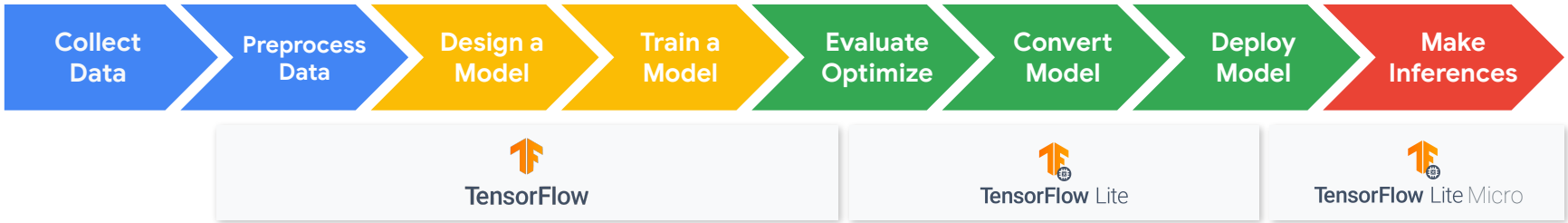
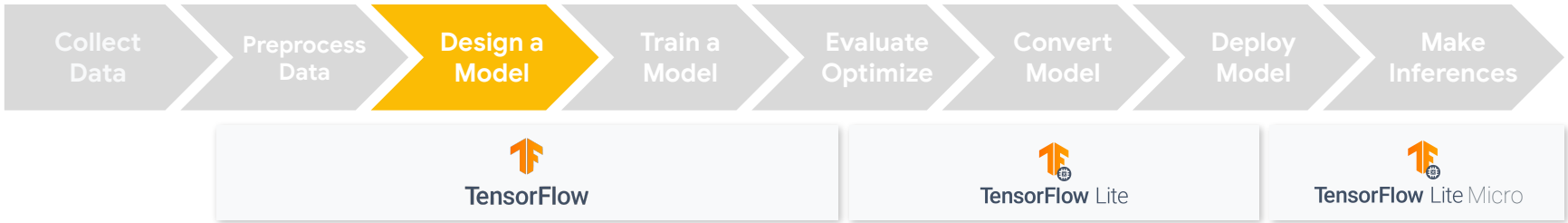
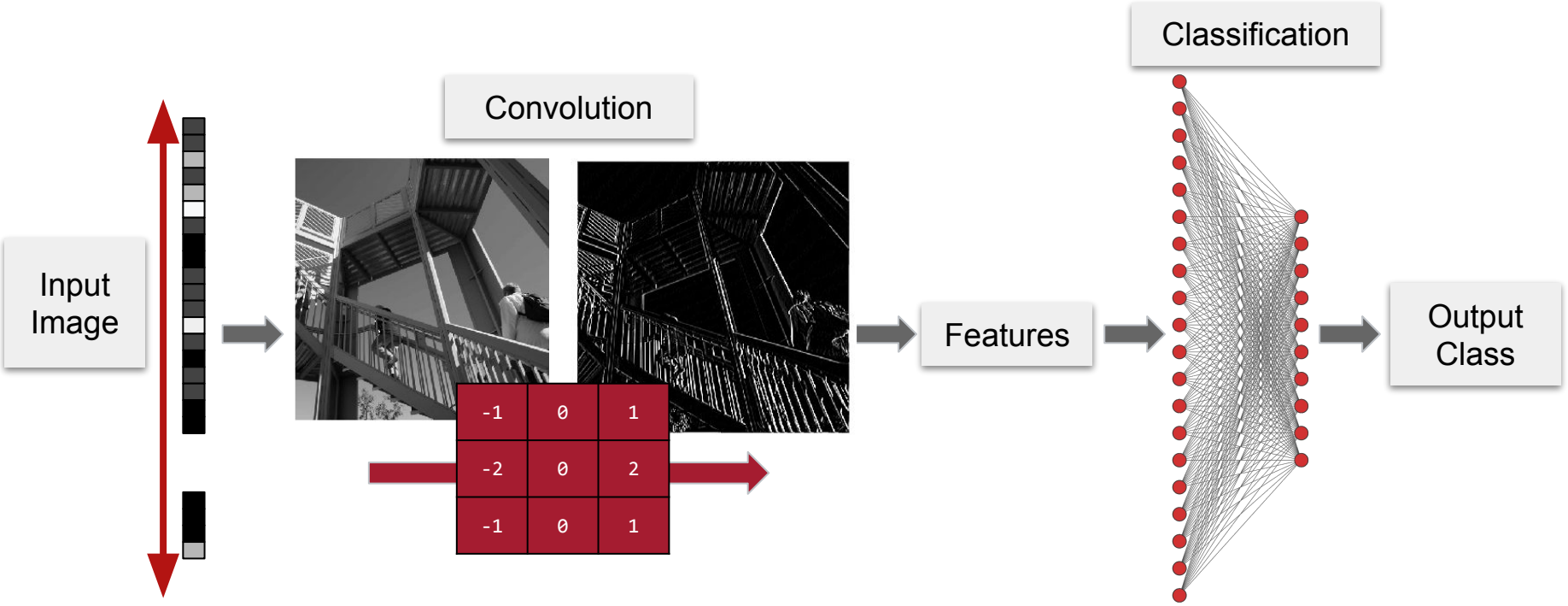
A Keyword Spotting Model

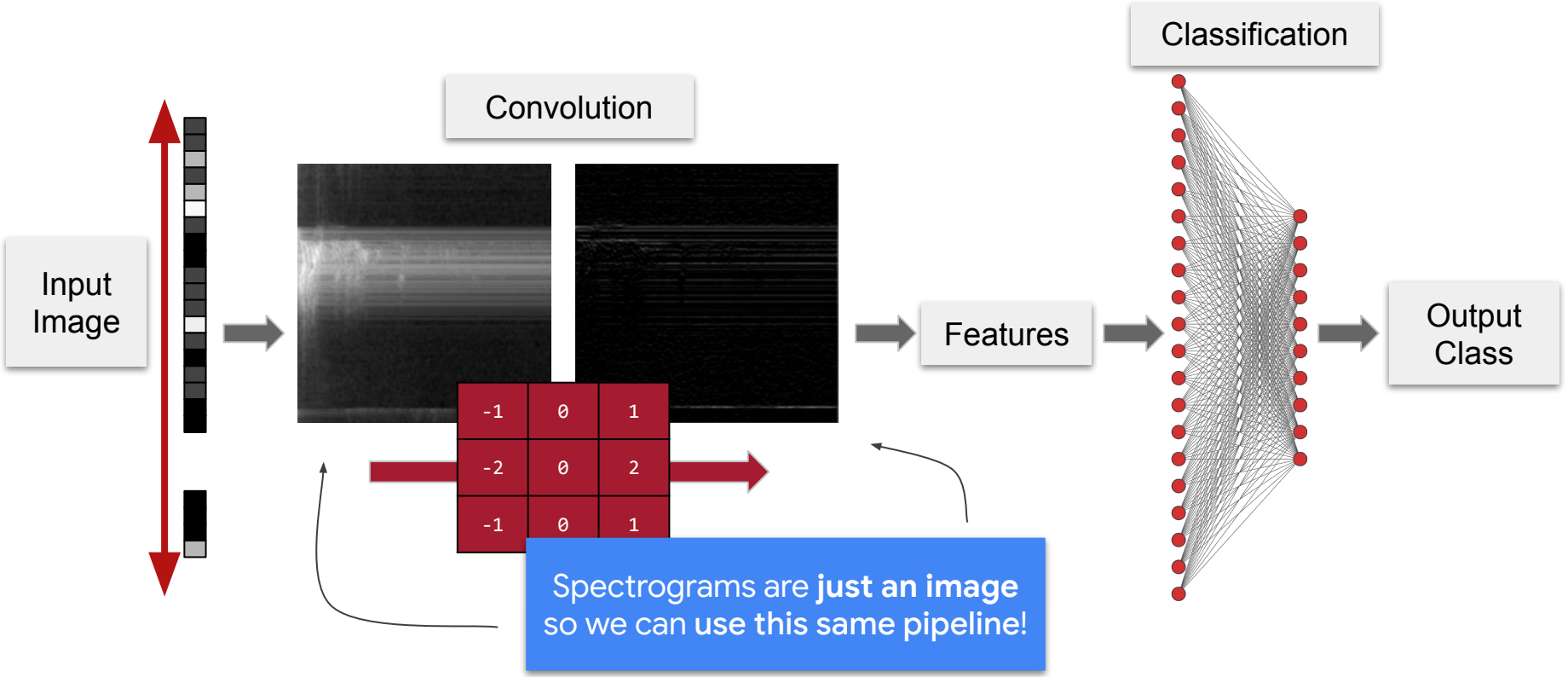

Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro

A model for **Keyword Spotting**

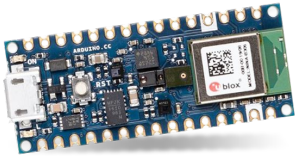


A model for **Keyword Spotting**



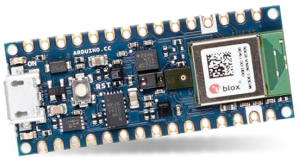
Can we use the **same model** we used in *Course 1* for **Horses v. Humans**?

