OPT2023: 15th Annual Workshop on Optimization for Machine Learning

Diversity-Adjusted Adaptive Step Size

Parham Yazdkhasti PARHAM.YAZDKHASTI@CISPA.DE
Xiaowen Jiang XIAOWEN.JIANG@CISPA.DE
Sebastian U. Stich STICH@CISPA.DE
CISPA, Germany

Abstract

Optimizing machine learning models often requires careful tuning of parameters, especially the learning rate. Traditional methods involve exhaustive searches or adopting pre-established rates, both with drawbacks. The former is computationally intensive, a concern amplified by the trend toward larger models like large language models (LLM). The latter risks suboptimal model training. Consequently, there is growing research on adaptive and parameter-free approaches to reduce reliance on manual step size tuning. While adaptive gradient methods like AdaGrad, RMSProp, and Adam aim to adjust learning rates dynamically, they still rely on learning rate parameters dependent on problem-specific characteristics. Our work explores the interplay between step size and gradient dissimilarity, introducing a "diversity-adjusted adaptive stepsize" that adapts to different levers of dissimilarity in sampled gradients within the SGD algorithm. We also investigate approximate algorithms to compute this step size efficiently while maintaining performance.

1. Introduction

Optimizing machine learning models often requires careful selection of optimization parameters, with the learning rate being a critical yet challenging parameter to tune. Conventional practices involve exhaustive searches or pre-established learning rates from prior work, both of which have their own challenges and issues. The first approach demands a lot of computation power. Considering the increasing trend towards employing larger models, such as large language models (LLM), this approach becomes even less feasible. The latter approach increases the risk of training a suboptimal model. Hence, there has been an increasing focus on research on adaptive and parameter-free methods to reduce the reliance on fine-tuning step sizes or eliminate this requirement completely.

Adaptive gradient methods, like Adagrad [4], RMSProp [10], and Adam [7], can address this challenge by dynamically adjusting the learning rate based on problem-specific characteristics. Nonetheless, these methods still maintain a learning rate parameter, the ideal value of which depends on problem-specific properties that are often challenging to discover. While these methods have proven to enhance optimizer performance in modern problems like training deep neural networks, each method has its own limitations. Adagrad is effective in sparse settings but struggles in dense and nonconvex scenarios due to rapid learning rate decay with dense gradients. Methods like RMSProp, Adam, and AdaDelta [14] address this by using exponential moving averages of recent squared gradients to adaptively adjust the learning rate, reducing reliance on past gradients. Exponential moving average (EMA)-based adaptive methods like Adam and RMSProp are widely used in deep learning but raise concerns about convergence and generalization. Research has shown that EMA-based methods may not converge to the optimal solution in simple convex settings with a constant minibatch size, as their effective learning rate can increase rapidly [15].

The findings from these studies show the importance of approaching adaptive methods with a new perspective. Consequently, our focus has shifted towards understanding the relationship between step size and the inherent gradient dissimilarity found in the problem. We employ the "diversity" measure proposed in [13] to quantify this dissimilarity, which naturally emerges in our theoretical analysis of the optimal step size. We will show that our proposed step size is optimal for smooth functions.

We propose a "diversity-adjusted adaptive step size". It can automatically adjust to batch sizes and only needs to be provided with an approximation of the L constant for L-smooth functions. Our proposed method can adapt to varying batch sizes without requiring manual parameter tuning, simplifying the training process of machine learning models. Recognizing that computing the exact value of our proposed step size can be computationally demanding, we introduce algorithms for approximating this value while preserving performance comparable to the exact value.

2. Related Works

In optimization techniques, many methods have been developed to address the challenges of setting appropriate parameters and achieving efficient convergence. This section provides an overview of relevant works.

Polyak's step size [5] is a theoretically grounded adaptive method that provides a universal solution when the optimal function value is known. An adaptive re-estimation procedure can recover the optimal convergence rate even when this value is unavailable. Polyak stepsize with decreasing stepsizes can recover the convergence rate of gradient descent in the deterministic setting, only if the step size is initialized properly.

The AdaGrad family [4, 12, 14] of optimization methods, including Adam [7], has been widely employed in a wide range of machine-learning applications. These methods dynamically adjust the learning rate based on problem-specific characteristics, making them one the most popular methods for training neural networks in practice. However, it is important to note that these AdaGrad-style algorithms still rely on tuning the stepsize, preventing them from being fully parameter-free.

