OPT2024: 16th Annual Workshop on Optimization for Machine Learning

Nonmonotone Line Searches Operate at the Edge of Stability

Curtis Fox[1,*] CURTFOX@CS.UBC.CA
Leonardo Galli[2,*] GALLI@MATH.LMU.DE
Mark Schmidt[1,†] SCHMIDTM@CS.UBC.CA
Holger Rauhut[2] RAUHUT@MATH.LMU.DE
University of British Columbia[1]*, LMU Munich*[2]*, Canada CIFAR AI Chair (Amii)*[†]
Equal Contribution[*]

Abstract

The traditional convergence analysis of Gradient Descent (GD) assumes the step size to be bounded from above by twice the reciprocal of the sharpness, i.e., the largest eigenvalue of the Hessian of the objective function. However, recent numerical observations on neural networks have shown that GD also converges with larger step sizes. In this case, GD may enter into the so-called edge of stability phase in which the objective function decreases faster than with smaller steps, but nonmonotonically. Interestingly enough, this same behaviour was already observed roughly 40 years ago when the first nonmonotone line search was proposed. These methods are designed to accept larger steps than their monotone counterparts (e.g., Armijo) as they do not impose a decrease in the objective function, while still being provably convergent. In this paper, we show that nonmonotone line searches are able to operate at the edge of stability regime right from the start of the training. Moreover, we design a new resetting technique that speeds up training and yields flatter solutions, by keeping GD at the edge of stability without requiring hyperparameter-tuning or prior knowledge of problem-dependent constants. To conclude, we observe that the large steps yielded by our method seem to mimic the behavior of the well-known Barzilai-Borwein method.

1. Introduction

Gradient descent (GD) and its stochastic variant stochastic gradient descent (SGD) [26] are crucial to the successes of deep learning activity today. In order to prove their convergence, it is common to impose strict requirements on the step sizes (e.g., infinitesimal [4], diminishing [6, 16, 18], bounded by twice the reciprocal of the sharpness [20]) that are not met by step size choices used in practice. In fact, depending on the local curvature, GD may be employed with larger steps and it may converge much faster [7]. It has been shown across a consistent set of experiments on various architectures that the training of neural networks via GD with a step size η goes through two distinct phases [7]. In the first phase (progressive sharpening), the loss function decreases monotonically while the sharpness (the largest eigenvalue of the Hessian of the training loss function) increases. In the second phase (edge of stability), the loss decreases nonmonotonically, while the sharpness stabilizes around $2/\eta$. While the ability of GD to converge nonmonotonically may be surprising, a deeper study reveals that the same behavior was already observed when employing nonmonotone line search methods [14, 35] (see the work by Galli [11] for a literature review). The first nonmonotone line search [14] was proposed in the 1980's to accept (large) step sizes that may not lead to monotonic decrease of the objective function f. The convergence of these methods can be proven without imposing a monotonic descent in the objective function [14]. Recent work showed that SGD

combined with a nonmonotone line search [12] converges faster than its monotone counterpart [31] when training deep neural networks.

The interest in large step sizes has rapidly grown as a consequence of the results in Cohen et al. [7]. Various effects of large step sizes have been proven in the recent literature, e.g., speeding up convergence [33], escaping sharp minima [19], learning "threshold-like" neurons [1], improving generalization [25]. On the other hand, the definition of "large" and "small" steps is incoherent between papers [19, 32], as these concepts are clearly problem-dependent. Moreover, it is a priori unclear if GD with a certain step η will hit the edge of stability or not. Both these open questions are related to GD's inability of adapting to f, at least when employed with a constant step η. The ability of adapting to the underlying problem and inherently estimating the local Lipschitz smoothness L, a.k.a. sharpness, was instead previously proven for a nonmonotone line search called the universal gradient method [21]. In this paper, we propose to study these unexplored connections to nonmonotone line searches to provide a problem-independent algorithm able to find and operate at the edge of stability. Our contributions are summarized below.

- We show empirically that nonmonotone methods operate at the edge of stability right from the start of the training for deep neural networks and yield steps that are always beyond $\frac{2}{L}$. Moreover, we design a new resetting technique that selects large enough initial step sizes to force the nonmonotone line search methods to remain at the edge of stability. The resulting method achieves fast convergence while finding solutions that are flatter than its monotone counterparts. This technique is problem independent and does not require hyper-parameter tuning.
- We propose a new on-the-fly estimation of the sharpness L, and show that it more closely approximates L than the value $2/\eta$, as indirectly suggested in Cohen et al. [7]. We draw a further surprising connection between large nonmonotone steps and Barzilai-Borwein (BB) [5] step sizes.

2. Related Works

The work by Cohen et al. [7] has led to a flurry of papers studying the edge of stability phenomenon. The self-stabilization of the sharpness around $2/\eta$ is explained by observing that, as the iterates diverge in the direction of the top eigenvector of the Hessian, the cubic term in the Taylor expansion of the loss function decreases the curvature until stability is restored [8]. However, this result assumes the occurrence of progressive sharpening, an observation that has no theoretical justification yet. The phenomenon of loss increase and sudden decrease with the annexed stabilization of the largest eigenvalue λ_1 was independently called catapult in Lewkowycz et al. [17]. With the focus on 2-dimensional functions, the authors of [32] propose a new measure of the regularity of a function and suggest that edge of stability, catapults and other effects of large step sizes correlate to high values of this degree of regularity.

