When Does Second-Order Optimization Speed Up Training?

Satoki Ishikawa & Rio Yokota

Tokyo Institute of Technology

Background

Q1: Does default first-order optimization work with various kinds of models?
A1: No. It is better to use second-order information depending on the training setting.
Q2: When should we use second-order optimization ?
A2: Second-order optimization is useful when the minibatch size is large or the dataset size is small.

What is Second-order Optimization?

Second-order optimization updates parameters by a preconditioned gradient:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{C}(\boldsymbol{\theta}_t)^{-e} \nabla_{\boldsymbol{\theta}_t} \mathcal{L}(\boldsymbol{\theta}_t), \tag{1}$$

where η is a learning rate and $\boldsymbol{C}(\boldsymbol{\theta})$ is the curvature matrix.

Major Second-order optimization

Gauss-Newton	$\boldsymbol{C}(\boldsymbol{\theta}) = \mathbb{E}[\nabla_{\theta_l}\mathcal{L}^\top \nabla_{\theta_l}\mathcal{L}] + \rho \boldsymbol{I}$
K-FAC [3]	$\boldsymbol{C}(\boldsymbol{\theta}) = \left(\mathbb{E}[\boldsymbol{\delta}_l \boldsymbol{\delta}_l{}^\top] + \rho_B \boldsymbol{I}\right) \otimes \left(\mathbb{E}[\boldsymbol{h}_{l-1} \boldsymbol{h}_{l-1}{}^\top] + \rho_A \boldsymbol{I}\right)$
Shampoo [1]	$\boldsymbol{C}(\boldsymbol{\theta}) = \left(\mathbb{E}[\boldsymbol{\delta}_l \boldsymbol{h}_{l-1}{}^\top \boldsymbol{h}_{l-1} \boldsymbol{\delta}_l{}^\top] + \rho_R \boldsymbol{I}\right) \otimes \left(\mathbb{E}[\boldsymbol{h}_{l-1} \boldsymbol{\delta}_l{}^\top \boldsymbol{\delta}_l \boldsymbol{h}_{l-1}{}^\top] + \rho_R \boldsymbol{I}\right)$

($\boldsymbol{h}_l$ is a forward signal and $\boldsymbol{\delta}_l$ is a backward signal)

Second-order Optimization is suitable for large batch training

For various architectures, second-order optimization methods, including K-FAC and Shampoo, outperform first-order optimization methods for larger batch sizes.

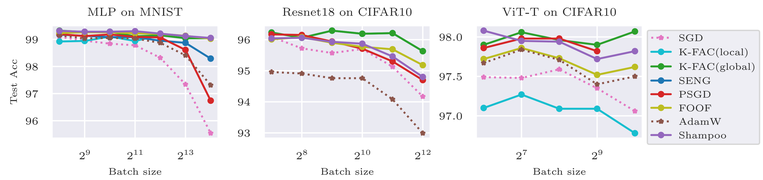

Figure 1. Second-order optimization performs better than first-order optimization for large batch training

Second-order Optimization that calculates curvature iteratively is not suitable for large batch training

In large batch training, the quality of the curvature of PSGD[2] does not improve enough due to lack of iterations. Therefore, PSGD is not suitable for large batch training.

We used the following index as a measure of curvature quality.

$$c(\boldsymbol{P}) = \mathbb{E}_{\delta\boldsymbol{\theta}}\left[\delta\hat{\boldsymbol{g}}^\top \boldsymbol{P} \delta\hat{\boldsymbol{g}} + \delta\boldsymbol{\theta}^\top \boldsymbol{P}^{-1} \delta\boldsymbol{\theta}\right] \tag{2}$$

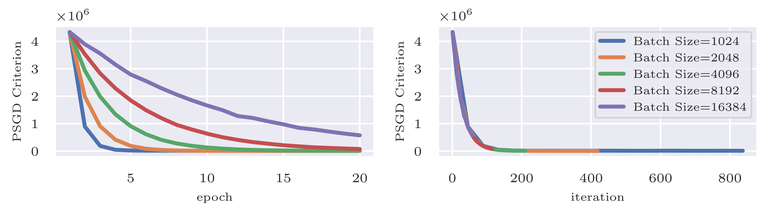

Figure 2. The quality of curvature is determined by the number of iterations and is independent of batch-size.

Second-order Optimization on large dataset

If the dataset size is very large, the benefits of second-order optimization can be negligible. This is a typical setting in language model training.

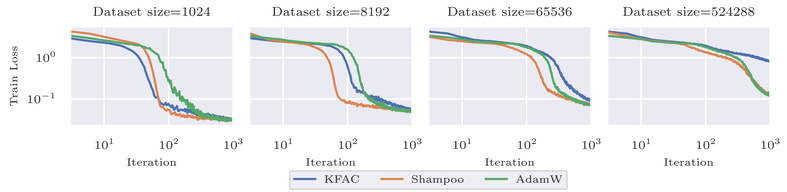

Figure 3. Train Curve Comparison of optimizers for different dataset sizes in training character-level language modeling.