Yin et al. [13] delve into the connection between batch size and gradient dissimilarity, introducing a novel metric called "diversity" to measure gradient dissimilarity. It is shown that to prevent saturation in minibatch Stochastic Gradient Descent (SGD), the batch size should be proportional to diversity. This is similar to the stepsize we derive in this work.

Schaul et al. [11] proposed adaptive step size, which is claimed to be the optimum step size for optimizing quadratic functions with SGD. This step size considers the present variation in data samples in addition to the curvature of the target function, and in this regard, it is similar to our proposed method. Unfortunately, all of the theoretical analysis of this paper is based on a strong assumption. Authors in this work assume that the function can be approximated to a quadratic function. Therefore, this work's proposed theories and convergence proofs are limited to quadratic functions. In contrast, our proposed theorems extend beyond quadratic functions and apply to a broader range of functions, including smooth and strongly convex functions. The authors also proposed an approximate algorithm for relaxing this constraint on the target function. This approximate algorithm simply uses an exponential moving average to estimate the first and second moments of the gradient.

Ivgi et al. [6] proposed an adaptive step size called DoG, incorporating a distance term with a cumulative sum over the past gradients. In contrast to the AdaGrad family of optimization methods,

their proposed method is fully parameter-free. However, this method uses cumulative sum over the past gradients, which can be problematic. It often leads to a rapid decay in the learning rate when gradients are dense, which is often the case in many machine-learning applications. This kind of cumulative summing also can be problematic in non-stationary cases.

3. Proposed Methods

This section introduces our adaptive step size for Stochastic Gradient Descent (SGD) and provides insights into its derivation.

Problem Setting. Our goal is to minimize a loss function $f : \mathbb{R}^d \to \mathbb{R}$, where f is defined as:

$$\arg\min_{\mathbf{x}\in\mathbb{R}^d} [f(\mathbf{x}) := \mathbb{E}_\xi[f(\mathbf{x}, \xi)]] \ .$$

We assume that f is lower bounded by a finite value $f^\star$. Stochastic Gradient Descent (SGD) is the most commonly used method for approximating the solution of a stated problem. The SGD update rule is defined as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, \xi_t) \, ,$$

where ξ_t denotes the randomness (e.g. data sample or mini-batch) selected uniformly at random at iteration t.

Stepsize Derivation. When we assume that the objective function f is L-smooth, it holds

$$\mathbb{E}_{\xi_t} f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq -\gamma_t \|\nabla f(\mathbf{x}_t)\|^2 + \frac{L}{2}\gamma_t^2 \mathbb{E}_{\xi_t}\left[\|\nabla f(\mathbf{x}_t, \xi_t)\|^2\right] \ . \tag{1}$$

By minimizing the right-hand side of Equation (1), we can derive the optimal step size that maximizes the reduction in function value at each iteration:

$$\gamma_t^\star = \arg\min_{\gamma_t} \left\{ -\gamma_t \|\nabla f(\mathbf{x}_t)\|^2 + \frac{L}{2}\gamma_t^2 \mathbb{E}_{\xi_t}\left[\|\nabla f(\mathbf{x}_t, \xi_t)\|^2\right] \right\} = \frac{\|\mathbb{E}_{\xi_t} \nabla f(\mathbf{x}_t, \xi_t)\|^2}{L\mathbb{E}_{\xi_t}\|\nabla f(\mathbf{x}_t, \xi_t)\|^2} \tag{2}$$

This leads us to propose our adaptive step size, defined as:

$$\gamma_t^\star = \frac{1}{L} \cdot \frac{\|\nabla f(\mathbf{x}_t)\|^2}{\|\nabla f(\mathbf{x}_t)\|^2 + \mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2} \tag{3}$$

