Quasi-Newton's method in the class gradient defined high-curvature subspace

Mark Tuddenham, Adam Prügel-Bennett, & Jonathon Hare
Vision, Learning, and Control Group, Electronics and Computer Science, University of Southampton

Class gradients

Classification problems using deep learning have been shown to have a high-curvature subspace in the loss landscape equal in dimension to the number of classes. Moreover, this subspace corresponds to the subspace spanned by the logit gradients for each class. An obvious strategy to speed up optimisation would be to use Newton's method in the high-curvature subspace and stochastic gradient descent in the co-space. We show that a naive implementation actually slows down convergence and we speculate why this might be.

Gur-Ari *et al.* [1] show that gradient descent is mostly contained in a small subspace, so although the problem is high-dimensional, the optimisation is limited by a low-dimensional high-curvature subspace. That subspace, spanned by the class gradients, intersects with the top-C eigenvectors of the Hessian [2], [3].
Fort and Ganguli [4] define the class gradient as

$$c_k = \frac{1}{|\mathcal{D}_k|} \sum_{\mu \in \mathcal{D}_k} \boldsymbol{\nabla} z_k^\mu \tag{1}$$

where $\mathcal{D}_k$ is the set of training examples,$\{(x^\mu, y^\mu)\}$, belonging to class k and z_k is the k^{th} logit.

Quasi-Newton's method

Let the logit gradients be decomposed as

$$\boldsymbol{\nabla} z_k^\mu = y_k^\mu \, c_k + \epsilon_k^\mu. \tag{2}$$

Then we can create a low-rank Hessian approximation as

$$H^{\text{Low Rank}} \approx \sum_{k=1}^{C} \left(\frac{1}{|\mathcal{B}|} \sum_{\mu \in \mathcal{B}} y_k^\mu \, p_k^\mu (1 - p_k^\mu) \right) c_k \, c_k^\intercal \tag{3}$$

where p_k^μ are the class probabilities and, following Fort and Ganguli [4], we ignore the terms of the Hessian that are sufficiently small since its eigensystem will remain unchanged [5].
Assuming that the class gradients are orthogonal we can define

$$H^{\text{Low Rank}} = \sum_{k=1}^{C} \lambda_k v_k \, v_k^\intercal, \tag{4}$$

where $v_k = c_k / |c_k|$ and $\lambda_k = \frac{1}{|\mathcal{B}|} \sum_{\mu \in \mathcal{B}} y_k^\mu \, p_k^\mu \, (1 - p_k^\mu) \, |c_k|^2$. The generalised inverse, $(H^{\text{Low Rank}})^\dagger$, is simply the reciprocal of $H^{\text{Low Rank}}$. Then, doing Newton's method in the high-curvature subspace, $\theta' \leftarrow \theta - (H^{\text{Low Rank}})^\dagger g^{\mathcal{B}}$, and stochastic gradient descent, with a learning rate η, in the co-space using a projection

$$\mathbf{P} = \mathbf{I} - \sum_{k=1}^{C} v_k \, v_k^\intercal \tag{5}$$

we get the update equation

$$\theta' \leftarrow \theta - \eta g^{\mathcal{B}} - \sum_{k=1}^{C} \left(\frac{1}{\lambda_k} - \eta \right) (v_k^\intercal g^{\mathcal{B}}) \, v_k. \tag{6}$$

To lower the stochasticity of the estimation of the eigensystem of $H^{\text{Low Rank}}$ we use an exponential moving average to estimate both v_k and λ_k.

Comparison to SGD

Figure 1: Typical validation loss when training a ResNet9 (6.5M parameters) for both SGD and quasi-Newton's method on CIFAR-10

- The final performances are similar, test loss is ≈ 0.50 and accuracy 87%.
- The quasi-Newton's method is more volatile and is slower to improve in general.
- One would expect to be able to use a higher learning rate with the quasi-Newton's method as that applies only to the low-curvature subspace, however, this is not true, and both methods diverge at $\eta > 10^{-1}$

To see if inter-batch noise causes the lack of improvement, we have trained a small CNN on MNIST so that full-batch training is feasible.

Figure 2: Comparison of batch sizes for a small CNN (400 parameters) on MNIST.

- Allthough our quasi-Newton's method may start learning faster than SGD, there comes a crossover point where SGD overtakes and continues to have better performance regardless of the batch size
- We can conclude that the noise incurred due to batch training is not a root of the performance deficit.

Conclusions

We are currently investigating the different explanations of why this naive method offers no improvement. Current ideas inculde

- The class vector subspace does not totally cover the top eigenvector subspace
- The structure and non-convexity of the landscape is such that rapid gradient descent actually slows down the search.

Common experience is that, when done right, optimisation strategies that compensate for different curvatures in the loss landscape can lead to a significant speedup.
Despite the failure of our naive implementation, the recognition that there exists a relatively low-dimensional subspace of high curvature which can be identified relatively easily suggests that it may be possible to improve on current methods.

References

[1] G. Gur-Ari, D. A. Roberts, and E. Dyer, "Gradient descent happens in a tiny subspace," *arXiv preprint arXiv:1812.04754*, 2018. [Online]. Available: `http://arxiv.org/abs/1812.04754`.

[2] V. Papyan, "Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 8819–8830, Jan. 2019. [Online]. Available: `http://arxiv.org/abs/1901.08244`.

[3] B. Ghorbani, S. Krishnan, and Y. Xiao, "An investigation into neural net optimization via hessian eigenvalue density," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 4039–4052, Jan. 2019. [Online]. Available: `http://arxiv.org/abs/1901.10159`.

[4] S. Fort and S. Ganguli, "Emergent properties of the local geometry of neural loss landscapes," Oct. 2019. [Online]. Available: `http://arxiv.org/abs/1910.05929`.

[5] F. Benaych-Georges and R. R. Nadakuditi, "The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices," *Advances in Mathematics*, vol. 227, pp. 494–521, 1 May 2011, ISSN: 10902082. DOI: `10.1016/j.aim.2011.02.007`.