The existing analyses [1, 19, 25, 32] only take into account GD with constant step sizes, while in this paper we will focus on the effects of large adaptive steps yielded by nonmonotone line search methods. In this context, line search methods were not considered until recently in the work by Roulet et al. [28]. In this concurrent work, the authors discuss monotone line searches and propose a curvature-aware step size scheduler that forces gradient descent to hit the edge of stability. However, this method relies on setting a hyperparameter, which when poorly selected can lead to the method diverging. To the best of our knowledge, none of the prior works consider nonmonotone line searches. Another connection to [28] is that both monotone and nonmonotone line searches are intrinsically curvature/sharpness-aware. In fact, a piece of evidence that went partially unnoticed in

the literature is that these methods yield step sizes that, in some scenarios, can go provably beyond the value $2/\lambda_1$ (see Lemma 1 below and the discussion around it).

3. Methods

In this paper, we focus on GD with different step size choices: constant step sizes, Armijo line search [3] as described in (1), and nonmonotone methods of the type described in (2).

$$f(w_k - \eta_k \nabla f(w_k)) \leq f(w_k) - c\eta_k \| \nabla f(w_k)\|^2, \quad \text{with } c \in (0,1), \tag{1}$$

$$\begin{gathered} f(w_k - \eta_k \nabla f(w_k)) \leq C_k - c\eta_k \|\nabla f(w_k)\|^2, \quad \text{with } c \in (0,1), \\ C_k = \max\left\{\tilde{C}_k; f(w_k)\right\}, \ \tilde{C}_k = \frac{\xi Q_k C_{k-1} + f(w_k)}{Q_{k+1}}, \ Q_{k+1} = \xi Q_k + 1. \end{gathered} \tag{2}$$

For both line search methods, we follow the classical backtracking procedure of starting with an initial step size $\eta_{k,0}$ and reducing it by $\beta \in (0,1)$ until the condition is fulfilled.

In the optimization literature, the value $\eta_{k,0}$ has often played a secondary role [9, 22, 31]. However, this parameter directly controls the final step size, i.e., $\eta_k = \eta_{k,0}\beta^{l_k}$, where $l_k \in \mathbb{N}$ is the smallest integer for which (1) or (2) is satisfied and counts the amount of cuts or backtracks of the line search procedure. When no backtracks are performed ($l_k = 0$), the final step size is just the initial step size for the given iteration, i.e., $\eta_k = \eta_{k,0}$. Beyond that, if $\eta_{k,0}$ is too small (e.g., $\eta_{k,0} << 2/L$), $\eta_{k,0}$ will never be reduced and GD will ultimately behave exactly like a GD with a small constant step. In the context of edge of stability, this means that the iterates may never hit the nonmonotone phase [7]. We corroborate this argument by reporting a deterministic version of Lemma 1 from [12].

Lemma 1 *Let f be L-Lipschitz smooth. Given an initial step size $\eta_{k,0} > 0$, then the step size η_k returned by (2) and (1) is either*

$$\begin{cases} \eta_k = \eta_{k,0} & \text{if } l_k = 0, \\ \eta_k \geq \frac{2\beta(1-c)}{L} & \text{if } l_k > 0. \end{cases} \tag{3}$$

This lemma ensures that if the initial step size $\eta_{k,0}$ is cut at least once (i.e., $l_k > 0$), η_k will be larger than $\frac{2\beta(1-c)}{L}$. With the values of β and c being $\frac{1}{2}$ (also used in the experiments below), the resulting bound ensures that if $l_k > 0$, $\eta_k \geq \frac{1}{2L}$. Additionally, one could also choose $\beta \approx 1$ and $c \approx 0$ to ensure $\eta_k \gtrsim \frac{2}{L}$, but the line search procedure would become very expensive in terms of backtracks as with $\beta \approx 1$ we will barely move away from $\eta_{k,0}$. On the other hand, nonmonotone line searches may both help in reducing the amount of backtracks and choosing a larger step size. While a provable lower bound on η_k specific to nonmonotone line searches is still elusive and will be the focus of future work, the numerical results (last row of Figure 2) show that these methods yield steps that are always larger than $\frac{2}{L_k}$, with peaks of $\frac{6}{L_k}$, where L_k is the local Lipschitz-smoothness constant ($L_k \leq L$).

Lemma 1 is thus suggesting that to provably achieve large enough step sizes, one needs to ensure a cut at each iteration. To obtain this, we propose a new resetting technique that yields large initial steps $\eta_{k,0}$ without prior knowledge of the Lipschitz smoothness constant L. When combined with Armijo (1) or the nonmonotone line search (NLS) (2), we call the resulting methods Armijo-noTune and NLS-noTune, as they automatically select both $\eta_{k,0}$ and η_k. These methods select η_k so that the

corresponding line search condition (either (1) or (2)) is satisfied with equality (approximately) by selecting a large enough initial step size $\eta_{k,0}$. In particular, for the Armijo-noTune and NLS-noTune methods, the initial step size for each iteration is set as follows:

$$\eta_{k+1,0} = \begin{cases} \eta_{k,0} \cdot \frac{1}{\delta_k} & \delta_k \sim \mathcal{N}(\frac{1}{2}, 1), \quad \delta_k \in [0.1, 0.9] \qquad \text{if } l_k = 0, \\ \eta_{k,0} & \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if } l_k > 0. \end{cases}$$

Simply put, if the line search does not perform any backtracks, the initial step size is increased. Otherwise, the initial step size is left unchanged. We compare these methods with the deterministic versions of SLS [31] and PoNoS [12], called Armijo and Polyak NLS (PoNLS) respectively. See Appendix B for more details on the methods discussed.

