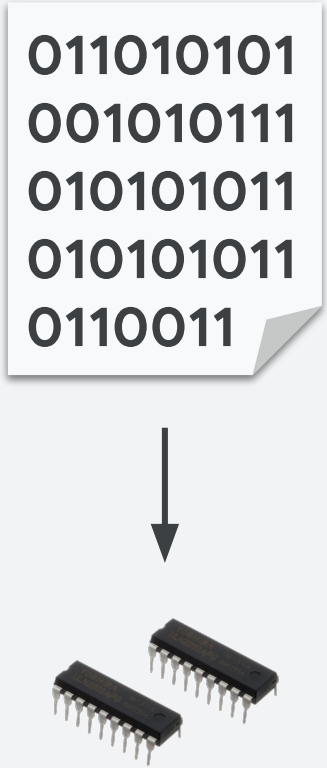
TFLite Micro: NN Operations

The OpsResolver

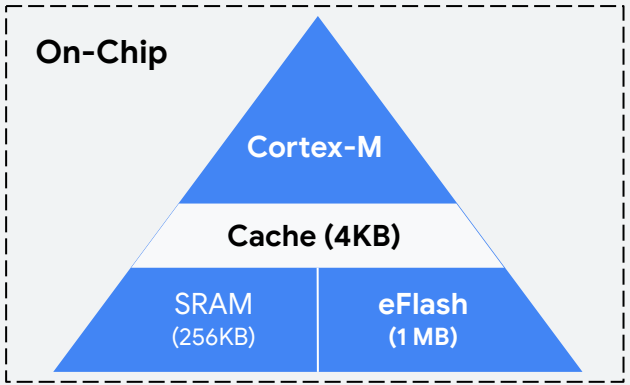
Why Care About **Binary Size**?

- **Executable code** used by a framework takes up space in Flash



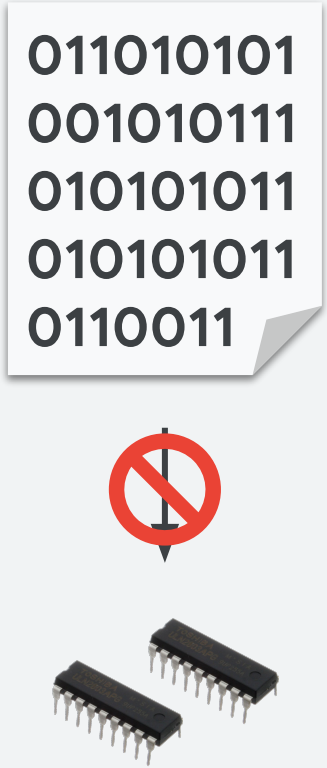
Why Care About **Binary Size**?

