OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Provable Benefit of Sign Descent: A Minimal Model Under Heavy-Tail Class Imbalance

Robin Yadav[1,3] ROBINY12@STUDENT.UBC.CA
Shuo Xie[1] SHUOX@TTIC.EDU
Tianhao Wang[1,2] TIANHAOWANG@UCSD.EDU
Zhiyuan Li[1] ZHIYUANLI@TTIC.EDU
[1] *Toyota Technological Institute at Chicago*
[2] *University of California, San Diego*
[3] *University of British Columbia*

Abstract

Adaptive optimization methods (such as Adam) play a major role in LLM pretraining, significantly outperforming Gradient Descent (GD). Recent studies have proposed new smoothness assumptions on the loss function to explain the advantages of adaptive algorithms with structured preconditioners, e.g., coordinate-wise or layer-wise, and steepest descent methods w.r.t. non-euclidean norms, e.g., ℓ_∞ norm or spectral norm, over GD. However, it remains unclear how these smoothness assumptions manifest in language modelling tasks. In this work, we aim to analyze the benefit of ℓ_∞-norm descent (a.k.a. sign descent) directly from properties of the data distribution, namely, heavy-tailed class imbalance. We propose a minimal yet representative setting of next-token prediction, where we can provably show faster convergence of coordinate-wise algorithms such as Sign descent (steepest descent w.r.t. ℓ_∞ norm) over normalized GD (steepest descent w.r.t. to ℓ_2 norm) in the presence of heavy tail class imbalance.

1. Introduction

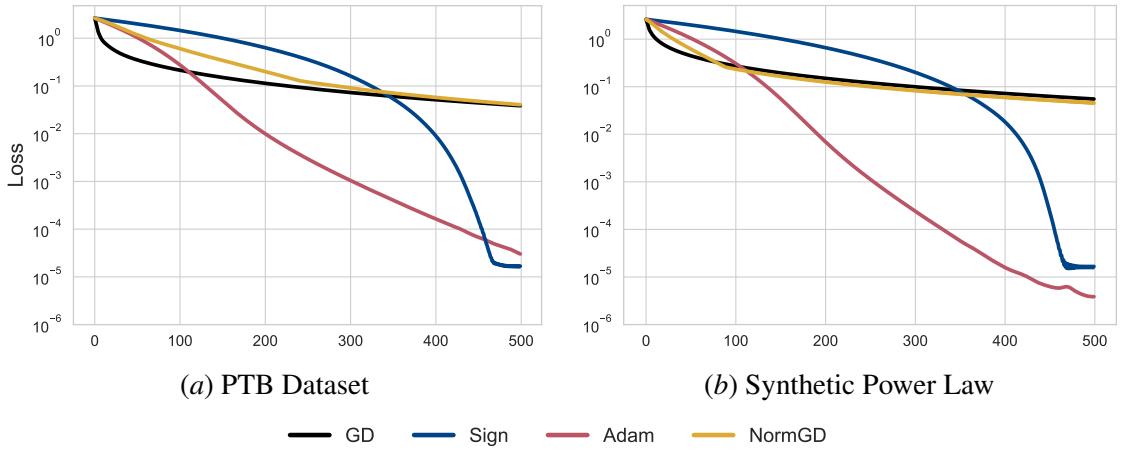
Adaptive coordinate-wise methods are the go-to class of optimizers for modern deep learning problems [2]. In particular, the Adam optimizer [5] and its variants [11] are prevalent in LLM pretraining, where they significantly surpass the performance of conventional rotationally invariant SGD methods [8, 16]. Despite this remarkable empirical success, we still lack a complete theoretical understanding of why Adam converges faster than SGD for language modelling tasks.

Recently, a growing body of work has explored new assumptions under which adaptive coordinate-wise algorithms and non-euclidean steepest descent methods achieve faster convergence than SGD [3, 4, 10, 16]. Specifically, these studies introduce new smoothness assumptions on the loss function, typically expressed as an upper bound on its Hessian. However, it remains unclear how these smoothness assumptions manifest in language modelling tasks and what properties of the dataset or network architecture they emerge from.

Recently, Kunstner et al. [8] identified *heavy-tailed class imbalance* in language datasets as a key property that induces a performance gap between Adam and SGD. In language data, word frequency typically follows Zipf's law; the k-th most frequent word has frequency $p_k \propto \frac{1}{k}$ [13]. Next token prediction suffers from heavy-tailed class imbalance because word frequency is inherited by the tokens. Under such conditions, SGD makes slow progress on low-frequency classes, which dominate the loss, resulting in poor overall performance. On the other hand, Adam is less sensitive to this issue

and reduces loss on all classes, regardless of their frequency, leading to faster overall convergence. Interestingly, Adam outperforms GD even when training just the classification head of a simple one-layer transformer (embedding and attention weights frozen) in the full-batch setting [8].

In this work, we take the initial steps towards providing a complete analysis that explains the benefit of coordinate-wise adaptive algorithms over GD on language tasks. We avoid going down the route of proposing intermediate smoothness assumptions. Instead, we analyze the benefit of sign-based methods and non-Euclidean steepest descent directly from the properties of network architecture and data distribution, namely, heavy-tailed class imbalance. Inspired by the simple transformer setting in Kunstner et al. [8], we aim to design the simplest possible language modelling problem where we can provably show faster convergence of coordinate-wise algorithms such as Sign descent (steepest descent w.r.t. ℓ_∞ norm) over GD.



(*a*) PTB Dataset (*b*) Synthetic Power Law

Figure 1: **GD and NormGD struggle to optimize a simple softmax unigram model with heavy-tail class imbalance** . This result holds on a real-world dataset and synthetically generated data following a power-law distribution $p_k \propto \frac{1}{k}$.

In summary, our main contributions are:

1. We introduce a simple convex, smooth problem with heavy-tail class imbalance, as shown in Figure 1, where coordinate-wise algorithms such as Adam outperform GD by a large margin.
2. Within this minimal setting, we show that sign descent with weight decay provably converges faster than normalized GD with weight decay.

2. Background and Preliminaries

Notations. We say a function f is L-smooth w.r.t. norm $||\cdot||$ if for all $x, y \in \mathbb{R}^d$ we have $||\nabla f(x) - \nabla f(y)||_\star \leq L||x-y||$ where $||\cdot||_\star$ denotes the dual norm. Let the smallest such constant be denoted by $L_{||\cdot||}(f)$. We use $\|\cdot\|_p$ to denote the the ℓ_p norm for $p \in [1, \infty]$. For a positive semi-definite matrix $\mathbf{A}$, the induced matrix norm is $\|x\|_{\mathbf{A}} = \sqrt{x^T \mathbf{A} x}$. We denote the set of minimizers of f as $\arg\min f$. We denote the softmax activation with $\sigma : \mathbb{R}^d \to \mathbb{R}^d$.

Steepest Descent. Steepest descent with respect to a norm $||\cdot||$ is a general algorithm which iteratively minimizes a local quadratic upper bound on the loss i.e. $x_{t+1} = \arg\min_{x \in \mathbb{R}^d} \nabla f(x_t)^T (x -$

$x_t)+\frac{1}{2\eta_t}||x-x_t||^2$. If we constrain the update direction to have unit norm, we obtain normalized steepest descent (NSD). The update step of NSD with weight decay factor λ is $x_{t+1} = (1-\lambda\eta_t)x_t - \eta_t\Delta_t$ where Δ_t is the normalized steepest descent direction. Sign Descent and Normalized GD are instances of normalized steepest descent w.r.t. ℓ_∞ and ℓ_2 norms, respectively. We restate the convergence result for normalized steepest descent with weight decay (NSD-WD) provided by Xie and Li [15] in the next theorem. In Section 3, it will be useful in comparing different normalized steepest descent on a specific problem once $x_\star$ and L are obtained.

