2024 Summer semester RML Term paper: Sources of Nondeterminism: Optimizer-Level

Yu-Chuan Cheng
6 LP, Scientific Computing, Heidelberg University
yu-chuan.cheng@stud.uni-heidelberg.de

Abstract

This term paper reviews the current state of optimization methods in deep learning, focusing on the performance and reproducibility of various optimizers. We review recent comprehensive benchmarks of deep learning optimizers, which reveal that optimizer performance is highly task-dependent, with no single method dominating across all problems. Despite the proliferation of new optimization techniques, established methods like Stochastic Gradient Descent (SGD) with Momentum and ADAM remain strong baselines.

We learn the trade-offs between computational effort and reproducibility in optimization studies, drawing on recent theoretical frameworks for studying reproducibility in convex optimization. The more ideal conditions we assume (smooth and strong convex), the higher reproducibility we can have. we also explore practical approaches to achieving reproducibility in deep learning model training, including methods for managing software and hardware-related randomness.

We conclude that while progress has been made in understanding and improving optimization methods and reproducibility in machine learning, further research is needed to address the complex interplay between performance, generalization, and reproducibility in advanced machine learning tasks.

1 Introduction

1.1 Short summary (Schmidt et al.)

This paper presents a comprehensive benchmark of deep learning optimizers, evaluating 15 popular methods across 8 diverse problems. The researchers conducted over 50,000 individual training runs to assess optimizer performance under various conditions, including different hyperparameter tuning budgets and learning rate schedules. Their findings reveal that optimizer performance is highly task-dependent, with no single method dominating across all problems. Notably, they observed that using multiple optimizers with default parameters performs similarly to extensively tuning a single optimizer. Despite the proliferation of new optimization methods, the study found that ADAM remains a strong baseline, with newer techniques failing to consistently outperform it in general.

The authors conclude that the field of deep learning optimization has reached a saturation point, suggesting that there are now "enough optimizers." They argue that future research should focus on more significant improvements rather than incremental changes to existing methods. The authors acknowledge limitations in their work, such as not covering very large-scale problems or specialized tasks like GANs.

1.2 Short summary (Ahn et al.)

This paper introduces a formal framework for studying reproducibility in convex optimization, focusing on the challenges posed by noisy or error-prone operations such as inexact gradient computations or initialization. The authors define a quantitative measure called "(ϵ, δ)-deviation" to assess the reproducibility of optimization procedures under various conditions. They analyze several convex optimization settings, including smooth, nonsmooth, and strongly convex objective functions, establishing tight bounds on the limits of reproducibility in each case. Their analysis reveals a fundamental trade-off between computation and reproducibility, showing that more computation is generally necessary and sufficient for better reproducibility.

The study examines different sources of irreproducibility, including stochastic gradient oracles, non-stochastic inexact gradient oracles, and inexact initialization. The authors provide matching upper and lower bounds for various optimization settings, demonstrating that gradient descent methods with small learning rates often achieve optimal reproducibility. They also extend their analysis to machine learning contexts, such as finite-sum minimization (for training loss) and stochastic convex optimization (for population loss). The paper's findings highlight the challenges of achieving reproducibility, especially for non-smooth functions, and suggest that introducing smoothness in optimization problems may improve reproducibility. Overall, this work provides a theoretical foundation for understanding and potentially improving the reproducibility of optimization algorithms in various settings, including machine learning applications.

2 Discussion

If we want the relatively optimal solution, stochastic Gradient Descent (SGD) with Momentum remains the dominant optimizer in machine learning. This preference likely stems from its established history and its ability to yield more optimal solutions when generalization performance is prioritized.

(Zhou et al.) provide a theoretical analysis of adaptive gradient and non-adaptive algorithms using Lévy-driven stochastic differential equations (SDEs). They find that, for the same basin, SGD exhibits shorter escaping times compared to ADAM. This is primarily because ADAM smooths gradients, leading to lighter gradient noise tails and diminishing the anisotropic structure in gradient noise due to its geometry adaptation.