4. Numerical Results

Following the experimental setup in Roulet et al. [27], we train all models on a subset of 4096 instances of the CIFAR10 dataset. We address the classification task with 3 different models, a Convolutional Neural Network (CNN) with 2 convolutional layers and a final linear layer, a resnet34 [15], and a vgg11 [30]. In all the experiments, we train the models with full batch GD and the mean square error loss function. Finally, for all line searches, we set $\beta = 0.5$. See Appendix B for more experimental details.

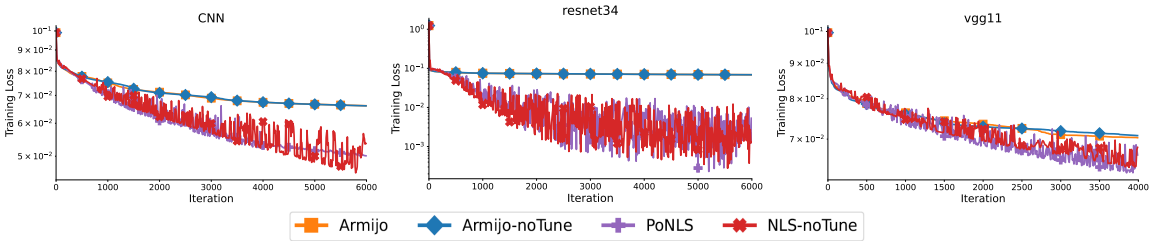

Figure 1: Training loss for gradient descent with 2 variants of Armijo line search and 2 variants of nonmonotone line search respectively. Non-monotone line searches converge faster than their monotone counterparts.

In Figure 1, we plot the training loss of Armijo, Armijo-noTune, PoNLS and NLS-noTune. Figure 1 confirms that large (nonmonotone) steps improve the speed of convergence [12] and hints that both Armijo-noTune and NLS-noTune are at least competitive with Armijo and PoNLS, respectively.

In Figure 2, we plot the training loss, the sharpness, and the sharpness * step size for Armijo-noTune, NLS-noTune and GD with 3 constant steps, i.e., GD-small, GD-medium and GD-large. Earlier work showed that monotone Armijo line search does not hit the edge of stability, as it keeps increasing the sharpness over time [27]. However, we show that nonmonotone line searches almost directly operate at the edge of stability. We see that the training loss decreases nonmonotonically, its sharpness is stable, and the sharpness * step size values are consistently above the edge of stability threshold of 2. Moreover, NLS-noTune yields flatter solutions, as in the majority of the iterations it yields the largest step sizes of the methods discussed (see Figure 5 in Appendix C).

In the top row of Figure 3, we plot for both Armijo-noTune and NLS-noTune the sharpness and the newly proposed approximation of L (L_{approx}) obtained by treating the Descent Lemma [20] as

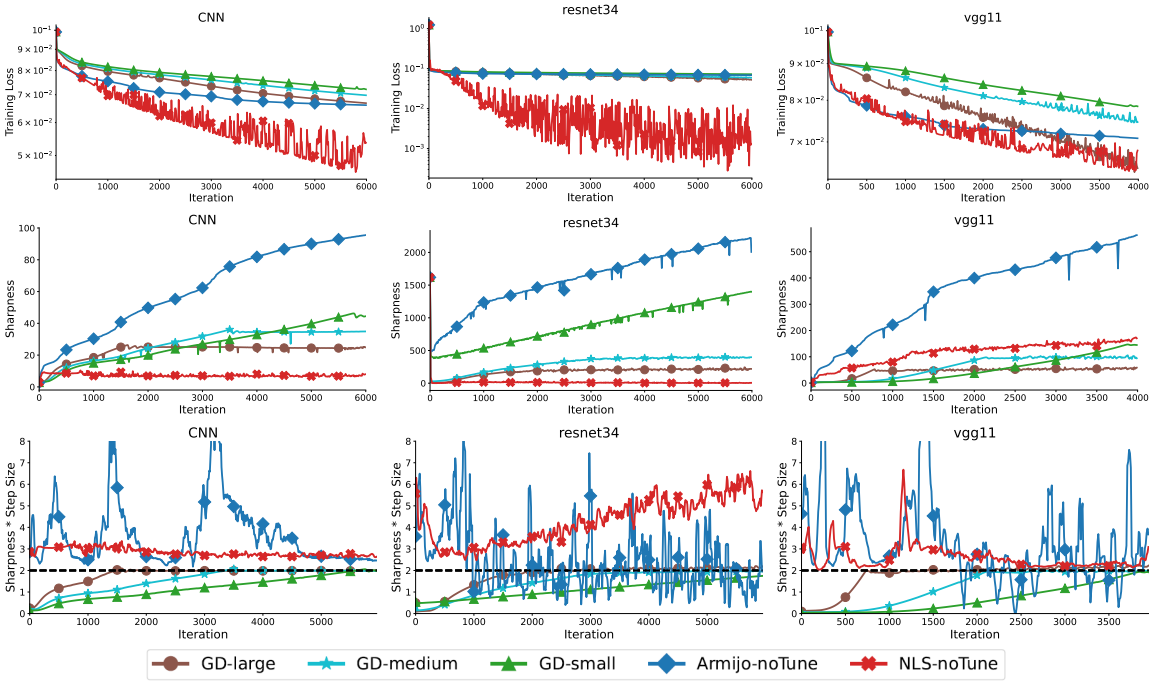

Figure 2: Training loss (Top Row), sharpness (Middle Row), and sharpness * step size (Bottom Row) for gradient descent with Armijo-noTune line search, NLS-noTune line search, and 3 constant step sizes. The sharpness of the monotone Armijo-noTune line search increases with time and the algorithm does not hit the edge of stability. The sharpness of the nonmonotone NLS-noTune method stabilizes and the sharpness * step size is consistently above the threshold of 2 for all three models.

an equality and solving for L. The details of L_{approx} are given in Appendix A. Interestingly, the sharpness and L_{approx} values are very close to each other for NLS-noTune, while not as close for Armijo-noTune. Moreover, for the NLS, L_{approx} better approximates the sharpness than the value of $2/\eta$, indirectly suggested by the edge of stability literature [7] (see Figure 6 in Appendix C).

