

# Introducing TensorFlow:

A framework for machine learning

---

Department of Computer Science, City University of Hong Kong

# Table of contents

1. Introduction
2. Setup
3. Demo
4. Reference
5. Q&A

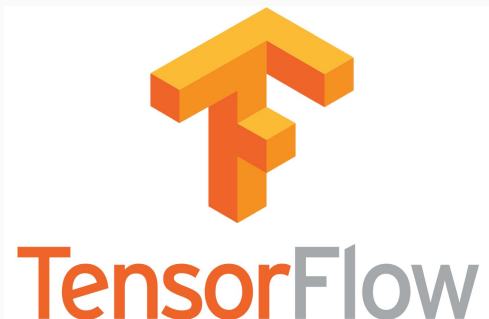
# Introduction

---

# Introduction

What is TensorFlow?

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.



# Introduction

## Why Tensorflow?

- Easy to use



### Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.



### Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.



### Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

# Introduction

## Why Tensorflow?

- Easy to use
- Widely used



Figure 1: Companies using TensorFlow

# Introduction

## Why Tensorflow?

- Easy to use
- Widely used



### TensorFlow

Learn the foundation of TensorFlow with tutorials for beginners and experts to help you create your next machine learning project.



### For JavaScript

Use TensorFlow.js to create new machine learning models and deploy existing models with JavaScript.



### For Mobile & IoT

Run inference with TensorFlow Lite on mobile and embedded devices like Android, iOS, Edge TPU, and Raspberry Pi.



### For Production

Deploy a production-ready ML pipeline for training and inference using TensorFlow Extended (TFX).

**Figure 2:** Different platforms of TensorFlow

# Introduction

## Why Tensorflow?

- Easy to use
- Widely used
- Resources and community support

### #PoweredByTF DevPost Challenge



Build something amazing with TF 2.0, share it with the world, and win prizes!

### 2.0 feature tracker



See details of in-progress, planned, and completed development activities for TensorFlow 2.0.

### Google Summer of Code



Get paid to work on an open-source TensorFlow project this summer! Open to undergrad and graduate students.

### YouTube



Our YouTube Channel focuses on machine learning and AI with TensorFlow. Explore a number of new shows, including TensorFlow Meets, Ask TensorFlow, and Coding TensorFlow.

### Twitter



For up-to-date news and updates from the community and the TensorFlow team, follow @tensorflow on Twitter.

### TensorFlow announcements



Join the TensorFlow announcement mailing list to learn about the latest release updates, security advisories, and other important information from the TensorFlow team.



# Introduction

What is TensorFlow.js?

TensorFlow.js is a library for developing and training ML models in JavaScript, and deploying in browser or on Node.js



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

# Setup

---

# Setup

There are two main ways to get TensorFlow.js up running in your browser:

- Using script tags (Add this code in your XX.html file)



```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.0.0"></script>
```

- Installation from NPM and using a build tool like Parcel, WebPack, or Rollup.



```
yarn add @tensorflow/tfjs  
npm install @tensorflow/tfjs
```

# Demo

---

Linear regression:

- Objective function:

$$h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

where  $\theta_0$  and  $\theta_1$  are the parameters that we want to get.

- Loss function:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2,$$

The loss function is **Mean squared error** (MSE), and we want to find a  $\theta$  that minimize this loss function.

Linear regression:

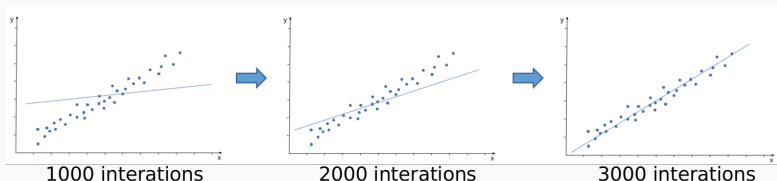
- Gradient Descent:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

where  $\alpha$  is the learning rate (e.g. 0.001). We repeat this process until  $\theta$  convergence.

- Learning process:

We randomly initialize a  $\theta$  and update it using gradient descent method.

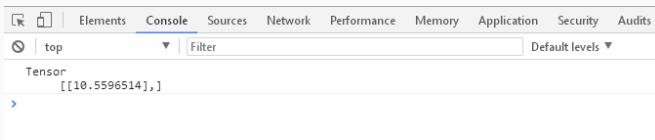


## Linear regression:

```
1 <html>
2   <head>
3     <!-- Load TensorFlow.js -->
4     <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.0.0"> </script>
5
6     <!-- Place your code in the script tag below. You can also use an external .js file -->
7     <script>
8       // Notice there is no 'import' statement. 'tf' is available on the index-page
9       // because of the script tag above.
10
11       // Define a model for linear regression.
12       const model = tf.sequential();
13       model.add(tf.layers.dense({units: 1, inputShape: [1]}));
14
15       // Prepare the model for training: Specify the loss and the optimizer.
16       model.compile({loss: 'meanSquaredError', optimizer: 'sgd'});
17
18       // Generate some synthetic data for training.
19       const xs = tf.tensor2d([1, 2, 3, 4, 5], [5, 1]);
20       const ys = tf.tensor2d([1, 3, 5, 7, 9], [5, 1]);
21
22       // Train the model using the data.
23       model.fit(xs, ys, {epochs: 100}).then(() => {
24         // Use the model to do inference on a data point the model hasn't seen before:
25         // Open the browser devtools to see the output
26         model.predict(tf.tensor2d([6], [1, 1])).print();
27       });
28     </script>
29   </head>
30
31   <body>
32   </body>
33 </html>
```

## Linear regression:

- Save the code above in a .html file (e.g. LRdemo.html), open it with browser, then open devtools (F12-Console) to see the output. (Code can be found in [https://ymq115599.github.io/Demo\\_LR.html](https://ymq115599.github.io/Demo_LR.html))



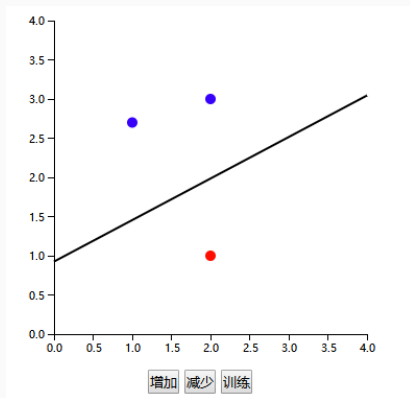
- Change the dataset (xs and ys) and see the difference.

```
18      // Generate some synthetic data for training.  
19      const xs = tf.tensor2d([1, 2, 3, 4, 5], [5, 1]);  
20      const ys = tf.tensor2d([1, 3, 5, 7, 9], [5, 1]);
```



## Linear regression:

- \*Can you visualize the result in webpage? (add codes in `< body >` part, a demo visualizing Support Vector Machine can be found in [https://ymq115599.github.io/SVM\\_Demo.html](https://ymq115599.github.io/SVM_Demo.html))



## Reference

---

# References

<https://tensorflow.google.cn/>

<https://www.tensorflow.org/js/tutorials/setup>

<https://js.tensorflow.org/api/latest/>

<https://www.tensorflow.org/js/guide>

## Q&A

---