What is TF Lite Micro?

As an engineer, it is important to understand (or at least have a good idea), about the inner workings of the software you use to know its capabilities and limitations. In the video (MLSys 2021: TensorFlow Lite Micro TFLM) and paper (TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems), you will learn more about the challenges that led to the development of TF Lite Micro. Professor Vijay Janapa Reddi, faculty member at Harvard and also of the TensorFlow team at Google, based on the work of Pete Warden from Google, who leads the team that works on TF Lite Micro, will introduce TF Lite Micro and give us a sneak peek into its internal workings.

TensorFlow has become the most popular deep learning framework, superseding other popular frameworks such as PyTorch and Keras. TensorFlow, developed by Google, contains a Python frontend with highly optimized C++ code at its core, making it simple to program, fast, and efficient. The library has a large developer community and is now seen as the *de facto* standard for most machine learning applications.

Despite this, TensorFlow is not suitable for every scenario. The standard TensorFlow library is ~400 MB in size, and even running a relatively small model (e.g., 200 MB) can take up a considerable amount of random access memory (> 1 GB). Such large storage and memory requirements make running simple models on lightweight systems largely intractable.

Recognizing this issue, Google developed a more lightweight framework, TensorFlow Lite, also sometimes referred to as TensorFlow Mobile. The TFLite binary is approximately 1 MB in size, considerably more compact than the original library, making it possible to run deep learning models on mobile devices such as smartphones and small computers as the Raspberry Pi. This compression was achieved by removing superfluous functionality that is largely unnecessary for mobile deployment.

While this is an improvement, our problem still remains: even TFLite is not suitable for every scenario. Many important deep learning applications exist at the microcontroller-level, which are significantly more resource-constrained than mobile devices, often equipped with less than 1 MB of storage and 256 KB RAM. Clearly, deploying TFLite models is not feasible for microcontrollers, so an alternative solution was needed.

TF Lite Micro takes the compression of the TensorFlow library to the extreme, removing all but essential functionality. In fact, the core runtime of the library takes up only 16 KB, several orders of magnitude smaller than TFLite. With such a small memory footprint, this lightweight framework makes it possible to deploy deep learning models on the smallest of microcontrollers, such as an Arduino Nano.

However, this is not without its complications. Deploying models with TF Lite Micro is fraught with new and unique challenges when building models. For example, since all functionality for plotting and debugging is removed, troubleshooting model issues is

difficult. Additionally, since many microcontrollers do not have floating point units or use 8-bit arithmetic, the model weights and activations must be suitably quantized on the microcontroller system. Since model training requires near machine precision to perform gradient descent, this largely precludes on-device training. Thus, TF Lite Micro models must first be trained on a device with greater computational resources before being ported to the microcontroller, adding an additional stage to the machine learning workflow.

Despite this, the benefits provided by TF Lite Micro - the ability to perform machine learning inference on microcontroller devices - far exceed the challenges, heralding a new era of machine learning that is often referred to as tiny machine learning.