In the bottom row of Figure 3, we compare the sharpness with two approximations of L proposed in the work by Shi and Guo [29], i.e., the following Barzilai and Borwein [5] (BB) formulas

$$L_{BB_1} = \frac{|y_k^T s_k|}{||s_k||^2} \quad L_{BB_2} = \frac{||y_k||^2}{|y_k^T s_k|}, \quad \text{with } s_k := w_k - w_{k-1}, y_k := \nabla f(w_k) - \nabla f(w_{k-1}).$$

We plot the relative error (i.e., $\frac{|\text{sharpness} - L_{BB}|}{\text{sharpness}}$) of approximating the sharpness with L_{BB_1} or L_{BB_2}. The values L_{BB_1} and L_{BB_2} yield good estimations of the sharpness when employing NLS-noTune on two out of the three experiments (CNN and vgg11). When the monotone counterpart is instead employed, only L_{BB_2} approximates L well on one out the three models (CNN). Consistently among these experiments, we can observe that L_{BB_1} and L_{BB_2} are very close to each other only when large nonmonotone steps are used. Based on this observation and following the derivation in the Appendix A, our results suggest that GD with the large (nonmonotone) steps yielded by NLS-noTune approximately behave like GD with a BB step, without computing Rayleigh quotients.

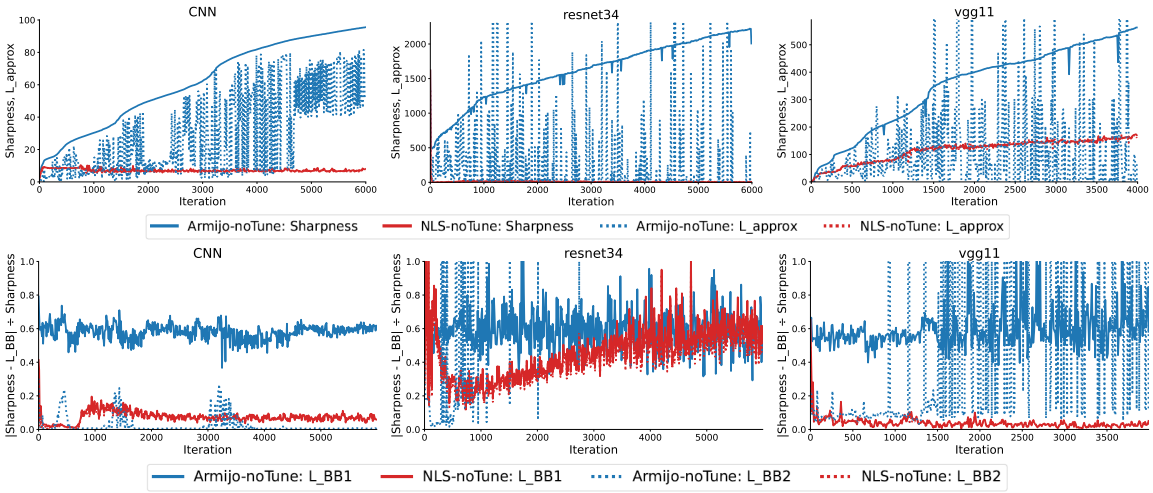

Figure 3: Top Row: Sharpness and an approximation of the Lipschitz constant via the Descent Lemma, which we call L_{approx}. The sharpness and L_{approx} values are very close to each other for NLS-noTune method, but not for the monotone Armijo-noTune method. Bottom Row: Relative error committed by Barzilai-Borwein formulas L_{BB_1} and L_{BB_2} in approximating the sharpness. For the CNN and vgg11 models, these formulas are good approximations of the sharpness when using the NLS-noTune method, while they are poor for the monotone Armijo-noTune method.

5. Conclusion

We analyze the sharpness trajectory of nonmonotone line searches and compare these results to that of their monotone counterparts. We see that unlike the monotone case, nonmonotone line searches are able to hit the edge of stability and continue operating in this regime throughout training. This property is enforced by a newly designed resetting technique that automatically selects large initial step sizes without prior knowledge of the Lipschitz smoothness L. The resulting method achieves fast training convergence and yields solutions that are generally flatter than those yielded by monotone methods. Despite the ongoing debate on the relationship between sharpness and the generalization abilities of neural networks [2], low sharpness solutions were recently proven to be approximately equivalent (at least for deep linear networks) to nuclear-norm-regularized solutions [13]. This inductive bias suggests that flatter solutions are more likely to be sparse and, thus, overall desirable over denser solutions.

Following up on Figure 3, we intend to further assess the ability of line search methods to estimate the local Lipschitz constant L in other settings and exploit these good estimations to design new adaptive (stochastic) line searches that will be completely hyperparameter-free following some existing work [23, 29]. Moreover, we intend to investigate further the connection between large nonmonotone steps and BB steps. In light of Figure 3 and its consequences, we suspect that the advantages of BB [24] on nonquadratic functions may not be traced back to its ability of occasionally hitting an eigenvalue of the Hessian [10], but instead to its large steps.

Acknowledgments

The work was partially supported by the Canada CIFAR AI Chair Program and NSERC Discovery Grant RGPIN-2022-036669.

References

[1] Kwangjun Ahn, Sebastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via edge of stability. In *Advances in Neural Information Processing Systems*, volume 36, pages 19540–19569, 2023.

