TensorFlow Report

Group 21

1 TF and Spark

TensorFlow is a library to improve the performance of computation and therefore by natural a great choice for machine learning tasks. In this case, we can run 1000 iterations within a few seconds with the sparse optimization it provides. Spark, on the other hand, is not specifically designed for machine learning tasks. It provides good fault tolerance, batch operation performance and scalability.

Pros of TF:

1 It supports matrix, vector operation, compatible with numpy.
2 We can develop application in both sync and async style.
3 It has optimization for sparse data, which is a common case in machine learning tasks.
4 Many ML libraries built on top on TF such as Keras.

Cons of TF:

1 The computational graph is python, so that it is slow compared to other popular frameworks such as PyTorch.

Pros of Spark:

1 Spark supports many types of workloads besides machine learning and iterative algorithm, for example, interactive queries and streaming.
2 All the operations iterate in batches, make it faster.

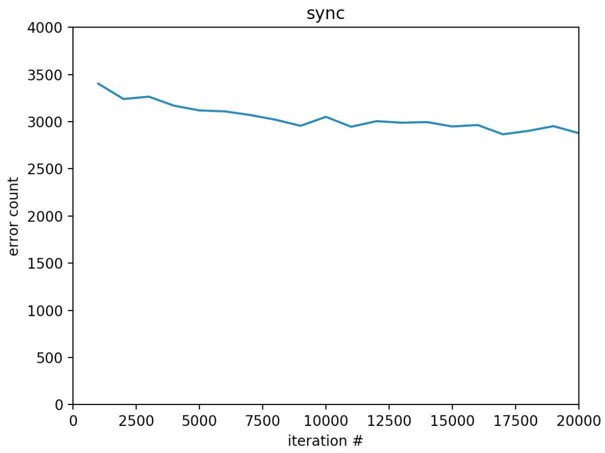
Cons of Spark:

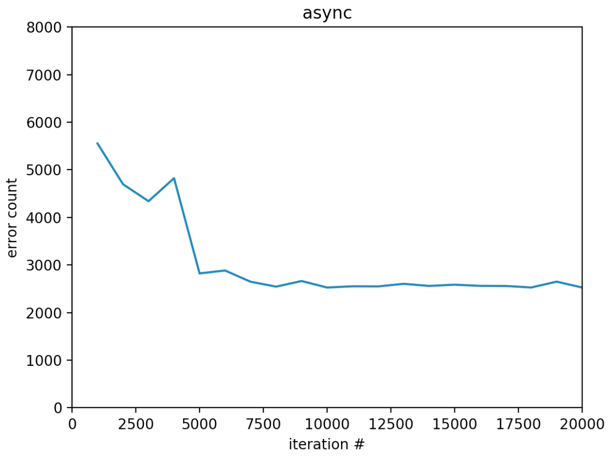
1 Does not allow users to write async application.

2 Plotting

It took us a few seconds to run about 1000 iterations. Although the instruction indicates to run 2e7 iterations, we only run 4e4 because it can be way faster and demonstrate the result as well. We applied to the test set to get error every 1000 iterations, testing size size is 1e4.

The plot can be found as below, it is clear that the model is converging as the error decreases. However the async mode faces a decresing in error rate rapidly when start, while the sync mode start from a rather low error rate and decreate gradually.





3 Bottleneck

Please find the stat for running 1e4 iterations below.

	CPU util	Network read	Network write
sync	100%	140M	140M
async	70%	182M	182M

We can conclude that the bottleneck for sync is **CPU**, whereas bottleneck for async mode is **network IO**. It makes sense because in sync mode, we wait for the straggler and then compute the gradients of data points from all workers. Whereas in async mode, each worker computes and transmits as fast as they can.

4 Extra credit

The only change of batch mode and single point mode is whether to parse and process the input data in batches.

Please find the stat of 2e4 iterations of sync mode, 1e5 iterations of async mode below. Note that all the tests are skipped for convenience. However, since our master node does not work after we tried every possible solution including rebooting the machine(by asking our nice TA), we only have 4 workers.

	single	Batch of 100	Batch of 1000	Batch of 10000
sync	128s	78s	73s	71s
async	140s	80s	78s	79s

Obviously, batching the data can fasten the training. The batch size can also influence the runtime. In sync mode, it seems the 1e4 batch size used the shortest time. Whereas for async mode, 1000 seems to the best size. However, since the iteration number is only 2e4 and it is relatively small, the runtime can be influenced heavily by overhead.

As a result, I believe there is a trade off and the fastest batch size can vary on different problems. We did not dive deeper on the influence of batch sizes to error rate.