SciTinyML
Scientific Use of Machine Learning on Low Power Devices

Regional Workshop - Africa

Anomaly Detection and Post-Processing Hands-On Lab

Marcelo Rovai
Professor, UNIFEI - Brazil

Shawn Himel
Senior DevRel Engineer, Edge Impulse
Anomaly Detection
What is Anomaly Detection?

In data analysis, anomaly detection is the identification of rare items, events or observations which raise suspicions because they differing significantly from the majority of the data.
Machine working well

Class 1

Outlier

Machine NOT working well

Machine working well

Class 1

Machine working well

Class 2
Outlier

Lift

Maritime

Ups!
Spectral Analysis

NN Classifier

K-Means

Classes

- Lift
- Terrestrial
- Maritime
- Idle

- Anomaly
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

### Time series data
- 
  - Create Impulse
  - Spectral features
  - 1st classifier

### Spectral Analysis
- 
  - Window size
  - Window increase

### Classification (Keras)
- 
  - Frequency (Hz)
  - Zero-pad data

### Output features
- 
  - 4 (idle, lift, maritime, terrestrial)

---

**Add a learning block**

Some learning blocks have been hidden based on the data in your project.

- **Classification (Keras)**
  - Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.
  - Author: Edgimpulse Inc. • Recommended: Yes

- **Anomaly Detection (K-means)**
  - Finds outliers in new data. Good for recognizing unknown states, and to complement classifiers.
  - Author: Edgimpulse Inc. • Recommended: Yes

- **Regression (Keras)**
  - Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.
  - Author: Edgimpulse Inc. • Recommended: Yes
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Live Classification

1 Arduino-CLI

1 Web-USB or EI-CLI

Serial
2. Update the firmware

The development board does not come with the right firmware yet. To update the firmware:

1. Download the latest Edge Impulse firmware, and unzip the file.
2. Open the flash script for your operating system (flash_windows.bat, flash_mac.command or flash_linux.sh) to flash the firmware.
3. Wait until flashing is complete, and press the RESET button once to launch the new firmware.

3. Setting keys

From a command prompt or terminal, run:

```
edge-impulse-desean
```

This will start a wizard which will ask you to log in, and choose an Edge Impulse project. If you want to switch projects run the command with `--clean`.

Alternatively, recent versions of Google Chrome and Microsoft Edge can collect data directly from your development board, without the need for the Edge Impulse CLI. See this blog post for more information.

4. Verifying that the device is connected

That's all! Your device is now connected to Edge Impulse. To verify this, go to your Edge Impulse project, and click Devices. The device will be listed here.
The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT288050.

(base) MacBook-Pro-de-Marcelo:~ mjroval$ /Users/mjroval/Downloads/arduino-nano-33-ble-sense/flash_mac.command
Flash board...
Device : NRF52840-QIAA
Version : Arduino Bootloader (SAM-BA extended) 2.0 [Arduino:IXXYS]
Address : 0x0
Pages : 256
Page Size : 4096 bytes
Total Size : 1024KB
Planes : 1
Lock Regions : 0
Locked : none
Security : false
Erase flash

Done in 0.001 seconds
Write 280848 bytes to flash (69 pages)
[100% (69/69 pages)]
Done in 18.982 seconds

Flashed your Arduino Nano 33 BLE development board.
To set up your development with Edge Impulse, run 'edge-impulse-daemon' to your impulse on your development board, run 'edge-impulse-run-impulse' layout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.

[Process completed]
Alternative option with **Edge Impulse CLI**
Test: terrestrial
Deploy

You can deploy your impulse to any device. This makes the model run without an internet connection, minimize latency, and runs with minimal power consumption. Read more.

Create library

Turn your impulse into optimized source code that you can reuse on any device.

Select optimizations (optional)

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.

Enable EON™ Compiler

Same accuracy, up to 50% less memory. Open source.