Theorem 2.1 *For any minimizer $x_\star$, suppose we run normalized steepest descent with weight decay of $\lambda \leq \frac{1}{\|x_\star\|}$ and learning rate of $\eta_t = \frac{2}{\lambda(t+1)}$. Suppose $B = \max\{\frac{1}{\lambda}, \|x_0\|\}$. Then the iterates $\{x_t\}_{t=1}^T$ satisfy,*

$$f(x_T) - f^\star \leq \frac{2L(1+B\lambda)^2}{\lambda^2(T+2)}.$$

In particular, if we initialize $x_0 = 0$ and select λ optimally, i.e., $\lambda = 1/\min_{x_\star \in \arg\min f} \|x_\star\|$, we have,

$$f(x_T) - f^\star \leq \frac{\mathcal{C}_{\|\cdot\|}(f)}{T+2},$$

where we define $\mathcal{C}_{\|\cdot\|}(f) \triangleq 8L \min_{x_\star \in \arg\min f} \|x_\star\|^2$ as the complexity of convex function f under norm $\|\cdot\|$.

3. Softmax Unigram Model

In this section, we construct a convex and smooth problem where we can provably show that Sign descent converges faster than normalized GD. Although this problem is simple, it effectively captures the advantage of sign-based methods and ℓ_∞ smoothness over GD in the presence of language data with heavy-tailed class imbalance. Concretely, let the vocabulary consist of d tokens, and let $p \in \mathbb{R}^d$ denote the vector of token proportions (sorted in decreasing order), where p_k represents the proportion of token k in the dataset. We impose the following assumption on p, which characterizes the notion of heavy-tailed class imbalance.

Assumption 3.1 *For $k \in [d]$, we assume that $p_k = \frac{k^{-1}}{\sum_j j^{-1}}$ i.e. $p_k \propto \frac{1}{k}$.*

Now, let's consider the following minimization problem,

$$\boxed{f(\theta) = \mathrm{KL}(p \parallel \mathrm{softmax}(\theta))} \tag{1}$$

Minimizing f corresponds to learning a unigram model of the data where the tokens are observed from a categorical distribution specified by p. In fact, it is equivalent to learning a "transformer" model with zero attention layers where every token is mapped to the same embedding vector. Despite this simplification, it captures the optimization difficulty that rotation-invariant algorithms such as GD face when training on language data. In Figure 1, we compare the performance of sign-based methods and GD when p satisfies Assumption 3.1 with $d = 10^3$ and observe a significant gap in performance. Now that we have this minimal setting, we benefit from being able to determine tight lower and upper bounds on the smoothness constants as presented in the following lemma.

Lemma 3.1 *We have the following bounds on the $L_{\|\cdot\|_2}(f)$ and $L_{\|\cdot\|_\infty}(f)$ smoothness constants of f in Eq. 1,*

$$\frac{1}{2} \leq L_{\|\cdot\|_2}(f) \leq 1, \text{ and } \quad L_{\|\cdot\|_\infty}(f) = 1.$$

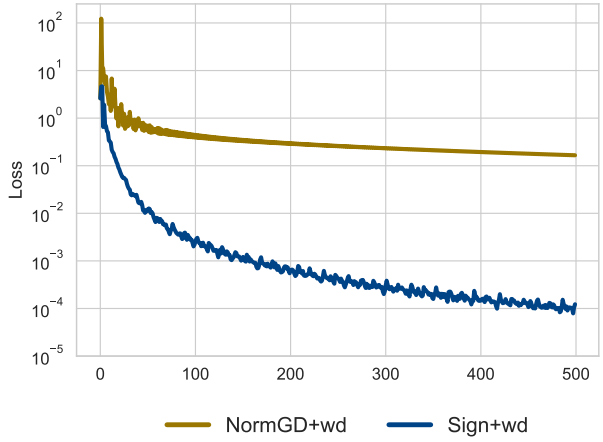


Figure 2: Performance of NSD with weight decay when minimizing f with $d = 10^3$. For each optimizer, we set $\lambda = \frac{1}{\min_{\theta_\star \in \arg\min f} \|\theta_\star\|}$ and use a learning rate of $\eta_t = \frac{2}{\lambda(t+1)}$.

To have a complete theoretical justification of the convergence rates, we must compute the optimal weight decay factor λ for each norm. Thus, in the following lemma, we establish the minimal norm solution for the ℓ_∞ and ℓ_2 norms.

Lemma 3.2 *For the optimal set of $\theta_\star$ of f, we have that,*

$$\min_{\theta_\star \in \arg\min f} \|\theta_\star\|_\infty = \frac{1}{2}\log(d), \text{ and } \quad \min_{\theta_\star \in \arg\min f} \|\theta_\star\|_2 = \sqrt{d \operatorname{Var}_{k\sim \mathrm{Unif}[d]}[\log(k)]}.$$

For language modelling problems, the vocabulary size d tends to be very large. We use this fact to prove the following lemma, which will allow us to estimate the minimal ℓ_2 norm of the optimal solution $\min_{\theta_\star \in \arg\min f} \|\theta_\star\|_2$.

Lemma 3.3 *For large d, we have that,*

$$\operatorname{Var}_{k\sim \mathrm{Unif}[d]}[\log(k)] = \Theta(1).$$

With Lemma 3.1 characterizing $L_{\|\cdot\|_2}(f)$ and $L_{\|\cdot\|_\infty}(f)$ and Lemma 3.2 characterizing the minimal norm solutions $\|\theta_\star\|_\infty$ and $\|\theta_\star\|_2$, we can compare the complexity of f under the ℓ_∞ and ℓ_2 norms. The next theorem demonstrates that for large d, the complexity of f under the ℓ_∞ norm is much smaller than the complexity of f under the ℓ_2 norm.

Theorem 3.1 *Suppose we have f defined in Equation (1). Then for large d,*

$$\mathcal{C}_{\|\cdot\|_\infty}(f) = 2\log(d)^2 \ll d \sim \mathcal{C}_{\|\cdot\|_2}(f).$$

Using Theorem 2.1 together with Theorem 3.1, we show that the convergence rate bound for normalized GD with weight decay is much smaller than that for Sign descent with weight decay.

Corollary 3.1 *Consider optimizing f in Equation (1) with large d and initialized at $\theta_0 = 0$. Then the iterates of Sign descent with weight decay $\lambda_\infty = \frac{2}{\log(d)}$ and learning rate $\eta_t = \frac{1}{\lambda_\infty (t+1)}$ satify,*

$$f(\theta_T) - f^\star \leq \frac{2\log(d)^2}{T+1}.$$

Furthermore, the iterates of normalized GD with weight decay $\lambda_2 = \frac{1}{\sqrt{d \operatorname{Var}_{k\sim \operatorname{Unif}[d]}[\log(k)]}}$ and learning rate $\eta_t = \frac{1}{\lambda_2 (t+1)}$ satisfy,

$$f(\theta_T) - f^\star \leq \frac{d}{T+1}.$$

Our theoretical investigation predicts that Sign descent should converge much faster than normalized GD (with weight decay). In Figure 2, we verify our results by showing the empirical performance of sign descent against normalized GD (with weight decay) when optimizing f using theoretically justified learning rates and λ according to Theorem 2.1. As suggested by Theorem 3.1, sign descent with weight decay significantly outperforms normalized GD with weight decay. We also show that our analysis and experimental results hold for a modified softmax example in Appendix C.