```
model = tf.keras.models.Sequential([
    # Five convolutional + pooling layers to extract features
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a DNN and classify
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

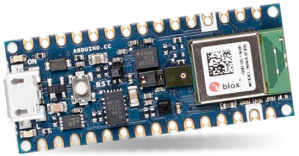
```
# Convert the model to TFLite and print the size
tf.saved_model.save(model, SAVED_MODEL)
float_converter = tf.lite.TFLiteConverter.from_saved_model(SAVED_MODEL)
float_tflite_model = float_converter.convert()
float_tflite_model_size = open(FLOAT_MODEL_TFLITE,"wb").write(float_tflite_model)
print("Float model is %d bytes" % float_tflite_model_size)
```



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

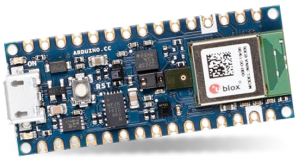
```
# Convert the model to TFLite and print the size
tf.saved_model.save(model, SAVED_MODEL)
float_converter = tf.lite.TFLiteConverter.from_saved_model(SAVED_MODEL)
float_tflite_model = float_converter.convert()
float_tflite_model_size = open(FLOAT_MODEL_TFLITE,"wb").write(float_tflite_model)
print("Float model is %d bytes" % float_tflite_model_size)
```

6822684 bytes ~ 6,800 KB



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

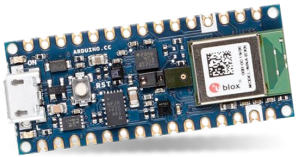
```
# Quantize the model and print the size
converter = tf.lite.TFLiteConverter.from_saved_model(SAVED_MODEL)
converter.optimizations = [tf.lite.Optimize.DEFAULT] # INT8 Quantization
converter.representative_dataset = representative_data_gen # extract test dataset
tflite_model = converter.convert()
tflite_model_size = open(MODEL_TFLITE, "wb").write(tflite_model)
print("Quantized model is %d bytes" % tflite_model_size)
```



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

```
# Quantize the model and print the size
converter = tf.lite.TFLiteConverter.from_saved_model(SAVED_MODEL)
converter.optimizations = [tf.lite.Optimize.DEFAULT] # INT8 Quantization
converter.representative_dataset = representative_data_gen # extract test dataset
tflite_model = converter.convert()
tflite_model_size = open(MODEL_TFLITE, "wb").write(tflite_model)
print("Quantized model is %d bytes" % tflite_model_size)
```

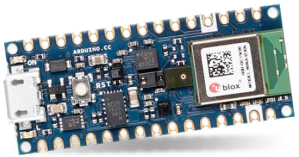
1722272 bytes ~ 1,700 KB



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

```
# Quantize the model and print the size
converter = tf.lite.TFLiteConverter.from_saved_model(SAVED_MODEL)
converter.optimizations = [tf.lite.Optimize.DEFAULT] # INT8 Quantization
converter.representative_dataset = representative_data_gen # extract test dataset
tflite_model = converter.convert()
tflite_model_size = open(MODEL_TFLITE, "wb").write(tflite_model)
print("Quantized model is %d bytes" % tflite_model_size)
```

NEED SMALLER MODEL

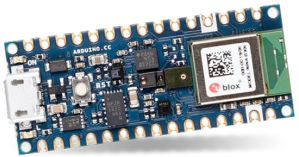


Our board [Course 3 Kit] only has **256KB** of RAM (memory)

TinyConv Model

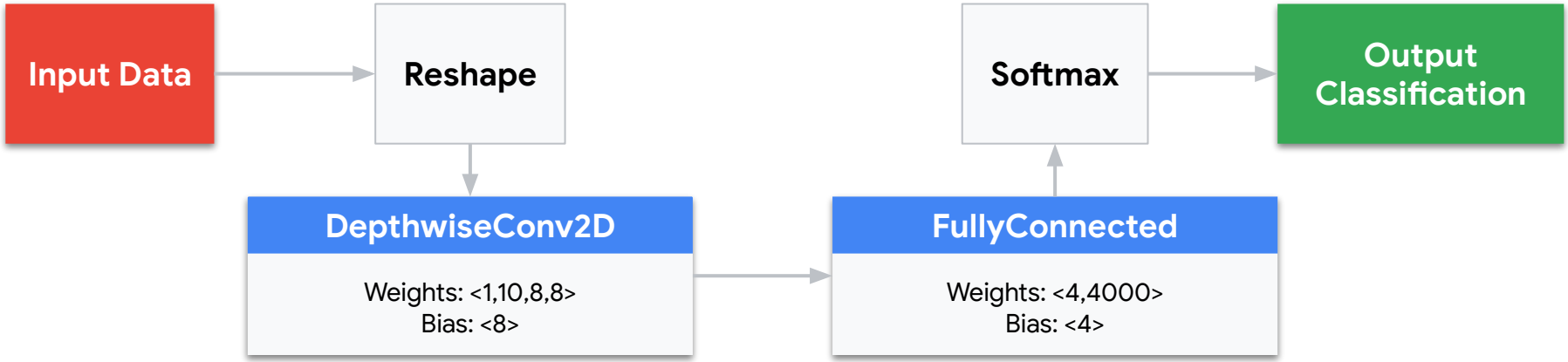
tiny_conv model

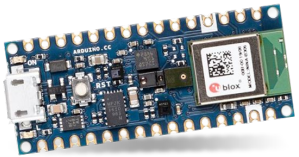
One Conv2D followed by a single Dense Layer!



