Distributed BERT Pre-Training & Fine-Tuning With Intel Optimized Tensorflow on Intel Xeon Scalable Processors

Muhammed Emin Ozturk
Computer Science
University of Utah
Salt Lake City, Utah

Wei Wang
Intel Corporation
Santa Clara, California

Maciej Szankin
Intel Corporation
San Diego, California

Lei Shao
Intel Corporation
Santa Clara, California

Abstract—**Distributed computing has become a key component in the field of Data Science, allowing for faster prototyping and accelerated time to market of numerous workloads. This work examines the distributed training performance of BERT, a state of the art language model for Neural Language Processing (NLP), in the tasks of pre-training and fine-tuning on general-purpose Intel CPUs. The effects using Intel-optimized TensorFlow optimizations on Intel Architectures with both FP32 and BFLOAT16 floating-point format are included in the analysis. Results show that the distributed TensorFlow BERT model with LAMB optimizer can maintain high accuracy while getting good performance speedups from scaling to a larger amount of Intel Xeon CPUs.**

1. Introduction and Motivation

Advances in Natural Language Processing (NLP) have greatly benefited from Deep Learning progress over over a couple of years. Ian Tenney et al. [1] describes improvements which have been made to the traditional NLP pipeline using Transformer [2] models, Bidirectional Encoder Representations from Transformers (BERT) [3] in particular. The latter model has become increasingly popular as a go-to architecture for many text processing solutions mostly due to it's innovative approach to training. BERT's training is split into two phases, which allows for applying it to problems without abundance of domain related data. In the first phase the model is pre-trained on a large corpus of generic data, allowing for obtaining relations between words and sentences, and only in the second phase a fine-tuning on a smaller domain related dataset takes place.

With the increasing importance of Machine Learning (ML) across industries, a whole new market for hardware accelerated ML has been created [4]. To provide a fair comparison of such accelerators, MLPerf [5] has been introduced to provide a standardized way of measuring their training and inference performance. In the most recent revision [6] MLPerf has been extended to include BERT in its training benchmark suite.

This work presents BERT pre-training and fine-tuning training with distributed TensorFlow (with Horovod) on Intel CPUs and shows the results in terms of accuracy and performance speedups. This study also is one of the first attempts to gain insights into BERT's training in BFLOAT16 precision on Intel Xeon scalable processors.

2. BERT: Bidirectional Encoder Representations from Transformers

BERT is considered as a milestone achievement in Natural Langugate Processing (NLP) community due to the fact that it outperforms previously proposed methods [7]. BERT model training usually contains two steps: Pretrainig and Fine-tuning [7]. We used Wikipedia dataset for BERT Pre-Training and SQuAD v1.1 dataset for BERT Fine-Tuning due to their free availability. In this work, we experimented with BERT-Large. BERT-Large model consists of 24 layers, 16 attention heads and 340 million parameters. BERT-Large is compute intensive and it requires more time to be trained. Hence, we enabled distributed BERT-Large training with LAMB optimizer on Intel Xeon CPUs to evaluate the results.

3. Methodology

3.1. Intel-Optimized TensorFlow

TensorFlow framework is open-source and is maintained by Google. Intel optimized TensorFlow targeting Intel Xeon architectures by integrating oneAPI Deep Neural Networks (oneDNN) into TensorFlow and performing various other optimizations including graph fusions etc. oneDNN comes with high-performance DNN kernels that support various data types including FP32, INT8, and Bfloat16. In this work, we focus on evaluating FP32 and Bfloat16 with Intel-optimized TensorFlow on BERT.

3.2. Distributed Training With TensorFlow And Horovod

Data parallelism is a popular approach to distributed DNN training. On Intel Xeon processors, distributed training with TensorFlow usually employs horovod [8]. We enabled distributed BERT training by adding the horovod APIs. One important issue to solve with data parallelism is that it could cause accuracy degradation with the default Adam optimizer. Training BERT with large batch without losing accuracy is possible with LAMB optimizer as introducted by You *et al.* [9]. Hence, LAMB optimizer was used in experiments. Several Learning Rate (LR) and Batch Size (BS) are tested for Pre-Training & Fine-Tuning in this work. Since Intel optimized TensorFlow enabled Bfloat16 support, we also experimented BERT distributed training with Bfloat16 data type with distributed training.

3.3. Experimental Setup

We experimented distributed FP32 BERT Fine-Tuning and Pretraining with up to 128 Intel® Xeon® Platinum 8260L CPUs, each of which had 24 cores. For BFLOAT16 BERT Fine-Tuning & Pre-Training, we used a machine with 8 socket Intel® Xeon® Platinum 8380 CPUs inside. Each CPU had 28 cores. We used an Intel-optimized TensorFlow version that featured the best Bfloat16 and FP32 support which can be found via https://github.com/Intel-tensorflow/tensorflow/tree/bf16/base. The horovod version was 0.19.1. We set and tune the following OpenMP environment variables to gain maximum performance on Intel architectures: KMP_AFFINITY, KMP_BLOCKTIME, OMP_NUM_THREADS. We also set the following flags to take advantage of the graph level parallelism and operator level parallelism: intra_op_parallelism_threads and inter_op_parallelism_threads.

