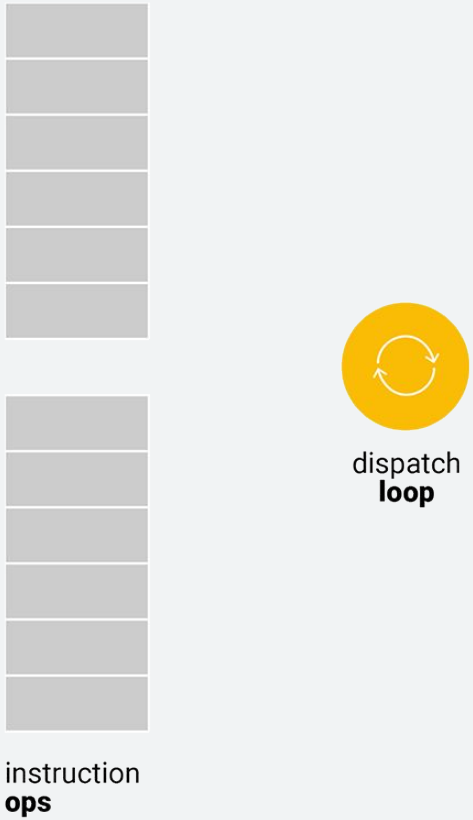
TFLite Micro: Interpreter

TFLite Micro Design

- TFLite Micro uses an **interpreter** design
- Store the model as data and loop through its ops at **runtime**



dispatch
loop

instruction
ops

Interpreter
(generally **slower** than compiled code)

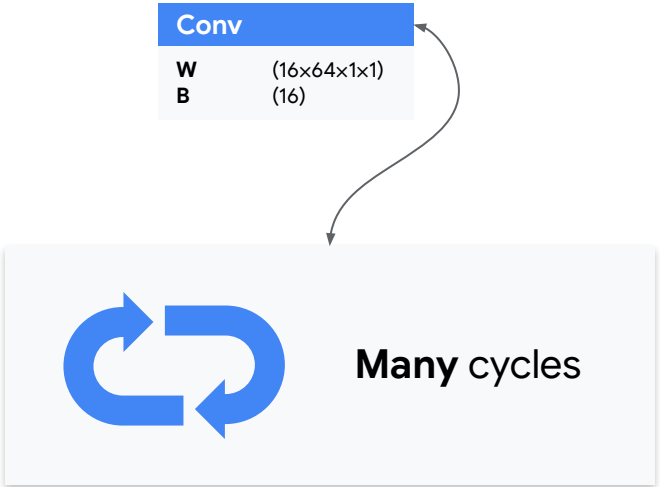
```
int main() {
    function_a();
    function_b();

    printf("done!\n");
}

void function_a() {
    doSomething();
    saveTheWorld();
    machineLearning++;

    printf("a is complete\n");
}

void functon_b() {
    x = 50;
    y = 249;
    z = 141;

    int result = run_conv(x,y,z);

    result += 61;

    printf("b is complete\n");
}
```

C/C++
code

one time
compilation

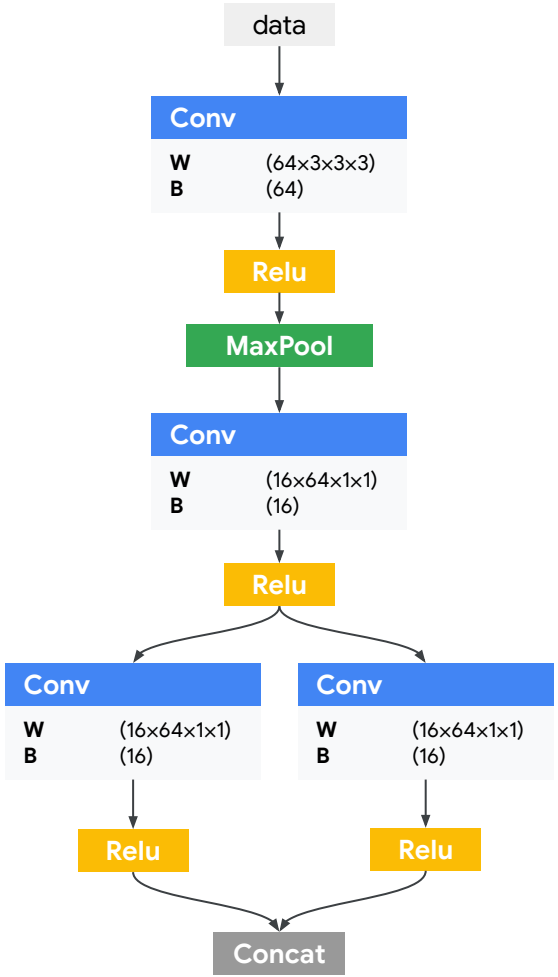
compiled
machine
code

Compiler
(generally **faster** than interpreted code)



ML is **Different**

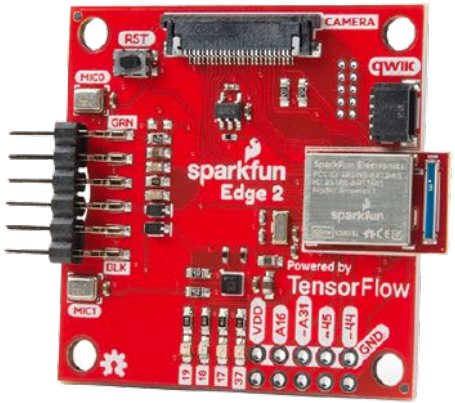
- Each layer like a **`Conv`** or **`softmax`** can take tens of thousands or even millions of cycles to complete execution



ML is **Different**

- Parsing overhead is **relatively small** for the TFMicro interpreter when we consider the **overall network graph**

Model	Total Cycles	Calculation Cycles	Interpreter Overhead
Visual Wake Words (Ref)	18,990.8K	18,987.1K	**< 0.1%**
Google Hotword (Ref)	36.4K	34.9K	**4.1%**



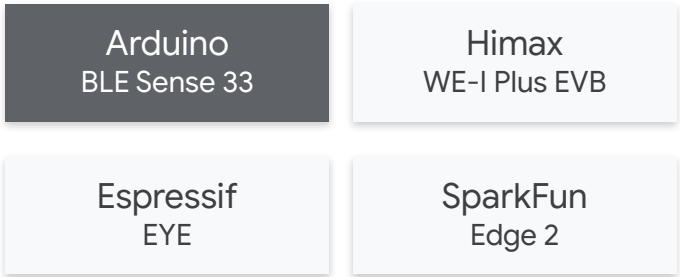
Sparkfun Edge 2
(Apollo 3 **Cortex-M4**)

Interpreter **Advantages**

- Change the model **without recompiling** the code

Interpreter **Advantages**

- Change the model **without recompiling** the code
- **Same operator code** can be used across multiple **different models** in the system



Interpreter **Advantages**

- Same **portable** model serialization format can be used **across a lots of systems**.

TFLite Micro
Interpreter Execution

```
if (op_type == CONV2D) {
  Convolution2d(conv_size, input, output, weights);
} else if (op_type == FULLY_CONNECTED) {
  FullyConnected(input, output, weights)
}
```