OPT2023: 15th Annual Workshop on Optimization for Machine Learning

# Adagrad Promotes Diffuse Solutions In Overparameterized Regimes

**Andrew Rambidis** ANDREW.RAMBIDIS@MAIL.MCGILL.CA
**Jiayi Wang** KELSEY.WANG@MAIL.MCGILL.CA
*McGill University*

## Abstract

With the high use of over-parameterized data in deep learning, the choice of optimizer in training plays a big role in a model's generalization ability due to solution selection bias. This work focuses on the adaptive gradient optimizer Adagrad, in the over-parameterized least-squares regime. We empirically find that when using sufficiently small step sizes, Adagrad promotes diffuse solutions in the sense of uniformity among the coordinates of the solution. Additionally, we theoretically show that Adagrad's solution, under the same conditions, exhibits greater diffusion compared to the solution obtained through gradient descent (GD) by analyzing the ratio of their updates. Lastly, we empirically compare the performance of Adagrad and GD on generated datasets. We observe a consistent trend that Adagrad promotes more diffuse solutions, which aligns with our theoretical analysis.

## 1. Introduction

Many modern applications of deep learning rely on over-parameterized modelling [1, 2, 16, 22]. In such settings, the model has more features than observations to fit the training data on. Under this regime, the space of possible solutions which minimize the loss function is very large. Importantly enough, the choice of optimal solution has an effect on the generalization performance of the trained model [9]. The optimal solution an optimization algorithm converges to is not equal across algorithms [20]. Instead, different classes of optimization algorithm will have their own implicit biases towards particular solutions depending on the regime [5, 18]. For small-batch methods such as stochastic gradient descent (SGD), a plethora of theoretical analyses on its behaviours exist, and has been shown to promote solutions which generalize well [13, 24].

In this paper, we want to continue the idea of studying the algorithmic behaviours of optimizers; specifically the class of optimizers known as adaptive-methods [6, 11, 14, 21]. Particularly, we focus on the diagonalized **Ada**ptive **Grad**ient Algorithm (Adagrad) [6], under the over-parameterized least-squares problem:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2} \|Ax - b\|_2^2 \tag{1}$$

where $x \in \mathbb{R}^d$, $A \in \mathbb{R}^{n \times d}$ such that $d > n$ (i.e. over-parameterized), and $b \in \mathbb{R}^n$. Under this regime, we aim to study the type of solutions produced by Adagrad. This can help us gain a better understanding on the 'why' of the strong generalization performance, and popularity of adaptive methods [4, 15]. In recent years, the research focus has changed towards exploring the implicit bias of Adagrad. Wang et al. [19] provided a detailed theoretical analysis on Adagrad's convergence direction. Furthermore, it has also been empirically demonstrated and theoretically proven that the

solutions produced by Adagrad are influenced by various hyperparameters such as initialization and step-size [8, 17]. These works support our analysis on the diffuse pattern of Adagrad's solutions.

While the regime of deep learning warrants the use of more complex neural network (NN) structures, it has been shown that studying under regime (1) is not arbitrary due to connections between the linear model and neural networks [10, 12]. That is, studying the over-parameterized linear regime can allow one to perform more simple and interpretable analysis of algorithmic behaviours which still have relevant connection to NNs.

Our contributions are: **1)** we empirically show evidence that, when applying Adagrad with sufficiently small step-size under regime (1), Adagrad promotes solutions which are diffuse in the sense of uniformity among the entries of the solution (the idea of diffusivity is formalized in Section 2); **2)** we theoretically show that, under the same assumptions as contribution 1), Adagrad promotes more diffuse solutions than that of gradient descent (GD), and support this analysis with empirical evidence.

## 2. Background

In this section, we briefly recall the diagonal Adagrad update rule, which we will simply refer to as "Adagrad", then discuss the preliminary background surrounding the idea of diffusivity of a vector.

**Adagrad** Let $\mathbf{g}_k \in \mathbb{R}^d$ denote the gradient of the least-squares loss function at iteration $k$, i.e. $\mathbf{g}_k := \nabla \ell(x_k) = \nabla \frac{1}{2}\|Ax_k - b\|_2^2$. Under this notation, the Adagrad update rule for problem regime (1) is:

$$x_{k+1} = x_k - \eta G_k \cdot \mathbf{g}_k \tag{2}$$

where $x_k \in \mathbb{R}^d$, $\eta \in \mathbb{R}$ is a constant step-size, and $G_k \in \mathbb{R}^{d\times d}$ is the Adagrad preconditioner matrix with respect to problem (1) and explicitly presented as:

$$G_k = Diag\left(\sum_{\tau=1}^{k} \mathbf{g}_\tau \mathbf{g}_\tau^T + \epsilon\right)^{-1/2} \tag{3}$$

where $\epsilon$ is a small perturbation parameter to prevent division by zero. A brief intuition behind this preconditioner matrix $G_k$ is that it dynamically applies feature-specific learning rates to each feature based on how sparse that feature is. Specifically, the Adagrad preconditioner will assign higher learning rates towards sparse features, and lower learning rates towards non-sparse features.