[2] Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A Modern Look at the Relationship between Sharpness and Generalization. In *International Conference on Machine Learning*, 2023.

[3] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[4] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7413–7424, 2019.

[5] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[6] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[7] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

[8] Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability. In *International Conference on Learning Representations*, 2022.

[9] Chen Fan, Gaspard Choné-Ducasse, Mark Schmidt, and Christos Thrampoulidis. Bisls/sps: Auto-tune step sizes for stable bi-level optimization. In *Advances in Neural Information Processing Systems*, 2023.

[10] Roger Fletcher. On the barzilai-borwein method. In *Optimization and control with applications*, pages 235–256. Springer, 2005.

[11] Leonardo Galli. *Nonmonotone Techniques for Smooth Optimization*. PhD thesis, Università degli Studi di Firenze, 2020.

[12] Leonardo Galli, Holger Rauhut, and Mark Schmidt. Don't be so monotone: Relaxing stochastic line search in over-parameterized models. In *Advances in Neural Information Processing Systems*, 2023.

[13] Khashayar Gatmiry, Zhiyuan Li, Ching-Yao Chuang, Sashank Reddi, Tengyu Ma, and Stefanie Jegelka. The Inductive Bias of Flatness Regularization for Deep Matrix Factorization. In *Advances in Neural Information Processing Systems*, 2023.

[14] Luigi Grippo, Francesco Lampariello, and Stefano Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world. *Transactions on Machine Learning Research*, 2022.

[17] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: The catapult mechanism. arXiv preprint arXiv:2003.02218, 2020.

[18] Panayotis Mertikopoulos, Nadav Hallak, Ali Kavis, and Volkan Cevher. On the Almost Sure Convergence of Stochastic Gradient Descent in Non-Convex Problems. In *Advances in Neural Information Processing Systems*, volume 33, pages 1117–1128. Curran Associates, Inc., 2020.

[19] Amirkeivan Mohtashami, Martin Jaggi, and Sebastian Stich. Special Properties of Gradient Descent with Large Learning Rates. arXiv preprint arXiv:2205.15142, 2023.

[20] Jurij Evgen'evič Nesterov. *Lectures on Convex Optimization*. Number volume 137 in Springer Optimization and Its Applications. Springer Nature, Cham, second edition edition, 2018. ISBN 978-3-319-91577-7.

[21] Yu Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1-2):381–404, 2015. ISSN 0025-5610, 1436-4646.

[22] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[23] Vassilis P Plagianakos, George D Magoulas, and Michael N Vrahatis. Deterministic nonmonotone strategies for effective training of multilayer perceptrons. *IEEE Transactions on Neural Networks*, 13(6):1268–1284, 2002.

[24] Marcos Raydan. The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997. ISSN 1052-6234, 1095-7189.

[25] Yinuo Ren, Chao Ma, and Lexing Ying. Understanding the Generalization Benefits of Late Learning Rate Decay. In *International Conference on Artificial Intelligence and Statistics*, 2024.

[26] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[27] Vincent Roulet, Atish Agarwala, and Fabian Pedregosa. On the Interplay Between Stepsize Tuning and Progressive Sharpening. arXiv preprint arXiv:2312.00209, 2023.

[28] Vincent Roulet, Atish Agarwala, Jean-Bastien Grill, Grzegorz Swirszcz, Mathieu Blondel, and Fabian Pedregosa. Stepping on the edge: Curvature aware learning rate tuners. *arXiv preprint arXiv:2407.06183*, 2024.

[29] Z.-J. Shi and J. Guo. A new family of conjugate gradient methods. *Journal of Computational and Applied Mathematics*, 224(1):444–457, 2009.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *Advances in Neural Information Processing Systems*, pages 3732–3745, 2019.

[32] Yuqing Wang, Zhenghao Xu, Tuo Zhao, and Molei Tao. Good regularity creates large learning rate implicit biases: Edge of stability, balancing, and catapult. arXiv preprint arXiv:2310.17087, 2023.

[33] Jingfeng Wu, Peter L. Bartlett, Matus Telgarsky, and Bin Yu. Large Stepsize Gradient Descent for Logistic Loss: Non-Monotonicity of the Loss Improves Optimization Efficiency. arXiv preprint arXiv:2402.15926, 2024.

[34] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020.

[35] Hongchao Zhang and William W Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14(4):1043–1056, 2004.

Appendix A. Mathematical Details

A.1. Proof of Lemma 1

Lemma 1 *Let f be L-Lipschitz smooth. Given an initial step size $\eta_{k,0} > 0$, then the step size η_k returned by (2) and (1) is either*

$$\begin{cases} \eta_k = \eta_{k,0} & \text{if } l_k = 0, \\ \eta_k \geq \frac{2\beta(1-c)}{L} & \text{if } l_k > 0. \end{cases} \tag{4}$$

Proof Let us give the proof only for (2). For the use of (1), simply replace C_k with $f(w_k)$. Let us denote $g_k := \nabla f(w_k)$. As f is L-smooth, we apply the Descent Lemma on $w_k - \eta_k g_k$ and w_k to get

$$\begin{aligned} f(w_k - \eta_k g_k) &\leq f(w_k) + g_k^T(w_k - \eta_k g_k - w_k) + \frac{\eta_k^2 L}{2}\|g_k\|^2 \\ &= f(w_k) - \left(\eta_k - \frac{\eta_k^2 L}{2}\right)\|g_k\|^2, \end{aligned}$$

which can be rewritten as

$$f(w_k - \eta_k g_k) \leq p_k(\eta_k), \quad \text{with } p_k(\eta) := f(w_k) - \left(\eta - \frac{\eta^2 L}{2}\right)\|g_k\|^2. \tag{5}$$

