Gauss-Newton under adversarial attacks

Marshall Jiang
marshall.jiang@gmail.com

Abstract

The use of gradient-only optimizer is considered standard in training deep neural networks due to its flexibility and speed. However, there has been interest in using Hessian information in accelerating gradient descent with methods like generalized Gauss-Newton, K-FAC and BFGS at the forefront. These methods generally converge faster to a local minima, assuming correct parameters are chosen, and these minima usually correspond to different ones found by gradient descent. In particular, there has been evidence suggesting that Gauss-Newton might converge to a minima with poor generalizability. We present early results that this can actually be good for greater adversarial robustness.

1 Introduction

With the extensive use of neural networks in a variety of fields, the safety and security concerns rise. One particular threat is so called "adversarial attacks" whereby an attacker perturbs data which causes a trained model to making an incorrect prediction [4]. While trivial examples like email spam evasion might be annoying, increasingly high stakes usage start to arise in security or in self-driving cars. This makes robustness against attacks of critical importance.

Adversarial examples usually lie on the decision boundaries learned by the neural network, as a small perturbation of an image or prompt will cause the network to misclassify. This suggests that curvature of the loss landscape can be used as a tool to understand the generation or detection of adversarial attacks [8, 7]. While there are other ways to increase robustness, most commonly known as "adversarial training" where actual attacks are performed while training, we only focus on the curvature analysis here.

In this document, we present a fairly early exploration into using Gauss-Newton as tool to enhance robustness. It has been noted that some minima found by certain second-order methods may be sharper and correspond to poorer generalizability on clean data [3, 1]. In this work, we present preliminary findings that suggest relationship: that a second-order optimization approach, specifically one based on the generalized Gauss-Newton approximation, may lead to a model that is more robust to adversarial attacks due to having poorer generalizability. We explore this trade-off and use the curvature analysis, where curious results are obtained.

2 Background

2.1 Adversarial Attacks

An adversarial attack involves the deliberate creation of an adversarial example, which is a data sample that has been minimally perturbed to cause a machine learning model to misclassify it. These perturbations are scaled to be small enough to be perceptible to human senses, but can have massive consequences.

One of the easiest types of attack is the Fast-Gradient Sign Method (FGSM) [6]. FGSM simply generates a perturbation by taking a single step in the gradient of the loss with *respect to the image*.

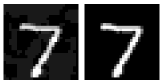

Figure 1: Figure of a succesful adversarial attack with $\varepsilon = 0.01$ on MNIST. The left is classified as a 3 while the right is the original image.

In other words, it finds the direction where changing the image will cause the largest increase in loss. Mathematically, this is simply

$$\bar{x} := x + \varepsilon \text{sign}(\nabla_x \mathcal{L}(x, \theta)) \tag{1}$$

where ε a small parameter, $\mathcal{L}(x, \theta)$ is the loss against the input x and θ the network parameters, and $\bar{x}$ the perturbation. Note that the gradient is with respect to the image and not the parameters. A successful attack is shown in fig. 1.

2.2 Generalized Gauss-Newton

The Gauss-Newton method is a classical optimization method, which was developed initially for least squares, but now have been adapted for neural networks [2]. Rather than the usual SGD updates

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta \mathcal{L}(x, \theta)$$

where η is the learning rate, generalized Gauss-Newton consists of using an approximation to the Hessian

$$\theta_{n+1} = \theta_n - \eta(\lambda I + J^T H J)^{-1} \nabla_\theta \mathcal{L}(x, \theta)$$

where J the Jacobian of the neural network, H the Hessian of the loss with respect to the network, and λ some positive regularizer. The main idea is that the approximation to the true Hessian provides more information about the loss landscape than just the gradient.

3 Methodology and analysis

We built and trained three different models using different optimizers: SGD, Adam and Gauss-Newton. While all three models have the exact same architecture and start with the same initial seeds, the use of different optimizers will result in them ending up with different behaviors. A simple, small convolution neural network is chosen for sake of computational costs. These models are trained on the MNIST dataset, and a small grid search was conducted to find a candidate learning rate. This hyperparameter search is by no means exhaustive, especially for the Gauss-Newton case, where there are considerably more levers to tweak. All models are trained such that they achieve roughly 95% or more accuracy on the test set.

