TensorFlow.js and Node-RED: The Low-Code Approach to ML Apps for IoT

—

va barbosa
@vabarbosa

Paul Van Eck
@pvaneckw

ibm.biz/tfjs-nodered
IoT & Connected Devices

# IoT Challenges

<table>
<thead>
<tr>
<th>Variety of edge devices</th>
<th>Quantity of edge devices</th>
<th>Different scale</th>
<th>Edge devices restrictions</th>
</tr>
</thead>
</table>

Node-RED is an open source flow-based programming tool for wiring together IoT devices.

https://nodered.org
TensorFlow.js is an open source library to train and deploy machine learning models with JavaScript.
// load a graph model from a URL
const model = await tf.loadGraphModel(MODEL_URL)

// create a tensor from an image
const imageTensor = tf.browser.fromPixels(imageNode)

// insert a dimension into the tensor's shape
const inputTensor = imageTensor.expandDims()

// execute the inference for the input tensors
return model.predict(inputTensor)

// split the prediction into the heatmap and pafmap arrays
const [heatmaps, pafmaps] = prediction.unstack()[0].split([HEATMAP_COUNT, PAFMAP_COUNT], 2)
Why add TensorFlow.js to Node-RED?

ML to IoT with ease
- Intuitive visual programming model
- Lower barrier to entry

Privacy
- Data security
- Data does not leave device

Connectivity
- Offline, unreliable
- Low bandwidth
Alternative to TensorFlow Lite?

Node.js
- 12+ million JS developers
- 200k+ NPM packages

Adoption, Support
- Cloud services (e.g., IBM Cloud)
- IoT platforms (e.g., AT&T IoT Platform)

Integration, Contribution
- Raspberry Pi, MultiTech Conduit, Ubidots, HomeKit, Opto 22, etc.
TensorFlow.js models in Node-RED
Components of a Node-RED module

- What a node does
- Node's properties, edit dialog, help text
- Package as npm module

JavaScript

HTML

package.json

https://medium.com/codait/creating-custom-node-red-nodes-for-your-api-the-easy-way-10770ccd8923
Packaging an Object Detection Model

```json
{
    "name": "node-red-contrib-tfjs-object-detection",
    "version": "1.0.0",
    "description": "Object detection model that aims to localize and identify multiple objects in a single image.",
    "dependencies": {
        "@tensorflow-models/coco-ssd": "^2.0.0",
        "@tensorflow/tfjs-node": "1.2.11"
    },
    "main": "object-detection.js",
    "scripts": {
        "test": "echo \"Error: no test specified\" && exit 1"
    },
    "keywords": [
        "tfjs",
        "node-red"
    ],
    "author": "pvaneck@us.ibm.com",
    "license": "Apache-2.0",
    "node-red": {
        "nodes": {
            "object-detection": "object-detection.js"
        }
    }
}
```

https://github.com/tonanhngo/nodered-tfjs/tree/master/object-detection
Packaging an Object Detection Model

```javascript
<script type="text/javascript">
  RED.nodes.registerType('object-detection',{  
    category: 'analysis-function',
    color: '#FF9BF6',
    defaults: {
      name: { value: "Object Detection Model" },
      modelUrl: { value: "" }
    },
    inputs:1,
    outputs:1,
    icon: "font-awesome/fa-object-group",
    label: function() {
      return this.name||"object-detection";
    }
  });
</script>

<script type="text/x-red" data-template-name="object-detection">
  <div class="form-row">
    <label for="node-input-name"><i class="icon-tag"></i> Name</label>
    <input type="text" id="node-input-name" placeholder="Node name"/>
  </div>
  <div class="form-row">
    <label for="node-input-model"><i class="fa fa-globe"></i> Model URL</label>
    <input type="text" id="node-input-modelUrl" placeholder="e.g. http://localhost/model/model.json..."/>
  </div>
  <small>
    If no URL is specified, then a model will be downloaded from a default public endpoint.
  </small>
</script>

<script type="text/x-red" data-help-name="object-detection">
  <p>
    Localize and identify multiple objects in a single image. This uses the TensorFlow.js port
    of the COCO-SSD model. This model is capable of detecting 90 classes of objects.
  </p>
</script>

https://github.com/tonanhngo/nodered-tfjs/tree/master/object-detection
Packaging an Object Detection Model