Note that (5) is valid for any η. Let us rewrite (2) as

$$f(w_k - \eta_k g_k) \leq q_k(\eta_k), \quad \text{with } q_k(\eta) := C_k - c\eta\|g_k\|^2.$$

Now, the backtracking procedure in (2) admits two possible output:
Case 1: $l_k = 0$. In this case, we have $\eta_k = \eta_{k,0}$.
Case 2: $l_k > 0$. In this case, we have $\eta_k < \eta_{k,0}$ with $f(w_k - \frac{\eta_k}{\beta} g_k) > q_k(\frac{\eta_k}{\beta})$. Then, we have that $q_k(\frac{\eta_k}{\beta}) \leq p_k(\frac{\eta_k}{\beta})$ because $q_k(\frac{\eta_k}{\beta}) > p_k(\frac{\eta_k}{\beta})$ would lead to a contradiction. In fact

$$f\left(w_k - \frac{\eta_k}{\beta} g_k\right) > q_k\left(\frac{\eta_k}{\beta}\right) > p_k\left(\frac{\eta_k}{\beta}\right) \geq f\left(w_k - \frac{\eta_k}{\beta} g_k\right)$$

is false. Thus, it has to be $q_k(\frac{\eta_k}{\beta}) \leq p_k(\frac{\eta_k}{\beta})$, from which we get that

$$f(w_k) - c\frac{\eta_k}{\beta}\|g_k\|^2 \leq C_k - c\frac{\eta_k}{\beta}\|g_k\|^2 \leq f(w_k) - \left(\frac{\eta_k}{\beta} - \frac{\eta_k^2 L}{2\beta^2}\right)\|g_k\|^2$$

and consequently

$$-c \leq -\left(1 - \frac{\eta_k L}{2\beta}\right) \Leftrightarrow \eta_k \geq \frac{2\beta(1-c)}{L},$$

which leads to (4). ■

A.2. Derivation of L_{approx}

Note that L_{approx} is computed by treating the descent lemma

$$f(w_{k+1}) \leq f(w_k) - \eta_k(1 - \frac{L\eta_k}{2})\| \nabla f(w_k)\|^2,$$

as an equality, and solving for L. Then

$$L_{\text{approx}} = \frac{2\left(f(w_{k+1}) - f(w_k)\right)}{\eta_k^2||\nabla f(w_k)||^2} + \frac{2}{\eta_k}.$$

A.3. Barzilai-Borwein Approximation Derivation

As discussed in Section 4, when using large monotone steps, we observe that $L_{BB_1} \approx L_{BB_2}$. Using this, we can derive by Cauchy-Schwarz that there exists $\alpha \in \mathbb{R} : s_k \approx \alpha y_k$. This implies that GD with large step sizes loosely solves a secant equation (as in Quasi-Newton methods [22]) where the Hessian is approximated by a scaled identity matrix (as in the BB method [5]). In particular, if we assume that $L_{BB_1} = L_{BB_2}$ then

$$\frac{||\nabla f(w_{k+1}) - \nabla f(w_k)||^2}{|(w_{k+1} - w_k)^T(\nabla f(w_{k+1}) - \nabla f(w_k)|} = \frac{|(w_{k+1} - w_k)^T(\nabla f(w_{k+1}) - \nabla f(w_k)|}{||w_{k+1} - w_k||^2}$$

which can be rewritten as

$$\begin{aligned} ||\nabla f(w_{k+1}) - \nabla f(w_k)||^2 \cdot ||w_{k+1} - w_k||^2 &= |(w_{k+1} - w_k)^T(\nabla f(w_{k+1}) - \nabla f(w_k))|^2 \\ &\leq ||w_{k+1} - w_k||^2 \cdot ||\nabla f(w_{k+1}) - \nabla f(w_k)||^2 \end{aligned}$$

by Cauchy-Schwarz. In particular, it means that Cauchy-Schwarz holds as an equality which implies that $w_{k+1} - w_k$ and $\nabla f(w_{k+1}) - \nabla f(w_k)$ are linearly dependent, i.e.,

$$\exists \alpha : \quad w_{k+1} - w_k = \alpha(\nabla f(k+1) - \nabla f(w_k))$$

as required. In other words, our results suggest that GD with large (nonmonotone) steps yielded by NLS-noTune approximately behave like GD with a BB steps, without computing any Rayleigh quotients.

Appendix B. Additional Experimental Details

B.1. Models

All the models use the PyTorch default for the initialization of model parameters. We include bias parameters in all of our models. The CNN model consists of the following structure:

- convolutional layer with 3 input channels and 32 output channels
- ReLU activation
- average pooling layer
- convolutional layer with 32 input channels and 32 output channels
- ReLU activation
- average pooling layer
- linear layer with input size 2048 and output of size 10

For the resnet34 and vgg11 experiments, we use the Pytorch implementations of the resnet34 and vgg11 models. Similarly to Roulet et al. [27], we remove all batch normalization layers in the resnet34 experiments, and do not use any dropout in the vgg11 experiments.

B.2. Optimization

For all line search methods, $c = 0.5$ and $\beta = 0.5$. The maximum step size is set to 10 for all line search methods. For the Armijo line search, the initial step size is set to 1 on each iteration. For the PoNLS line search, the initial step size is set using the Polyak step size on each iteration, see the work by Galli et al. [12]. We do not use any regularization in our experiments. For the CNN and resnet34 experiments, we train for 6000 epochs. For the vgg11 experiments, we train for 4000 epochs.