**Diffusion** The following definition presents our method of measuring how diffuse a vector is:

**Definition 1 (Local Sparsity Metric)** *Given a non-zero vector $x \in \mathbb{R}^d \setminus \{0\}$, we define the local sparsity metric (LSM) to be the measure of a vector's relative sparsity via the following metric: $LSM : \mathbb{R}^d \setminus \{0\} \to \mathbb{R}$ where*

$$LSM(x) := \frac{\|x\|_2}{\|x\|_1}. \tag{4}$$

This metric and the theory behind the relationship between the $\ell_2$ and $\ell_1$ norms comes from the theory of compressive sensing in which one's main goal is to find the sparsest solution to an over-parameterized system of linear equations. The work on their connection is covered by a Russian

paper written by Garnaev and Gluskin [7] and summarized in English by Zhang in his technical report [23]. With respect to diffuse vectors, we say that a lower value of the LSM implies a more diffuse vector; the higher it is, the more locally sparse our vector is. To better clarify Definition 1, we present the following example: Consider the two vectors

$$\begin{aligned} w_1 &= (0.25, 0.25, 0.15, 0.35)^T \\ w_2 &= (0.90, 0.02, 0.02, 0.06)^T. \end{aligned} \tag{5}$$

the LSM states that $w_1$ is more diffuse than $w_2$, i.e. $LSM(w_1) < LSM(w_2)$. To see this, notice that the entries of $w_1$ are relatively close together in magnitude, that is, the "total weight" of the entries is diffuse pretty evenly among the entries, while $w_2$ has a strong localization around the first entry. The following proposition presents us with bounds for the LSM:

**Proposition 2** *For any non-zero vector $x \in \mathbb{R}^d \setminus \{0\}$, its LSM is bounded as follows:*

$$\frac{1}{\sqrt{d}} \leq LSM(x) \leq 1. \tag{6}$$

As was mentioned above, the closer the LSM of a vector is to the lower bound, the more diffuse said vector is, while the closer it is to the upper bound, the more localized. In fact, the bounds are attainable. To see this consider the vectors $x_{low} = (\pm a, \ldots, \pm a)^T \in \mathbb{R}^d$, and $x_{high} = (\pm a, 0, \ldots, 0)^T \in \mathbb{R}^d$, where $a \in \mathbb{R} \setminus \{0\}$. Then,

$$\begin{aligned} LSM(x_{low}) &= \frac{\|x_{low}\|_2}{\|x_{low}\|_1} = \frac{\sqrt{a^2 + \cdots + a^2}}{|a| + \cdots + |a|} = \frac{\sqrt{d}|a|}{d|a|} = \frac{\sqrt{d}}{d} = \frac{1}{\sqrt{d}}, \\ LSM(x_{high}) &= \frac{\|x_{high}\|_2}{\|x_{high}\|_1} = \frac{\sqrt{a^2}}{|a|} = 1. \end{aligned} \tag{7}$$

## 3. Analysis of the Uniformity Metric on Adagrad and Gradient Descent

In this section, we perform comparative analysis of the LSM value produced by the Adagrad solution to that of the solution generated by GD under the same setting. We show theoretically that, under the over-parameterized least-squares regime, and for small enough step-size, Adagrad promotes LSM values closer to the lower bound than GD. This result supports the experimental comparisons we provide in Section 4.

**Theorem 3** *Let $A \in \mathbb{R}^{n \times d}$ such that $n < d$ and let $A(:, i) \in \mathbb{R}^n$ denote the i-th column of $A$ for $i = 1, \ldots, d$. Let $x_k^{ada}, x_k^{gd} \in \mathbb{R}^d$ denote the $k$-th iterate generated by Adagrad and gradient descent, respectively, and $b \in \mathbb{R}^n$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be the least squares function*

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2. \tag{8}$$

*Finally, consider the magnitudes of both the Adagrad and gradient descent updates at the i-th coordinate:*

$$|\Delta x_k^{ada}(i)| = \eta \cdot \left| \frac{1}{\sqrt{\sum_{\tau=0}^{k} (\mathbf{g}_\tau^{ada}(i))^2 + \epsilon}} \cdot \mathbf{g}_k^{ada}(i) \right| \tag{9}$$

