Studying K-FAC Heuristics by Viewing Adam through a Second-Order Lens

Ross M. Clarke
José Miguel Hernández-Lobato

University of Cambridge

ICML 2024

Second-Order Optimisation

ML Training

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$$

$$\mathbf{g}_t = \left[\frac{\partial f}{\partial \theta}\right]_{\boldsymbol{\theta}_t} \qquad \mathbf{C}_t \in \left\{[\mathbf{H}]_{\boldsymbol{\theta}_t}, [\mathbf{F}]_{\boldsymbol{\theta}_t}, [\mathbf{G}]_{\boldsymbol{\theta}_t}, \cdots\right\}$$

First-Order Optimisation

$$\theta_{t+1} = \theta_t - \mathbf{u}\left(\mathbf{g}_t\right)$$

Second-Order Optimisation

$$\theta_{t+1} = \theta_t - \mathbf{u}\left(\mathbf{g}_t, \mathbf{C}_t^{-1}\right)$$

Second-Order Optimisation

ML Training

$$\theta^\star = \arg\min_{\theta} f(\theta)$$

$$\mathbf{g}_t = \left[\frac{\partial f}{\partial \theta}\right]_{\theta_t} \qquad \mathbf{C}_t \in \left\{ [\mathbf{H}]_{\theta_t}, [\mathbf{F}]_{\theta_t}, [\mathbf{G}]_{\theta_t}, \cdots \right\}$$

First-Order Optimisation

$$\theta_{t+1} = \theta_t - \mathbf{u}(\mathbf{g}_t)$$

Second-Order Optimisation

$$\theta_{t+1} = \theta_t - \mathbf{u}\left(\mathbf{g}_t, \mathbf{C}_t^{-1}\right)$$

Current Landscape

A Trade-Off

- First-order optimisers most popular and cheaper
- Second-order optimisers theoretically faster to converge

Observations

- Second-order methods often suffer instability
- K-FAC[1] performs surprisingly well
- Heuristics are *essential* components of second-order optimisers

[1] Martens and Grosse (2015), “Optimizing Neural Networks with Kronecker-factored Approximate Curvature”

Idea

Could we apply second-order optimisers’ heuristics to first-order methods?

Current Landscape

A Trade-Off

- First-order optimisers most popular and cheaper
- Second-order optimisers theoretically faster to converge

Observations

- Second-order methods often suffer instability
- K-FAC[1] performs surprisingly well
- Heuristics are *essential* components of second-order optimisers

[1] Martens and Grosse (2015), "Optimizing Neural Networks with Kronecker-factored Approximate Curvature"

Idea

Could we apply second-order optimisers' heuristics to first-order methods?

Current Landscape

A Trade-Off

- First-order optimisers most popular and cheaper
- Second-order optimisers theoretically faster to converge

Observations

- Second-order methods often suffer instability
- K-FAC[1] performs surprisingly well
- Heuristics are *essential* components of second-order optimisers

[1] Martens and Grosse (2015), "Optimizing Neural Networks with Kronecker-factored Approximate Curvature"

Idea

Could we apply second-order optimisers' heuristics to first-order methods?

Heuristics Borrowed from K-FAC

$$M(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_{t-1}) + (\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\top \mathbf{g}_t + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\top (\mathbf{C}_t + \lambda_t \mathbf{I})(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$$

Adaptive Learning Rate

$$\alpha_t = \arg\min_{\alpha} M(\boldsymbol{\theta}_{t-1} - \alpha \mathbf{d}_t) = \frac{\mathbf{g}_t^\top \mathbf{d}_t}{\mathbf{d}_t^\top (\mathbf{C}_t + \lambda_t \mathbf{I})\mathbf{d}_t}$$

Adaptive Levenberg-Marquardt Damping[2, 3, 4]

$$\rho = \frac{f(\boldsymbol{\theta}_t) - f(\boldsymbol{\theta}_{t-1})}{M(\boldsymbol{\theta}_t) - M(\boldsymbol{\theta}_{t-1})}; \qquad \lambda_{t+1} = \begin{cases} \omega_{\text{dec}} \lambda_t & \text{if } \rho > \frac{3}{4} \\ \lambda_t & \text{if } \frac{1}{4} \le \rho \le \frac{3}{4} \\ \omega_{\text{inc}} \lambda_t & \text{if } \rho < \frac{1}{4} \end{cases}$$

[2] Levenberg (1944), “A Method for the Solution of Certain Non-Linear Problems in Least Squares”
[3] Marquardt (1963), “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”
[4] Roweis (1996), *Levenberg-Marquardt Optimization*

Heuristics Borrowed from K-FAC

$$M(\theta) = f(\theta_{t-1}) + (\theta - \theta_{t-1})^\mathsf{T}\mathbf{g}_t + \frac{1}{2}(\theta - \theta_{t-1})^\mathsf{T}(\mathbf{C}_t + \lambda_t\mathbf{I})(\theta - \theta_{t-1})$$