B.3. Calculation of Sharpness

For the experiments that compute the sharpness, the sharpness is computed using the *eigenvalues* function in the PyHessian package [34], with the *maxIter* parameter set to 100, *tol* parameter set to 10^{-3}, and the *top_n* parameter set to 1.

B.4. Plotting

For all of our plots, we plot the corresponding value every 10th epoch. For all the sharpness * step size plots, we also take the average over a window of 50 points to smooth the results for each point. For the plots in Figure 3 (bottom row), as well as Figures 6, 7, and 8, we instead take the average over a window of 10 points. To achieve this, we use the numpy function *convolve*, with *v* parameter set to $\frac{\text{np.ones(window-size)}}{\text{window-size}}$ and the *mode* parameter set to "valid".

Appendix C. Additional Experiments

In Figure 4, we see that for both monotone line searches, the sharpness continues to increase a lot with time. For the nonmonotone line searches, we do not see this large increase in sharpness with time. We show in Figure 4 that although the monotone line searches lead to values above the threshold of 2 for sharpness * step size, we see that this is due to larger sharpness values as well as smaller stepsizes (see Figures 4 and 5) than those seen with the nonmonotone line searches.

In Figure 6 we plot the relative error (i.e., $\frac{|\text{sharpness}-\text{approx}|}{\text{sharpness}}$) of approximating the sharpness where approx is either L_{approx} discussed in Section 4 or $2/\eta$, where η is the step size. In the case of the monotone line search, both choices seem to be poor at approximating the sharpness. For the nonmonotone line search, L_{approx} is comparable to or a better approximation of the sharpness than the value of $2/\eta$. An overlooked side effect of the edge of stability phenomenon is that GD estimates the sharpness L with the value $2/\eta$ while in this phase. In the case of the CNN model and VGG11 model, L_{approx} achieves a low relative error of around 10% or less in most iterations.

In Figures 7 and Figure 8, we consider an additional approximation of L proposed in [29], i.e., the following Barzilai-Borwein (BB) [5] formula

$$L_{BB_3} = \frac{||y_k||}{||s_k||} \quad \text{with } s_k := w_k - w_{k-1}, y_k := \nabla f(w_k) - \nabla f(w_{k-1}).$$

We also consider the L_{BB_1} and L_{BB_2} formulas as well as L_{approx} as before. Finally, we plot the relative error (i.e., $\frac{|\text{sharpness}-\text{approx}|}{\text{sharpness}}$) of approximating the sharpness where approx is one of the 4 metrics dicussed. In Figure 7 we focus on the monotone Armijo free method, and in Figure 8 we look at the nonmonotone NLS method. We see in Figures 7 and Figure 8 that L_{approx} is at least

competitive with L_{BB_1}, L_{BB_2}, and L_{BB_3} for the non-monotone line search, but this is not the case for the monotone line search. Additionally, for the monotone line search, we see that in most cases all 4 of the chosen metrics seem to be poor approximations of the sharpness.

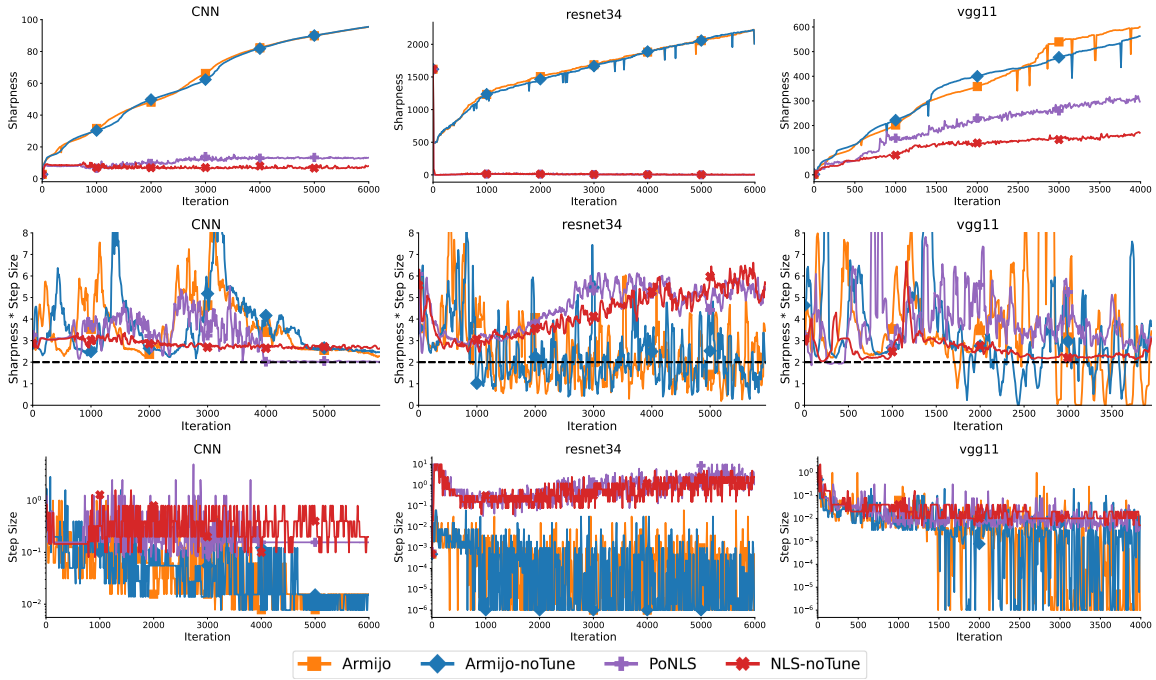

Figure 4: In this plot we show the sharpness (Top Row), sharpness * step size (Middle Row), and step size (Bottom Row) for 2 variants of gradient descent with Armijo line search and 2 variants of gradient descent with nonmonotone line search.

