Simple Text Classification with TensorFlow.js (Node.js)

1. Prerequisites

- Make sure you have **Node.js** installed.
- Create a new folder for your project.

2. Setup

Open your terminal and run:

```
mkdir tfjs-text-classification
cd tfjs-text-classification
npm init -y
npm install @tensorflow/tfjs
```

---

3. Create a file called `index.js` with the following content:

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

// Sample training data
const sentences = [
  "I love this movie",
  "This is a bad product",
  "I enjoyed the experience",
  "Terrible service"
];

const labels = [1, 0, 1, 0];  // 1 = positive, 0 = negative

// Build vocabulary
const vocabulary = {};
let index = 2;  // reserve 0 for padding, 1 for unknown

sentences.forEach(sentence => {
  sentence.toLowerCase().split(' ').forEach(word => {
    if (!(word in vocabulary)) {
      vocabulary[word] = index++;
    }
  });
});

// Convert text to sequence of numbers
function textToSequence(text) {
  const words = text.toLowerCase().split(' ');
  return words.map(word => vocabulary[word] || 1);
}
```

```
// Prepare training data sequences
const sequences = sentences.map(textToSequence);

// Padding sequences
const maxLen = Math.max(...sequences.map(seq => seq.length));
const paddedSequences = sequences.map(seq => {
  const pad = Array(maxLen - seq.length).fill(0);
  return pad.concat(seq);
});

// Convert to tensors
const xs = tf.tensor2d(paddedSequences, [paddedSequences.length, maxLen]);
const ys = tf.tensor2d(labels, [labels.length, 1]);

// Define model
const model = tf.sequential();

model.add(tf.layers.embedding({
  inputDim: index,
  outputDim: 8,
  inputLength: maxLen
}));

model.add(tf.layers.flatten());

model.add(tf.layers.dense({
  units: 8,
  activation: 'relu'
}));

model.add(tf.layers.dense({
  units: 1,
  activation: 'sigmoid'
}));

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

// Train and test
async function trainAndTest() {
  console.log('Training model...');
  await model.fit(xs, ys, {
    epochs: 20,
    verbose: 1
  });

  // Test prediction
  const testSentence = "I hate this";
  const testSeq = textToSequence(testSentence);
  const paddedTestSeq = Array(maxLen -
testSeq.length).fill(0).concat(testSeq);
  const input = tf.tensor2d([paddedTestSeq], [1, maxLen]);

  const prediction = model.predict(input);
```

```
  const predValue = (await prediction.data())[0];

  console.log(`Prediction for "${testSentence}": ${predValue.toFixed(4)}`);
  console.log(predValue > 0.5 ? 'Positive' : 'Negative');
}

trainAndTest();
```

---

4. How to Run

Run this in your terminal:

```
node index.js
```

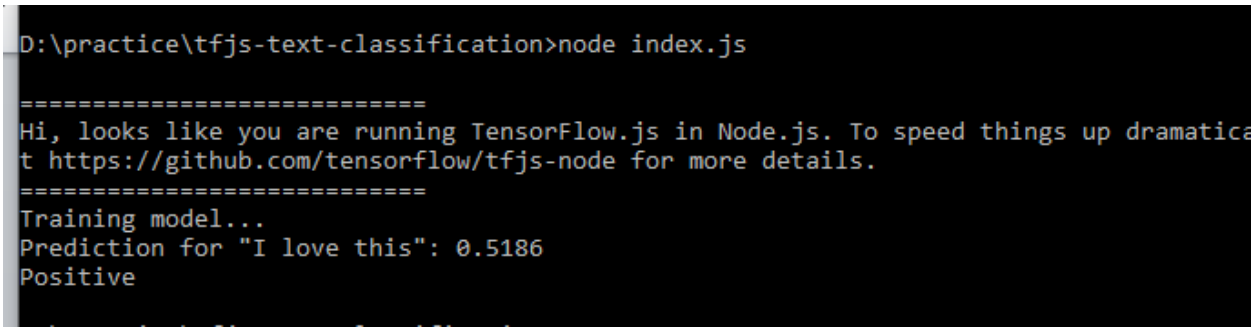
What you should see:

- The model will train for 20 epochs (very fast for this small data).
- Then it will print a prediction for the test sentence `"I hate this"`.
- The output will be a number between 0 and 1, interpreted as positive (> 0.5) or negative (<= 0.5).

Notes:

- This is a **toy example** to demonstrate the workflow.
- For real projects:
  - Use much more data.
  - Use proper tokenization and preprocessing.
  - Consider batch training.
  - Use pretrained models or embeddings for better accuracy.

Output:-


D:\practice\tfjs-text-classification>node index.js
==========
Hi, looks like you are running TensorFlow.js in Node.js. To speed things up dramatica
t https://github.com/tensorflow/tfjs-node for more details.
==========
Training model...
Prediction for "I love this": 0.5186
Positive