4. Challenges in Extending to Adam

Prior works have constructed simple, ill-conditioned, diagonal quadratic functions where Adam and sign descent outperform (normalized) GD [8, 15, 16]. Our work is motivated by distilling transformer models on language data to a minimal setting where we can provably show faster convergence of adaptive coordinate-wise methods. Although diagonal quadratics may appear simpler than our formulation, we argue that they are less representative of language modeling. In particular, our construction incorporates the softmax projection, a core component of transformer architectures. Thus, our setting is simple but also serves as an abstraction of training on language data with heavy-tailed class imbalance.

Adaptive smoothness. Xie et al. [16, 17] propose another explanation of the gap between coordinate-wise algorithms and GD. Namely, they argue that the commonly assumed ℓ_2 smoothness in convergence analysis is not a tight enough characterization of the loss function to explain the optimization benefits of Adam over SGD. To rectify the existing gap in theory, they generalize the notion of smoothness to adaptive methods with general structured preconditioners, including Adam, blockwise Adam, and one-sided shampoo.

Definition 4.1 (Adaptive Smoothness) *The adaptive smoothness of a function f w.r.t. a subalgebra $\mathcal{K}$ is defined as the smallest smoothness of f w.r.t. all norm $\|\cdot\|_{\mathbf{A}}$ where $\mathbf{A} \in \mathcal{K}, \mathbf{A} \succeq 0, \operatorname{Tr}(\mathbf{A}) \leq 1$, that is,*

$$\mathcal{L}_{\mathcal{K}}(f) = \min_{\substack{\mathbf{A}\in\mathcal{K} \\ \mathbf{A}\succeq 0 \\ \operatorname{Tr}(\mathbf{A})\leq 1}} L_{\|\cdot\|_{\mathbf{A}}}(f) = \min_{\substack{\mathbf{A}\in\mathcal{K} \\ \forall x, -\mathbf{A}\preceq\nabla^2 f(x)\preceq\mathbf{A}}} \operatorname{Tr}(\mathbf{A}). \tag{2}$$

In particular, when the subalgebra $\mathcal{K}$ is the set of diagonal matrices, we call the above smoothness notion the diagonal adaptive smoothness*, and it is the minimal trace of the diagonal matrix that dominates the Hessian:*

$$\mathcal{L}_{diag}(f) = \min_{\substack{\mathbf{A} \text{ diag} \\ \forall x, \nabla^2 f(x) \preceq \mathbf{A}}} \operatorname{Tr}(\mathbf{A}). \tag{3}$$

The diagonal adaptive smoothness of f is equivalent to anisotropic smoothness [10], *i.e.* 1-smoothness under the norm induced by $\mathbf{A}^\star$ where $\mathbf{A}^\star$ is the matrix that achieves the minimum trace in Equation (2). The convergence analysis for convex problems in Xie et al. [17] suggests that adaptive coordinate-wise algorithms such as Adam optimize faster than rotationally-invariant SGD methods when $\mathcal{L}_{\text{diag}}(f) \ll dL_{\|\cdot\|_2}(f)$. The following lemma provides a lower bound for the adaptive smoothness constant of f in Equation (1).

Lemma 4.1 *Consider f defined in Eq. 1. Then the diagonal adaptive smoothness of f is at least $\frac{d}{2}$.*

This lemma demonstrates that adaptive smoothness is not much smaller than $dL_{\|\cdot\|_2}(f)$ for f given in Equation (1). Consequently, adaptive smoothness alone cannot account for the advantages of Adam over rotationally invariant algorithms such as SGD in this setting. This is a key distinction between the softmax unigram model and potentially simpler diagonal quadratics setups. For instance, consider the simple diagonal quadratic example proposed in Kunstner et al. [8], $g(\theta) = \sum_{k=1}^{d} p_k \theta_k^2$ where $p \in \mathbb{R}^d$ satisfies Assumption 3.1. The Hessian of g is $\nabla^2 g(\theta) = \operatorname{diag}(p)$, and its adaptive smoothness is $\mathcal{L}_{\text{diag}}(g) = 1$, which is much smaller than $dL_{\|\cdot\|_2}(f)$ for large d. Thus, while adaptive smoothness predicts faster convergence of Adam over SGD for diagonal quadratics with heavy-tailed class imbalance, it fails to do so for the softmax unigram model, highlighting the difference of our setup from quadratic models. We leave the theoretical explanation of why Adam outperforms GD empirically on this softmax unigram model to future work.

5. Related Works

Prior work provides both convergence analyses and empirical evidence aimed at explaining the performance gap between Adam and SGD. Zhang et al. [18] suggests that SGD struggles more with heavy-tailed gradient noise, while Kunstner et al. [7] shows that the gap persists even in the full-batch setting, with deterministic Adam resembling sign descent [1]. Kunstner and Bach [6] proves scaling laws for sign and gradient descent on a linear bigram model with quadratic loss. Other explanations include coordinate-wise clipping reducing directional sharpness [12] and block-wise Hessian heterogeneity in transformers, favoring Adam [19]. Finally, Levy et al. [9], Ward et al. [14] show that Adam can achieve optimal convergence rates without relying on problem-dependent constants.

6. Conclusion and Future Works

We focus on a simplified setting of language modelling where we can provably show that non-Euclidean steepest descent methods converge faster than GD with weight decay. Future work includes extending the analysis to more complex setups, such as the softmax linear bigram model, which we already have some initial results in Appendix B. It also remains to develop adaptive smoothness assumptions that better capture the gap between Adam and SGD.

7. Acknowledgements

This work is supported by Darpa AIQ grant and OpenAI superalignment grant.

References

[1] Lukas Balles and Philipp Hennig. Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients. In *Proceedings of the 35th International Conference on Machine Learning*, pages 404–413. PMLR, July 2018. URL https://proceedings.mlr.press/v80/balles18a.html. ISSN: 2640-3498.

[2] Jeremy Bernstein and Laker Newhouse. Old Optimizer, New Norm: An Anthology, December 2024. URL http://arxiv.org/abs/2409.20325. arXiv:2409.20325 [cs].

[3] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed Optimisation for Non-Convex Problems. In *Proceedings of the 35th International Conference on Machine Learning*, pages 560–569. PMLR, July 2018. URL https://proceedings.mlr.press/v80/bernstein18a.html. ISSN: 2640-3498.

[4] Ruichen Jiang, Devyani Maladkar, and Aryan Mokhtari. Provable Complexity Improvement of AdaGrad over SGD: Upper and Lower Bounds in Stochastic Non-Convex Optimization, June 2025. URL http://arxiv.org/abs/2406.04592. arXiv:2406.04592 [math].

[5] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

[6] Frederik Kunstner and Francis Bach. Scaling Laws for Gradient Descent and Sign Descent for Linear Bigram Models under Zipf's Law, May 2025. URL http://arxiv.org/abs/2505.19227. arXiv:2505.19227 [cs].

[7] Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise Is Not the Main Factor Behind the Gap Between Sgd and Adam on Transformers, But Sign Descent Might Be. September 2022. URL https://openreview.net/forum?id=a65YK0cqH8g.

[8] Frederik Kunstner, Alan Milligan, Robin Yadav, Mark Schmidt, and Alberto Bietti. Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on Language Models. November 2024. URL https://openreview.net/forum?id=msW3fL8J1D.

[9] Kfir Levy, Ali Kavis, and Volkan Cevher. STORM+: Fully Adaptive SGD with Recursive Momentum for Nonconvex Optimization. In *Advances in Neural Information Processing Systems*, volume 34, pages 20571–20582. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ac10ff1941c540cd87c107330996f4f6-Abstract.html.

