

# NPU User's Guide

NXP Semiconductors

November 9, 2022

## Introduction

Model inference using the Neural Processing Unit is currently supported via the eIQ TensorFlow Lite Micro library (see the eIQ TensorFlow Lite Library User's Guide). Standard quantized TensorFlow Lite models has to be converted using the NPU model conversion tool to be prepared for acceleration on the specialized hardware. A custom TensorFlow Lite Micro operator implementation then executes the transformed model nodes using the NPU, while unsupported operations are executed using either the CMSIS-NN or reference operator implementations. See the NPU model conversion tool's documentation (included in the tool's distribution package) for the list of supported operators.

## Model conversion

The NPU model conversion command line tool analyzes a quantized TensorFlow Lite model and transforms the supported operators into specialized NPU nodes. The following is an example of the command line tool arguments for model conversion:

```
neutron-converter --input mobilenet_v1_0.25_128_quant.tflite \  
--output mobilenet_v1_0.25_128_quant_npu.tflite
```

For the complete list of parameters see the conversion tool's documentation included in the distribution package.

## Inference

Running an inference using a model converted for the NPU requires registration of a custom operator implementation. First the header file with the custom operator implementation interface has to be included:

```
#include "tensorflow/lite/micro/kernels/micro_ops.h"  
#include "tensorflow/lite/micro/all_ops_resolver.h"  
#include "tensorflow/lite/micro/kernels/neutron/neutron.h"
```

Next, the specialized implementation has to be registered in the operator resolver object:

```
static tflite::AllOpsResolver microOpResolver;  
microOpResolver.AddCustom(tflite::GetString_NEUTRON_GRAPH(),  
    tflite::Register_NEUTRON_GRAPH());
```

Now the specialized NPU nodes from the converted model will be executed using this newly registered implementation.

## Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2022 NXP

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.