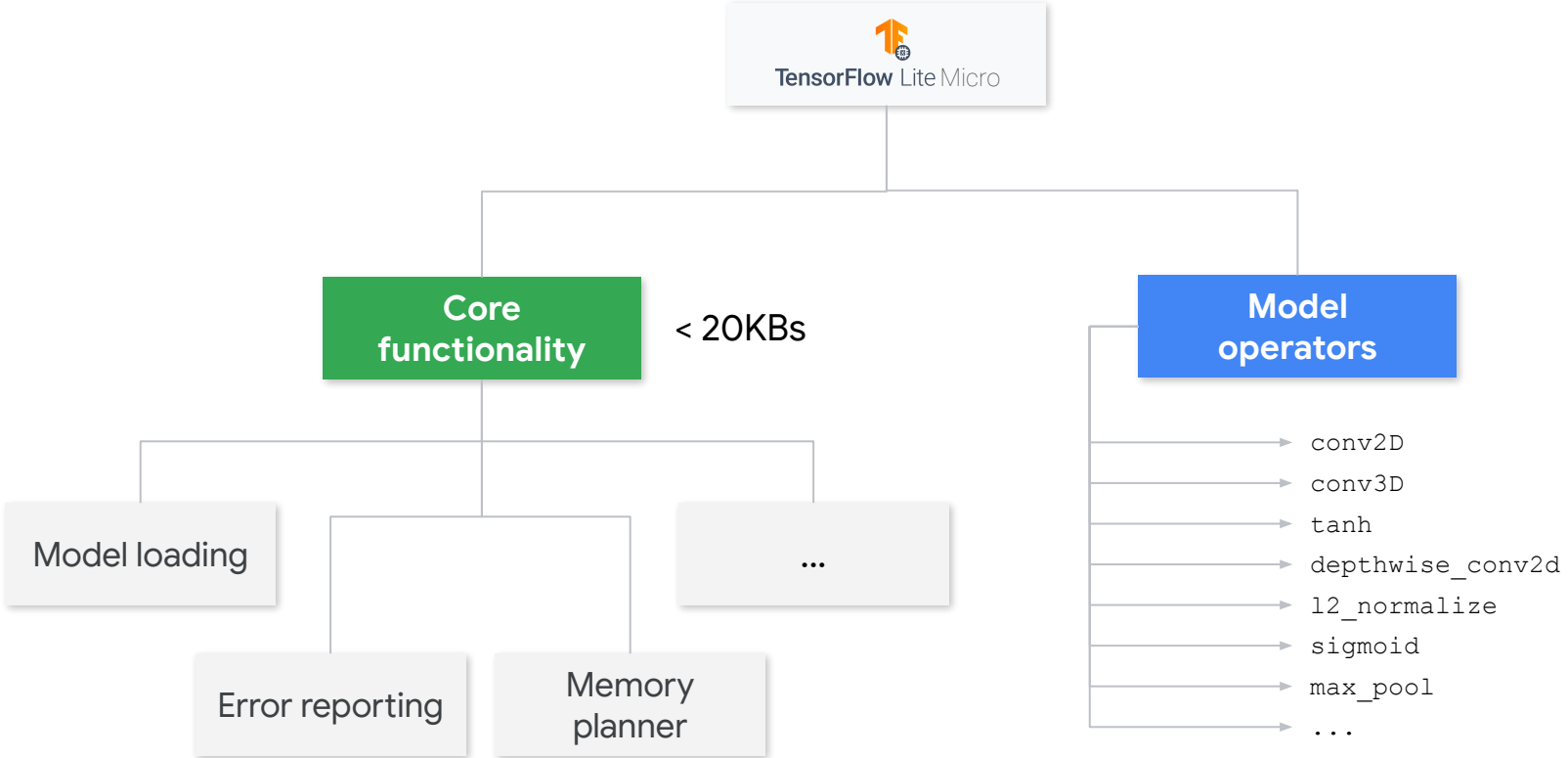
- **Executable code** used by a framework takes up space in Flash
- **Flash is a limited resource** on embedded devices and often just tens of kilobytes available



Why Care About **Binary Size**?

- **Executable code** used by a framework takes up space in Flash
- **Flash is a limited resource** on embedded devices and often just tens of kilobytes available
- If compiled **code is too large**, it **won't be usable** by applications.