[10] Yuxing Liu, Rui Pan, and Tong Zhang. AdaGrad under Anisotropic Smoothness. October 2024. URL https://openreview.net/forum?id=4GT9uTsAJE.

[11] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs].

[12] Yan Pan and Yuanzhi Li. Toward Understanding Why Adam Converges Faster Than SGD for Transformers, May 2023. URL https://arxiv.org/abs/2306.00204v1.

[13] Steven T. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130, October 2014. ISSN 1531-5320. doi: 10.3758/s13423-014-0585-6. URL https://doi.org/10.3758/s13423-014-0585-6.

[14] Rachel Ward, Xiaoxia Wu, and Leon Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. *Journal of Machine Learning Research*, 21(219):1–30, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/18-352.html.

[15] Shuo Xie and Zhiyuan Li. Implicit bias of AdamW: ℓ_∞-norm constrained optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 54488–54510, Vienna, Austria, July 2024. JMLR.org.

[16] Shuo Xie, Mohamad Amin Mohamadi, and Zhiyuan Li. Adam Exploits ℓ_∞-geometry of Loss Landscape via Coordinate-wise Adaptivity, June 2025. URL http://arxiv.org/abs/2410.08198. arXiv:2410.08198 [cs].

[17] Shuo Xie, Tianhao Wang, Sashank Reddi, Sanjiv Kumar, and Zhiyuan Li. Structured Preconditioners in Adaptive Optimization: A Unified Analysis, July 2025. URL http://arxiv.org/abs/2503.10537. arXiv:2503.10537 [cs].

[18] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are Adaptive Methods Good for Attention Models? In *Advances in Neural Information Processing Systems*, volume 33, pages 15383–15393. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b05b57f6add810d3b7490866d74c0053-Abstract.html.

[19] Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, and Zhi-Quan Luo. Why Transformers Need Adam: A Hessian Perspective. *Advances in Neural Information Processing Systems*, 37:131786–131823, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ee0e45ff4de76cbfdf07015a7839f339-Abstract-Conference.html.

Appendix A. Proofs for Unigram Model

A.1. Proof of Lemma 3.1

First, we can express f as,

$$f(\theta) = \sum_{i=1}^{d} \left[p_i \log(p_i) - p_i\theta_i + p_i \log\left(\sum_{j=1}^{d} \exp(\theta_j) \right) \right]$$

Computing the derivative of f w.r.t to θ_k and using the fact that $\sum_{i=1}^{d} p_i = 1$ we obtain,

$$\frac{\partial f}{\partial \theta_k} = -p_k + \sum_{i=1}^{d} p_i \frac{\exp(\theta_k)}{\sum_j \exp(\theta_j)} = -p_k + \sigma(\theta)_k \sum_{i=1}^{d} p_i = \sigma(\theta)_k - p_k.$$

Therefore, $\nabla f(\theta) = \sigma(\theta) - p$. Thus, the Hessian of f is simply the Jacobian of the softmax function. It's not too hard to compute,

$$\frac{\partial \sigma_i}{\partial \theta_i} = \sigma(\theta)_i(1 - \sigma(\theta)_i), \text{ and } \quad \frac{\partial \sigma_j}{\partial \theta_i} = -\sigma(\theta)_i \sigma(\theta)_j, \text{ for } i \neq j.$$

Therefore, $\nabla^2 f(\theta) = \operatorname{diag}(\sigma(\theta)) - \sigma(\theta)\sigma(\theta)^\top$.

A.1.1. UPPER BOUNDS

Now we can compute the upper bounds on the smoothness constants of f. Firstly, note that,

$$L_{\|\cdot\|_p}(f) = \sup_{\theta \in \mathbb{R}^d} \sup_{\|u\|_p \leq 1} u^\top \nabla^2 f(\theta) u.$$

Then, $\nabla^2 f(\theta) \preceq \operatorname{diag}(\sigma(\theta)) \preceq \mathbf{I}_d$. Thus $u^\top \nabla^2 f(\theta) u \leq u^T u = \|u\|_2^2$. Therefore, $L_{\|\cdot\|_2}(f) \leq 1$. For $\|u\|_\infty \leq 1$,

$$u^\top \nabla^2 f(\theta) u \leq \; u^\top \operatorname{diag}(\sigma(\theta)) u = \sum_{k=1}^{d} \sigma(\theta)_k u_k^2 \leq \sum_{k=1}^{d} \sigma(\theta)_k \, |u_k| \leq \|\sigma(\theta)\|_1 \|u\|_\infty \leq 1.$$

Since $\|u\|_\infty \leq 1$ we have that $u_k^2 \leq |u_k|$. The second-to-last inequality is because of Cauchy-Schwarz, and then we note that $\|\sigma(\theta)\|_1 = 1$.

A.1.2. LOWER BOUNDS

Next, we compute the lower bounds on $L_{\|\cdot\|_2}(f)$ and $L_{\|\cdot\|_\infty}(f)$. Consider $\theta(t) : \mathbb{R} \to \mathbb{R}^d$ as follows $\theta(t) = [t \; t \; -t \; \cdots \; -t]$. Then, define,

$$s(t) := \sigma(\theta(t))_1 = \sigma(\theta(t))_2 = \frac{e^t}{2e^t + (d-2)e^{-t}}.$$

For any $r \in \mathbb{R}^2$ with $\|r\|_2 = 1$ define $u = [r_1 \; r_2 \; 0 \; \cdots \; 0] \in \mathbb{R}^d$. Let,

$$\mathbf{A}(t) = \begin{bmatrix} s(t)(1-s(t)) & -s(t)^2 \\ -s(t)^2 & s(t)(1-s(t)) \end{bmatrix}$$

Then, $u^T \nabla^2 f(\theta(t)) u = r^T \mathbf{A}(t) r$. Since r is arbitrary, we see that,

$$\sup_{\|u\|_2=1} u^T \nabla^2 f(\theta(t)) u \geq \sup_{\|r\|_2=1} r^T \mathbf{A}(t) r.$$

Therefore, we must compute the largest eigenvalue of $\mathbf{A}(t)$. Solving the equation $\det(\mathbf{A}(t) - \lambda \mathbf{I}) = 0$, we get the following characteristic polynomial,

$$\lambda^2 - 2\lambda s(t)(1 - s(t)) + s(t)^2 - 2s(t)^3 = 0.$$

Solving for λ, we get the following two solutions,

$$\lambda_1 = s(t)(1 - s(t)) + s(t)^2 = s(t), \text{ and } \quad \lambda_2 = s(t)(1 - s(t)) - s(t)^2 = s(t) - 2s(t)^2.$$

Since $s(t) > 0$, it is clear that $\lambda_1 = s(t)$ is the greatest eigenvalue of $\mathbf{A}(t)$. Observe that $\lim_{t\to\infty} s(t) = \frac{1}{2}$. Now, we apply the following reasoning,

$$\begin{aligned} L_{\|\cdot\|_2}(f) = \sup_{\theta \in \mathbb{R}^d} \sup_{\|u\|_2=1} u^T \nabla^2 f(\theta) u &\geq \lim_{t\to\infty} \sup_{\|u\|_2=1} u^T \nabla^2 f(\theta(t)) u \\ &\geq \lim_{t\to\infty} \sup_{\|r\|_2=1} r^T \mathbf{A}(t) r \\ &= \lim_{t\to\infty} s(t) = \frac{1}{2}. \end{aligned}$$

