Investigation into Training Dynamics of Learned Optimizers (Student Abstract)

Jan Sobotka, Petr Šimánek

Faculty of Information Technology, Czech Technical University in Prague, Thákurova 9, Prague 160 00, Czech Republic
sobotj11@fit.cvut.cz, simanpe2@fit.cvut.cz

Abstract

Modern machine learning heavily relies on optimization, and as deep learning models grow more complex and data-hungry, the search for efficient learning becomes crucial. Learned optimizers disrupt traditional handcrafted methods such as SGD and Adam by learning the optimization strategy itself, potentially speeding up training. However, the learned optimizers' dynamics are still not well understood. To remedy this, our work explores their optimization trajectories from the perspective of network architecture symmetries and proposed parameter update distributions.

Introduction

We focus on the learning-to-optimize (L2O) method and the particular architectural design introduced by Andrychowicz et al. (2016). Concretely, the goal is to meta-learn an optimizer M, implemented as a two-layer LSTM network with parameters ϕ, to optimize a vector of optimizee neural network parameters θ in order to minimize a loss function $L(\theta)$. At each time step t, the optimizer has access to the gradient $\nabla L(\theta_t)$ and produces an update $\mathbf{g}_t$ to get θ_{t+1}. The phase of learning the optimizer's parameters ϕ is usually referred to as meta-training and the subsequent evaluation with frozen parameters ϕ is known as meta-testing.

Although learned optimizers have shown great potential, major practical difficulties still persist. Furthermore, since the field is in its nascent stages, many fundamental questions remain unanswered, and an extensive investigation into their training dynamics is still lacking, which hinders well-informed further progress.

To tackle this, we empirically analyze the impact of symmetries introduced by optimizee architectures and examine the heavy-tailedness and variation of noise in the predicted parameter updates.

Methods

Symmetry-induced constraints. As shown by Kunin et al. (2021), numerous symmetries in neural network architectures impose stringent geometric constraints on gradients. We primarily focus on rescale symmetry which arises from the Leaky ReLU activation function, and on scale symmetry present in networks with batch normalization. For example, the geometric constraint for the rescale symmetry is that the gradients are everywhere perpendicular to the parameter vector with an inverted sign of the outgoing weights.

Heavy-tailed distribution of gradient and update noise. The work of Simsekli, Sagun, and Gurbuzbalaban (2019) has demonstrated that the distribution of gradient noise in SGD converges to a heavy-tailed α-stable random variable. It is known that the density of this distribution decays with a power law tail like $|x|^{-\alpha-1}$ where $\alpha \in (0, 2]$ is called the tail-index: as α gets smaller, the distribution has a heavier tail.

Update covariance. To investigate the noise (variation) in the mini-batch updates for different optimizers, we study their update covariance $\mathbf{K} = \frac{1}{N}\sum_{i=0}^{N}(\mathbf{g}_i - \mathbf{g})(\mathbf{g}_i - \mathbf{g})^T$ where $\mathbf{g}_i$ is the parameter update on sample $\mathbf{x}_i$, $\mathbf{g}$ is the full-batch update, and N is the number of training samples.

Experiments

We meta-train and meta-test on optimizee feed-forward neural networks with 1 hidden layer of 20 neurons with either Leaky ReLU or batch normalization followed by ReLU. Both optimizee models have the softmax function at the output layer and are trained on the MNIST classification task with the cross-entropy loss function and batch size of 128. The optimizer's parameters ϕ are trained to minimize the sum of the optimizee's unrolled losses over 20 mini-batches.

Deviations from the Geometric Constraints

To assess the importance of learned optimizers being free from the geometric constraints that might bind classical optimizers, we track the progression of optimizers' update deviations from these constraints.

As can be seen in Figure 1, the deviations of L2O from the geometric constraints that come from the two symmetries are much larger than those of SGD and Adam. But most strikingly, the increase in this symmetry breaking is largest for L2O at the beginning of optimization, whereas for Lion, it increases more gradually and achieves higher values later.

To get a deeper insight into how L2O leverages the freedom of parameter updates, we meta-train with an additional regularization loss that penalizes the absolute size of

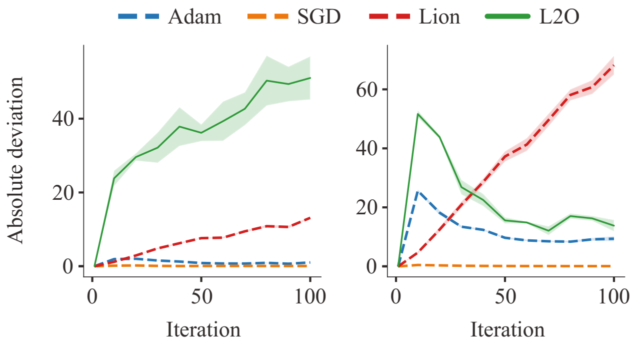

Figure 1: Deviations from the geometric constraints. Left: Rescale symmetry breaking (Leaky ReLU optimizee). Right: Scale symmetry breaking (optimizee with ReLU and batch normalization).

the L2O's update deviations from the geometric constraints on gradients. The performance for various regularization strengths β is shown in Figure 2.

