

1. Which Devices support TensorFlow Lite for Inference? (Check all that apply)

1 / 1 point

☐ RISC

☒ Coral

✓ Correct

☒ Raspberry Pi

✓ Correct

☒ Sparkfun Edge

✓ Correct

2. With a Raspberry Pi, how can you use TensorFlow?

1 / 1 point

☐ Inference Only

☐ Training Only

☐ It doesn't work on Pi

☒ Inference and Training

✓ Correct

3. If you only want to do inference on a Pi, what's the best way?

1 / 1 point

- ☐ Do nothing, the Pi base image has TensorFlow in it
- ☐ Compile all of TensorFlow from Source and run it
- ☐ Install the full TensorFlow with Pip install
- ☒ Install the standalone interpreter using pip

 **Correct**

4. When using ImageNet on a Raspberry Pi for Image Classification, how many classes are supported?

1 / 1 point

- ☐ 800
- ☐ 100
- ☐ 500
- ☒ 1000

 **Correct**

5. How do you initialize the standalone interpreter in Python?

1 / 1 point

- ☐ `tf.lite.load(saved_model)`
- ☒ `tf.lite.Interpreter(directory_of_lite_Model)`
- ☐ `tf.lite.load(lite_model)`
- ☐ `tf.lite.Interpreter(directory_of_saved_model)`

6. How do you get the input tensors for a model with the standalone interpreter?

1 / 1 point

- ☐ Call `get_input_tensors()` after initializing the interpreter
- ☐ Call `get_input_tensors()` after calling `allocate_tensors()` on the interpreter
- ☐ Call `get_input_details()` after initializing the interpreter
- ☒ Call `get_input_details()` after calling `allocate_tensors()` on the interpreter

 **Correct**

7. How do you perform inference using the interpreter?

0 / 1 point

- ☐ Call `invoke()`, and pass it the input tensor
- ☒ Just call `invoke()`, TensorFlow can do the rest
- ☐ Call `invoke()`, and pass it both the input and output tensors
- ☐ Set the Input tensor with the `set_tensor` command and then call `invoke()`

 **Incorrect**

8. How do you read the results of inference using the interpreter?

1 / 1 point

- ☐ Call `invoke()`, pass it the input and output tensors, and then read the output tensor
- ☐ Call `invoke()`, pass it the input tensor, read the results
- ☒ Call `invoke()`, and then call `get_tensor()` on the interpreter to read the output
- ☐ Call `invoke()`, and the the output will be rendered automatically