Presentation Abstract

Formal Approach to Editing a Tensorflow Computational Graph for Large Model Support

Tung D. Le[1,a)] Haruki Imai[1,b)] Yasushi Negishi[1,c)] Kiyokuni Kawachiya[1,d)]

Presented: November 1, 2018

Deep neural networks are becoming larger and their training consumes a huge memory space. While accelerators such as GPUs are suitable for training neural networks, they have limited memory. Meanwhile, host memory that is about 32 times bigger than GPU memory is not fully utilized during training. Moreover, modern IBM machines for AI are integrated with NVLinks that provide very fast connection between CPUs and GPUs. This motivates us to propose a new method to fully utilize host memory as well as NVLinks to support training very large models. In this presentation, we present a formal method for rewriting the computational graph of a neural network, in which swap-out and swap-in operations are inserted into the graph for temporarily storing intermediate results on CPU memory. In particular, we first revise the concept of a computational graph in TensorFlow by defining a concrete semantics for variables in a graph. We then formally show how to derive swap-out and swap-in operations from an existing graph, and finally present rules to optimize the graph. To show the advantage of our method, we trained a neural network, 3DUNet, for detecting brain tumors. We used an IBM Power8 machine coupled with a NVIDIA Tesla P100 GPU (16 GB memory). Power8 is directly connected to the GPU by 80 GB/s duplex links (NVLinks). We were able to train 3DUNet using four 3D images of size of 192^3 per mini-batch. Meanwhile, the vanilla TensorFlow 1.8 were only able to train 3DUNet using one 3D images of size of 144^3 per mini-batch.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

[1] IBM Research - Tokyo, Chuo, Tokyo 103–8510, Japan
[a)] tung@jp.ibm.com
[b)] imaihal@jp.ibm.com
[c)] negishi@jp.ibm.com
[d)] kawatiya@jp.ibm.com