Now, we compute the lower bound for $L_{\|\cdot\|_\infty}(f)$. We follow the same steps as above but instead consider $r \in \mathbb{R}^2$ with $\|r\|_\infty = 1$. Again, $u^\top \nabla^2 f(\theta(t)) u = r^\top \mathbf{A}(t) r$. Therefore,

$$\sup_{\|u\|_\infty=1} u^T \nabla^2 f(\theta(t)) u \geq \sup_{\|r\|_\infty=1} r^T \mathbf{A}(t) r.$$

Of course, $r_1, r_2 \in \{-1, 1\}$ and $\mathbf{A}(t)$ is symmetric, so it is easy to determine the supremum. In fact, it is achieved with $\bar{r} = [1, -1]$, $\bar{r}^\top \mathbf{A}(t) \bar{r} = 2s(t)$. Applying the same inequalities as before,

$$\begin{aligned} L_{\|\cdot\|_\infty}(f) = \sup_{\theta \in \mathbb{R}^d} \sup_{\|u\|_\infty=1} u^T \nabla^2 f(\theta) u &\geq \lim_{t\to\infty} \sup_{\|u\|_\infty=1} u^T \nabla^2 f(\theta(t)) u \\ &\geq \lim_{t\to\infty} \sup_{\|r\|_\infty=1} r^T \mathbf{A}(t) r \\ &= \lim_{t\to\infty} 2s(t) = 1. \end{aligned}$$

A.2. Proof of Lemma 3.2

Of course, for a given $\theta \in \arg\min f$ we must have that $\sigma(\theta) = p$. We note that σ is invariant up to a constant shift. Since $p_k = k^{-1} / \sum_i i^{-1}$, the set of optimal solutions is described by,

$$\theta_k = -\log(k) + c, \quad \text{for } k \in [d],$$

where $c \in \mathbb{R}$ is a fixed constant. Now, we find the c that minimizes $\|\theta_\star\|_\infty$ for $\theta_\star \in \arg\min f$. Since $\log$ is an increasing function, it is not too hard to see that the value of $\|\theta_\star\|_\infty = \max\{|c|, |-\log(d) + c|\}$ is achieved in the first or last entry. Therefore, the c that minimizes this quantity must be the midpoint *i.e.* $c = \frac{\log(d)}{2}$. Thus, $\min_{\theta_\star \in \arg\min f} \|\theta_\star\|_\infty = \frac{\log(d)}{2}$.

Next, we find the c that minimizes $\|\theta_\star\|_2$ for $\theta_\star \in \arg\min f$. Define $\gamma : \mathbb{R} \to \mathbb{R}$ as follows,

$$\gamma(c) = \sum_{k=1}^{d} (-\log(k) + c)^2 = \|\theta_\star\|_2^2.$$

Then,

$$\gamma'(c) = 2\sum_{k=1}^{d} (c - \log(k)), \text{ and } \quad \gamma''(c) = 2d.$$

Solving for $\gamma'(c) = 0$, we find that the optimal solution $c = \frac{1}{d}\sum_{k=1}^{d} \log(k)$. Since $\gamma''(c) > 0$, we indeed confirm that it is a minimum. Therefore,

$$\min_{\theta_\star \in \arg\min f} \|\theta_\star\|_2 = \sqrt{\sum_{k=1}^{d} \left(-\log(k) + \frac{1}{d}\sum_{j=1}^{d} \log(j) \right)^2} = \sqrt{d \operatorname{Var}_{k\sim \mathrm{Unif}[d]}[\log(k)]}$$

A.3. Proof of Lemma 3.3

Let $V_d = \operatorname{Var}_{k\sim \mathrm{Unif}[d]}[\log(k)]$. Now, observe that,

$$V_d = \frac{1}{d}\sum_{k=1}^{d} \log(k)^2 - \left(\frac{1}{d}\sum_{k=1}^{d} \log(k) \right)^2.$$

Let,

$$A = \sum_{k=1}^{d} \log(k)^2 = \sum_{k=2}^{d} \log(k)^2, \text{ and } \quad B = \sum_{k=1}^{d} \log(k) = \sum_{k=2}^{d} \log(k).$$

For increasing function $g : \mathbb{R} \to \mathbb{R}$, we have that $\int_1^d g(x) \le \sum_{k=2}^{d} g(k) \le \int_2^{d+1} g(k)$. Then note,

$$\int \log(x)dx = x\log(x) - x, \text{ and } \quad \int \log(x)^2 dx = x[\log(x)^2 - 2\log(x) + 2].$$

Now, we compute upper and lower bounds on $A_l \le A \le A_u$ and $B_l \le B \le B_u$:

$$\begin{aligned}
A_l &= d[\log(d)^2 - 2\log(d) + 2] - 2 \\
A_u &= (d+1)[\log(d+1)^2 - 2\log(d+1) + 2] - 2[\log(2)^2 - 2\log(2) + 2] \\
B_l &= d\log(d) - d + 1 \\
B_u &= (d+1)\log(d+1) - (d+1) - (2\log(2) - 2)
\end{aligned}$$

$$
\begin{aligned}
V_d &\le \frac{1}{d}A_u - \left(\frac{1}{d}B_l\right)^2 \\
&\le \frac{d+1}{d}[\log(d+1)^2 - 2\log(d+1) + 2] - (\log(d) - 1)^2 \\
&= \log(d+1)^2 - \log(d)^2 - 2\log(d+1) + 2\log(d) + 2 - 1 + \frac{1}{d}[\log(d+1)^2 - 2\log(d+1) + 2] \\
&\le \log(d+1)^2 - \log(d)^2 + 1 + \frac{1}{d}\log(d+1)^2 \\
&= [\log(d+1) + \log(d)][\log(d+1) - \log(d)] + 1 + \frac{1}{d}\log(d+1)^2 \\
&\le 2\log(d+1)\log(1+\frac{1}{d}) + 1 + \frac{1}{d}\log(d+1)^2 \\
&\le 2 * d * \frac{1}{d} + 1 + \frac{1}{d}\log(d+1)^2 \\
&\le 3 + 2 = 5
\end{aligned}
$$

The last inequality is because $\max_{d>0} \frac{1}{d}\log(d+1)^2 = 2$.

$$
\begin{aligned}
V_d &\ge \frac{1}{d}A_l - (\frac{1}{d}B_u)^2 \\
&\ge \log(d)^2 - 2\log(d) + 2 - \frac{2}{d} - \frac{1}{d^2}[(d+1)\log(d+1) - (d+1) - (2\log(2) - 2)]^2 \\
&\ge \log(2)^2 - 2\log(2) + 2 - 1 - \frac{1}{4}[3\log 3 - 3 - 2\log 2 + 2]^2 > 0
\end{aligned}
$$

when $d \ge 2$ because the function is an increasing function of d when $d \ge 1$.