Interestingly, as regularization increases, the L2O's speed of optimization significantly drops. The same observation for the effect of regularization can be made for most of the optimizee architectures on which L2O was not meta-trained.

Distribution of Gradients and Parameter Updates

In this section, we 1) investigate how heavy-tail is the gradient and update noise by estimating the parameter α using the estimator implementation from Simsekli, Sagun, and Gurbuzbalaban (2019); and 2) monitor the progression of the largest eigenvalue of the update covariance $\mathbf{K}$. The results are shown in Figure 3.

First, we see that the noise in the updates from L2O is generally less heavy-tailed and has higher variation than the updates from baseline optimizers such as SGD and Adam. Second, the distribution of L2O parameter updates is much less heavy-tailed than that of the gradients. This shows that L2O effectively attenuates the heavy-tail portion of deviations in the gradient estimates on its input, taking a less jittery optimization trajectory. More importantly, this noise filtering is much more pronounced than for the baseline optimizers.

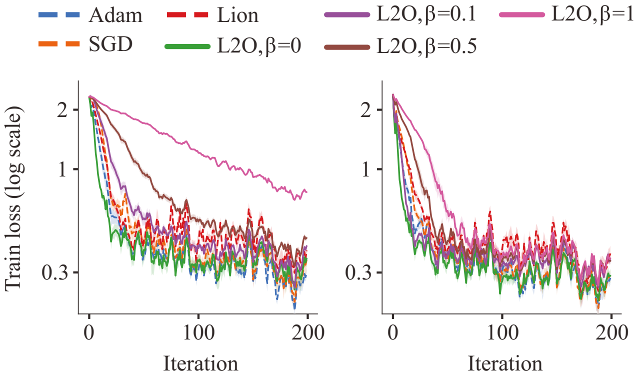

Figure 2: Performance after symmetry breaking regularization. Left: Rescale symmetry breaking regularized (Leaky ReLU optimizee). Right: Scale symmetry breaking regularized (optimizee with ReLU and batch normalization).

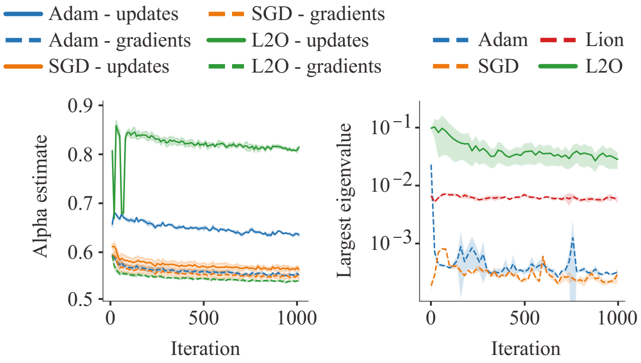

Figure 3: Heavy-tailedness and update covariance. Left: Gradient and update noise (Leaky ReLU optimizee). Right: Update covariance (Leaky ReLU optimizee).

Conclusion

One of the most pronounced features of learned optimizers is their rapid symmetry breaking at the beginning of the optimization run. Remarkably, the good performance of L2O in the initial phase of training correlates with this very well, as also demonstrated by the symmetry breaking regularization which severely hindered the optimizer.

Another aspect is the less heavy-tailed distribution of L2O updates despite the gradients exhibiting very heavy-tailed behavior. Together with the high variation of updates across different samples, as shown by large maximum eigenvalues of update covariance, this points to one interesting observation: L2O appears to act as a stabilizing force in the optimization process. While the inherent stochasticity and heavy-tailed nature of gradients might lead to erratic updates and slow convergence, the noise clipping of L2O seems to mitigate these issues.

Acknowledgments

This work was supported by the Student Summer Research Program 2023 of FIT CTU in Prague.

References

Andrychowicz, M.; Denil, M.; Colmenarejo, S. G.; Hoffman, M. W.; Pfau, D.; Schaul, T.; Shillingford, B.; and de Freitas, N. 2016. Learning to Learn by Gradient Descent by Gradient Descent. In *Proceedings of the 30th NIPS*, NIPS'16, 3988–3996. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Kunin, D.; Sagastuy-Brena, J.; Ganguli, S.; Yamins, D. L.; and Tanaka, H. 2021. Neural Mechanics: Symmetry and Broken Conservation Laws in Deep Learning Dynamics. In *ICLR 2021*.

Simsekli, U.; Sagun, L.; and Gurbuzbalaban, M. 2019. A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th ICML*, volume 97 of *Proceedings of Machine Learning Research*, 5827–5837. PMLR.