Large Model Support for Deep Learning in Caffe and Chainer

Minsik Cho
IBM T. J. Watson Research Center
Yorktown Heights, NY
minsikcho@us.ibm.com

Tung D. Le
IBM Tokyo Research Lab
Tokyo, Japan
tung@jp.ibm.com

Ulrich A. Finkler
IBM T. J. Watson Research Center
Yorktown Heights, NY
ufinkler@us.ibm.com

Haruiki Imai
IBM Tokyo Research Lab
Tokyo, Japan
imaihal@jp.ibm.com

Yasushi Negishi
IBM Tokyo Research Lab
Tokyo, Japan
negishi@jp.ibm.com

Taro Sekiyama
IBM Tokyo Research Lab
Tokyo, Japan
sekiym@jp.ibm.com

Saritha Vinod
IBM Systems
Bangalore, India
sarithavn@in.ibm.com

Vladimir Zolotov
IBM T. J. Watson Research Center
Yorktown Heights, NY
zolotov@us.ibm.com

Kiyokuni Kawachiya
IBM Tokyo Research Lab
Tokyo, Japan
kawatiya@jp.ibm.com

David S. Kung
IBM T. J. Watson Research Center
Yorktown Heights, NY
kung@us.ibm.com

Hillery C. Hunter
IBM T. J. Watson Research Center
Yorktown Heights, NY
hhunter@us.ibm.com

ABSTRACT

Deep learning is both compute- and data-intense, and recent breakthroughs have largely been fueled by the fp32 compute capacity of modern GPUs. This has made GPUs the prevalent tool for training deep neural networks, but GPUs have only small amounts of costly 3D-stacked HBM DRAM as their local memory. Working out of a small memory imposes a limit on the maximum learning capacity a neural network can have (i.e., the number of learnable parameters) and the maximum size and number of samples a network can consume at a given time. The field of deep learning is evolving in many new directions, and research teams are exploring both very large neural networks and attempting to apply deep learning to real datasets, including high-resolution images. Those exploring both the boundaries of neural networks and use of real datasets today generally will find that their deep learning software won't support what they wish to train, and if it does, they find performace to be intolerably slow. In this paper, we present the idea of large model support, and its implementation in two popular deep learning frameworks, Caffe and Chainer. The key idea is to use GPU memory as an application-level cache w.r.t. the host memory so that a large network (e.g., many parameters or many layers) can be trained with real-world samples (e.g., HD-images). Although our large model support scheme may degrade the performance of training due to the communication overhead between the system CPUs and GPUs, the overhead in general is observed to reduce significantly with the use of a faster communication link between the CPU and GPU (NVLink and next-Gen NVLink). Our experimental results show that our large model support in Caffe and Chainer performs very well, and can train 2 to 6 times larger ImageNet models.

SysML'18, February 2018, Stanford, CA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/xxxxxx

ACM Reference Format:
Minsik Cho, Tung D. Le, Ulrich A. Finkler, Haruiki Imai, Yasushi Negishi, Taro Sekiyama, Saritha Vinod, Vladimir Zolotov, Kiyokuni Kawachiya, David S. Kung, and Hillery C. Hunter. 2018. Large Model Support for Deep Learning in Caffe and Chainer. In *Proceedings of ACM Conference (SysML'18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/xxxxxx

1 INTRODUCTION

Deep learning has become the de-facto technique for an increasing number of cognitive applications, including vision, speech, and language translation [1, 6, 7]. Its success is driven by the availability of an enormous volume of data and advances in deep neural networks, which in turn make deep learning one of the most computationally demanding AI applications [1, 3, 8]. Hardware accelerators like GPUs and their accompanying software stacks have provided a significant amount of speed-up [10]. However, GPUs have a much smaller memory space (12-16GB) due to the expense of HBM DRAM, chip pinout required to drive high memory bandwidth, and wirability of the silicon interposers which carry the DRAM. In contrast, CPUs use a far more scalable type of DRAM memory (DDR3 or DDR4) and can easily have 64-512GB memory capacity. GPUs have had similar memory capacity for the past 2-3 generations, but deep neural network models have gotten deeper and wider to achieve higher learning capacity. For example, [5] proposes a Resnet with 1001 layers, and Neural Machine Translation models [2] get unrolled into a large number of layers. Therefore, a complex neural network which would be perfectly trained on CPUs may never be trained on GPUs due to the limited device memory. Using the full