In a secondary experiment, we trained three additional models, one for each optimizer, on a “poisoned“ dataset. This dataset was constructed by taking 10% of the original training data and perturbing it with the wrong label. Poisoned dataset is especially important now due to how LLMs and foundation models need massive amounts of data to train, meaning that the dataset will have many incorrect or, worse, specially designed data, which can cause the training to misalign [5].

We refer the reader to the code base for the exact details.[1]

3.1 Adversarial Robustness

With the models trained, all models are subject to adversarial attacks by FGSM. We perform a light sweep over the ε parameter of eq. (1) over $\{0.1, 0.05, 0.01\}$. In Table 1, we show the results of the FGSM attacks on the clean dataset. It's clear that SGD and Adam have no discernible differences between the percentages, but Gauss-Newton results in a clear decrease in the number of succesful attacks.

This trend is also seen in the poisoned dataset context, but to a lesser extent. We show the results from the FGSM attack on the models trained on poisoned dataset in table 2. It seems that while Adam and SGD actually improved robustness, perhaps from being less prone to overfitting, Gauss-Newton did not significantly change at all.

[1] Available at `https://github.com/runiteking1/aisec-project/`

Table 1: Performance of Models Under FGSM Attack for Varying Epsilon (Clean Data)

Optimizer	**Correct Tests**	$\varepsilon = 0.1$	$\varepsilon = 0.05$	$\varepsilon = 0.01$
Adam	9778	7396 (75.64%)	3142 (32.13%)	253 (2.59%)
SGD	9611	7268 (75.62%)	2687 (27.96%)	227 (2.36%)
Gauss-Newton	9585	4937 (51.51%)	1334 (13.92%)	159 (1.66%)

Table 2: Performance of Models Under FGSM Attack for Varying Epsilon (Poisoned Data)

Optimizer	**Correct Tests**	$\varepsilon = 0.1$	$\varepsilon = 0.05$	$\varepsilon = 0.01$
Adam	9776	6259 (64.02%)	2105 (21.53%)	186 (1.90%)
SGD	9553	6016 (62.97%)	1775 (18.58%)	173 (1.81%)
Gauss-Newton	9581	4888 (51.02%)	1367 (14.27%)	162 (1.69%)

3.2 Analysis

Beyond simple accuracy metrics, we sought to understand the geometric properties of the local minima that each optimizer converged to, and how these properties relate to adversarial robustness. We performed two key analyses to measure curvature.

3.2.1 Input Gradient Norm and Logit Margin Distributions

To understand the local geometry of the model's decision boundary in the input space, we measured the distribution of the input gradient norm and the logit margin for each model on the clean test set.

The input gradient norm is defined as $\left\|\nabla_x \mathcal{L}(x, \theta)\right\|^2$ where $\mathcal{L}$ again is the loss, θ the final model weights, and x the input image, the norm of the gradient at the loss. A high norm indicates that the model's output is highly sensitive to a small change in the input, suggesting that the data point is located in a region of high curvature. The common assumption is that high norms usually mean an easier attack vectors, as one the same amount of change can cause greater change in the loss value. However, this is *not* what we show the distribution over the test set in fig. 2. In fact, it seems that the Gauss Newton resulted in significantly larger average norm compared to Adam! This suggests that GN converges to a far different type of local minima, where the decision boundaries are more seperated.

As for the logit margin, it is a measure of the model's confidence. Mathematically, it's the largest logit (the prediction) minus the second largest logit. For example, if we call the model f, and an input image x, the output of the network could look like

$$f(x) = [.2, 5, .3, .5, 20, .3, -.2, .7, 1.4, -3.4].$$

The logit margin would be $20 - 5 = 15$ representing the "sureness" of a model. Thus, the bigger the difference, the more sure a model generally is. In fig. 3, we observe that GN model actually, unintuitively, resulted in a model which is more unsure in absolute magnitude. Again, this suggests additional work is needed to examine this interesting phenomena.

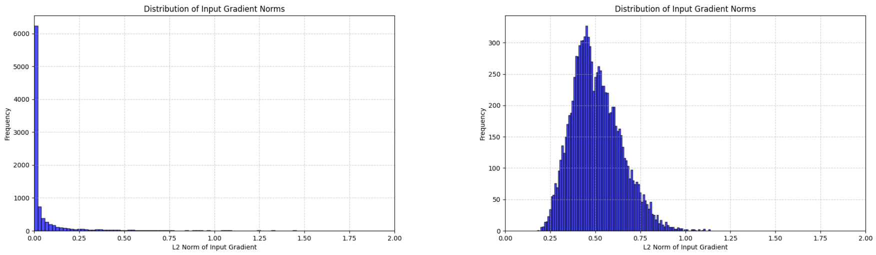

Figure 2: Figure of distribution of input gradient norms on the test set with Adam on the left and GN on right. Note that the Adam has significantly smaller norms, meaning the curvature near the model is more flat compared to GN.

