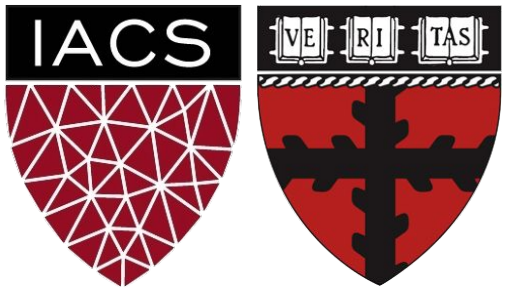
Lecture 5: Data - TF Data, TF Records

Advanced Practical Data Science, MLOps

AC215

Pavlos Protopapas
Institute for Applied Computational Science, Harvard



Communication

- Exercise 1 was due before class today.
- Exercise 2 released today and due 09/23 2 PM
- Submit Milestone 1 on Github by Friday 09/17
- Any project group concerns email helpline - ac215.fall2021@gmail.com

Outline

1. Some of your questions
2. Complete Tutorial: TF Data & TF Records
3. Start on Exercise

Some of your questions

How do we train the model on multiple GPUs? The figure on slide 42 suggests that each processor is working on one batch asynchronously. Don't we have to wait for the model weights to be updated using the results from the first batch of data before starting the training on the second batch?

When training a model on multiple GPUs there are two scenarios. Slide 41: Single machine(node) , multiple GPUs and Slide 42: Multi machine (Node), multi GPU
In both these scenarios almost all of your TensorFlow code (TF Data + Model) will remain the same. The additional code you would need is to use a distribution strategy using tf.distribute.Strategy, read more on it here(https://www.tensorflow.org/guide/distributed_training & https://keras.io/guides/distributed_training/)

After the CPU sends data to the GPU, is that memory that was used for the data immediately cleaned and replaced with the next round of pre-fetched data?
Once CPU passes all of the data to either on GPU or to multiple GPUs then CPU will release the memory and start working on the next batch. But the CPU will need to still keep track of some metadata on which mini batch went to which GPU and other metadata to manage training

Some of your questions

Are the parameters sent back from the GPU to the CPU as an .hdf5 file?

No .hdf5 file is only used when the model needs to be saved on disk. The CPU has access to the GPU memory and will load the model parameters and instructions on what to execute.

Is there a tf.data, tf.records equivalent in PyTorch? Can these be used with PyTorch?
Pytorch has DataLoaders, they aren't like TF Records. You can use TFRecords with Pytorch also.

I do hope there is a bit more detail about what you expect on Friday for Milestone 1. I see the list of things on canvas, but don't know that we have enough to go on to really know how much we should be writing here and how do we gauge how long things will take or the schedule. I'm sure we'll figure out something!
1-2 page document should suffice for Milestone1. Please feel free to ask on Ed, if this isn't clear.

Some of your questions

How does TFRecord exactly improve the speed of I/O?

Consider we have 1000 images in our dataset. We need to read 1000 files for every epoch. So 1000 i/o. Now if we “zip” these files into chunks of 100 images each then we have 10 TF record files. So Tensorflow data need only read “10” files instead of 1000. So TF Records reduces the number of I/O and speeds up the overall data reading. It does not speed up a single I/O

Just to confirm, using TFRecords implies that we are training multiple models based on different subsets of the training data, correct?

No, TF Records is just an efficient file format that is used by TF Data to read data efficiently during training. Now training can be to one GPU or multiple GPUs.

Outline

1. Some of your questions
2. Complete Tutorial: TF Data & TF Records
3. Start on Exercise