

tensorflow.js

tensorflow.js is a JavaScript library for training and deploying ML models in the browser and on **node.js**.

tensorflow.js via node.js on Mac

In order to run tensorflow.js in node.js on Mac, you should install tensorflow.js as following:

```
$ npm install @tensorflow/tfjs-node
```

If your system NVIDIA GPU, it will be recommended to install below package as well.

```
$ npm install @tensorflow/tfjs-node-gpu
```

The below is the example of tensorflow.js based on node.js

```
const tf = require('@tensorflow/tfjs');

// Load the binding:
require('@tensorflow/tfjs-node'); // Use '@tensorflow/tfjs-node-gpu' if running with GPU.

// Train a simple model:
const model = tf.sequential();
model.add(tf.layers.dense({units: 100, activation: 'relu', inputShape: [10]}));
model.add(tf.layers.dense({units: 1, activation: 'linear'}));
model.compile({optimizer: 'sgd', loss: 'meanSquaredError'});

const xs = tf.randomNormal([100, 10]);
const ys = tf.randomNormal([100, 1]);

model.fit(xs, ys, {
  epochs: 100,
  callbacks: {
    onEpochEnd: async (epoch, log) => {
      console.log(`Epoch ${epoch}: loss = ${log.loss}`);
    }
  }
});
```

tensorflow.js via script tag

The below is the example of Tensorflow.js coding as a part of HTML documentation.

tensorflow-example.html

```
<html>
  <head>
    <!-- Load TensorFlow.js -->
    <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.13.3/dist/tf.min.js"> </script>

    <!-- Place your code in the script tag below. You can also use an external .js file -->
    <script>
      // Notice there is no 'import' statement. 'tf' is available on the index-page
      // because of the script tag above.

      // Define a model for linear regression.
      const model = tf.sequential();
      model.add(tf.layers.dense({units: 1, inputShape: [1]}));

      // Prepare the model for training: Specify the loss and the optimizer.
      model.compile({loss: 'meanSquaredError', optimizer: 'sgd'});

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

      // Train the model using the data.
      model.fit(xs, ys, {epochs: 10}).then(() => {
        // Use the model to do inference on a data point the model hasn't seen before:
        // Open the browser devtools to see the output
        model.predict(tf.tensor2d([5], [1, 1])).print();
      });
    </script>
  </head>

  <body>
  </body>
</html>
```

Reference URL: <https://js.tensorflow.org/>