Course Summary

Congratulations on completing the course! You are now equipped with all of the tools necessary to understand and approach the scaled deployment of TinyML applications with MLOps! To see just how far we have come, let's review the key things we learned throughout this course.

This course focused on MLOps in the context of TinyML. Because MLOps is becoming an important aspect of applied machine learning in industry, analogous to DevOps for the software engineering world, we had to differentiate between MLOps for BigML and TinyML. We defined BigML in terms of hardware, software, and network capabilities, such as having access to memory off-chip, using traditional software workflows, and utilizing cloud infrastructure. In contrast, TinyML involves resource-constrained devices that operate at the edge of the cloud, and the tiny memory footprint precludes the use of traditional software workflows, stimulating the need for novel lightweight frameworks such as TensorFlow Lite Micro.

These differences between BigML and TinyML presented novel challenges in the implementation of TinyMLOps. For example, continuous training becomes largely untenable due to (1) the inability to store data on-device and (2) the lack of numeric precision (e.g., lack of a floating point unit as part of the processor architecture) required to propagate gradient terms during the training process. The limited communication also presents additional challenges, since communication processes are power-intensive, making it costly to stream data to a centralized storage unit. Because of these limitations, novel techniques needed to be developed to combat these issues, such as the use of federated learning across a network of devices. Further challenges were identified such as scalability and interoperability, which led to novel methods such as TinyMLaaS.

To this end, throughout the course we learned both how traditional MLOps works for BigML applications, and how these MLOps approaches are translated into the world of TinyML. Based on the challenges we identified above, we presented the current state-of-the-art solutions that TinyML engineers use to approach these problems, covering the gamut from training operationalization and model development through to prediction serving and continuous monitoring. Since TinyML is a proto-engineering field, it is inevitable that these tools and techniques will change over time as the field evolves and advances. However, knowing the current approaches and their corresponding challenges will make it easier to rationalize and understand further advancements in the field - who knows, maybe you will be the one producing those advancements!

Whether you took this course to develop new skills for your career, to enrich your resume, or just for personal development, finishing this course is no small feat. Whatever the case, we hope that this will just be the start of your journey deploying TinyML applications at scale and that this course will act as a stepping stone for many future advancements in TinyML. We wish you success in your future careers, and remember - the future of ML is tiny and bright.