*and*

$$|\Delta x_k^{gd}(i)| = \eta \cdot \left|\mathbf{g}_k^{gd}(i)\right| \tag{10}$$

*respectively, where both algorithms start at* $x_0 = \mathbf{0}$. *For some* $\delta > 0$, *assume that there exist coordinates,* $i$, *such that*

$$|A(:,i)^T b| \geq \delta \|A(:,i)\|. \tag{11}$$

*That is, we consider the coordinates in which* $|A(:,i)^T b|$ *are at least a constant factor greater than the norm of the column* $\|A(:,i)\|$. *Then, for* $\eta > 0$ *chosen small enough such that, if* $\eta_1 > 0$ *satisfies* $\|I - \eta_1 A^T A\|_2 \leq 1$, *and* $\eta_2 > 0$ *satisfies* $\|I - \eta_2 G_0 A^T A\|_2 \leq 1$, *we set* $\eta \leq \min\{\eta_1, \eta_2\}$, *where* $G_0$ *is the Adagrad preconditioner matrix at iteration* $k = 0$. *For iterations* $0 < k \leq \min\left\{\left(\frac{\delta}{2\eta\|A\|\cdot\|A^T b\|}\right), \left(\frac{\delta}{2\eta\|A\|\cdot\|G_0\|\cdot\|A^T b\|}\right)\right\}$ *the following bounds hold*

$$\left|\frac{3}{\sqrt{k}\cdot\sqrt{(\frac{3}{2}(A^T b)_i)^2 + \epsilon k^{-1/2}}}\right| \leq \left|\frac{\Delta x_k^{ada}(i)}{\Delta x_k^{gd}(i)}\right| \leq \left|\frac{3}{\sqrt{k}\cdot\sqrt{(\frac{1}{2}(A^T b)_i)^2 + \epsilon k^{-1/2}}}\right| \tag{12}$$

*for* $i$ *satisfying our assumption, and where* $(A^T b)_i$ *is the i-th coordinate of the vector* $A^T b$.

Theorem 3 states that, for feature columns $A(:,i)$ satisfying our assumptions, the ratio between the updates $\Delta x_k^{ada}$ and $\Delta x_k^{gd}$, at said feature columns, varies inversely to the magnitude of the feature. Intuitively, this means that for features that show up less frequently in the data, Adagrad will promote a larger step for said feature in the update compared to GD. On the other hand, for frequently appearing features, GD will promote a larger step for said feature when compared to Adagrad. This suggests that the solution attained by Adagrad will have entries that are more diffuse (or less locally sparse) when compared to gradient descent. Adagrad achieves this by reducing influence of the weights of frequent features, and simultaneously pushing the weight of infrequent features closer to those of the frequent ones. The proof of Theorem 3 can be found in Appendix A.

## 4. Experiments

We now cover our experimental findings in which we observe that Adagrad promotes diffuse solutions for small enough step-size on over-parameterized least-squares.

We present two tables: 1, and 2. Table 1 shows the LSM of the minimum 2-norm, Adagrad, and stochastic Adagrad (batch size = 1) solutions of over-parameterized least-squares, at the initial point $x_0 = 0$, for varying step-sizes in the range 0.01 to 0.00001. The LSM values present are taken to be the average over 5 runs in which for a particular run $i$, a data matrix $A_i$ and solution vector $b_i$ are randomly generated via a standard normal distribution with $A_i \in \mathbb{R}^{50\times 50'000}$ and $b_i \in \mathbb{R}^{50}$, $\forall i \in [5]$. Table 2 performs the same experiment, only now we are keeping step-size fixed to $\eta = 0.001$, and varying the dimension. Note that for both tables, we utilize the minimum $\ell_2$-norm solution as GD converges to the same solution in over-parameterized least-squares when $x_0 = 0$.

We see in Table 1 that as we decrease the step-size from 0.01 to 0.00001, the LSM of Adagrad decreases. We note that for higher values of $\eta$, the LSM of Adagrad is larger than that of GD's solution, then, after passing below a certain step-size threshold, Adagrad's LSM decreases past

| Learning rate | Lower bound | 2-Norm | Adagrad | S-Adagrad | Upper bound |
|---|---|---|---|---|---|
| $\eta = 0.01$ | 0.00447 | **0.00561** | 0.01703 | 0.00569 | 1 |
| $\eta = 0.005$ | 0.00447 | **0.00561** | 0.01326 | 0.00576 | 1 |
| $\eta = 0.001$ | 0.00447 | **0.00561** | 0.00583 | 0.00578 | 1 |
| $\eta = 0.0005$ | 0.00447 | 0.00560 | **0.00477** | 0.00595 | 1 |
| $\eta = 0.0001$ | 0.00447 | 0.00561 | **0.00449** | 0.00565 | 1 |
| $\eta = 0.00001$ | 0.00447 | 0.00561 | **0.00448** | 0.00554 | 1 |