```javascript
// called when the runtime loads the node on start-up
// the argument provides the module access to the Node-RED runtime api
module.exports = function(RED) {

  const cocoSsd = require('@tensorflow-models/coco-ssd');
  const tf = require('@tensorflow/tfjs-node');

  // called whenever a new instance of the node is created
  // the argument passed contains the node-specific properties set in the flow editor
  function ObjectDetectionNode(config) {
    RED.nodes.createModel(this, config);

    this.modelUrl = config.modelUrl;
    const node = this;

    node.status({fill: 'yellow', shape: 'dot', text: 'Loading model...'});

    cocoSsd.load({modelUrl: node.modelUrl}).then(model => {
      node.loadedModel = model;
      node.status({fill: 'green', shape: 'dot', text: 'Model is ready'});
      console.log('Object Detection Model Loaded. ');
    });

    // register a listener to the input event which gets called whenever a message arrives at the node
    node.on('input', function(msg) {
      const imgTensor = tf.node.decodeImage(new Uint8Array(msg.payload), channels = 3);
      node.loadedModel.detect(imgTensor).then(predictions => {
        msg.payload = predictions;
        node.send(msg);
      });
    });
  }

  // register with the runtime
  RED.nodes.registerType("object-detection", ObjectDetectionNode);
}

https://github.com/tonanhngo/nodered-tfjs/tree/master/object-detection
```
### Keep in Mind

<table>
<thead>
<tr>
<th>Model</th>
<th>Performance</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>– Storing / Serving</td>
<td>– Loading / Caching</td>
<td>– Audio / Video</td>
</tr>
<tr>
<td></td>
<td>– Worker Thread</td>
<td>– Sensors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Input / Output processing</td>
</tr>
</tbody>
</table>
Example Flows
object detection with raspberry pi
twitter sentiment analysis
Other Use Cases

What are some potential real-life use cases?
Worker safety with hard hat detection

Photo by CEphoto, Uwe Aranas or alternatively © CEphoto, Uwe Aranas
Classifying faces for targeted advertising
DroneAid

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="SOS" /></td>
<td>Immediate Help Needed</td>
<td><img src="image" alt="Shelter" /></td>
<td>Shelter Needed</td>
</tr>
<tr>
<td><img src="image" alt="OK" /></td>
<td>No Help Needed</td>
<td><img src="image" alt="First Aid Kit" /></td>
<td>First Aid Kit Needed</td>
</tr>
<tr>
<td><img src="image" alt="Water" /></td>
<td>Water Needed</td>
<td><img src="image" alt="Area with Children" /></td>
<td>Area with Children in Need</td>
</tr>
<tr>
<td><img src="image" alt="Food" /></td>
<td>Food Needed</td>
<td><img src="image" alt="Area with Elderly" /></td>
<td>Area with Elderly in Need</td>
</tr>
</tbody>
</table>

https://github.com/code-and-response/droneaid
Recap

Node-RED
- Low-code platform for IoT
- Flexible and extensible
- Visual programming

TensorFlow.js
- Machine Learning in JavaScript
- Data Privacy
- Network connectivity

Node-RED + TensorFlow.js
- Easy packaging of TF.js models for Node-RED usage
- Rapid building of AI-enabled IoT applications
- Leverage TensorFlow.js, Node-RED, Node.js, and NPM communities
- Democratize AI
Links

https://js.tensorflow.org
https://nodered.org

https://github.com/yhwang/node-red-contrib-ds2-tfjs
https://github.com/vabarbosa/tfjs-node-red
https://github.com/tonanhngo/nodered-tfjs

https://medium.com/codait/creating-custom-node-red-nodes-for-your-api-the-easy-way-10770ccd8923