A.4. Proof of Lemma 4.1

Consider any $i < j \in [d]$. Let $\sigma(\theta^{(i,j)})$ be the vector where the i-th and j-th entries are $\frac{1}{2}$ and every other entry is 0. For a diagonal matrix $\mathbf{A} = \mathrm{diag}(\alpha_1, \cdots, \alpha_d)$ to dominate $\nabla^2 f(\theta^{(i,j)}) = \mathrm{diag}(\sigma(\theta^{(i,j)})) - \sigma(\theta^{(i,j)})^\top \sigma(\theta^{(i,j)})$, it must hold that

$$\alpha_i x_i^2 + \alpha_j x_j^2 \ge \frac{1}{4}x_i^2 - \frac{1}{2}x_i x_j + \frac{1}{4}x_j^2$$

for any $x_i, x_j \in \mathbb{R}$. It is equivalent to $(\alpha_i - 1/4)(\alpha_j - 1/4) \ge 1/16$. For any α_i and α_j satisfying this constraint, it always hold that

$$\alpha_i + \alpha_j = \frac{1}{2} + (\alpha_i - 1/4) + (\alpha_j - 1/4) \ge \frac{1}{2} + 2\sqrt{(\alpha_i - 1/4)(\alpha_j - 1/4)} \ge \frac{1}{2} + \frac{1}{2} = 1.$$

Furthermore, there is a lower bound of $\mathrm{Tr}(\mathbf{A})$ as following

$$\mathrm{Tr}(\mathbf{A}) = \sum_{i=1}^{d} \alpha_i = \frac{1}{d-1}\sum_{i<j}(\alpha_i + \alpha_j) \ge \frac{1}{d-1}\frac{d(d-1)}{2} = d/2$$

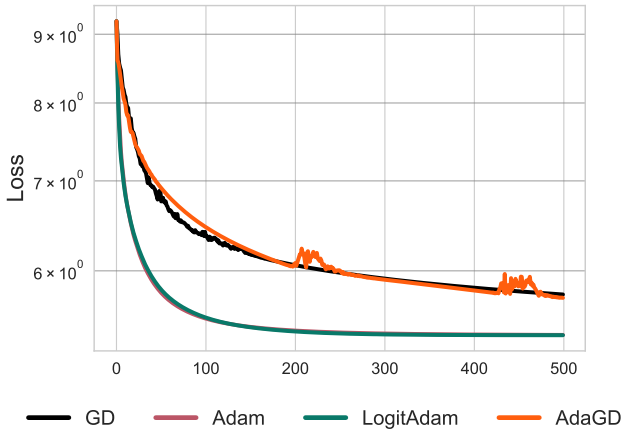


Figure 3: Rotationally invariant methods such as GD and AdaGD struggle to optimize a softmax linear bigram model under heavy-tail class imbalance. On the other hand, coordinate-wise adaptive methods such as Adam and Logit Adam optimize much faster.

Appendix B. Softmax Linear Model

In this section, we take the first steps to extend our theoretical investigation of the softmax unigram model to more complex setups. Here, we consider learning a softmax linear bigram model under heavy-tailed class imbalance. Suppose we have a dataset $\{(x_i, y_i)\}_{i=1}^n$ with $y_i \in [c]$ and $x_i \in \mathbb{R}^{d_e}$ where c is the number of classes and d_e is the embedding dimension. Let w_k be the k-th row of the weight matrix $\mathbf{W}$ and $\sigma(\cdot)_k$ be the k-th element of the softmax output. Our objective is to minimize the cross entropy loss $f : \mathbb{R}^{c \times d_e} \to \mathbb{R}$,

$$f(\mathbf{W}) = -\frac{1}{n} \sum_{i=1}^{n} \log \sigma(\mathbf{W} x_i)_{y_i}, \tag{4}$$

In Figure 3, we optimize f on the PennTreeBank dataset. Note that each token in the dataset is mapped to a fixed random vector sampled from a d_e-dimensional standard normal distribution. The dataset has a vocabulary size of $c = 10^3$ and $d_e = 256$. We find that coordinate-wise adaptive algorithms, such as Adam and Logit Adam, outperform rotationally invariant methods, GD and AdaGD. Logit Adam *i.e.* "Logit-wise" Adam, is a variant of blockwise Adam designed for optimizing the softmax linear model. We partition the weight matrix by the rows and keep track of the EMA of the ℓ_2 norms of the row vectors i.e. $v_{t+1,k} = \beta_2 v_{t,k} + (1-\beta_2)\|\nabla_{w_k} f(\mathbf{W})\|_2^2$ for $k \in [c]$. Interestingly, Logit Adam performs almost identically to Adam while only keeping track of c learning rates instead of $c \times d_e$ learning rates.

Now that we have verified that adaptive coordinate-wise algorithms such as Adam can outperform GD on a linear bigram model with heavy-tail class imbalance, we follow the steps done for the softmax unigram model theory. Analagous to Lemma 3.1, Lemma B.1 provides bounds on the $L_{\|\cdot\|_\infty}(f)$ and $L_{\|\cdot\|_2}(f)$ smoothness constant of f. Specifically, with Lemma B.1, we can show that $L_{\|\cdot\|_\infty}(f) \ll c d_e L_{\|\cdot\|_2}(f)$ assuming that $c \gg d_e$. This is a reasonable assumption to make for the softmax linear bigram model, as vocabulary sizes in language modelling are quite large. Given that $L_{\|\cdot\|_\infty}(f) \ll c d_e L_{\|\cdot\|_2}(f)$, we can expect sign-based steepest descent methods to outperform GD. We provide the statement and proof of Lemma B.1 below.

Lemma B.1 *Let f be in Eq. 4. Let $\sigma_i := \sigma(\mathbf{W}x_i)$, $\sigma_i^k := \sigma(\mathbf{W}x_i)_k$, $\Lambda = \frac{1}{n}\sum_{i=1}^n \|x_i\|_2^2$, and $\otimes$ denote the Kronecker product. Then,*

1. *the hessian $\nabla^2 f(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \left[(\mathrm{diag}(\sigma_i) - \sigma_i\sigma_i^T) \otimes x_i x_i^T\right]$.*
2. *we have $\frac{\Lambda}{4d_e} \leq L_{\|\cdot\|_2}(f) \leq \Lambda$ and $L_{\|\cdot\|_\infty}(f) \leq d_e\Lambda$*

Proof Firstly, we can express f as,

$$f(\mathbf{W}) = -\frac{1}{n}\sum_{i=1}^n \left[w_{y_i}^\top x_i - \log\left(\sum_{j=1}^c e^{w_j^\top x_i}\right)\right].$$

Then we compute the derivative w.r.t to w_k,

$$\nabla_{w_k} f(\mathbf{W}) = -\frac{1}{n}\sum_{i=1}^n \mathbb{1}[y_i = k]x_i + \frac{1}{n}\sum_{i=1}^n \sigma_i^k x_i.$$

Then, we can compute the Hessian of f. Specifically,

$$\nabla^2_{w_k} f(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \sigma_i^k(1-\sigma_i^k)x_i x_i^\top,$$

When $k \neq j$, we have that,

$$\nabla_{w_k}\nabla_{w_j} f(\mathbf{W}) = \sum_{i=1}^n -\sigma_i^k\sigma_i^j x_i x_i^\top.$$

If we look at the structure of the Hessian for each example, we see that it resembles the Hessian of the unigram softmax model. In fact, each example we see that the Hessian is $(\mathrm{diag}(\sigma_i) - \sigma_i^\top\sigma_i) \otimes x_i x_i^\top$. Therefore, we get,

$$\nabla^2 f(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \left[(\mathrm{diag}(\sigma_i) - \sigma_i\sigma_i^T) \otimes x_i x_i^T\right]$$

Next, we compute the upper bounds on the Hessian. Since,

$$\nabla^2 f(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \mathrm{diag}(\sigma_i)\otimes x_i x_i^\top - \frac{1}{n}\sum_{i=1}^n \sigma_i\sigma_i^T \otimes x_i x_i^T \implies \nabla^2 f(\mathbf{W}) \preceq \frac{1}{n}\sum_{i=1}^n (\mathrm{diag}(\sigma_i)\otimes x_i x_i^T)$$