Table 1: LSM for different learning rates with dataset size $n = 50, d = 50'000$. For each learning rate, the lowest LSM besides the lower bound is marked bold.

| Dataset | Lower bound | 2-Norm | Adagrad | S-Adagrad | Upper bound |
|---|---|---|---|---|---|
| $n = 50, d = 1000$ | 0.03162 | 0.03960 | **0.03231** | 0.03920 | 1 |
| $n = 50, d = 10000$ | 0.01000 | 0.01253 | **0.01010** | 0.01285 | 1 |
| $n = 50, d = 50000$ | 0.00447 | **0.00561** | 0.00583 | 0.00645 | 1 |
| $n = 50, d = 100000$ | 0.00316 | **0.00396** | 0.00507 | 0.00415 | 1 |

Table 2: LSM for different size of dataset with learning rate $\eta = 0.001$. For each size of dataset (each row), the lowest LSM besides the lower bound is marked bold.

that of GD's and stabilizes to a value close to the LSM's lower bound. This observation agrees with Theorem 3 as we expect **Adagrad to promote directions towards sparser features**, while dampening directions towards frequent ones for step-sizes that satisfy the bound. Table 1 suggests that, past such a bound, not only can we not guarantee a lower LSM for Adagrad (vs. GD), but it seems to increase towards the upper bound of the LSM. Such a behaviour indicates that the step-size hyperparameter acts more so as a control parameter which allows one to control for desired diffusivity.

Furthermore, Table 2 shows us that **increasing the dimension further also affects the LSM**. Namely, if we fix $\eta$ and increase the dimension $d$, the distance between the LSM of Adagrad's solution and the lower bound increases.

Note that stochastic Adagrad does not seem to exhibit this behaviour. In fact, stochastic Adagrad seems to select a solution in which the LSM is similar to that of GD.

Additionally, we expand the analysis to discuss if there are any advantages to diffuse solutions. We provide an example case of interpolating a line in Appendix C in which a diffuse solution outperforms locally sparse ones.

## 5. Conclusion and Future Work

In this work, we proposed the LS metric, and provided experimental evidence showing that smaller choices of step size for Adagrad, in overparameterized linear regression, yield solutions with diffuse coordinates. Furthermore, we presented comparative analysis and experiments comparing the solution diffusivity of Adagrad to gradient descent and found that Adagrad promotes more diffuse solutions than that of gradient descent, with respect to LSM. Additionally, the experiments of Ap-

pendix C presents more promising generalization performance of diffuse solutions when applied to radial basis function (RBF) interpolation. Such observations warrants further investigation of diffuse solutions and their benefits to other classes of problems. As such, the next steps of our research looks to analyze *why* diffuse solutions generalize better in RBF interpolation, and to further explore other classes of problems in which diffuse solutions yield better generalization.

## References

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

[2] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR, 2018.

[3] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003. doi: 10.1017/CBO9780511543241.

[4] Zaiyi Chen, Yi Xu, Enhong Chen, and Tianbao Yang. Sadagrad: Strongly adaptive stochastic gradient methods. In *International Conference on Machine Learning*, pages 913–921. PMLR, 2018.

[5] Dami Choi, Christopher J Shallue, Zachary Nado, Jaehoon Lee, Chris J Maddison, and George E Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

[6] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 07 2011.

[7] Andrei Yur'evich Garnaev and Efim Davydovich Gluskin. The widths of a euclidean ball (russian). *Doklady Akademii Nauk SSSR*, 277(5):1048–1052, 1984.

[8] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018.

[9] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[10] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

[11] Geoffrey Hinton. Neural networks for machine learning: Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude. Lecture Notes, 2018.

[12] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[13] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

[14] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR: international conference on learning representations*, pages 1–15. ICLR US., 2015.

[15] Mingrui Liu, Youssef Mroueh, Jerret Ross, Wei Zhang, Xiaodong Cui, Payel Das, and Tianbao Yang. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. *arXiv preprint arXiv:1912.11940*, 2019.

[16] OpenAI. Gpt-4 technical report. Technical report, OpenAI, 2023.

[17] Qian Qian and Xiaoyuan Qian. The implicit bias of adagrad on separable data. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Gal Vardi. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66 (6):86–93, 2023.

[19] Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In *International Conference on Machine Learning*, pages 10849–10858. PMLR, 2021.