Adaptive Learning Rate

$$\alpha_t = \arg\min_{\alpha} M(\theta_{t-1} - \alpha\mathbf{d}_t) = \frac{\mathbf{g}_t^\mathsf{T}\mathbf{d}_t}{\mathbf{d}_t^\mathsf{T}(\mathbf{C}_t + \lambda_t\mathbf{I})\mathbf{d}_t}$$

Adaptive Levenberg-Marquardt Damping[2, 3, 4]

$$\rho = \frac{f(\theta_t) - f(\theta_{t-1})}{M(\theta_t) - M(\theta_{t-1})}; \qquad \lambda_{t+1} = \begin{cases} \omega_{\text{dec}}\lambda_t & \text{if } \rho > \frac{3}{4} \\ \lambda_t & \text{if } \frac{1}{4} \le \rho \le \frac{3}{4} \\ \omega_{\text{inc}}\lambda_t & \text{if } \rho < \frac{1}{4} \end{cases}$$

[2] Levenberg (1944), "A Method for the Solution of Certain Non-Linear Problems in Least Squares"
[3] Marquardt (1963), "An Algorithm for Least-Squares Estimation of Nonlinear Parameters"
[4] Roweis (1996), *Levenberg-Marquardt Optimization*

Heuristics Borrowed from K-FAC

$$M(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_{t-1}) + (\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\mathsf{T}\mathbf{g}_t + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\mathsf{T}(\mathbf{C}_t + \lambda_t \mathbf{I})(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$$

Adaptive Learning Rate

$$\alpha_t = \arg\min_{\alpha} M(\boldsymbol{\theta}_{t-1} - \alpha \mathbf{d}_t) = \frac{\mathbf{g}_t^\mathsf{T}\mathbf{d}_t}{\mathbf{d}_t^\mathsf{T}(\mathbf{C}_t + \lambda_t \mathbf{I})\mathbf{d}_t}$$

Adaptive Levenberg-Marquardt Damping[2, 3, 4]

$$\rho = \frac{f(\boldsymbol{\theta}_t) - f(\boldsymbol{\theta}_{t-1})}{M(\boldsymbol{\theta}_t) - M(\boldsymbol{\theta}_{t-1})}; \qquad \lambda_{t+1} = \begin{cases} \omega_{\text{dec}}\lambda_t & \text{if } \rho > \frac{3}{4} \\ \lambda_t & \text{if } \frac{1}{4} \leq \rho \leq \frac{3}{4} \\ \omega_{\text{inc}}\lambda_t & \text{if } \rho < \frac{1}{4} \end{cases}$$

[2] Levenberg (1944), “A Method for the Solution of Certain Non-Linear Problems in Least Squares”
[3] Marquardt (1963), “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”
[4] Roweis (1996), *Levenberg-Marquardt Optimization*

AdamQLR: First-Order Optimisation with Second-Order Heuristics

Adam[5]

$\mathbf{m}_0, \mathbf{v}_0 \leftarrow \mathbf{0}$

for $t = 1, 2, \cdots$ until θ converged **do**

$\mathbf{g}_t \leftarrow \nabla_\theta f(\theta_{t-1})$

$\mathbf{m}_t \leftarrow \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t$

$\mathbf{v}_t \leftarrow \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)(\mathbf{g}_t \odot \mathbf{g}_t)$

$\widehat{\mathbf{m}}_t \leftarrow \frac{\mathbf{m}_t}{1-\beta_1^t}$

$\widehat{\mathbf{v}}_t \leftarrow \frac{\mathbf{v}_t}{1-\beta_2^t}$

$\mathbf{d}_t \leftarrow \frac{\widehat{\mathbf{m}}_t}{\sqrt{\widehat{\mathbf{v}}_t}+\epsilon}$

Perform QLR Heuristics

$\theta_t \leftarrow \theta_{t-1} - \alpha_t \mathbf{d}_t$

end for

[5]Kingma and Ba (2015), "Adam: A Method for Stochastic Optimization"