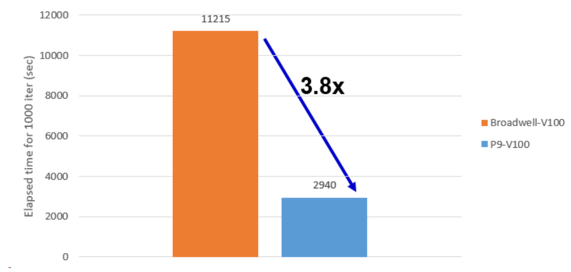

Figure 1: Caffe LMS on 4 V100 GPUs

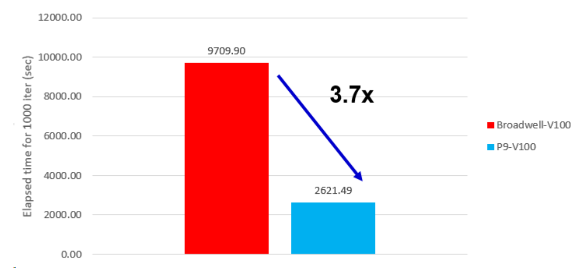

Figure 2: Chainer LMS on 4 V100 GPUs

capability of a system (both the CPU memory and the GPU memory) together, in order to enable deep learning to continue to push boundaires, motivates our large model support (LMS).

The key idea in LMS is to treat GPU memory as an application-level cache w.r.t the host main memory. Basically, all data reside on host memory and are copied to GPU memory only when needed. After the GPU memory is used, depending on whether it has been modified or has future usage, it can be either copied back to CPU memory or simply discarded. To efficiently utilize GPU memory, our LMS implementation keeps a large memory pool so that different memory pieces from CPU memory can share the same GPU memory chunk. Therefore, at any moment, the GPU only holds data necessary to process one operation, for example, the forward propagation of one operation in a neural network. If the memory requirement from any operation is larger than the GPU memory, then even LMS will fail as well. In theory, LMS should be able to handle a deep neural network of an arbitrary capacity, as long as all the data from the largest operation can fit into the GPU memory. A similar idea has been proposed for Tensorflow in [4], but we discuss LMS implementation in Caffe and Chainer and share our results.

2 EXPERIMENTAL RESULTS

We have successfully implemented LMS functionality in Caffe [7] and Chainer [9] as part of the PowerAI deep learning software distribution. We have open sourced our implementations, and they are available at the following github, [12] for Caffe, [11] for Chainer, respectively. To demonstrate LMS functionality, we obtained the results of running 1000 iterations of an enlarged GoogLeNet model (mini-batch size=5) on an enlarged ImageNet Dataset (crop size of 2240x2240) on two platforms:

- POWER9 AC922 system with next-Gen NVLink, CPU at 2.25 GHz with 1024 GB memory, 4x V100-SXM2 GPUs on Red Hat Enterprise Linux 7.4 for Power Little Endian (POWER9) with CUDA 9.1/ CUDNN 7
- Intel Xeon E5-2640 v4 at 2.4 GHz with 1024 GB memory, 4x V100-PCIe GPUs on Ubuntu 16.04. with CUDA .9.0/ CUDNN 7

The key difference between two platforms is the next-Gen NVLink which connects CPUs and GPUs with 150GB/s bandwidth, while a PCIe connection provides 16GB/s.

Fig. 1 shows the elapsed runtime for Caffe with LMS for the first 1000 iterations. We observed that Caffe-LMS on P9 runs about 3.8x faster than on Xeon E5-2640 due to the NVLink 2.0. The same observation is made when we compare Chainer-LMS runs on both platforms as in Fig. 2.

We also observed that LMS can improve the training performance by maximizing GPU utilization. For Resnet-152 on Caffe, the maximum batch size without LMS was 32 and the corresponding throughput was 91.2 images/sec. With LMS, we were able to increase the batch size to 48 and improved the throughput to 121.2 images/sec in spite of the CPU-GPU communication overhead.

REFERENCES

[1] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. 2015. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *CoRR* abs/1512.02595 (2015).
[2] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. 2017. Massive Exploration of Neural Machine Translation Architectures. *CoRR* abs/1703.03906 (2017). arXiv:1703.03906 http://arxiv.org/abs/1703.03906
[3] Jianmin Chen, Rajat Monga, Samy Bengio, and Rafal Józefowicz. 2016. Revisiting Distributed Synchronous SGD. *CoRR* abs/1604.00981 (2016).
[4] Jun Yang Minghui Qiu Yang Gu Chen Meng, Minmin Sun. 2017. Training Deeper Models by GPU Memory Optimization on TensorFlow. In *NIPS Workshop on ML Systems*.
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity Mappings in Deep Residual Networks. *CoRR* abs/1603.05027 (2016). http://arxiv.org/abs/1603.05027
[6] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. 448–456.
[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).
[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*. 1097–1105.
[9] Yusuke Niitani, Toru Ogawa, Shunta Saito, and Masaki Saito. 2017. ChainerCV: a Library for Deep Learning in Computer Vision. *CoRR* abs/1708.08169 (2017).
[10] NVidia. 2017. https://devblogs.nvidia.com/parallelforall/inside-volta.
[11] IBM PowerAI. 2017. https://github.com/cupy/cupy/pull/694 and 3762.
[12] IBM PowerAI. 2017. https://github.com/ibmsoe/caffe/tree/master-lms.