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

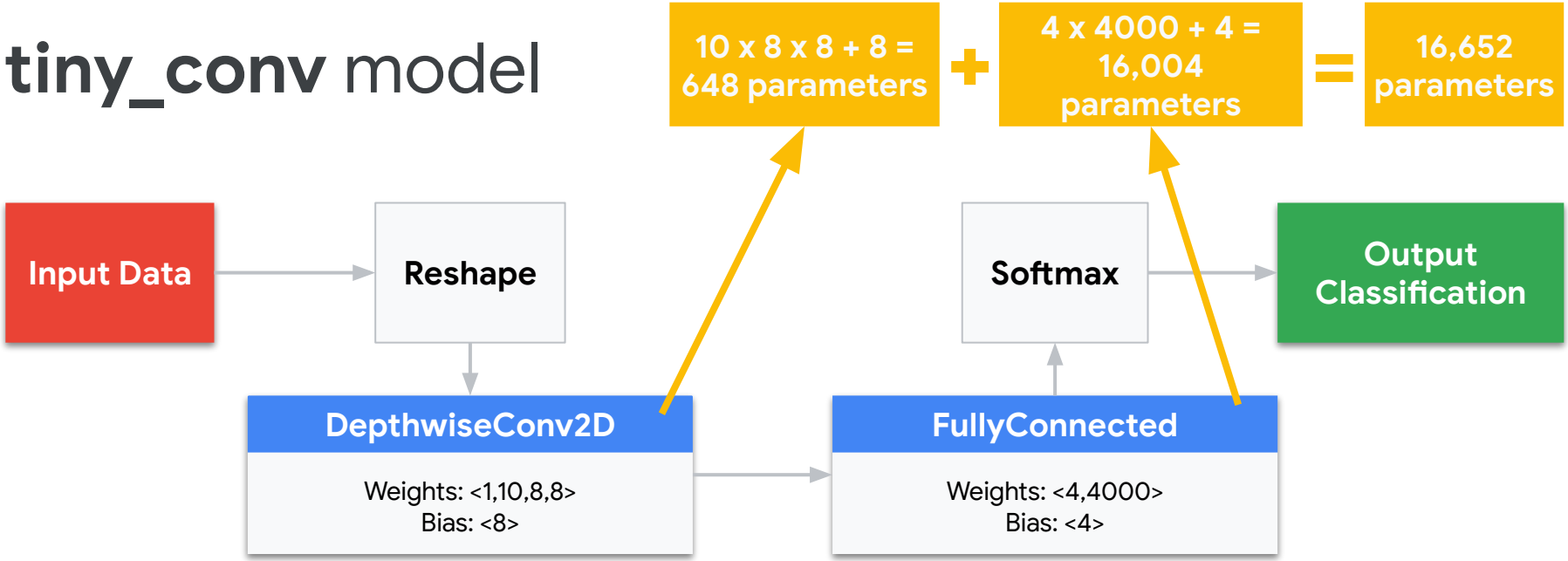
tiny_conv model

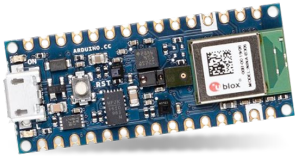




Our board [Course 3 Kit] only has **256KB** of RAM (memory)

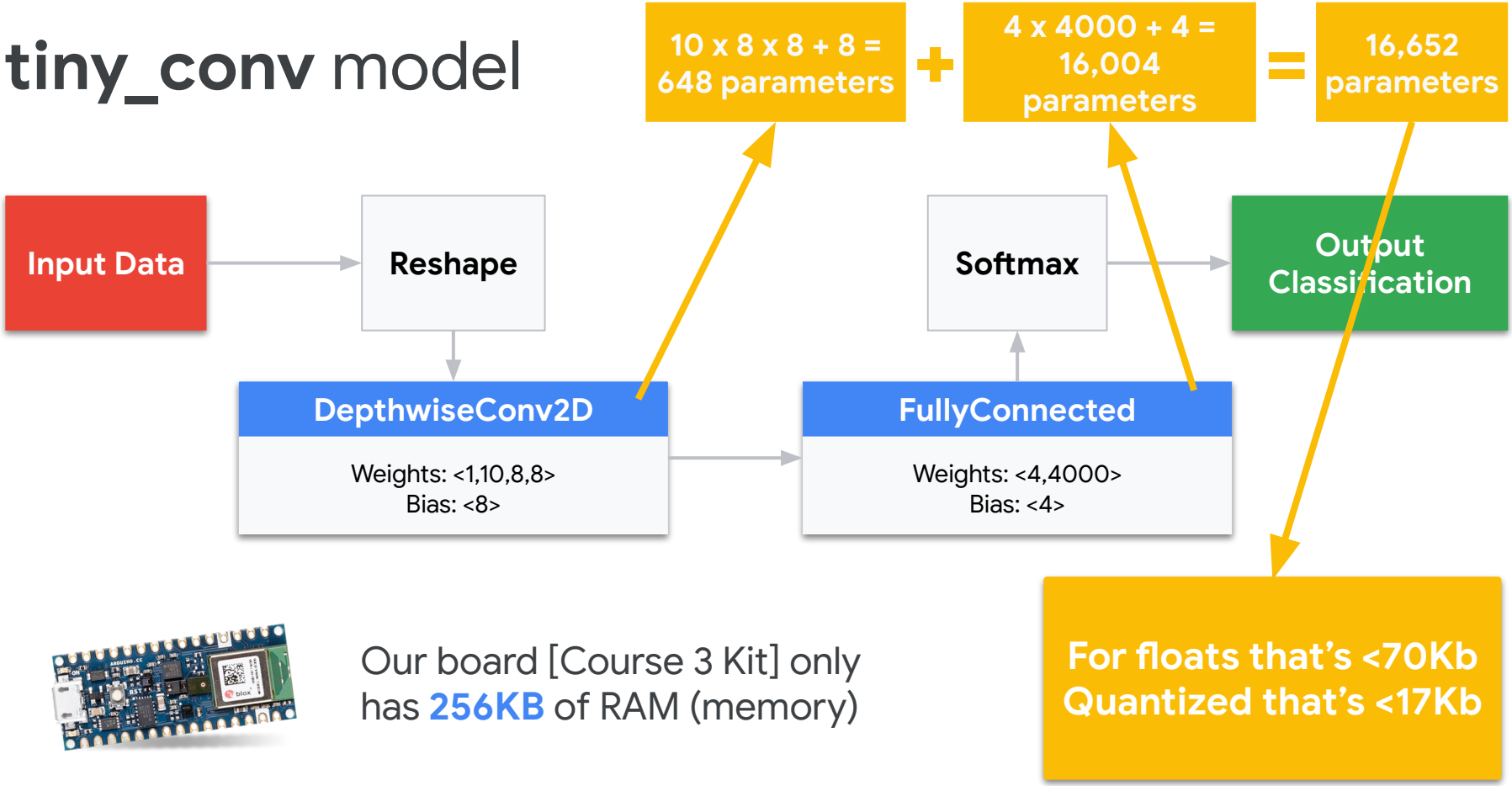
tiny_conv model





Our board [Course 3 Kit] only has **256KB** of RAM (memory)

tiny_conv model

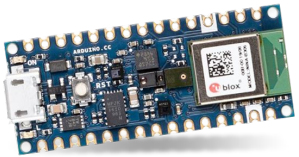


Our board [Course 3 Kit] only has **256KB** of RAM (memory)

tiny_conv model

One Conv2D followed by a single Dense Layer!

<70 KBs for float and only <17 KBs quantized



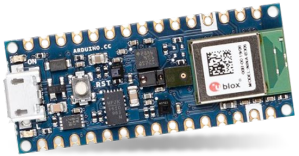
Our board [Course 3 Kit] only has **256KB** of RAM (memory)

tiny_conv model

One Conv2D followed by a single Dense Layer!

FAST INFERENCE

<70 KBs for float and only <17 KBs quantized



Our board [Course 3 Kit] only has **256KB** of RAM (memory)