By defining the diversity measure in the following way :

$$D_t = \frac{\mathbb{E}_{\xi_t}\|\nabla f(\mathbf{x}_t, \xi_t)\|^2}{\|\mathbb{E}_{\xi_t} \nabla f(\mathbf{x}_t, \xi_t)\|^2} \tag{4}$$

we can establish a connection between this measure and our proposed step size:

$$\gamma_t^\star = \frac{1}{LD_t} \tag{5}$$

Discussion. Note that the proposed stepsize $\gamma_t^\star$ depends on the unknown parameters $\|\nabla f(\mathbf{x}_t)\|^2$, $\mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2$ and L. Computing the gradient and the variance exactly in each iteration would not be efficient, as this would amount to compute a full-batch gradient on all the data (except for some toy problems). Therefore, the stepsize $\gamma_t^\star$ in (3) should be viewed as an idealized step

size, that can offer valuable insights into a novel approach to developing adaptive step sizes. To bridge the gap between theory and practice, we discuss methods to estimate approximations of these parameters, and design variants of the exact method to make it suitable for practical applications in Section C. Surprisingly, we observe that $\gamma_t^\star$ can be approximated with good precision with an acceptable computational cost. For brevity, we will discuss the main properties of the ideal stepsize $\gamma_t^\star$ in the main text.

These results show that the critical factor for determining the stepsize is the gradient variance scaled by the full gradient norm. Additionally, it implies that as the noise variance increases, the optimization algorithm should adopt a more conservative approach, taking smaller steps. Conversely, when the noise variance is small in comparison to the full gradient norm, the optimizer can act more confidently and pursue more aggressive steps. Another benefit of the proposed step size is its capability to adapt to different batch sizes automatically, without requiring any manual tuning.

Equation (3) bears similarities to previous research in this field. Specifically, for optimizing quadratic functions with diagonal and similar Hessian matrices, our step size simplifies to the adaptive step size proposed in [11]. This indicates that the method presented in [11] can be considered a special case of our adaptive step size. Our proposed approach also shares similarities with the work introduced in [13]. They advocate that the optimal batch size for running the SGD algorithm should be proportional to diversity, suggesting that gradient noise variance should be scaled inversely with diversity. Our adaptive step size also leverages the diversity measure to regulate gradient noise variance. For a more detailed exploration of these connections please refer to Appendix B.

4. Convergence Rate

Theorem 1 *Let f be an L-smooth function (8) with $\mathbf{x}^\star$ being its minimizer, and let assumption (10) hold. If the proposed step size in equation (3) is chosen, then the iterates generated by Stochastic Gradient Descent (SGD) satisfy:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{2L(f(\mathbf{x}_0)-f^\star)}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{T}}\,. \tag{6}$$

Theorem 2 *Suppose f is an L-smooth (8) and μ-strongly convex (9) function with $\mathbf{x}^\star$ being its minimizer, and let assumption (10) hold. If we choose the proposed step size in equation (3), then the iterates generated by Stochastic Gradient Descent (SGD) satisfy:*

$$\mathbb{E}\left[\left\|\mathbf{x}_T-\mathbf{x}^\star\right\|^2\right] \leq \frac{2L}{\mu}\cdot\frac{\max\{\frac{\sigma^2}{\mu^2}\,,\left\|\mathbf{x}_0-\mathbf{x}^\star\right\|^2\}}{T}\,. \tag{7}$$

The adaptive step size we have introduced is specifically designed for non-convex and smooth functions. Consequently, while it provides acceptable convergence rates for such functions, it may not yield the best possible rate for strongly convex functions. However, it is worth noting that variations of our proposed step size could potentially offer a solution for strongly convex functions.

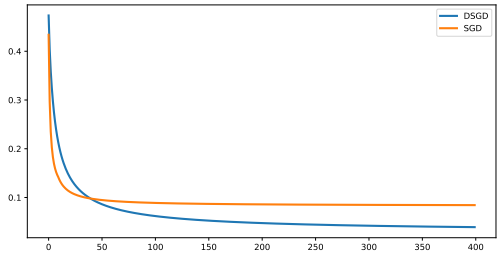

Figure 1: Convergence of the average iterates of DSGD vs. constant stepsize SGD. The average iterate of constant stepsize SGD only converges to a neighborhood of the solution.

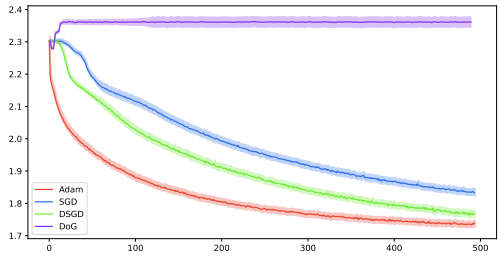

Figure 2: Comparing DSGD with Adam, vanilla SGD, and DoG on LeNet with the CIFAR-10 dataset. Loss vs. number of iterations.