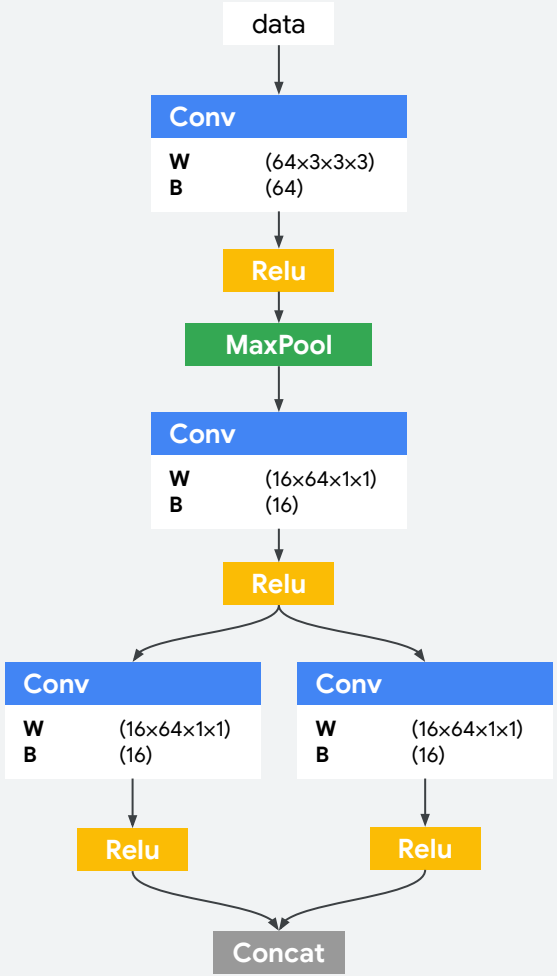

TensorFlow Lite Micro
Core functionality
< 20KBs
Model operators
Model loading
...
Error reporting
Memory planner
conv2D
conv3D
tanh
depthwise_conv2d
l2_normalize
sigmoid
max_pool
...

Optimizing Operator Usage in TFL Micro

- There are **many operators in TensorFlow** (~1400 and growing)

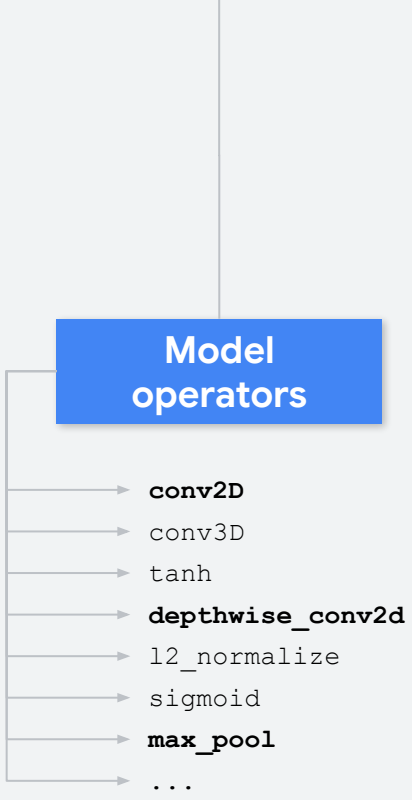
Optimizing Operator Usage in TFL Micro

- There are **many operators in TensorFlow** (~1400 and growing)
- **Not all operators are used** or even needed to perform inference



Optimizing Operator Usage in TFL Micro

- There are **many operators in TensorFlow** (~1400 and growing)
- **Not all operators are used** or even needed to perform inference
- Bring in or **load only those that are important** to conserve memory usage

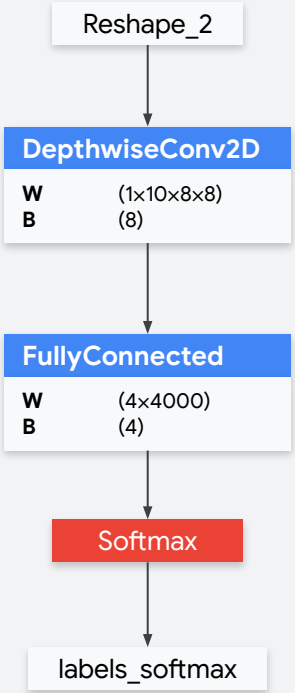


How to **Reduce the Size** Taken by Ops?

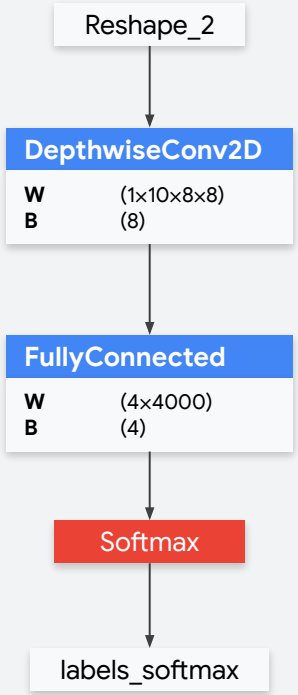
Allow developers to specify which ops they want to be included in the binary

```
tflite::MicroMutableOpResolver<4>
op_resolver(error_reporter);
if (op_resolver.AddDepthwiseConv2D() != kTfLiteOk) {
    return;
}
```