QLR Heuristics

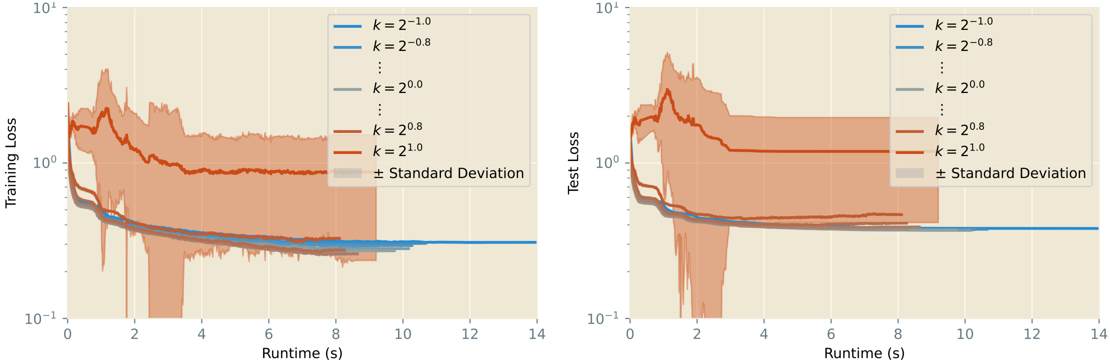
$$\alpha_t = \frac{\mathbf{g}_t^\mathsf{T} \mathbf{d}_t}{\mathbf{d}_t^\mathsf{T} (\mathbf{C}_t + \lambda_t \mathbf{I}) \mathbf{d}_t}$$

$$\rho = \frac{f(\theta_t) - f(\theta_{t-1})}{M_t(\theta_t) - M_t(\theta_{t-1})}$$

$$\lambda_{t+1} = \begin{cases} \omega_{\text{dec}} \lambda_t & \text{if } \rho > \frac{3}{4} \\ \lambda_t & \text{if } \frac{1}{4} \leq \rho \leq \frac{3}{4} \\ \omega_{\text{inc}} \lambda_t & \text{if } \rho < \frac{1}{4} \end{cases}$$

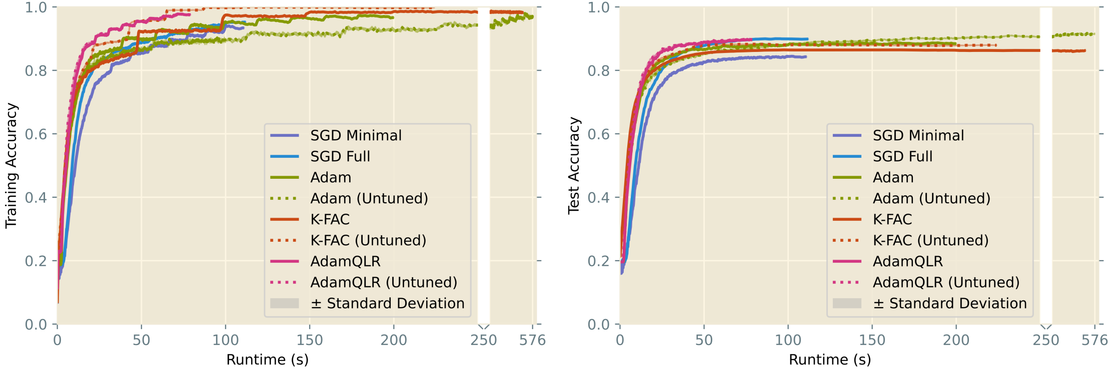
Results (AdamQLR Sensitivity Study)

Fashion-MNIST on 784/50/10 MLP



Results (Optimisation Performance)

SVHN on ResNet-18



Summary

Contributions

- AdamQLR: a hybrid first- and second-order optimiser
- Task-dependent effect of second-order heuristics
- Partial ablation study of K-FAC
- Robustness to hyperparameters

Open Questions

- Why does relative performance vary so much?
- In what other ways might we combine second-order heuristics with first-order methods?

References I

1. James Martens and Roger Grosse. "Optimizing Neural Networks with Kronecker-factored Approximate Curvature". In: *International Conference on Machine Learning*. International Conference on Machine Learning. 2015, pp. 2408–2417. URL: http://proceedings.mlr.press/v37/martens15.html (visited on 11/20/2018).
2. Kenneth Levenberg. "A Method for the Solution of Certain Non-Linear Problems in Least Squares". In: *Quarterly of Applied Mathematics* 2.2 (1944), pp. 164–168. ISSN: 0033-569X, 1552-4485. DOI: 10.1090/qam/10666. URL: https://www.ams.org/qam/1944-02-02/S0033-569X-1944-10666-0/ (visited on 02/02/2023).
3. Donald W. Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441. ISSN: 0368-4245. DOI: 10.1137/0111030. URL: https://epubs.siam.org/doi/10.1137/0111030 (visited on 05/14/2023).

References II

4. Sam Roweis. *Levenberg-Marquardt Optimization*. Technical Report. New York University, 1996, p. 5. URL: `https://cs.nyu.edu/~roweis/notes/lm.pdf` (visited on 08/09/2022).
5. Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015*. 3rd International Conference on Learning Representations, ICLR 2015. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: `http://arxiv.org/abs/1412.6980` (visited on 11/17/2021).

Find out more

https://arxiv.org/abs/2310.14963

ICML 2024 Poster Session 4
13:30 – 15:00 CEST
Tuesday 23rd July

4-9, Hall C
Messe Wien