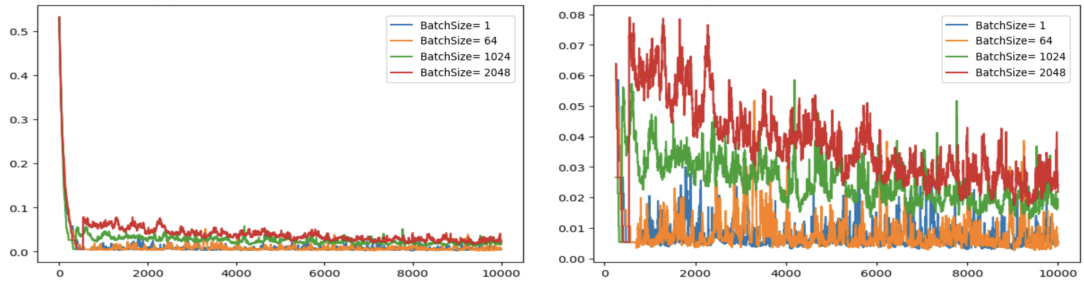

Figure 3: This figure shows how our proposed adaptive step size adapts to different batch sizes automatically. The left figure illustrates stepsize vs. iterations, while the right panel provides the same plot with the initial 100 iterates excluded for clarity.

5. Experiments

For calculating the diversity measure in both of the experiments in Sections 5.2 and 5.3 without having to compute full batch gradients, we employed an approximation algorithm outlined in Algorithm 1 (termed DSGD), see Appendix C.

5.1. Convergence of DSGD

Figure 1 compares our proposed adaptive step size (DSGD) against constant stepsize SGD. We plot the average of the gradient norms (from the first iteration to the iteration t). While SGD fails to converge, DSGD demonstrates convergence. For this experiment, a linear classifier was trained on a small portion of the MNIST [3] dataset. Zero-mean Gaussian noise with a variance of 0.5 was added to all of the data points. The noise was added to give us control over the stochastic gradient noise in the problem.

5.2. Comparing Approximate DSGD with Other Commonly Used Methods

In Figure 2, we conducted a performance comparison between our proposed method and other well-known optimization algorithms. Here we trained a LeNet deep neural network from scratch. The objective of this training was image classification using the CIFAR-10 dataset [8]. It is worth

noting that we fine-tuned hyperparameters for both the Adam and SGD algorithms. For the fine-tuning of the hyperparameters, The fine-tuning process involved testing ten different step sizes with logarithmic spacing. Furthermore, in this experiment, a hash table with a length of 100 ($C = 100$) was employed, and the exact evaluation period was configured to be 2000 ($K = 2000$).

5.3. DSGD Adaptes to Different Batch Sizes

One notable advantage of our proposed method is its ability to adapt to different batch sizes automatically. Figure 3 illustrates how our adaptive step size responds to changes in batch sizes when optimizing the same problem. For this experiment, we optimized the weights of a linear regression model using our proposed adaptive step size for the SGD algorithm. The model is optimized to fit the YearPredictionMSD [1] data set.

6. Conclusion

In this paper, we introduced an adaptive step size aimed at maximizing the decrease in function value at each iteration. This adaptive step size leverages the diversity measure to adapt according to varying levels of gradient noise. We provided a convergence proof for both smooth and non-convex functions. Additionally, we presented approximation methods to make our approach applicable in practical scenarios. While our method ensures convergence for strongly convex functions as well, it does not achieve the optimal convergence rate known for this class of functions. This observation could inspire future research in this direction.

References

[1] T. Bertin-Mahieux. YearPredictionMSD. UCI Machine Learning Repository, 2011. DOI: https://doi.org/10.24432/C50K61.

[2] Yair Carmon and Oliver Hinder. Making sgd parameter-free. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 2360–2389. PMLR, 02–05 Jul 2022.

[3] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[5] Elad Hazan and Sham Kakade. Revisiting the polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

[6] Maor Ivgi, Oliver Hinder, and Yair Carmon. Dog is sgd's best friend: A parameter-free dynamic step size schedule. *arXiv preprint arXiv:2302.12022*, 2023.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[9] Adam M Oberman and Mariana Prazeres. Stochastic gradient descent with polyak's learning rate. *arXiv preprint arXiv:1903.08688*, 2019.

