Accelerated Training in TensorFlow

When prototyping neural networks, training models can be the biggest drain of time. Some methods to speed up training are obvious, such as increasing the batch size or decreasing the number of epochs. However, both of these methods run the risk of reducing the performance of the network. Following are a number of methods to speed TensorFlow training without reducing the accuracy of the network:

1. Build Tensorflow from the source.

   The pip installation of Tensorflow is the most convenient to install, but the version you are installing is designed to run on many different computer hardware and not your specific computer. By building Tensorflow from the source, you can optimize it for the way your computer works. For instructions, see here: https://www.tensorflow.org/install/source

2. Use Accelerated Linear Algebra (XLA)

   XLA is a very simple speed-up for many models and improves training speed by 10-20%. It does so by compiling the graph in model-specific computation kernels, instead of the all-purpose kernel normally used by TensorFlow. To enable it in your model, simply run the following command after importing TensorFlow:

   tf.config.optimizer.set_jit(True)

3. Use Mixed-Precision Variables (GPU Specific)

   Unless otherwise specified, TensorFlow will default to float32 variables in training. This is not always necessary, so allowing TensorFlow to switch between float32 and float16 can speed training. To do this, you can add these lines of code to your code before you create your model:

   policy = tf.keras.mixed_precision.experimental.Policy('mixed_float16')
   tf.keras.mixed_precision.experimental.set_policy(policy)