Parallel Computing in Large Scale CNN Training

Convolutional Neural Network has been used widely in fields ranging from computer vision to climate analytics and astrophysics. It is now common to employ clusters of GPUs to train a single model. Training a large scale CNN takes days or weeks to converge. Current approaches to accelerating CNN training focus on distributed data-parallelism. In the paper, the author proposes new algorithms to perform convolution with distributed input/output channel/filter data.

The normal algorithm for training a CNN includes three phases: convolutional forward, backward filter and backward data. Mathematically represented as y=sum(x*w); dl/dw =sum(dl/dy*x); dl/dx=sum(dl/dy*w).

The first algorithm is called stationary-x. This algorithm does forward propagation locally, then performs a segmented reduce-scatter among each set of processors that has different channels for the same sample and spatial region, and then product the correct distribution for y. Back propagation begins with a segmented all gather to assemble the filters of dl/dy. Backward data then be computed locally and backward filter can be computed partially locally.

The second algorithm is called stationary-y. This algorithm performs all gather at the beginning of forward propagation and the do location convolution calculation. The backward data step is performed locally and then a reduce-scatter to complete the sum over filters. The backward filter computation is similar to stationary-x except the segmented allreduce aggregates gradient updates among processors with the same filters.

The third algorithm is called stationary-w. This algorithm first does a segmented all gather of the channels of x such that they match the channel distribution of w. Forward propagation is performed locally, and then a segmented reduce-scatter completes the summation and scatters data back over the processor grid. The backward-data phase is symmetric. The backward-filter phase first does a local computation and then aggregates the updates among corresponding processors.

The author uses the supercomputer Lassen which consists of 795 nodes each with two IBM POWER9 CPUs and four NVIDIA V100 GPUs with

NVLinks. The supercomputer is now located in Lawerence Livermore National Laboratory.

For the strong scalability, they test on the con_3 and con_5 layer and find that the scaling is sub-linear. And the forward propagation tends to scale better than back propagation.

For the weak scalability, they find in forward propagation ti works excellent. And it back propagation, the channel parallelism again outperforms sample parallelism.