Research Topics: Deep Learning Beyond the Gradient

Felix Dangel

Keywords: Automatic differentiation, Hessian, Fisher, K-FAC, second-order optimization, Newton's method

TL;DR: Contemporary deep learning (DL) algorithms rely on the gradient. During my PhD, I developed tools to efficiently compute quantities beyond the gradient—higher-order information—like gradient statistics over a mini-batch and Hessian approximations. This information enables a richer view onto the loss landscape, quantifying its noise (stochasticity) and curvature (geometry). During my Postdoc, I want to **explore the potential of higher-order information for building more powerful DL methods** and further **improve automatic differentiation**.

Interested in similar topics? Let's chat!

No time to read on? Don't worry. I'll try to find and reach out to you ;)

(details below, feel free to skip)

Improving Automatic Differentiation Frameworks

Next-generation ML frameworks should be able to seamlessly and efficiently compute not only the gradient, but also higher-order information. Currently, their computation often requires workarounds that are inefficient or complicated to implement, reducing the availability to users. I want to **identify shortcomings in existing AD libraries to improve their design and performance** and make higher-order information accessible to the community. Ideas:

- Make frameworks more aware of (i) linear algebra structure and (ii) hardware to automate many numerical tricks and optimizations (for instance optimizing contraction schedules, partial operand access/slicing, . . .).
- Develop representations for higher-order derivatives that simplify graph analysis and optimization.
- Explore the implications of new concepts, for instance distinguishing between co- and contra-variant indices of tensors rather than treating them as multi-dimensional arrays [1].

Establishing Second-order Methods

Note: 'Second-order method' might refer to optimization, or more broadly any method motivated by a quadratic Taylor approximation (for instance for model compression [2] or Laplace approximations [3]).

DL is dominated by first-order (gradient-based) methods. Exploring methods that rely on other information has been prohibitively expensive and complicated until recently due to software constraints. I want to **investigate the potential of higher-order information such as curvature and noise for DL applications**. Ideas:

- Identify challenges for second-order optimization methods in the DL regime and work on fixes.

  - What do "Newton's method" and "air travel" have in common? Both are very fast, but their worst-case is bad! (slide 97) Is noise a problem for their stability in the mini-batch setting? Can we develop techniques to improve stability, for instance removing the need for damping (fractional NGD [4]), applying momentum to Newton steps (determinantal averaging [5]), or designing other safeguard heuristics?
  - Is there theory on the impact of noise in the convex setting?
  - Is their implicit bias a problem for generalization?
  - Reducing computational cost, for instance with 'mixed training' (split NN parameters into two groups, train one group with a first-order method, the other with a second-order method).

- Identify new tasks where higher-order information is useful (or even required as first-order methods fail).
- Hyperparameter adaptation with higher-order information

Empirical Studies and Novel Approximations of Higher-order Information

Empirical investigations of DL quantities have observed pronounced patterns, for instance in the Hessian's spectrum [6]–[8]. I want to **investigate and understand the origin of structure in higher-order information**. This enables the development of efficient representations that specifically tackle the approximation of relevant terms. Ideas:

- Using tiny subspace, class-structure, and layer structure for new approximations. Numerous empirical works observed the Hessian's eigenvalue spectrum to cluster into groups [6]–[8]. The sizes of those groups are affected by the DNN's output dimension C (number of classes). [9] observed that after a few steps of training, the gradient resides mostly in the first group—the Hessian's top-C eigenspace. Keeping track of interactions between gradient and Hessian can therefore be sped up by projection onto that tiny subspace.
  - [10] attributed these groups to terms in a hierarchical decomposition of the Hessian. The term responsible for the top-C eigenvalues can be computed relatively efficiently. The tiny subspace seems to not change much during training. This allows for constructing light-weight, high quality Hessian approximations that can be recycled in later iterations (momentum).
- Understand why using gradients to approximate curvature seems to work in DL. For instance: The empirical Fisher (EF) is a popular approximation of the Fisher/Hessian built from gradients. [11] shows that, in general, it can be dangerous to use the EF instead of the Fisher for optimization. Empirically, however, the EF performs well in DL applications [2]. Also, the EF and Fisher/GGN look quite similar for a DNN (visually). Maybe both matrices share important properties in the DL setting, despite the concerns of [11]?
- Investigate evolution during training for phase-dependent algorithms. By monitoring such quantities during training, one can identify different regimes and design specialized methods for each [12].

References

[1] A. Kristiadi, F. Dangel, and P. Hennig, "The geometry of neural nets' parameter spaces under reparametrization." 2023.
[2] S. P. Singh and D. Alistarh, "Woodfisher: Efficient second-order approximation for neural network compression," 2020.
[3] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace redux - effortless bayesian deep learning," 2021.
[4] D. Huh, "Curvature-corrected learning dynamics in deep neural networks," 2020.
[5] M. Derezinski and M. W. Mahoney, "Distributed estimation of the inverse hessian by determinantal averaging," in *Advances in neural information processing systems (neurips)*, 2019.
[6] V. Papyan, "The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size." 2019.
[7] L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou, "Empirical analysis of the hessian of over-parametrized neural networks." 2018.
[8] L. Sagun, L. Bottou, and Y. LeCun, "Eigenvalues of the hessian in deep learning: Singularity and beyond." 2017.
[9] G. Gur-Ari, D. A. Roberts, and E. Dyer, "Gradient descent happens in a tiny subspace." 2018.
[10] V. Papyan, "Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet Hessians," 2019.
[11] F. Kunstner, P. Hennig, and L. Balles, "Limitations of the empirical fisher approximation for natural gradient descent," 2019.
[12] L. Tatzel, P. Hennig, and F. Schneider, "Late-phase second-order training," 2022.