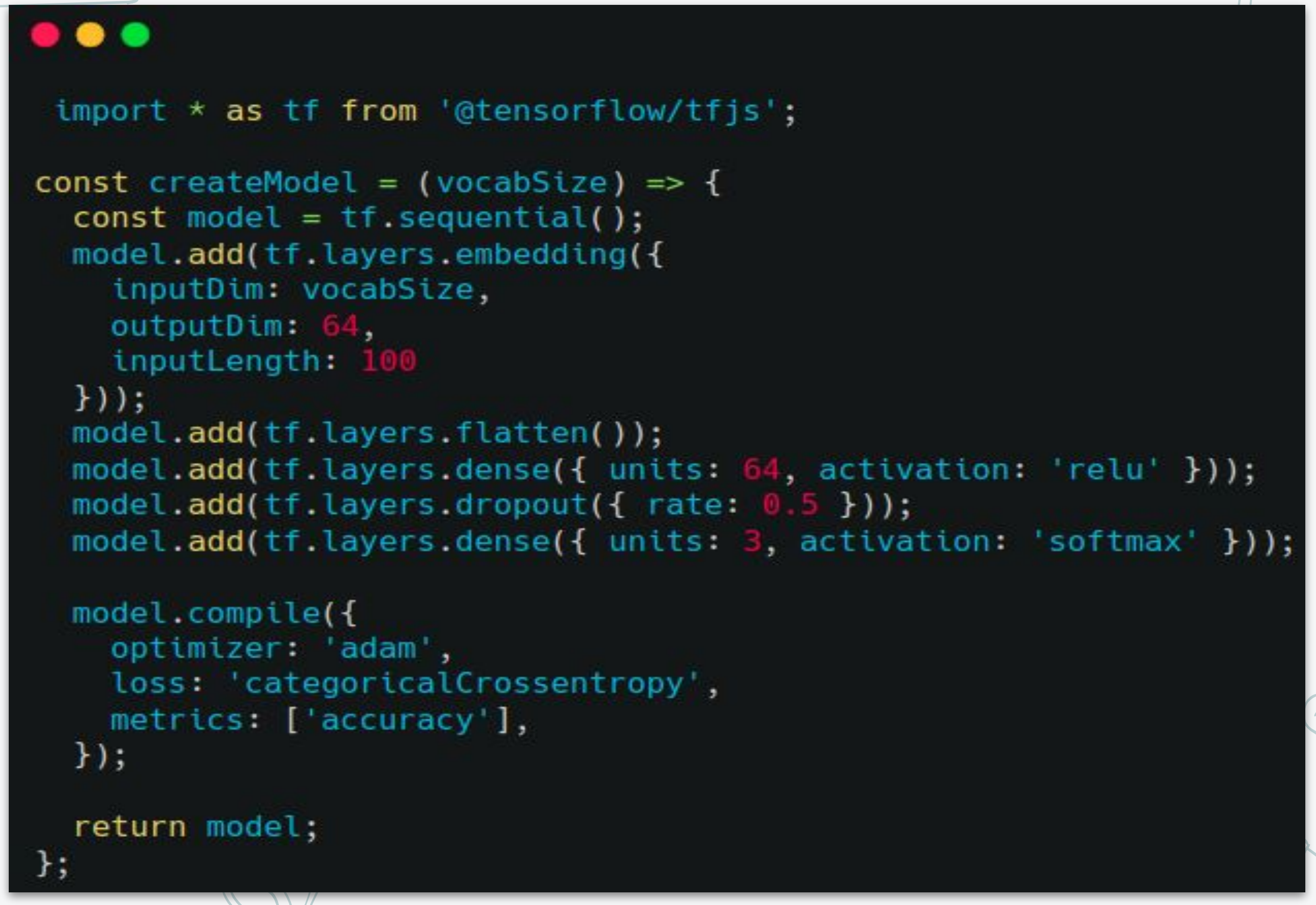


Install Libraries



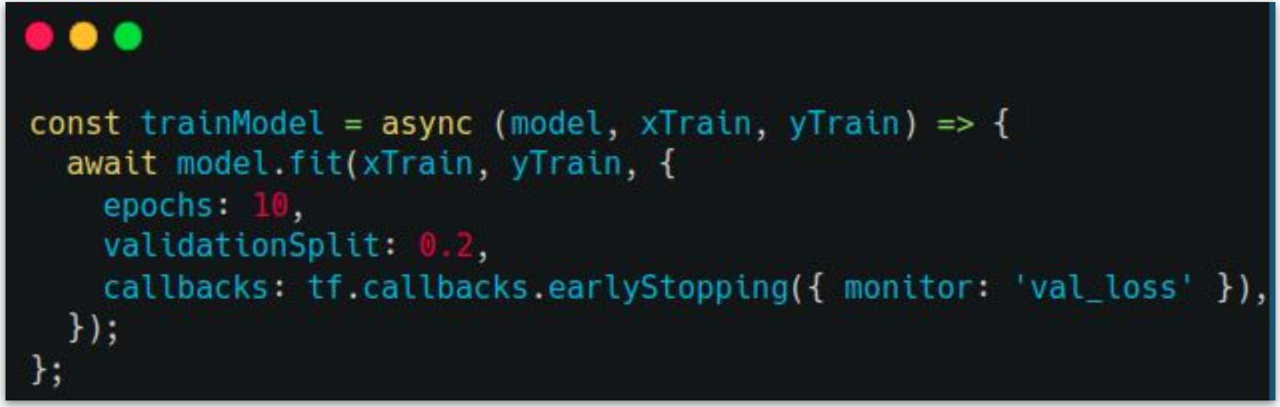
```
npm install @tensorflow/tfjs @tensorflow/tfjs-react-native react-native-camera-roll
```

Define your model



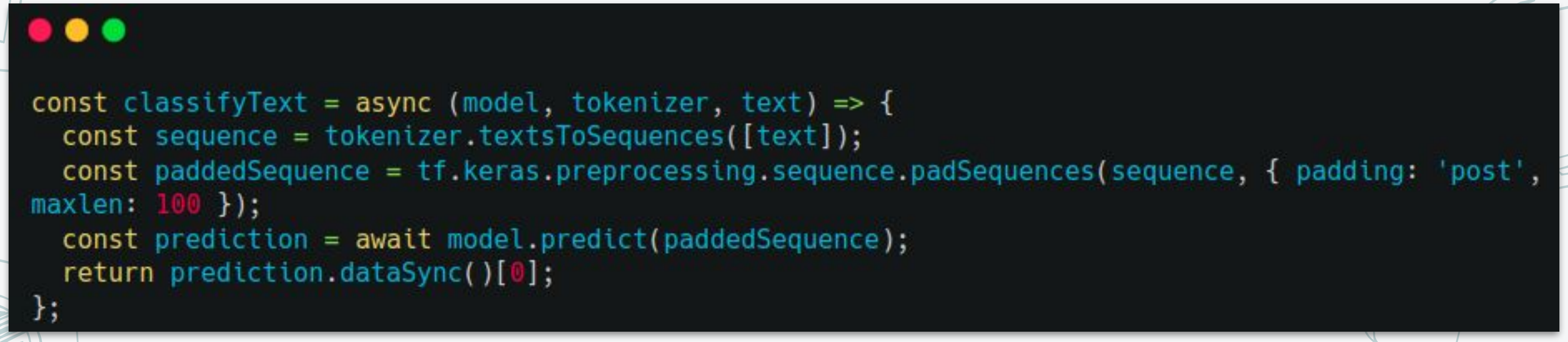
```
 import * as tf from '@tensorflow/tfjs';

const createModel = (vocabSize) => {
  const model = tf.sequential();
  model.add(tf.layers.embedding({
    inputDim: vocabSize,
    outputDim: 64,
    inputLength: 100
  }));
  model.add(tf.layers.flatten());
  model.add(tf.layers.dense({ units: 64, activation: 'relu' }));
  model.add(tf.layers.dropout({ rate: 0.5 }));
  model.add(tf.layers.dense({ units: 3, activation: 'softmax' }));

  model.compile({
    optimizer: 'adam',
    loss: 'categoricalCrossentropy',
    metrics: ['accuracy'],
  });

  return model;
};
```

Train your model



```
const trainModel = async (model, xTrain, yTrain) => {
  await model.fit(xTrain, yTrain, {
    epochs: 10,
    validationSplit: 0.2,
    callbacks: tf.callbacks.earlyStopping({ monitor: 'val_loss' }),
  });
};
```