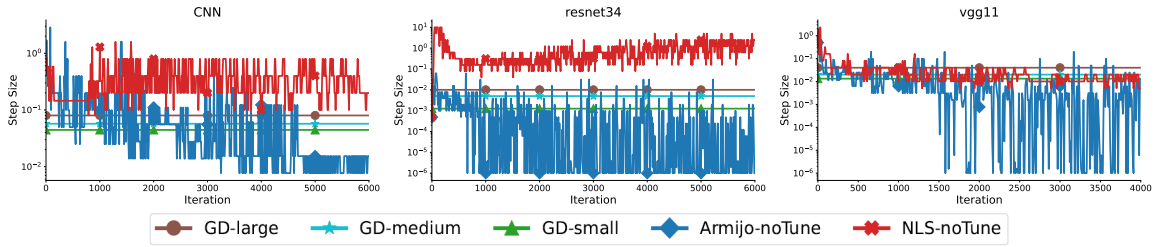

Figure 5: We plot the step size for gradient descent with Armijo-noTune line search, NLS-noTune line search, and 3 constant step sizes. The nonmontone NLS-noTune method uses larger (often significantly larger) step sizes than the Armijo-noTune method.

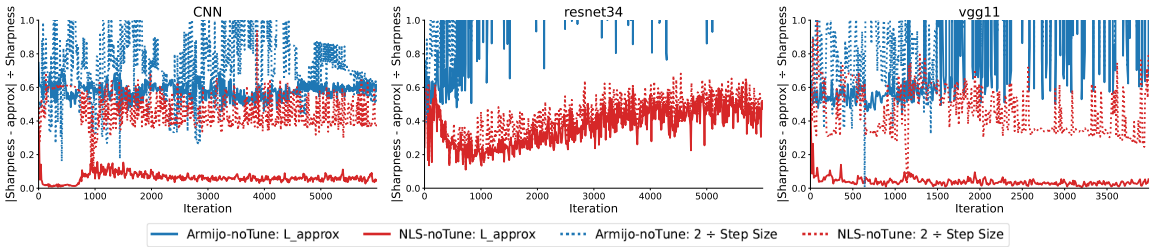

Figure 6: We plot the relative error in approximating the sharpness using $2/\eta$ and an approximation of the Lipschitz constant via the Descent Lemma, which we call L_{approx}. We observe that both metrics are poor approximations of the sharpness when using the monotone Armijo-noTune method. For the nonmonotone NLS-noTune method, L_{approx} achieves low relative error and is about as good or better at approximating the sharpness than the $2/\eta$ metric.

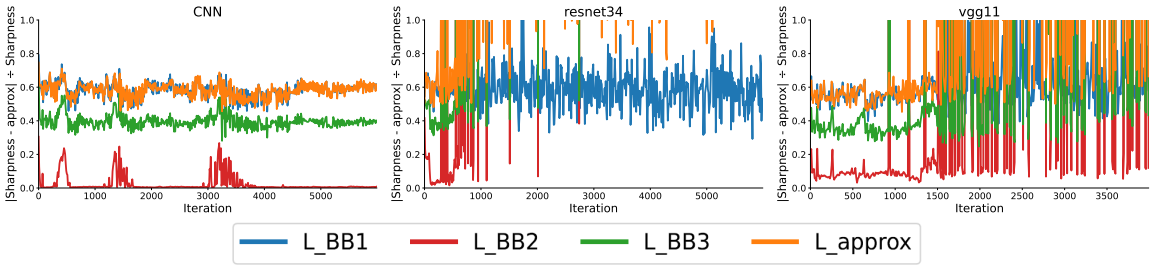

Figure 7: In this figure, we focus only on the Armijo-noTune method. We plot the relative error in approximating the sharpness using the L_{BB_1}, L_{BB_2}, and L_{BB_3} formulas as well an approximation of the Lipschitz constant via the Descent Lemma, which we call L_{approx}. In all cases other than using the L_{BB_2} formula with the CNN model, all 4 metrics are poor approximations of the sharpness.

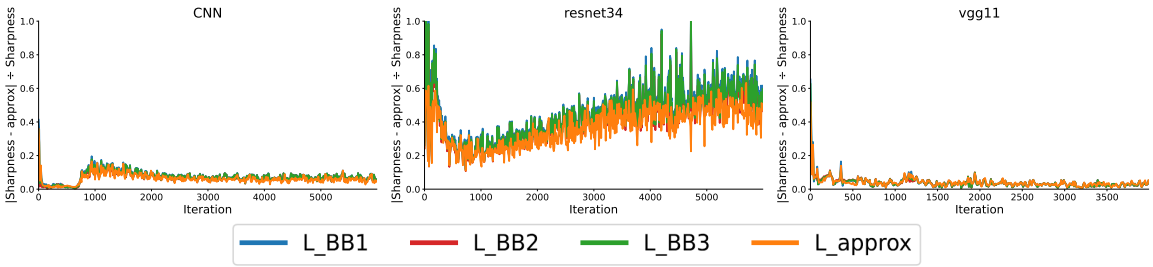

Figure 8: In this figure, we focus only on the NLS-noTune method. We plot the relative error in approximating the sharpness using the L_{BB_1}, L_{BB_2}, and L_{BB_3} formulas as well an approximation of the Lipschitz constant via the Descent Lemma, which we call L_{approx}. For the CNN and vgg11 models, the 4 metrics have low relative error in approximating the sharpness. For each of the 3 models, the relative error for each of the metrics are very close in value.