[10] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[11] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International conference on machine learning*, pages 343–351. PMLR, 2013.

[12] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21(1):9047–9076, 2020.

[13] Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1998–2007. PMLR, 2018.

[14] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[15] Zhenxun Zhuang. *Adaptive strategies in non-convex optimization*. PhD thesis, Boston University, 2022.

Appendix A. Appendix

Assumption 1 *(L-smooth) The function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if*

$$f(\mathbf{x}) - f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \geq \frac{1}{2L}\left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\|^2, \quad \mathbf{x},\mathbf{y} \in \mathbb{R}^d \tag{8}$$

Assumption 2 *(μ-convex) The function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if*

$$f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \geq \frac{\mu}{2}\left\|\mathbf{x}-\mathbf{y}\right\|^2, \quad \mathbf{x},\mathbf{y} \in \mathbb{R}^d \tag{9}$$

Assumption 3 *The function $f(\mathbf{x},\xi)$ has a bounded gradient variance if*

$$\mathbb{E}_{\xi_t}\left\|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t,\xi_t)\right\| \leq \sigma^2 \tag{10}$$

Proof [Proof of Theorem 1]

From the quadratic upper bound, we know

$$\begin{aligned}
\mathbb{E}\, f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) - \frac{\left\|\nabla f(\mathbf{x}_t)\right\|^2}{2L\,\mathbb{E}_\xi\left\|\nabla f(\mathbf{x}_t,\xi)\right\|^2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \\
&\leq \min_\gamma\left[f(\mathbf{x}_t) - \gamma\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma^2 L}{2}\mathbb{E}_\xi\left\|\nabla f(\mathbf{x}_t,\xi)\right\|^2\right] \\
&\leq f(\mathbf{x}_t) - \gamma\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma^2 L}{2}\mathbb{E}_\xi\left\|\nabla f(\mathbf{x}_t,\xi)\right\|^2 \\
&\leq f(\mathbf{x}_t) + (\frac{L}{2}\gamma^2 - \gamma)\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma^2 L}{2}\sigma^2
\end{aligned}$$

After taking an expectation and sum on both sides of the inequality, and setting the step size to $\gamma = \frac{1}{L\sqrt{T}}$ we have

$$\gamma(1 - \frac{L}{2}\gamma) \sum_{t=0}^{T-1} \mathbb{E} \left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathbb{E}\left[f(\mathbf{x}_0) - f(\mathbf{x}_T)\right] + \frac{\gamma^2 LT}{2}\sigma^2$$

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq 2\frac{\mathbb{E}\left[f(\mathbf{x}_0) - f(\mathbf{x}_T)\right]}{T\gamma} + \gamma L\sigma^2$$

$$\leq 2\frac{Lf(\mathbf{x}_0)}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{T}}$$

■

Lemma 3 *Assume a transformation ψ is defined as:*

$$\psi_{\beta,r}(z) = \frac{\beta + rz}{\beta + z} z \tag{11}$$

for any $\alpha > 0$, $r \in [0, 1)$, $b > 0$ and $c = \frac{1-r}{\beta + \frac{r}{b}}$; then

$$\psi_{\beta,r}\left(\frac{1}{ck + b}\right) \leq \frac{1}{c(k+1) + b}\,. \tag{12}$$

Proof (This proof idea is from [9]). Consider $x = \frac{1}{y}$. Then, define S as the multiplicative inverse of $\psi_{\beta,r}$:

$$\psi_{\beta,r}(x) = \psi_{\beta,r}\left(\frac{1}{y}\right) = \frac{1}{y}\frac{\beta y + r}{\beta y + 1} := \frac{1}{S(y)}.$$

Next, we consider the sequence $a_k := ck + b$. We start by expanding the difference:

$$\begin{aligned} S\left(a_k\right) - a_{k+1} &= a_k \frac{\beta a_k + 1}{\beta a_k + r} - (a_k + c) \\ &= \frac{a_k(1 - r - c\beta) - cr}{\beta a_k + r} \\ &= \frac{kc(1 - r - c\beta) + (b(1 - r - c\beta) - cr)}{\beta ck + (\beta b + r)}. \end{aligned}$$

This difference is positive if both the denominator and numerator are positive for $k \geq 0$. The denominator's positivity follows from $b, c, r, \beta \geq 0$. The numerator is positive for all $k \geq 0$ provided

$$1 - r - c\beta \geq 0,$$

since the choice of c guarantees $b(1 - r - c\beta) - cr = 0$. In fact, the positivity of this expression also holds for our choice of c.