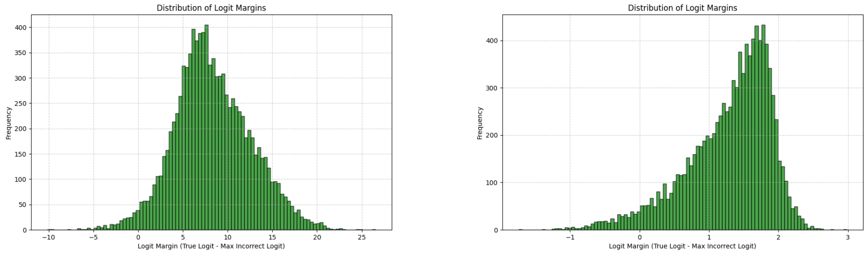

Figure 3: Figure of distribution of logit margins on the test set with Adam on the left and GN on right. The x-scale is different on the figures. Note that Adam seems to be generally quite confident while GN is more "timid" in its response.

3.2.2 SAM Analysis

Sharpness-Aware Minimization (SAM) is an optimization method that directly connects the geometry of a model's loss landscape to its adversarial robustness. It operates on the principle that a model's ability to withstand adversarial attacks on its input data is tied to the "flatness" of the minimum it converges to in the parameter space.

While traditional optimizers aim to find a minimum with the lowest possible loss, SAM seeks to minimize the worst case loss within a small neighborhood around the model's current parameters. Mathematically, this is the minimax problem

$$\min_{\theta} \max_{\|\varepsilon\| \leq \rho} \mathcal{L}(\theta + \varepsilon).$$

The max operator ensures that the landscape around the loss is flat and less "brittle."

We performed simplified SAM analysis and display the results in table 3. The GN model seems to have greater curvature at the loss, again, counterintuitively.

Table 3: Median Loss Increase from SAM Analysis for Adam and Gauss-Newton. We note that like the logit/gradient norms metrics, these results are unintuitive!

ρ	**GN Median Loss Increase**	**Adam Median Loss Increase**
0.001	0.0026	0.0009
0.005	0.0137	0.0048
0.01	0.0282	0.0101

4 Future Work

The results obtained from this work is intriguing: while the analysis by all accounts should result in a resounding victory for Adam, GN performed much better. However, this is very exploratory as MNIST is considered too easy of a dataset and GN is too slow of an optimizer for large scale problem. With more resources, one should run K-FAC, which is inherently more scalable, on CIFAR10 or Imagenet data. More experiment should also be done with the poisoned dataset idea.

Acknowledgments and Disclosure of Funding

Thanks to Qin Liu for helpful comments and guidance on possible analysis avenues.

References

[1] Shun-ichi Amari, Jimmy Ba, Roger Grosse, Xuechen Li, Atsushi Nitanda, Taiji Suzuki, Denny Wu, and Ji Xu. When does preconditioning help or hurt generalization? *arXiv preprint arXiv:2006.10732*, 2020.

[2] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. 2017.

[3] Davide Buffelli, Jamie McGowan, Wangkun Xu, Alexandru Cioba, Da-shan Shiu, Guillaume Hennequin, and Alberto Bernacchia. Exact, tractable gauss-newton optimization in deep reversible architectures reveal poor generalization. *Advances in Neural Information Processing Systems*, 37:133541–133570, 2024.

[4] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[5] Antonio Emanuele Cinà, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard A Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 55(13s):1–39, 2023.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015.

[7] Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, ZhiHua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. Flooding-X: Improving BERT's resistance to adversarial attacks via loss-restricted fine-tuning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5634–5644, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[8] Rui Zheng, Shihan Dou, Yuhao Zhou, Qin Liu, Tao Gui, Qi Zhang, Zhongyu Wei, Xuanjing Huang, and Menghan Zhang. Detecting adversarial samples through sharpness of loss landscape. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11282–11298, Toronto, Canada, July 2023. Association for Computational Linguistics.