[20] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

[21] Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012.

[22] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[23] Yin Zhang. A simple proof for recoverability of l1-minimization: Go over or under? Technical report, Rice University, 2005.

[24] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Chu Hong Hoi, et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Advances in Neural Information Processing Systems*, 33:21285–21296, 2020.

## Appendix A. Proof of Theorem 3

**Proof** [Theorem 3] Under our hypothesis, there exists coordinates, $i$, such that we satisfy (11). Under such $i$, we note that the $i$-th coordinate of $\nabla f(x)$ is

$$\mathbf{g}(i) = (A^TAx - A^Tb)_i = (A^TAx)_i - (A^Tb)_i$$

Now, for such $i$ respecting (11), we must have for small enough iterations $k$ that the following inequality holds:

$$\left|(A^TAx_k)_i\right| \leq \frac{1}{2}\left|(A^Tb)_i\right| \tag{13}$$

for both $x_k^{ada}$ and $x_k^{gd}$. We show this holds for both gradient descent and Adagrad under our hypothesis and assumptions. First we show this holds for gradient descent which has the following update:

$$x_k = x_{k-1} - \eta A^T(Ax_{k-1} - b).$$

This can be rewritten into the following recursive form:

$$x_k = (I - \eta A^TA)x_{k-1} + \eta A^Tb \tag{14}$$

which can be expanded and written in terms of $x_0$:

$$x_k = (I - \eta A^TA)x_0 + A^Tb \cdot \sum_{i=1}^{k-1}(I - \eta A^TA)^i \tag{15}$$

$$= A^Tb \cdot \sum_{i=0}^{k-1}(I - \eta A^TA)^i \tag{16}$$

since $x_0 = 0$. Applying the 2-norm on both sides, we have

$$\|x_k\| = \eta\|A^Tb \cdot \sum_{i=0}^{k-1}(I - \eta A^TA)^i\| \tag{17}$$

$$\leq \eta\|A^Tb\| \cdot \sum_{i=0}^{k-1}\|I - \eta A^TA\|^i \tag{18}$$

$$\leq \eta\|A^Tb\| \cdot \sum_{i=0}^{k-1} 1 \tag{19}$$

$$= \eta k\|A^Tb\|. \tag{20}$$

where the inequality from (18) to (19) is due to our choice of $\eta$. Now, for $i$ satisfying our assumption, we have

$$|(A^TAx_k)_i| = |A(:,i)^T(Ax_k)| \tag{21}$$

$$\leq \|A(:,i)\| \cdot \|A\| \cdot \|x_k\| \tag{22}$$

$$\leq \frac{1}{\delta}|A(:,i)^Tb| \cdot \|A\| \cdot k\eta\|A^Tb\| \tag{23}$$

$$= k \cdot \left(\frac{\eta\|A\| \cdot \|A^Tb\|}{\delta}\right)|A(:,i)^Tb|. \tag{24}$$

By our choice of $k$ in the hypothesis, we must have that

$$|(A^TAx_k)_i| \leq k \cdot \left(\frac{\eta\|A\| \cdot \|A^Tb\|}{\delta}\right)|A(:,i)^Tb| \leq \frac{1}{2}|A(:,i)^Tb|. \tag{25}$$

Now we show the case for Adagrad. Recall the Adagrad update to be:

$$x_k = x_{k-1} - \eta G_k A^T(Ax_{k-1} - b). \tag{26}$$

Then similar to gradient descent we can express the update as a recursion:

$$x_k = (I - \eta G_k A^T A)x_{k-1} + \eta G_k A^T b \tag{27}$$

and can be expanded to be written in terms of $x_0$

$$x_k = (I - \eta G_k A^T A)x_0 + \eta G_k A^T b \cdot \sum_{i=0}^{k-1}(I - \eta G_k A^T A)^i \tag{28}$$

$$= \eta G_k A^T b \cdot \sum_{i=0}^{k-1}(I - \eta G_k A^T A)^i \tag{29}$$

where the last equality comes from the fact that $x_0 = 0$. From here, we apply the 2-norm to get the following bound on $\|x_k\|$:

$$\|x_k\| \leq \eta\|G_k A^T b\| \cdot \sum_{i=0}^{k-1}\|I - \eta G_k A^T A\|^i. \tag{30}$$