Supporting this, (Wilson et al.) argues that the solutions identified by adaptive methods often generalize worse than those found by SGD, despite sometimes achieving better training performance. This is evident in Figure 2 of (Zhou et al.), where SGD and the Heavy Ball method achieve approximately 2% lower error rates under the "War and Peace" dataset. One possible explanation for this difference is the limited training epochs. (Schmidt et al.) focus on budgets of 25, 50, and 75 epochs, which may not be sufficient for medium to large models, such as those used in ImageNet, GANs, and reinforcement learning. Although their analysis of basic models with popular optimizers is extensive, we cannot consider the results conclusive until all iconic machine learning methods are thoroughly examined.

In Figure 5 of (Schmidt et al.), they demonstrate that across a wide range of learning rates from 10^{-3} to 10^{-1}, test accuracy remains stable. This experiment was conducted with 10 different seeds, applying various final settings for each budget. (Ahn et al.) highlight the inherent trade-off between computational effort and reproducibility, which is a crucial consideration in optimization studies.

(Ahn et al.) introduce several points that require further solidification:

- **Limited scope of convexity:** While this provides a solid foundation, many real-world machine learning problems, particularly those involving deep neural networks, are nonconvex.

- **Focus on first-order methods:** The analysis needs to be extended to include other optimization algorithms, such as second-order methods or adaptive methods like Adam or AdaGrad, given the prevalent use of second-order methods today.

Although (Ahn et al.) provides a good starting point for understanding the upper and lower bounds on reproducibility in non-convex optimization, further research is necessary to implement these findings in practical scenarios and to navigate the trade-off between randomness and accuracy effectively.

Regarding reproducibility, (Goodman et al.) categorize reproducibility into three types:

- **Methods reproducibility:** This refers to the ability to replicate the exact experimental procedures used in the original study, focusing on methodological transparency and detail. In some of the deep learning research, authors did not even publish the hyperparameters they used. Therefore, sometimes we couldn't even replicate the methods they use in the study.

- **Results reproducibility:** This involves achieving the same results in an independent study that closely mirrors the original experiment, ensuring that the outcomes can be replicated. In the context of deep learning, the choice of optimizer would fall under this category.

- **Inferential reproducibility:** This pertains to drawing qualitatively similar conclusions from either an independent replication or a reanalysis of the original study, emphasizing the consistency of interpretations and conclusions.

Operationally, the concept of method reproducibility varies across different scientific disciplines. This paper explores examples from the biomedical, laboratory, and clinical sciences, each of which has its own interpretation of reproducibility. In the context of optimizers, several hyperparameters require careful consideration, including the learning rate, weight decay, beta coefficients (such as those used in the Adam optimizer), and epsilon—a small value added to prevent division by zero in optimizers like Adam and RMSprop—as well as batch size. While these hyperparameters are critical, the primary focus often lies on whether the results can be consistently reproduced without significant variation.

(Chen et al.) propose a systematic approach for training reproducible deep learning models. While their paper includes detailed steps, I will focus on the methods for probing upper and lower bounds and practical operations. The steps are as follows:

1. **Conducting initial training**
2. **Verifying model reproducibility**
3. **Profiling and diagnosing:** Software-related randomness can be traced to underlying system calls. For example, TensorFlow's environment variables can be adjusted to disable cuDNN's auto-tuning feature, ensuring deterministic operations. Hardware-related randomness is examined by checking if library functions during training trigger non-determinism at the hardware layer.
4. **Updating**
5. **Record-and-Replay:** This involves recording the random values returned by system calls during the training process and re-running the training with these recorded values, as in step one.

The paper also includes a case study on Convolutional Neural Networks (CNNs) using datasets like MNIST, CIFAR-10, and CIFAR-100. However, the study heavily emphasizes TensorFlow, with limited comparison to PyTorch. Including both frameworks would make the study more comprehensive. The paper concludes by reiterating the trade-off between minimizing randomness and the computational costs associated with record-and-replay techniques.