Available optimizations for NN Classifier

<table>
<thead>
<tr>
<th>Quantized (int8)</th>
<th>RAM USAGE</th>
<th>LATENCY</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7K</td>
<td>1 ms</td>
<td>19.0K</td>
<td>99.64%</td>
</tr>
</tbody>
</table>

Unoptimized (float32)

<table>
<thead>
<tr>
<th>RAM USAGE</th>
<th>LATENCY</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8K</td>
<td>1 ms</td>
<td>21.3K</td>
</tr>
</tbody>
</table>

Build
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 20 ms., Classification: 1 ms., Anomaly: 1 ms.):
  idle: 0.00000
  lift: 0.00000
  maritime: 0.99609
  terrestrial: 0.00000
**anomaly score: 0.620**

Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 20 ms., Classification: 0 ms., Anomaly: 2 ms.):
  idle: 0.00000
  lift: 0.00000
  maritime: 0.99609
  terrestrial: 0.00000
**anomaly score: 1.470**
Pos-processing

Turn on/off LEDs

- Idle: ==> All OFF
- lift: ==> Green ON
- maritime: ==> Red ON
- terrestrial: ==> Blue ON
- Anomaly ==> LED_BUILTIN ON
void setup()
{
    Serial.begin(115200);
    while (!Serial);

    Serial.println("IEST101 - Nano Motion Classification - Inference Test");
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(LEDRED, OUTPUT);
    pinMode(LEDGREEN, OUTPUT);
    pinMode(LEDBLUE, OUTPUT);

    // Ensure the LED is off by default.
    digitalWrite(LED_BUILTIN, LOW);
    digitalWrite(LEDRED, HIGH);
    digitalWrite(LEDGREEN, HIGH);
    digitalWrite(LEDBLUE, HIGH);

    if (!imu.begin()) {
        ei_printf("Failed to initialize IMU\n\n");
    } else {
        ei_printf("IMU initialized\n\n");
    }

    if (EE_CLASSIFIER_RAW_SAMPLES_PER_FRAME != 3) {
        ei_printf("ERR: EE_CLASSIFIER_RAW_SAMPLES_PER_FRAME should be equal to 3 (t return;
    }

    digitalWrite(LED_BUILTIN, LOW);
    digitalWrite(LEDRED, LOW);
    digitalWrite(LEDGREEN, LOW);
    digitalWrite(LEDBLUE, LOW);
}

void turn_off_leds()
{
    digitalWrite(LED_BUILTIN, HIGH);
    digitalWrite(LEDRED, HIGH);
    digitalWrite(LEDGREEN, HIGH);
    digitalWrite(LEDBLUE, HIGH);
}

void turn_on_leds(int pred_index) {
    switch (pred_index)
    {
    case 0: // Idle: [0] => All OFF
        turn_off_leds();
        break;

    case 1: // lift: [1] => Green ON
        turn_off_leds();
        digitalWrite(LEDGREEN, LOW);
        break;

        turn_off_leds();
        digitalWrite(LEDRED, LOW);
        break;

        turn_off_leds();
        digitalWrite(LEDBLUE, LOW);
        break;
    }

reset()
// Run the classifier
159  ei_impulse_result_t result = { 0 };  
160  err = run_classifier(&signal, &result, debug_nn);
162  if (err != EI_IMPULSE_OK) {
163      ei_printf("ERR: Failed to run classifier (%d)\n", err);
164      return;
166  }

  // print the predictions
168  ei_printf("Predictions \n");
169  ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.)",  
170      result.timing.dsp, result.timing.classification, result.timing.anomaly);
171  ei_printf("\n");
172  int pred_index = 0;
173  float pred_value = result.classification[0].value;
175  for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
176      ei_printf("%s: %.5f\n", result.classification[ix].label, result.classification[ix].value);
178      if (result.classification[ix].value > pred_value){
179          pred_index = ix;
180          pred_value = result.classification[ix].value;
181      }
182  }
183  ei_printf("Prediction: %s with probability %.2f\n", result.classification[pred_index].label, pred_value);
184  #if EI_CLASSIFIER_HAS_ANOMALY == 1
185  ei_printf(" Anomaly score: %.3f\n", result.anomaly);
186  if (result.anomaly > 0.5)
187      digitalWrite(LED_BUILTIN, HIGH);
189  else
190      digitalWrite(LED_BUILTIN, LOW);
192  #endif
label: idle
label: lift
label: maritime
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 20 ms., Classification: 1 ms., Anomaly: 1 ms.):
- idle: 0.00781
- lift: 0.12109
- maritime: 0.87109
- terrestrial: 0.00000

Prediction: maritime with probability 0.87

anomaly score: 0.902

Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 20 ms., Classification: 1 ms., Anomaly: 1 ms.):
- idle: 0.89453
- lift: 0.08984
- maritime: 0.01172
- terrestrial: 0.00781

Prediction: idle with probability 0.89

anomaly score: 0.248
Thanks