TinyConv Keyword Spotting Model



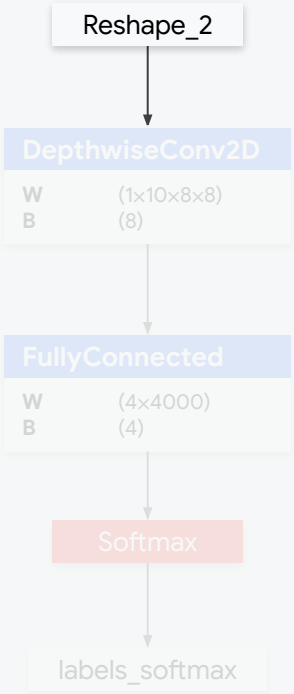
TinyConv Keyword Spotting Model




```
static tflite::MicroMutableOpResolver<4> micro_op_resolver(error_reporter);

if (micro_op_resolver.AddDepthwiseConv2D() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddFullyConnected() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddSoftmax() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddReshape() != kTfLiteOk) {
  return;
}
```

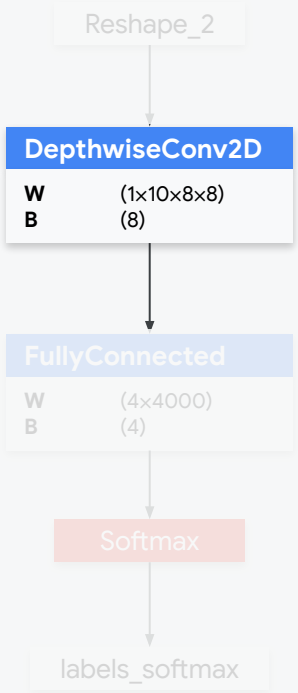
TinyConv Keyword Spotting Model




```
static tflite::MicroMutableOpResolver<4> micro_op_resolver(error_reporter);

if (micro_op_resolver.AddDepthwiseConv2D() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddFullyConnected() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddSoftmax() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddReshape() != kTfLiteOk) {
  return;
}
```

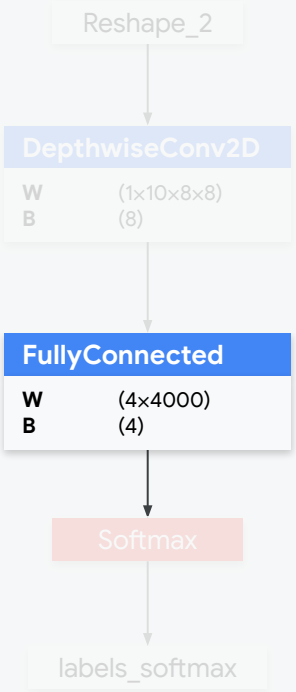
TinyConv Keyword Spotting Model




```
static tflite::MicroMutableOpResolver<4> micro_op_resolver(error_reporter);

if (micro_op_resolver.AddDepthwiseConv2D() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddFullyConnected() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddSoftmax() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddReshape() != kTfLiteOk) {
  return;
}
```

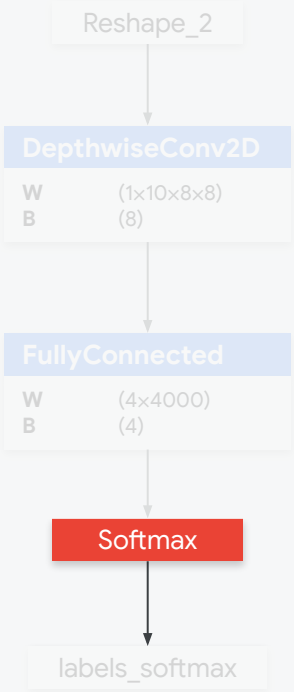
TinyConv Keyword Spotting Model




```
static tflite::MicroMutableOpResolver<4> micro_op_resolver(error_reporter);

if (micro_op_resolver.AddDepthwiseConv2D() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddFullyConnected() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddSoftmax() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddReshape() != kTfLiteOk) {
  return;
}
```