Hence, $S\left(a_k\right) - a_{k+1} \geq 0$. Then, using (10),

$$\psi_{\beta,r}\left(\frac{1}{a_k}\right) = \frac{1}{S\left(a_k\right)} \leq \frac{1}{a_{k+1}} = \frac{1}{c(k+1) + b},$$

and the lemma follows. ■

Lemma 4 *We have the following lower bound on the step size,*

$$\gamma_t = \frac{1}{L} \cdot \frac{\|\nabla f(\mathbf{x}_t)\|^2}{\|\nabla f(\mathbf{x}_t)\|^2 + \mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2} \geq \frac{1}{L} \cdot \frac{1}{1 + \frac{\sigma^2}{\mu^2}(\|\mathbf{x}_t - \mathbf{x}^\star\|^2)^{-1}} \tag{13}$$

Proof

$$\begin{aligned}
\gamma_t &= \frac{1}{L} \cdot \frac{\|\nabla f(\mathbf{x}_t)\|^2}{\|\nabla f(\mathbf{x}_t)\|^2 + \mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2} \\
&\overset{(10)}{\geq} \frac{1}{L} \cdot \frac{\|\nabla f(\mathbf{x}_t)\|^2}{\|\nabla f(\mathbf{x}_t)\|^2 + \sigma^2} \\
&= \frac{1}{L} \cdot \frac{1}{1 + \frac{\sigma^2}{\|\nabla f(\mathbf{x}_t)\|^2}} \\
&\overset{(9)}{\geq} \frac{1}{L} \cdot \frac{1}{1 + \frac{\sigma^2}{\mu^2}(\|\mathbf{x}_t - \mathbf{x}^\star\|^2)^{-1}}
\end{aligned}$$

■

Proof [Proof of Theorem 2] Here we prove the theorem by using induction. Let's assume:

$$\mathbb{E}\,\|\mathbf{x}_t - \mathbf{x}^\star\|^2 \leq \frac{1}{\alpha t + (\|\mathbf{x}_0 - \mathbf{x}^\star\|^2)^{-1}} \tag{14}$$

where

$$\alpha = \frac{\mu}{\frac{L\sigma^2}{\mu^2} + (L - \mu)\,\|\mathbf{x}_0 - \mathbf{x}^\star\|^2} \tag{15}$$

$$
\begin{aligned}
\mathbb{E}_{\xi_t}\left\|\mathbf{x}_{t+1}-\mathbf{x}^\star\right\|^2 &= \left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 + \mathbb{E}_{\xi_t}\left[\nabla f(\mathbf{x}_t,\xi_t)^T(\mathbf{x}_t-\mathbf{x}^\star)\right] + \gamma_t^2\,\mathbb{E}_{\xi_t}\left\|\nabla f(\mathbf{x}_t,\xi_t)\right\|^2 \\
&\overset{(9)}{\leq} (1-\mu\gamma_t)\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 - 2\gamma_t(f(\mathbf{x}_t)-f(\mathbf{x}^\star)) + \gamma_t^2\,\mathbb{E}_{\xi_t}\left\|\nabla f(\mathbf{x}_t,\xi_t)\right\|^2 \\
&\overset{(3)}{\leq} (1-\mu\gamma_t)\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 + \gamma_t\left(-2(f(\mathbf{x}_t)-f(\mathbf{x}^\star)) + \frac{1}{L}\left\|\nabla f(\mathbf{x}_t)\right\|^2\right) \\
&\overset{(8)}{\leq} (1-\mu\gamma_t)\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 + \gamma_t\left(-2(\frac{1}{2L}\left\|\nabla f(\mathbf{x}_t)\right\|^2) + \frac{1}{L}\left\|\nabla f(\mathbf{x}_t)\right\|^2\right) \\
&= (1-\mu\gamma_t)\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 \\
&\overset{(13)}{\leq} (1-\frac{\mu}{L}\cdot\frac{1}{1+\frac{\sigma^2}{\mu^2\|\mathbf{x}_t-\mathbf{x}^\star\|^2}})\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 \\
&= (\frac{1+\frac{\sigma^2}{\mu^2\|\mathbf{x}_t-\mathbf{x}^\star\|^2}-\frac{\mu}{L}}{1+\frac{\sigma^2}{\mu^2\|\mathbf{x}_t-\mathbf{x}^\star\|^2}})\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 \\
&= \frac{(1-\frac{\mu}{L})\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2+\frac{\sigma^2}{\mu^2}}{\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2+\frac{\sigma^2}{\mu^2}}\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2 \\
&\overset{(11)}{=} \psi_{\frac{\sigma^2}{\mu^2},(1-\frac{\mu}{L})}\left(\left\|\mathbf{x}_t-\mathbf{x}^\star\right\|^2\right) \\
&\overset{(14)}{\leq} \psi_{\frac{\sigma^2}{\mu^2},(1-\frac{\mu}{L})}\left(\frac{1}{\alpha t+(\|\mathbf{x}_0-\mathbf{x}^\star\|^2)^{-1}}\right) \qquad \text{(according to monotonicity of } \psi) \\
&\overset{(12)}{\leq} \frac{1}{\alpha(t+1)+(\|\mathbf{x}_0-\mathbf{x}^\star\|^2)^{-1}}
\end{aligned}
$$