Ratio of batch size to dataset size determines the importance of Second-order Information

If the batch size is sufficiently small compared to the dataset size, first-order optimization does not work well, while second-order optimization does.

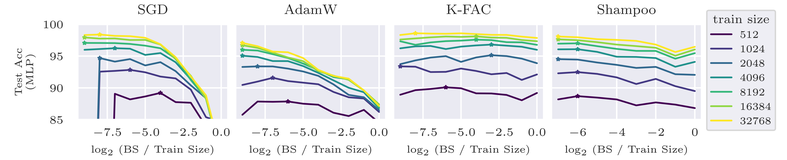

Figure 4. If the ratio of batch size to train size is not sufficiently large, the benefits of second-order optimization are not apparent.

Second-order Information is important in the early stages of training

We trained Deit-small on ImageNet using Shampoo for 300 epochs. We can see that if we update the curvature matrix only during 1 epoch (= 0.3% = 2500 iterations), we need not to update the curvature matrix after that. We compute statistics for every 10 iterations.

		Percentage of frequent updates (Preconditioner Interval = 10)							
		0%	0.01%	0.03%	0.1%	0.3%	1%	3%	10%
Preconditioner Interval	300	1.3	0.9	79.88	79.75	79.82	79.83	79.98	79.83
	1000	0.82	0.75	79.94	79.52	79.90	79.99	79.89	79.98
	3000	0.48	0.32	0.51	79.50	79.64	80.12	80.08	79.92
	10000	0.33	0.51	0.58	79.69	79.85	79.87	80.06	79.87
	30000	0.12	0.12	0.17	79.34	80.07	80.02	79.99	80.19

Table 1. In ViT-Pretraining, we can reduce the frequency of computing the inverse matrix in the early stage of training.

References

[1] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In ICML. PMLR, 2018.
[2] Xi-Lin Li. Preconditioned stochastic gradient descent. IEEE transactions on neural networks and learning systems, 2017.
[3] James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In ICML. PMLR, 2015.

Characteristics of hyperparameters for batch size

The smaller the batch size, the smaller the optimal damping. Since the smaller the damping is, the more the advantage of second-order optimization can be used, this proportionality also indicates that the advantage of second-order optimization is small when the batch size is small.

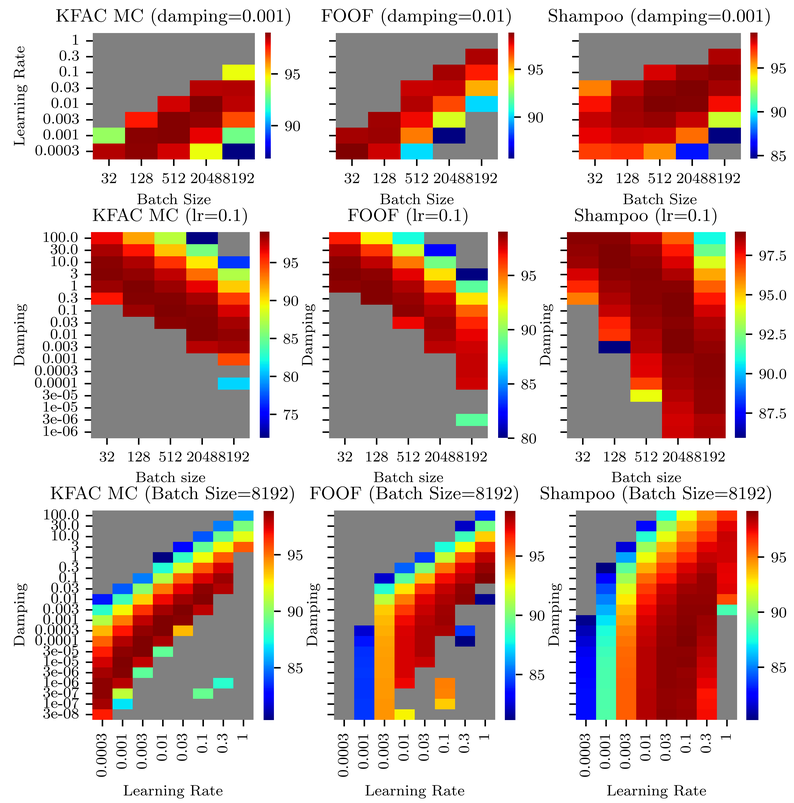

Figure 5. Learning rate, damping, and batch size are interrelated.

Summary

1. The larger the batch size, the smaller the dataset size, the more second-order optimization should be used.
2. It is in the early stages of learning that the second-order information matrix should be updated.
3. Damping is also proportional to batch size and learning rate.

E-Mail: riverstone@rio.gsic.titech.ac.jp