(Impagliazzo et al.) provide insight into how randomness is instrumental in balancing accuracy and reproducibility:

- **Randomized rounding:** They use randomized rounding schemes to achieve reproducibility while maintaining accuracy. For instance, in the statistical query algorithm (rSTAT), they partition the interval [0,1] and round estimates to midpoints, allowing different runs to converge on the same output with high probability while preserving accuracy within a chosen tolerance.
- **Balancing parameters:** By adjusting these parameters, one can trade-off between accuracy and reproducibility.
- **Two-stage sampling:** The heavy-hitters algorithm uses a two-stage sampling process. The first stage identifies potential candidates, while the second stage uses fresh samples to estimate probabilities, helping balance accuracy and reproducibility.
- **Recursive techniques:** The approximate median algorithm combines a moderately accurate reproducible algorithm with a highly accurate non-reproducible algorithm. This approach gradually improves accuracy while maintaining reproducibility.

By carefully incorporating randomness in these ways, the authors demonstrate how to achieve high reproducibility without sacrificing too much accuracy, and vice versa. The randomness allows for some flexibility in the output, which is key to achieving reproducibility across different sample sets, while the controlled nature of this randomness ensures that accuracy is maintained within specified bounds.

3 Conclusion

The field of machine learning optimization is complex, with no single optimizer dominating across all tasks. While ADAM remains a strong baseline, traditional methods like Stochastic Gradient Descent (SGD) with Momentum continue to be preferred in many scenarios, especially when generalization performance is prioritized.

Key points supporting this conclusion include:

- The benchmark study by (Schmidt et al.) showed that optimizer performance is highly task-dependent, and using multiple optimizers with default parameters can perform similarly to extensively tuning a single optimizer.

- Theoretical analysis by (Zhou et al.) suggests that SGD exhibits shorter escaping times compared to ADAM, which may contribute to better generalization.
- (Wilson et al.) argue that adaptive methods often generalize worse than SGD, despite sometimes achieving better training performance.

The discussion also highlights the importance of reproducibility in machine learning research. (Ahn et al.) introduced a formal framework for studying reproducibility in convex optimization, revealing a trade-off between computation and reproducibility. However, their work is limited to convex problems and first-order methods, leaving room for further research in non-convex scenarios and with other optimization algorithms.

The paper by (Chen et al.) proposes a systematic approach for training reproducible deep learning models, emphasizing the need to balance randomness and accuracy. This is further supported by (Impagliazzo et al.) work on incorporating controlled randomness to achieve reproducibility without sacrificing too much accuracy.

In conclusion, while progress has been made in understanding and improving optimization methods and reproducibility in machine learning, further research is still needed, especially in non-convex problems and with more diverse optimization algorithms. Although a non-convex problem has no general solution, optimizing the non-convex problems depends on specific cases. I would say that the discussion under convex conditions will be a nice outline of deep learning problems. The field continues grappling with balancing performance, generalization, and reproducibility in complex machine-learning tasks.

References

Kwangjun Ahn, Prateek Jain, Ziwei Ji, Satyen Kale, Praneeth Netrapalli, and Gil I. Shamir. Reproducibility in Optimization: Theoretical Framework and Limits.

Boyuan Chen, Mingzhi Wen, Yong Shi, Dayi Lin, Gopi Krishnan Rajbahadur, Zhen Ming, and Jiang. Towards Training Reproducible Deep Learning Models. In *Proceedings of the 44th International Conference on Software Engineering*, pages 2202–2214.

Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. What does research reproducibility mean? 8(341):341ps12–341ps12.

Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 818–831. Association for Computing Machinery.

Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9367–9376. PMLR.

Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning.

Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Hoi, and Weinan E. Towards Theoretically Understanding Why SGD Generalizes Better Than ADAM in Deep Learning.