By definition of $G_k$, we must have $\|G_0\| \geq \|G_k\|$ for all $k > 0$ since the denominator of any diagonal entry of $G_k$ grows each iteration. Therefore, by our choice of $\eta$ we continue to have

$$\leq \eta\|G_k A^T b\| \cdot \sum_{i=0}^{k-1}\|I - \eta G_0 A^T A\|^i \tag{31}$$

$$\leq \eta\|G_k A^T b\| \cdot \sum_{i=0}^{k-1} 1 \tag{32}$$

$$= \eta(k-1)\|G_k A^T b\| \tag{33}$$

$$\leq \eta k\|G_k\| \cdot \|A^T b\| \tag{34}$$

$$\leq \eta k\|G_0\| \cdot \|A^T b\|. \tag{35}$$

So by similar manipulation as in the gradient descent case, we have that

$$|(A^TAx_k)_i| \leq \|A(:,i)\| \cdot \|A\| \cdot \|x_k\| \tag{36}$$

$$\leq \frac{1}{\delta}|A(:,i)^Tb| \cdot \|A\| \cdot k\eta\|G_0\| \cdot \|A^Tb\| \tag{37}$$

$$= k \cdot \left(\frac{\eta\|A\| \cdot \|G_0\| \cdot \|A^Tb\|}{\delta}\right)|A(:,i)^Tb|. \tag{38}$$

Again, by our choice of $k$ in the hypothesis, we must have that

$$|(A^T A x_k)_i| \leq \frac{1}{2}|A(:,i)^T b|. \tag{39}$$

This concludes the work and we continue with the main proof:

Using (13), we derive the following upper and lower bounds for both $|\mathbf{g}_k^{ada}(i)|$ and $|\mathbf{g}_k^{gd}(i)|$:

$$\begin{aligned} |\mathbf{g}_k(i)| &= |(A^T A x_k)_i - (A^T b)_i| \\ &\leq |(A^T A x_k)_i| + |(A^T b)_i| \\ &\leq \frac{1}{2}|(A^T b)_i| + |(A^T b)_i| \\ &= \frac{3}{2}|(A^T b)_i| \end{aligned} \tag{40}$$

and

$$\begin{aligned} |\mathbf{g}_k(i)| &= |(A^T A x_k)_i - (A^T b)_i| \\ &\geq \big||(A^T b)_i| - |(A^T A x_k)_i|\big| \\ &\geq \left||(A^T b)_i| - \frac{1}{2}|(A^T b)_i|\right| \\ &= \frac{1}{2}|(A^T b)_i| \end{aligned} \tag{41}$$

where $x_k$ depends on whether we set $\mathbf{g}_k(i)$ to be with respect to Adagrad or GD. Combining (41) and (40) we get:

$$\frac{1}{2}|(A^T b)_i| \leq |\mathbf{g}_k(i)| \leq \frac{3}{2}|(A^T b)_i|. \tag{42}$$

We therefore have an upper and lower bound for $\Delta x_k^{gd}(i)$ with respect to $|(A^T b)_i|$:

$$\frac{\eta}{2}|(A^T b)_i| \leq \left|\Delta x_k^{gd}(i)\right| \leq \frac{3\eta}{2}|(A^T b)_i|. \tag{43}$$

Now consider the denominator of the diagonal entries $i$ satisfying our assumption, and denote it as $(G_k^{1/2})_i$:

$$(G_k^{1/2})_i := \sqrt{\sum_{\tau=0}^{k} \mathbf{g}_{\tau,i}^2 + \epsilon}. \tag{44}$$

Using (42) with respect to $\mathbf{g}_k^{ada}$, we establish the following bound for $(G_k^{1/2})_i$:

$$\sqrt{k} \cdot \sqrt{(\frac{1}{2}(A^T b)_i))^2 + \epsilon k^{-1/2}} \leq (G_k^{1/2})_i \leq \sqrt{k} \cdot \sqrt{(\frac{3}{2}(A^T b)_i))^2 + \epsilon k^{-1/2}}. \tag{45}$$

Taking the inverse of $(G_k^{1/2})_i$, our bounds become:

$$\frac{1}{\sqrt{k}\cdot\sqrt{(\frac{3}{2}(A^Tb)_i))^2+\epsilon k^{-1/2}}} \leq (G_k^{1/2})_i^{-1} \leq \frac{1}{\sqrt{k}\cdot\sqrt{(\frac{1}{2}(A^Tb)_i))^2+\epsilon k^{-1/2}}}. \tag{46}$$

We can then bound our update $\Delta x_k^{ada}(i)$ with respect to $|(A^Tb)_i|$:

$$\left|\frac{\eta}{2\sqrt{k}\cdot\sqrt{(\frac{3}{2}(A^Tb)_i))^2+\epsilon k^{-1/2}}}(A^Tb)_i\right| \leq \left|\Delta x_k^{ada}(i)\right| \leq \left|\frac{3\eta}{2\sqrt{k}\cdot\sqrt{(\frac{1}{2}(A^Tb)_i))^2+\epsilon k^{-1/2}}}(A^Tb)_i\right| \tag{47}$$