4. Experimental Results

Two fundamental metrics were used for evaluating the distributed BERT training performance with Intel optimized TensorFlow on Intel Xeon architectures: 1) the accuracy of the trained models, i.e., the F1 score and Exact-Match for fine-tuning task (SQuAD question-answering), and the masked language model (LM) accuracy and training loss for pretraining task 2) the performance speedup as compared to the baseline performance on a single machine with dual socket Intel Xeon CPUs.

4.1. BERT Fine-Tuning Results

Machine	#N	#PPN	#MPI	GBS	F1	EM	Speedup
CLX2S	1	2	2	64	93.01	86.74	1x
CLX4S	1	4	4	128	93.16	86.96	1.9x
CLX4S	2	4	8	256	92.91	86.67	3.45x
CLX2S	8	2	16	512	92.89	86.68	6.21x
CLX2S	16	2	32	1024	92.35	86.18	9.5x
CLX2S	32	2	64	2048	92.11	85.7	19x
CLX2S	64	2	128	4096	91.82	85.22	41.3x

TABLE 1. DISTRIBUTED BERT FINE-TUNING TRAINING RESULTS: ACCURACY & SPEEDUP WHEN INCREASING THE AMOUNT OF CPUS AND MPI WORKERS (ONE MPI WORKER PER CPU). CLX REFERS TO THE INTEL XEON PLATINUM 8260L CPUS AND 2S/4S MEANS DUAL-SOCKET AND FOUR-SOCKET SYSTEMS RESPECTIVELY.

We completed FP32 BERT Fine-Tuning experiment on a cluster of CLX2S and CLX4S machines using distributed Intel-BERT model with horovod and FP32/BF16 support. The CLX2S and CLX4S contains dual-socket and four-socket Intel Xeon Platinum 8260L, respectively. For fine-tuning, we increased the number of MPI processes from 2 to 128 while keeping the mini-batch size at 32. As shown in Table 1, even with a global bacth size (GBS) of 4096 (with 128 MPI workers), it achieved high accuracy with nice speedups (up to compared to the two socket baseline. In addition to accuracy evaluation of FP32 BERT Fine-Tuning with up to 64 CLX nodes, we also tested BFLOAT16 precision BERT Fine-Tuning performance on the 8S CPX machine. Speedups and accuracy metrics comparing with the two socket CLX baseline are shown in Table 2. Clearly, Bfloat16 achieved about 3x-4x speedup without losing accuracy.

Machine	#N	#PPN	#MPI	GBS	F1-S	EM	Speedup
CPX8S	1	2	2	64	92.96	86.58	3.8x
CPX8S	1	4	4	128	93.04	86.79	7.6x
CPX8S	1	8	8	256	93.02	86.98	12.6x

TABLE 2. BERT FINE-TUNING TRAINING RESULTS (ACCURACY AND SPEEDUP) WITH BF16 ON 8S CPX SYSTEM AS COMPARED TO 2S CLX SYSTEM.

4.2. BERT Pre-Training

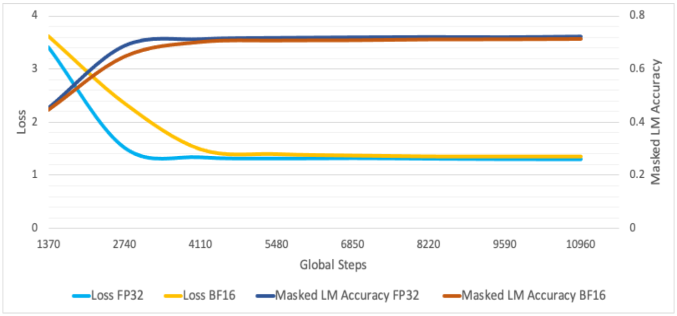

Figure 1. BERT Pre-Training Accuracy VS Global Steps For FP32 and BF16

For BERT pretraining, we adopted Google's 16 TPU worker hyper-parameter in MLPerf v0.7 training submission and tested with Bfloat16 on 8S CPX and FP32 on 16 CLX machines. As shown in Fig.1, accuracy metrics are tracked with respect to global steps. Our main goal for pre-training experiment was reaching MLPerf requirement of 0.712 masked LM accuracy. We evaluated every 1370 steps and observed both FP32 and Bfloat16 training surpassed the accuracy metric. We are investigating why FP32 converged faster (in terms of epochs) than BF16 as shown in Fig. 1.

References

[1] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[4] Z. Li, Y. Wang, T. Zhi, and T. Chen, “A survey of neural network accelerators,” *Frontiers of Computer Science*, vol. 11, no. 5, pp. 746–761, 2017.

[5] P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf *et al.*, “Mlperf training benchmark,” *arXiv preprint arXiv:1910.01500*, 2019.

[6] P. Mattson, V. J. Reddi, C. Cheng, C. Coleman, G. Diamos, D. Kanter, P. Micikevicius, D. Patterson, G. Schmuelling, H. Tang *et al.*, “Mlperf: An industry standard benchmark suite for machine learning performance,” *IEEE Micro*, vol. 40, no. 2, pp. 8–16, 2020.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[8] A. Sergeev and M. D. Balso, “Horovod: fast and easy distributed deep learning in TensorFlow,” *arXiv preprint arXiv:1802.05799*, 2018.

[9] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, , and C.-J. Hsieh, “Large batch optimization for deep learning: Training bert in 76 minutes,” *ICLR*, 2020.