Population Based Training as a Service

Ang Li[*]
DeepMind
Mountain View, CA

Ola Spyra
DeepMind
Mountain View, CA

Sagi Perel
Google Brain
Pittsburgh, PA

Valentin Dalibard
DeepMind
London, UK

Max Jaderberg
DeepMind
London, UK

Chenjie Gu
DeepMind
Mountain View, CA

David Budden
DeepMind
London, UK

Tim Harley
DeepMind
London, UK

Pramod Gupta
DeepMind
Mountain View, CA

Abstract

Population Based Training (PBT) is a recent approach that jointly optimizes neural network weights and hyperparameters which periodically copies weights of the best performers and mutates hyperparameters during training. We present in this extended abstract a generalized black-box service framework for Population Based Training. We argue that the black-box PBT service design has benefits in system scalability and required engineering effort. We perform a case study on WaveNet speech synthesis to demonstrate the effectiveness of our PBT service.

1 Introduction

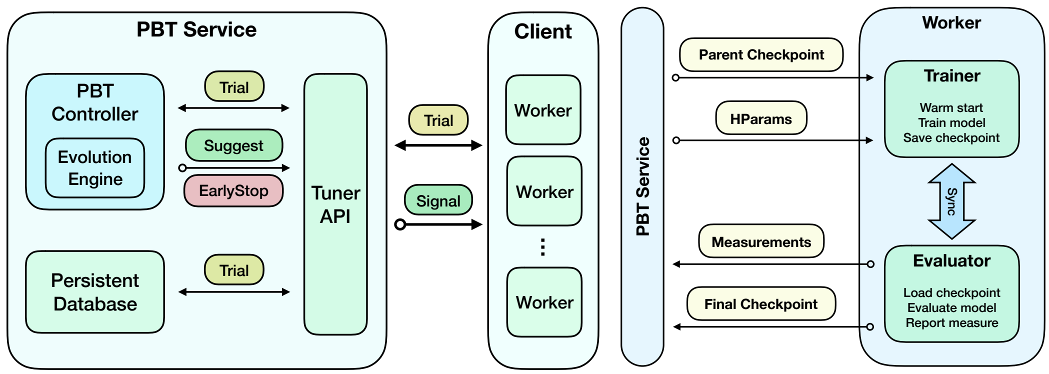
Neural network training often employs a two-stage procedure, *i.e.*, hyperparameter tuning [1, 2, 3, 4, 5, 6, 7] followed by a full model training. Although there have been systems that enable automatic optimization of the hyperparameters, the gap between the two stages relies heavily on human engineering which results in a lengthy and often inefficient model development cycle.

Population Based Training (PBT) has recently emerged and opened up a new direction in neural network training which jointly optimizes the hyperparameters while training the network weights at the same time [8]. Its core idea is to repeatedly replace a poorly performing model with a better performer and continue training with hyperparameters mutated from the better one. PBT allows for *training a model with both differentiable and non-differentiable objectives*, and also *allows all hyperparameters to be dynamic over time*, which is hard to achieve by traditional tuning methods.

We propose a generalized service framework for Population Based Training to improve training extensibility and scalability. The proposed framework is based on the idea of decomposing the whole model training into multiple *trials*, where in each trial a worker only trains for a limited amount of time. An important notion of the service is that each trial is dependent on one or more other trials, *e.g.*, the initial checkpoint can be the last checkpoint of another trial and the hyperparameters can be decided based on other trials' final measurements. A population controller is introduced into the system to oversee the whole population of trials. The controller also decides the hyperparameters and the checkpoint for warm-starting a worker in a new trial.

[*]Correspondence to: Ang Li <anglili@google.com>

NeurIPS Systems for ML workshop, Montréal, Canada, December 2018.



(a) (b)

Figure 1: PBT service diagram (a) and communication between a worker and the service (b).

2 PBT Service Framework

The proposed PBT service framework is inspired by the design of Vizier [7], a black-box hyperparameter tuning service. It is also a *stateless* service, by which we mean each of the requests to the service does not depend on any other. The PBT service diagram is shown in Figure 1(a). It is composed of a controller, a tuner API layer and a persistent database. The tuner API and the database are similar to the design of the Vizier service. The major information container about model training is called *Trial*, defined using a protocol buffer, which is passed between PBT controller, Tuner API, persistent database and the workers in the client. The PBT controller may suggest two kinds of actions to the client, *i.e.*, suggest a new trial or early-stop a trial. The communication between a worker and the service is shown in Figure 1(b) where the service passes hyperparameters and a parent checkpoint for the trainer to warm start with. After training and evaluation, the worker returns measurements and a final checkpoint path to the service.

In addition to the benefits of PBT algorithms, the proposed PBT service has several system-level advantages: (a) allows for tuning hyperparameters no matter whether they are defined in the computation graph or not; (b) sufficient scalability and flexibility in using low priority workers; and (c) maximizes flexibility with the machine learning model training frameworks.

3 Case Study on WaveNet

While the effectiveness of Population Based Training has been demonstrated by [8] with a variety of applications such as neural machine translation, generative adversarial networks and reinforcement learning in DM-Lab, Starcraft, etc, we perform a case study on a new application – speech synthesis using WaveNet [9], to conduct analysis on both accuracy and performance of the proposed PBT system. The results suggest that the performance of our PBT service-based approach is linearly scalable. A benchmark also suggests that PBT achieves faster convergence within a fixed computation budget, compared to popular methods such as grid search, CMA-ES [6] and GP-Bandit [10].

4 Conclusion

We presented a black-box service framework for Population Based Training. The proposed service design allows clients to train their models using PBT with minimal infrastructure effort. Our result suggests that the proposed PBT framework is feasible for large scale deep neural network training.

Acknowledgement. The authors would like to thank Alexander Vostrikov, Ali Razavi, Andy Bodart, Ben Coppin, Daniel Golovin, Daniel Visentin, Eddie Kessler, Eli Bixby, Gabriel Doliner, Harish Chandran, Joyce Chen, Karen Simonyan, Mariam Doliashvili, Matthieu Devin, Maxim Krikun, Oriol Vinyals, Salem Haykal, Sibon Li, Simon Osindero, Todd Hester, Tom Walters, Vivek Ramavajjala, Zhe Zhao and many other colleagues in Alphabet for their meaningful discussions and contributions.

References

[1] James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc., 2011.

[2] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305, February 2012.

[3] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pages 2951–2959, USA, 2012. Curran Associates Inc.

[4] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12(8):1889–1900, 2000.

[5] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18:185:1–185:52, 2017.

[6] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, June 2001.

[7] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D. Sculley. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1487–1495, New York, NY, USA, 2017. ACM.

[8] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[9] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016.

[10] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 1015–1022, USA, 2010. Omnipress.