We will compute an upper bound for the following quantity, $\sup_{||u||_2\leq 1} u^T\nabla^2 f(\mathbf{W})u$ that is independent of $\mathbf{W}$, which will give us an upper bound on $L_{\|\cdot\|_2}(f)$. We partition $u \in \mathbb{R}^{cd_e}$ into c blocks:

$u = [u^1 \quad u^2 \quad \cdots \quad u^k]$. Then for $||u||_2 \leq 1$,

$$\begin{aligned}
u^T H(W) u &\leq \frac{1}{n} \sum_{i=1}^{n} u^T (\text{diag}(\sigma_i) \otimes x_i x_i^T) u^T \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k (u^k)^T (x_i x_i^T) u^k \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k |(u^k)^T x_i|^2 \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k ||u^k||_2^2 ||x_i||_2^2 \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k ||x_i||_2^2 = \frac{1}{n} \sum_{i=1}^{n} ||x_i||_2^2 \left(\sum_{k=1}^{c} \sigma_i^k \right) = \frac{1}{n} \sum_{i=1}^{n} ||x_i||_2^2
\end{aligned}$$

We use the fact that $||u||_2 \leq 1$ implies that for each block $||u^k||_2 \leq 1$. We also note that for every $i \in [n]$ we know that $\sum_{k=1}^{c} \sigma_i^k = 1$ by a basic property of softmax.

To compute the upper bound on $L_{\|\cdot\|_\infty}(f)$, we apply the same reasoning as above but use the generalized Cauchy-Schwarz inequality,

$$\begin{aligned}
u^T H(W) u &\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k |(u^k)^T x_i|^2 \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k ||u||_\infty^2 ||x_i||_1^2 \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} \sigma_i^k ||x_i||_1^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} ||x_i||_1^2 \sum_{k=1}^{c} \sigma_i^k \\
&= \frac{1}{n} \sum_{i=1}^{n} ||x_i||_1^2 \leq \frac{d}{n} \sum_{i=1}^{n} ||x_i||_2^2
\end{aligned}$$

We use the fact that $||u||_\infty \leq 1$ implies that for each block $||u^k||_\infty \leq 1$. For every $i \in [n]$ we know that $\sum_{k=1}^{c} \sigma_i^k = 1$. In the last step, we employ the following fact about the ℓ_1 and ℓ_2 norms: $||x||_2 \leq ||x||_1 \leq \sqrt{d}||x||_2$.

Now, we compute the lower bound for $L_{\|\cdot\|_2}(f)$. Select a $q \in \mathbb{R}^{d_e}$ such that $q^T x_i \neq 0$ for all i with $||q||_2 = 1$. WLOG, we can assume that,

$$\sum_{i : q^T x_i > 0} ||x_i||_2^2 \geq \frac{1}{2} \sum_{i=1}^{n} ||x_i||_2^2.$$

If the above is not true, we can let $q := -q$, and we know that one side must have at least half the mass.

Let $\mathbf{W}(t) : \mathbb{R} \to \mathbb{R}^{c \times d_e}$ define the following weight matrices: $w_1 = w_2 = tq$ and $w_k = 0$ for $k > 2$.

Step 1: Compute a lower bound on the operator norm $||\nabla^2 f(W(t))||_2$ for fixed t. Let $r \in \mathbb{R}^d$ be a unit vector. Define $v \in \mathbb{R}^{cd}$ with $||v||_2 = 1$ as follows:

$$v = [\frac{1}{\sqrt{2}} r^T \quad -\frac{1}{\sqrt{2}} r^T \quad 0 \quad \cdots \quad 0]^T.$$

Then, of course, $||\nabla^2 f(W(t))||_2 \geq v^T \nabla^2 f(W(t)) v$. Now, let's compute this quadratic form. We can exploit the fact that the Hessian has a "block" structure composed of $c \times c$ blocks. The first block of v is $\frac{1}{\sqrt{2}} r$, the second block is $-\frac{1}{\sqrt{2}} r$ and the remaining $c - 2$ blocks are $0 \in \mathbb{R}^{d_e}$. Therefore, we only have to work with the 2×2 upper-left block sub-matrix to compute the quadratic form. For this $W(t)$, we have that for all $i \in [n]$,

$$\sigma_i^1 = \sigma_i^2 = \frac{\exp(tq^T x_i)}{c - 2 + 2\exp(tq^T x_i)}.$$

For clarity, we define $s_i := \sigma_i^1$. Recall that for computing the quadratic form, we do not need the values of σ_i^k for $k > 2$ because only the upper-left 2×2 blocks will matter in the computation. Evaluating the block computation,

$$v^T \nabla^2 f(W(t)) v = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} [s_i(1 - s_i) + s_i(1 - s_i) + 2s_i^2] r^T x_i x_i^T r = r^T \left[\frac{1}{n} \sum_{i=1}^n s_i x_i x_i^T \right] r.$$

Therefore, $||\nabla^2 f(W(t))||_2 \geq r^T \left[\frac{1}{n} \sum_{i=1}^n s_i x_i x_i^T \right] r$. Take $\sup_{||r||_2 = 1}$ of the right-hand side to get that,

$$\left\| \nabla^2 f(W(t)) \right\|_2 \geq \left\| \left[\frac{1}{n} \sum_{i=1}^n s_i x_i x_i^T \right] \right\|_2.$$

Now use the fact that $\lambda_{\max}(\mathbf{M}) \geq \frac{\mathrm{Tr}(\mathbf{M})}{d}$ for any matrix $\mathbf{M}$. Therefore,

$$||\nabla^2 f(W(t))||_2 \geq \frac{1}{d_e n} \sum_{i=1}^n s_i ||x_i||_2^2$$

Step 2: Take $t \to \infty$. Observe that s_i is really a function of t. In order to emphasize this we consider $\lim_{t \to \infty} s_i(t)$. Recall that we specifically chose q such that $q^T x_i \neq 0$ for all $i \in [n]$. Therefore, we only need to consider the case where $q^T x_i > 0$ or $q^T x_i < 0$. It is not too hard to see that,

$$\lim_{t \to \infty} s_i(t) = \begin{cases} \frac{1}{2} & q^T x_i > 0 \\ 0 & q^T x_i < 0 \end{cases}$$

Therefore,

$$\begin{aligned}\lim_{t\to\infty}\frac{1}{d_e n}\sum_{i=1}^{n} s_t||x_i||_2^2 &= \lim_{t\to\infty}\frac{1}{d_e n}\sum_{i:q^T x_i>0} s_t||x_i||_2^2 + \lim_{t\to\infty}\frac{1}{d_e n}\sum_{i:q^T x_i<0} s_t||x_i||_2^2 \\ &= \frac{1}{d_e n}\sum_{i:q^T x_i>0}\frac{1}{2}||x_i||_2^2 \\ &= \frac{1}{2d_e n}\sum_{i:q^T x_i>0}||x_i||_2^2 \\ &\geq \frac{1}{4d_e n}\sum_{i=1}^{n}||x_i||_2^2.\end{aligned}$$

Step 3: Chain inequalities. Realize that,

$$L_{\|\cdot\|_2}(f) = \sup_{W\in\mathbb{R}^{c\times d}}||\nabla^2 f(W)||_2 \geq \sup_t ||\nabla^2 f(W(t))||_2 \geq \sup_t \frac{1}{d_e n}\sum_{i=1}^{n} s_i(t)||x_i||_2^2.$$

Then,

$$\sup_t \frac{1}{d_e n}\sum_{i=1}^{n} s_i(t)||x_i||_2^2 \geq \lim_{t\to\infty}\frac{1}{d_e n}\sum_{i=1}^{n} s(t)||x_i||_2^2 \geq \frac{1}{4d_e n}\sum_{i=1}^{n}||x_i||_2^2.$$

Therefore,

$$L_{\|\cdot\|_2}(f) \geq \frac{1}{4d_e n}\sum_{i=1}^{n}||x_i||_2^2.$$

■

Appendix C. Additive Logistic Transformation Unigram Model

In this section, we consider a slightly modified version of the problem presented in Equation (1) that has a unique solution. Specifically, $\tilde{f}:\mathbb{R}^{d-1}\to\mathbb{R}^d$,

$$\tilde{f}(\theta) = \mathrm{KL}(p \parallel \tilde{\sigma}(\theta)), \tag{5}$$

where $\tilde{\sigma}$ is the additive logistic transformation. The additive logistic transformation is equivalent to computing the softmax over $[\theta_1\ \theta_2\ \cdots\ 0]$, so it has a unique inverse. This is beneficial because it allows us to simplify the analysis of $\tilde{f}$. We first start by providing lower bounds on the smoothness of $\tilde{f}$.