Using (47) and (43) we get the following final bound for our ratio:

$$\left|\frac{3}{\sqrt{k}\cdot\sqrt{(\frac{3}{2}(A^Tb)_i)^2+\epsilon k^{-1/2}}}\right| \leq \frac{|\Delta x_k^{ada}(i)|}{|\Delta x_k^{gd}(i)|} \leq \left|\frac{3}{\sqrt{k}\cdot\sqrt{(\frac{1}{2}(A^Tb)_i)^2+\epsilon k^{-1/2}}}\right|. \tag{48}$$

This concludes the proof. ■

## Appendix B. Proof of Proposition 2

**Proof** [Proposition 2] We note that by the sign invariance under the norms, it suffices to prove this for $x \geq 0$. So, let $x = (x_1, \ldots, x_d)^T > 0$. To show the lower bound, we have by the Cauchy-Schwartz inequality that

$$\begin{aligned} \|x\|_1 = \sum_{k=1}^{d} |x_k| \leq \left( \sum_{k=1}^{d} |x_k|^2 \right)^{1/2} \cdot \left( \sum_{k=1}^{d} 1^2 \right)^{1/2} = \sqrt{d}\|x\|_2 \\ \implies \frac{1}{\sqrt{d}} \leq \frac{\|x\|_2}{\|x\|_1}. \end{aligned} \tag{49}$$

For the upper bound, squaring both sides we get that

$$\|x\|_1^2 = (x_1 + \cdots + x_d)^2 \geq x_1^2 + \cdots + x_d^2 = \|x\|_2^2 \tag{50}$$

which holds due to our assumption that $x \geq 0$. Rearranging yields our upper bound:

$$\|x\|_2^2 \leq \|x\|_1^2 \implies \frac{\|x\|_2}{\|x\|_1} \leq 1. \tag{51}$$

This concludes the proof. ■

## Appendix C. Additional Experiment: Interpolation

In this additional experiment, we present an application for which having a low LSM score is advantageous in generalization performance. This experiment is based on radial basis function (RBF) interpolation [3]. We note that solving this problem is equivalent to solving a linear regression problem $Ax = b$, where $A$ is a matrix with entries satisfying $a_{ij} = \phi(\|r_i - \zeta_j\|_2)$, and $b_i = f(\zeta_i)$. Normally, $\phi$ would be a radial basis function. In our situation, we do not use a common RBF. Instead, we work with spike-like functions:

$$\phi(\|r - \zeta\|_2) := \max\left\{0, \frac{\alpha - \|r - \zeta\|_2}{\alpha^2}\right\} \tag{52}$$

where $\zeta \in \mathbb{R}^m$ is the centre of the spike, and $\alpha \in \mathbb{R}$ is a shaping parameter which affects both height and width of the spike. We note that 52 is constructed such that the integral evaluates to 1.

The goal of our experiment is to interpolate the "unknown" function $f(x) = x$. To do so, we set up our experiment by randomly generating a dataset $A \in \mathbb{R}^{2n\times d}$ $(2n < d)$ and labels $b \in \mathbb{R}^{2n}$. The entries

$$a_{ij} = \max\left\{0, \frac{\alpha - |r_i - \zeta_j|}{\alpha^2}\right\}$$

are generated by sampling $r_i \in \mathbb{R}$ using a $(0, 1)$-Uniform distribution, and the $\alpha_i \in \mathbb{R}$ are sampled from the same distribution but scaled by $0.1$. The centres $\zeta_j$ are also sampled using a $(0, 1)$-Uniform distribution, and the entries of $b$, $b_i$, are take to be the $r_i$ ($b_i = r_i$, since $f(r_i) = r_i$).

Using this generated dataset, we solve the underdetermined system $\bar{A}x = \bar{b}$ where $\bar{A} \in \mathbb{R}^{n\times d}$ and $\bar{b} \in \mathbb{R}^n$ are taken to be the reduced dataset used for training. We examine how well GD and Adagrad perform in generalizing the entire dataset by comparing their residuals:

$$\ell(x^*) = \|Ax^* - b\|_2. \tag{53}$$

We perform 5 runs and take the average of the residuals which have been normalized with respect to the $\ell_2$ norm of $b$, in order to keep consistency among the runs. Said runs operate on a dataset with dimension $2n = 100$, $d = 10'000$. Training is performed on the reduced dataset of $n$ samples, and residuals are calculated on the entire dataset (all $2n$ samples). Tables 3 and 4 display the LSM and residuals of the minimum 2-norm, and Adagrad solutions trained at the initial point $x_0 = 0$, for varying step-sizes in the range 0.001 to 1.

