Instead of treating the $\overline{\boldsymbol{W}}(t)$ as the finetuned global model directly, which is the strategy of model averaging, we treat global model update operation as a block-level stochastic optimization process and propose a Blockwise Model-Update Filtering (BMUF) technique to stabilize the learning process.

First, we calculate the model-update vector resulting from current data block by subtract initial model from the aggregated model:

$$\boldsymbol{G}(t) = \overline{\boldsymbol{W}}(t) - \boldsymbol{W}_{\boldsymbol{g}}(t-1)$$

Then calculate global model-update vector, which is a weighted sum of $\boldsymbol{G}(t)$ and previous global model update vector:

$$\Delta(t) = \zeta_t \boldsymbol{G}(t) + \eta_t \Delta(t-1)$$

This formulation is similar with SGD with momentum trick, so we call ζ_t block learning rate and η_t block momentum. ζ_t and η_t can be set automatically by an empirical formulation. Then we update global model by:

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) + \Delta(t)$$

Inspired by Nesterov momentum trick, we generate initial model for next data block by

$$\boldsymbol{W}_{\boldsymbol{g}}(t) = \boldsymbol{W}(t) + \eta_{t+1}\Delta(t)$$

Broadcast $\boldsymbol{W}_{\boldsymbol{g}}(t)$ to each worker and repeat IBPO and BMUF until all data blocks are processed, which is called one sweep. We can fine-tune the model by several sweeps until a stopping criterion is satisfied and obtain the final global model.

4.3 Discussion

1-SGD uses minibatch level parallelism and BMUF uses Block level parallelism with a momentum like update trick to overcome scale out challenges of simple model averaging, a wide variety of deep learning models can benefit. Due to the synchronization at mini-batch level, 1-bit is more sensitive to the I/O latency (because once a worker slows down due to I/O, the overall training speed of one-mini-batch slows down). BMUF on the other hand synchronizes at block level, thus its speed is less sensitive to burst I/O latency. However, 1-bit SGD can work as local model optimizer for BMUF for optimal scalability across multiple server / multiple GPU distributed computing environment.

5. Tutorial Session

In this tutorial, we assume the audience is familiar with the basics of deep learning. The session will focus specifically on text-based modeling of sequences. We encourage the audience to come prepared with the latest CNTK version installed on their machines, which can be done by following the instructions on the github site [2]. Tutorial details will be updated and archived. We will be using both slideware and Jupyter Python notebooks. The audience is expected to be familiar with Python and the Jupyter notebooks.

ACKNOWLEDGMENTS

We would like to thank Frank Seide, Principal Researcher at Microsoft Research and CNTK architect, Redmond USA and Qiang Huo, Principal Research Manager at Microsoft Research Asia (MSRA), Beijing, China for sharing their research materials on 1-bit SGD and BMUF algorithms, respectively. Additionally, we would like to thank Kai Chen from MSRA for BMUF algorithm summarization.

6. REFERENCES

[1] Sutskevar, I., Vinyals, O., and Le, Q.V.. "Sequence to sequence with neural networks," https://arxiv.org/pdf/1409.3215.pdf, 2014

[2] Cognitive Toolkit (formerly CNTK), https://github.com/Microsoft/CNTK/wiki

[3] Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D., "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Proceedings of Interspeech*, 2014.

[4] K. Chen and Q. Huo, "Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering," In *Proceedings of ICASSP*, 2016.

[5] Luong, M.T., Pham H. and Manning, C.D. Effective approaches to attention based neural machine translation. https://arxiv.org/abs/1508.04025.

[6] Shen, Y., Huang, P., Gao, J., and Chen, W., "ReasoNet: Learning to stop reading in machine comprehension," https://posenhuang.github.io/papers/reasonet_iclr_2017.pdf.

[7] The Stanford Question Answering Dataset (SQuAD), https://rajpurkar.github.io/SQuAD-explorer/

[8] Bahdanau, D., Cho, K., and Bengio. Y. "Neural machine translation by jointly learning to align and translate," *in Proceedings of the International Conference on Learning Representations*, 2015.

[9] Hill, F., Bordes, A., Chopra, S. and Weston. J., "The Goldilocks principle: Reading children's books with explicit memory representations," in *Proceedings of the International Conference on Learning Representations*, 2016.

[10] Dhingra, B, Liu, H., Cohen, W.W. and Salakhutdinov, R. "Gated-attention readers for text comprehension," *CoRR, abs/1606.01549*, 2016.

[11] Sordoni, A., Bachman, P., and Bengio, Y, "Iterative alternating neural attention for machine reading," CoRR, abs/1606.02245, 2016.

[12] Williams. R.J., "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, 8(3-4):229–256, 1992.

[13] Mnih, V., Heess, N., Graves, A et al., "Recurrent models of visual attention," In *Advances in Neural Information Processing Systems*, pp. 2204–2212, 2014.

[14] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G., "Achieving Human Parity in Conversational Speech Recognition," https://arxiv.org/abs/1610.05256.

[15] Caffe, https://github.com/BVLC/caffe.

[16] MxNet, https://github.com/dmlc/mxnet

[17] Tensorflow, https://github.com/tensorflow/tensorflow

[18] Theano, https://github.com/Theano/Theano

[19] Torch, https://github.com/torch/torch7/wiki/Cheatsheet

[20] Shi, S, Wang, Q., Xu., P., and Chu, X., "Benchmarking state-of-the-art deep learning software tools," https://arxiv.org/pdf/1608.07249v6.pdf

[21] Tutorial session titled, Scalable deep document / sequence reasoning with Cognitive Toolkit https://github.com/Microsoft/CNTK/wiki/WWW-2017-Tutorial