which concludes the induction.

■

Appendix B. Comparison to related works

Equation (3) bears similarities to some of the previously discussed related works in this research domain. For instance, for one-dimensional quadratic functions with identical Hessians (h) but different $\arg\min_x f_i(x) := x_i^\star$, equation (3) simplifies to:

$$\frac{1}{h}\cdot\frac{\|x_t-x^\star\|^2}{\|x_t-x^\star\|^2+\sigma^2}$$

Here, σ^2 represents the variance of $x_i^\star$. For multi-dimensional quadratic functions with identical diagonal Hessian matrices (H) but different $\arg\min_{\mathbf{x}} f_i(\mathbf{x}) := \mathbf{x}_i^\star$, we define Σ as the covariance of $\mathbf{x}_i^\star$. Assuming Σ is diagonal, the equation (3) can be expressed as:

$$\frac{1}{\|H\|}\cdot\frac{\|H(\mathbf{x}_t-\mathbf{x}^\star)\|^2}{\|H(\mathbf{x}_t-\mathbf{x}^\star)\|^2+\mathbb{E}_j\left\|H(\mathbf{x}_j^\star-\mathbf{x}^\star)\right\|^2} = \frac{1}{\|H\|}\cdot\frac{\sum_{i=0}^{n-1}h_i^2(x_t^{(i)}-(x^\star)^{(i)})^2}{\sum_{i=0}^{n-1}h_i^2(x_t^{(i)}-(x^\star)^{(i)})^2+\sum_{i=0}^{n-1}h_i^2\sigma_i^2}$$

Here, we define $h_i = H_{i,i}$ and $\sigma_i^2 = \Sigma_{i,i}$. Notably, all of these results align with the proposed findings in [11]. Therefore, it can be asserted that their proposed adaptive step size is a special case of our proposed step size.

In comparison to the work proposed in [13], there are notable similarities. To represent the Diversity of gradient norm at $\mathbf{x}_t$, we will use D_t. The authors in [13] suggest that the optimal batch size for running the SGD algorithm should be proportional to diversity. In other words, they propose a method that scales $\mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2$ with $\frac{1}{D_t}$. By considering this, we can rewrite the decrease lemma for nonconvex functions as follows:

$$\mathbb{E}_{\xi_t}\left[f(\mathbf{x}_{t+1})\right] \leq f(\mathbf{x}_t) - \frac{\gamma}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\gamma^2 L}{2} \cdot \frac{\mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2}{D_t}$$

On the other hand, our proposed method suggests that the step size in each iteration should be proportional to the inverse of diversity. In other words, we define our adaptive step size as $\gamma_t := \frac{1}{LD_t}$. With this assumption, we have:

$$\mathbb{E}_{\xi_t}\left[f(\mathbf{x}_{t+1})\right] \leq f(\mathbf{x}_t) - \frac{1}{2LD_t}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{1}{2L} \cdot \frac{\mathbb{E}_\xi \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \xi_t)\|^2}{D_t^2}$$

Clearly, both methods use the diversity measure to determine the strength with which they should reduce the present noise in the gradient. To be more precise, we can say that our proposed method employs the diversity measure to address the existing trade-off between the last two terms in the decrease lemma.