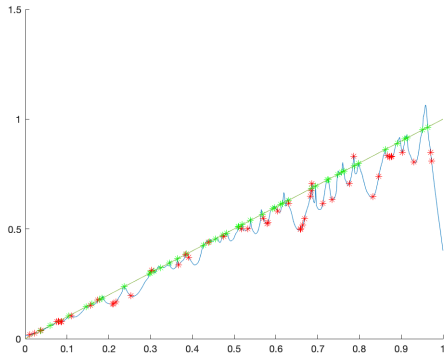
| Learning rate | Lower bound | 2-Norm | Adagrad | Upper bound |
|---|---|---|---|---|
| $\eta = 1$ | 0.01 | **0.0139** | 0.0148 | 1 |
| $\eta = 0.1$ | 0.01 | 0.0144 | **0.0126** | 1 |
| $\eta = 0.01$ | 0.01 | 0.0138 | **0.0123** | 1 |
| $\eta = 0.001$ | 0.01 | 0.0139 | **0.0124** | 1 |

Table 3: LSM for different learning rates with dataset size $n = 100, d = 10'000$. For each learning rate, the lowest LSM besides the lower bound is marked bold.

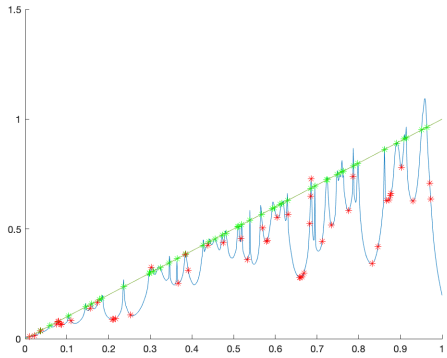
| Learning rate | 2-Norm | Adagrad |
|---|---|---|
| $\eta = 1$ | 0.2506 | **0.1044** |
| $\eta = 0.1$ | 0.3174 | **0.1406** |
| $\eta = 0.01$ | 0.2338 | **0.0962** |
| $\eta = 0.001$ | 0.2497 | **0.0888** |

Table 4: Residuals of different learning rates on entire dataset (n=100, d=10'000) with training data size n = 50, d = 10'000. For each learning rate, the lowest residual is marked bold.

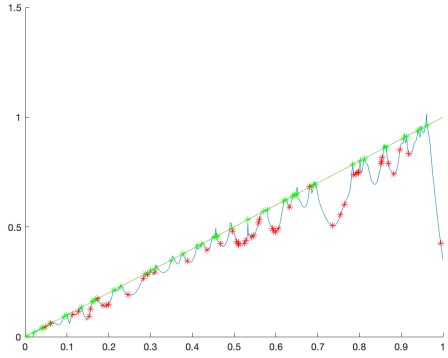
Following said tables are plots shown in Figure 1 which visualize the performance of said optimizers over two particular runs; the first run is ran for step-size $\eta = 0.001$, the latter is ran for step-size $\eta = 1$.



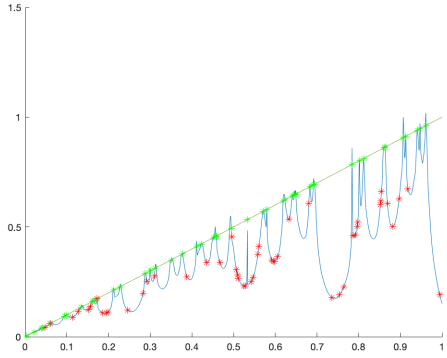
**Adagrad,** $\eta = 0.001$



**2-Norm,** $\eta = 0.001$



**Adagrad,** $\eta = 1$



**2-Norm,** $\eta = 1$

Figure 1: Plots of interpolation performance of Adagrad and minimum 2-Norm solutions on interpolation problem with $2n = 100$, $d = 10'000$. Green line represents the true function $y = x$, the green dots are the training samples, and the red dots are the interpolated values at the test points.

As can be seen from the information given in Tables 3 and 4, having lower LSM does have an impact on the interpolation performance. We notice that as we increase the step-size $\eta$ from $0.001$ to $1$, both the LSM and residuals of Adagrad increases showing some form of correlation between the two. We note, however, that diffusivity alone does not dictate the success Adagrad displays when comparing performance to GD. This is supported by the fact that for $\eta = 1$, the LSM of Adagrad is larger than the 2-norm, yet Adagrad still exhibits a lower residual score. This shows that, while diffusivity affects performance, it is not the sole explanatory factor.