Make Predictions Using your model

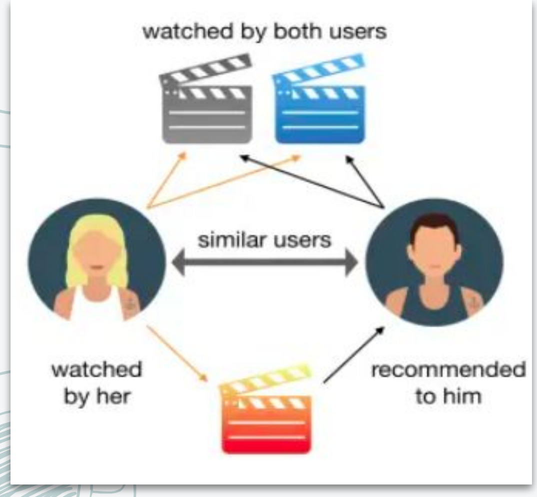


```
const classifyText = async (model, tokenizer, text) => {
  const sequence = tokenizer.textsToSequences([text]);
  const paddedSequence = tf.keras.preprocessing.sequence.padSequences(sequence, { padding: 'post',
maxlen: 100 });
  const prediction = await model.predict(paddedSequence);
  return prediction.dataSync()[0];
};
```

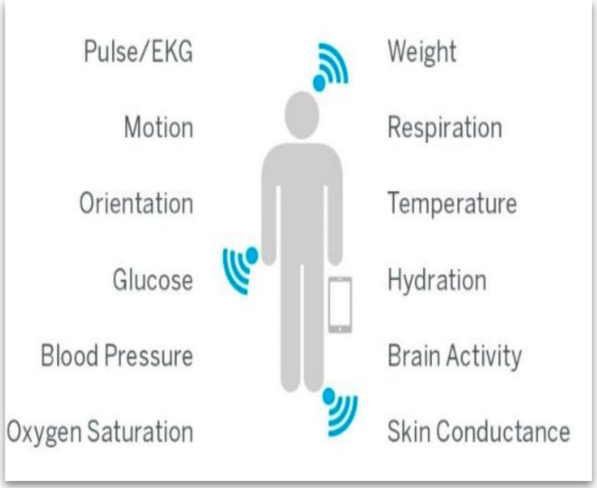
CRUCIAL QUESTION!

What are the potential applications of ML on edge devices?

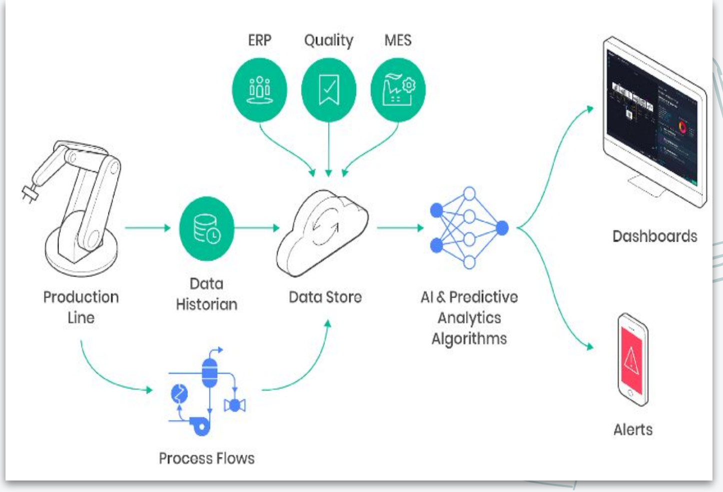
Use Cases Of: Machine Learning on Edge



Recommendation Systems



Patient Monitoring Systems



Predictive Maintenance

Img Src: Google Photos

CRUCIAL QUESTION!

What are the challenges of building ML models on edge devices?

Challenges

- Memory Constraints
- Data Quality
- Limited Computation Resources

CRUCIAL QUESTION!

What are the best practices for building and deploying bulky machine learning models one edge devices?

Best practices

Optimize

Optimize your model's runtime using quantization or pruning techniques

Find

Find the right params for your inference pipeline be it it batch size, epochs etc

Benchmark

Benchmark your end-to-end pipeline/application to figure out any bottlenecks

Use

Use concurrent or multiple processes to revamp your code for optimization

Frameworks

Use mobile-friendly frameworks such as pytorch lite, or tensorflow lite

Reuse

Avoid redundant computations via data-reuse among multiple tasks

Resources

- ✖ Research paper: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
- ✖ Tutorials: MIT 6.S191 Introduction to Deep Learning by Prof Alexander Amini
- ✖ Book: Learning TensorFlow.js
- ✖ Course: https://www.coursera.org/learn/browser-based-models-tensorflow

Thank you!

Feel free to shoot me any questions, or just DM for a quick hi!

@iamrashminagpal