In comparison to the approach introduced in [6], although both step sizes leverage the norm of the stochastic gradient to adjust the step size, the underlying motivations differ significantly. While the adaptive step size in DoG is inspired by the theoretical results presented in [2], our proposed method is primarily motivated by the concept that an adaptive step size for SGD should be capable of adjusting to the current noise levels in the gradient.

Appendix C. Methods for Approximating DSGD

In this section assume we are optimizing a finite sum problem where

$$f(\mathbf{x}_t) = \sum_{i=1}^{n} f_i(\mathbf{x}_t)$$

Finite sum problems are a special case of 3. Many machine learning problems can be represented in the finite-sum formulation. Many machine learning problems can be represented in the finite-sum formulation. Therefore, we present the approximate algorithms with the assumption that they will be used to solve these types of problems, making them more compatible with the problems' settings in practice.

C.1. Using a Hash Table

As we mentioned earlier, computing the exact value of our proposed adaptive step size is not practical and it is only feasible for very limited problems. Therefore here we propose a method for approximating the diversity measure. This method uses a hash table to keep track of the last C stochastic gradients, where C is the length of the table. In the ideal case, we would C should be

equal to the length of the data set. In this case, we could keep a record of the most recently evaluated stochastic gradient ($\nabla f_i(\mathbf{x})$ for the ith data) for each data point in the data set. However, it is obvious that we can not do that for large datasets. So instead of a full memory, here we use a hash table to address this problem. Using a hash table ensures that available space for memorizing past gradients is distributed evenly between all data points. We use H_j notation to represent the jth entry of our hash table.

Algorithm 1 DSGD with Hash Table

Parameters: smoothness coefficient, L ; hash table size, C; the total number of iterations, T; the period of evaluation of the full gradient, K

Initialize: $\mathbf{x}_0 \in \mathbb{R}^d$

for $t = 0, 1, 2, ..., T$ **do**

  if $t \equiv 0 \pmod{K}$ **then**

    for $i = 0, 1, ..., C$ **do**

      Sample i uniformly at random from $\{1, 2, ..., n\}$

      $j \equiv i \pmod{C}$

      $H_j = \nabla f_i(\mathbf{x}_0)$

    end

  end

  Sample i uniformly at random from $\{1, 2, ..., n\}$

  $j \equiv i \pmod{C}$

  $H_j = \nabla f_i(\mathbf{x}_t)$

  $d_t = \frac{\frac{1}{n}\sum_{j=1}^{n} \|H_j\|^2}{\|\frac{1}{n}\sum_{j=1}^{n} H_j\|^2}$

  $\gamma_t = \frac{1}{L.d_t}$

  $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f_i(\mathbf{x}_t)$

end

C.2. Stochastic Updates

Here we suggest another method to approximate the diversity measure. We propose an algorithm where we sometimes skip updating the step size and retain the last computed value. We believe that significant changes in the step size are uncommon. Thus, we introduce the following algorithm: the adaptive step size is updated only with a probability of p. It's important to note that, despite its reduced memory usage and faster execution compared to Algorithm 1, this algorithm does not approximate the diversity measure as precisely.

Algorithm 2 DSGD with Stochastic Updates

Parameters: smoothness coefficient, L ; hash table size, C; the total number of iterations, T; the probability of updating the stepsize, p

Initialize: $\mathbf{x}_0 \in \mathbb{R}^d$

$\mathbf{w}_0 \leftarrow \mathbf{x}_0$

for $i = 0, 1, ..., T$ **do**

Sample k_t at random from the distribution $Ber(p)$

$$\mathbf{w}_t = \begin{cases} \mathbf{x}_t & \text{if } k_t = 1 \\ \mathbf{w}_{t-1} & \text{if } k_t = 0 \end{cases}$$

$$\tilde{D}_t = \frac{\frac{1}{n}\sum_i^n \|\nabla f_i(\mathbf{w}_t)\|^2}{\|\frac{1}{n}\sum_i^n \nabla f_i(\mathbf{w}_t)\|^2}$$

Sample i uniformly at random from $\{1, 2, ..., n\}$

$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{L \cdot d_t}\nabla f_i(\mathbf{x}_t)$

end