Lemma C.1 *Consider $\tilde{f}$ in Equation (5). Then,*

$$\frac{1}{2} \leq L_{\|\cdot\|_2}(\tilde{f}) \leq 1, \text{ and } \quad L_{\|\cdot\|_\infty}(\tilde{f}) = 1.$$

In particular, the smoothness constants of f in Equation (1) and $\tilde{f}$ are the same.

Proof Expand $\tilde{f}$ as,

$$\tilde{f}(\theta) = \sum_{k=1}^{d-1} \left[-p_k \theta_k + p_k \log\left(1 + \sum_{j=1}^{d-1} e^{\theta_j}\right)\right] + p_d \log\left(1 + \sum_{j=1}^{d-1} e^{\theta_j}\right).$$

Then,

$$\frac{\partial \tilde{f}}{\partial \theta_i} = -p_i + \sum_{k=1}^{d-1} p_k \tilde{\sigma}(\theta)_i + p_d \tilde{\sigma}(\theta)_i = \tilde{\sigma}(\theta)_i - p_i.$$

Therefore, $\nabla \tilde{f}(\theta) = \tilde{\sigma}(\theta)_{1:d-1} - p_{1:d-1}$. Therefore, the Hessian of $\tilde{f}$ is the Jacobian of $\tilde{\sigma}$, which is just the $d-1 \times d-1$ sublock of the Hessian of f. So,

$$\nabla^2 \tilde{f}(\theta) = \operatorname{diag}(\tilde{\sigma}(\theta)_{1:d-1}) - \tilde{\sigma}(\theta)_{1:d-1} \tilde{\sigma}(\theta)_{1:d-1}^\top.$$

To compute the smoothness constants of $\tilde{f}$, we can apply the same reasoning we did to compute the upper and lower bounds for the smoothness constants of f. ■

In the next lemma, we provide the norm of the optimal solution of $\tilde{f}$.

Lemma C.2 *For the optimal set of $\theta_\star$ of $\tilde{f}$, we have that,*

$$\min_{\theta_\star \in \arg\min \tilde{f}} \|\theta_\star\|_\infty = \log(d), \text{ and } \quad \min_{\theta_\star \in \arg\min f} \|\theta_\star\|_2 = \sqrt{\sum_{k=1}^{d-1} \log\left(\frac{k}{d}\right)^2}.$$

Proof Suppose $p \in \mathbb{R}^d$ is a probability vector *i.e.* $\sum_i p_i = 1$. Then the inverse of the additive logistic transformation is given by,

$$\tilde{\sigma}(p)^{-1} = \left[\log\left(\frac{p_1}{p_d}\right) \ \log\left(\frac{p_2}{p_d}\right) \ \cdots \ \log\left(\frac{p_{d-1}}{p_d}\right)\right]$$

The minimizer of $\tilde{f}$ must satisfy $\tilde{\sigma}(\theta_\star) = p$. Thus, $\tilde{f}$ has a unique minimizer. Since $p_k = \frac{k^{-1}}{\sum_i^d i^{-1}}$,

$$\theta_\star = \left[\log\left(\frac{1}{d}\right) \ \log\left(\frac{2}{d}\right) \ \cdots \ \log\left(\frac{d-1}{d}\right)\right].$$

Then, it's not too hard to see that $\|\theta_\star\|_\infty = \log(d)$ and $\|\theta_\star\|_2 = \sqrt{\sum_{k=1}^{d-1} \log\left(\frac{k}{d}\right)^2}$. ■

The following theorem characterizes the complexity of $\tilde{f}$ under ℓ_2 and ℓ_∞ norms.

Theorem C.1 *For large d, we have that,*

$$\mathcal{C}_{\|\cdot\|_\infty}\left(\tilde{f}\right) = 8\log(d)^2 \ll 4\sum_{k=1}^{d-1} \log\left(\frac{k}{d}\right)^2 \leq \mathcal{C}_{\|\cdot\|_2}\left(\tilde{f}\right).$$

Proof Use the fact that $L_{\|\cdot\|_\infty}(\tilde{f}) = 1$ and $\frac{1}{2} \leq L_{\|\cdot\|_2}(\tilde{f})$ together with the norms computed in Lemma C.2. ■

Now that we know the complexity of $\tilde{f}$, we can compare the upper bound on the convergence rate for sign descent with weight decay and normalized GD with weight decay.

Corollary C.1 *Consider optimizing $\tilde{f}$ in Equation (5) with large d and initialized at $\theta_0 = 0$. Then the iterates of Sign descent with weight decay $\lambda_\infty = \frac{1}{\log(d)}$ and learning rate $\eta_t = \frac{1}{\lambda_\infty(t+1)}$ satify,*

$$f(x_T) - f^\star \leq \frac{8\log(d)^2}{T+1}.$$

Furthermore, the iterates of normalized GD with weight decay $\lambda_2 = \frac{1}{\sqrt{\sum_{k=1}^{d-1}\log\left(\frac{k}{d}\right)^2}}$ and learning rate $\eta_t = \frac{1}{\lambda_2(t+1)}$ satisfy,

$$f(x_T) - f^\star \leq \frac{4}{T+1}\sum_{k=1}^{d-1}\log\left(\frac{k}{d}\right)^2.$$

Our theoretical investigation for $\tilde{f}$ suggests that sign descent with weight decay optimizes much faster normalized GD with weight decay. In Figure 4 we very the results of Corollary C.1.

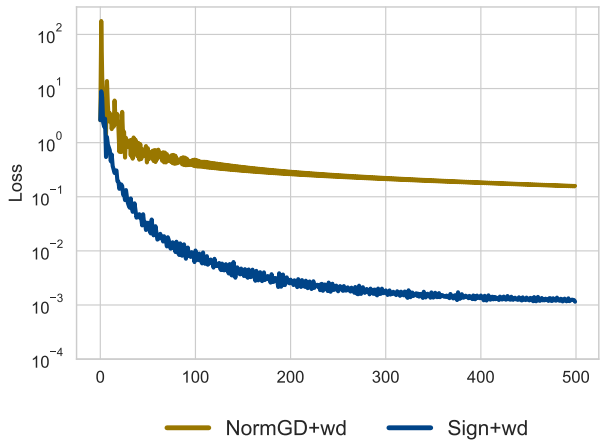


Figure 4: Performance of NSD with weight decay when minimizing f with $d = 10^3$. For each optimizer, we set $\lambda = \frac{1}{\min_{\theta_\star \in \arg\min f}\|\theta_\star\|}$ and use a learning rate of $\eta_t = \frac{2}{\lambda(t+1)}$.