TinyConv Keyword Spotting Model

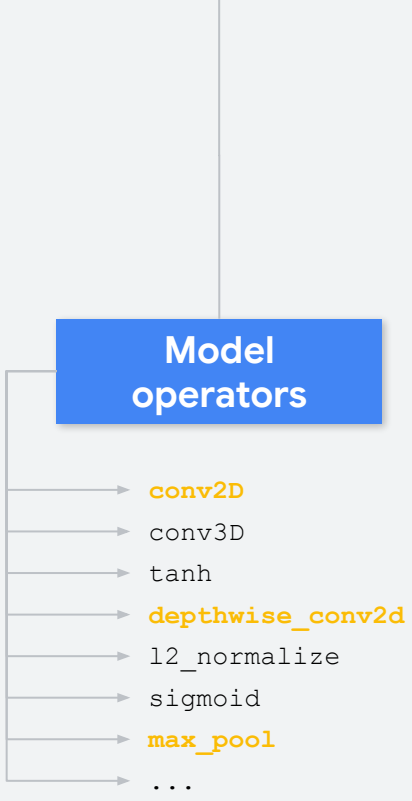



```
static tflite::MicroMutableOpResolver<4> micro_op_resolver(error_reporter);

if (micro_op_resolver.AddDepthwiseConv2D() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddFullyConnected() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddSoftmax() != kTfLiteOk) {
  return;
}
if (micro_op_resolver.AddReshape() != kTfLiteOk) {
  return;
}
```

Optimizing Operator Usage in TFL Micro

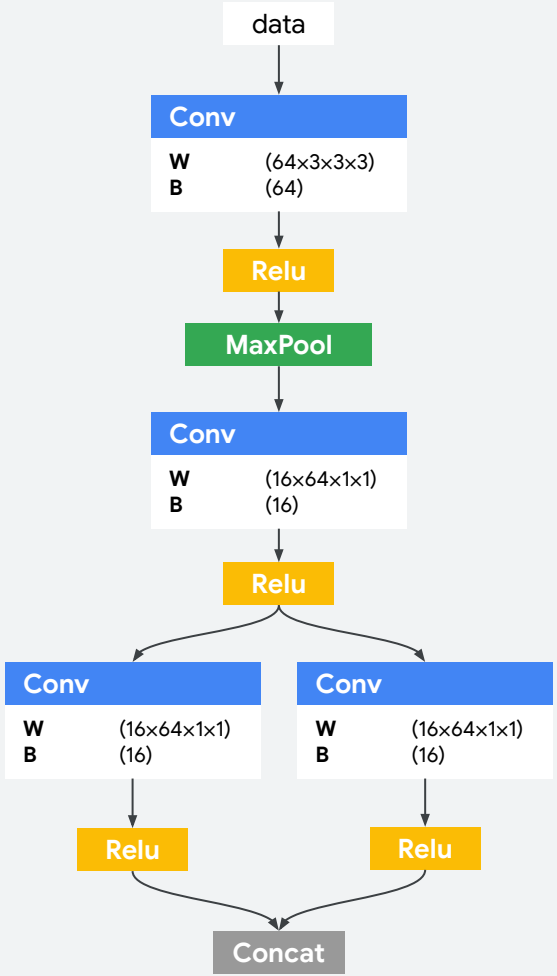
- There are **many operators in TensorFlow** (~1400 and growing)
- **Not all operators are used** or even needed to perform inference
- Bring in or **load only those that are important** to conserve memory usage



Which Ops
to Include?

NETRON

https://netron.app



If memory is not an issue, you can choose to simply include all operators, both used and unused, at the expense of increased memory consumption

```
static tflite::AllOpsResolver resolver;

// Build an interpreter to run the model with.
static tflite::MicroInterpreter static_interpreter(
    model, resolver, tensor_arena, kTensorArenaSize, error_reporter);
interpreter = &static_interpreter;
```

Memory **Improvements**

- Selective op registration **reduces memory consumption by 30%**
- **Memory reduction varies by model**, depending on the operators used by the model