THE SIMPLER THE BETTER: VANILLA SGD REVISITED

Anonymous authors
Paper under double-blind review

ABSTRACT

The stochastic gradient descent (SGD) method, first proposed in 1950's, has been the foundation for deep-neural-network (DNN) training with numerous enhancements including adding a momentum or adaptively selecting learning rates, or using both strategies and more. A common view for SGD is that the learning rate should be eventually made small in order to reach sufficiently good approximate solutions. Another widely held view is that the vanilla SGD is out of fashion in comparison to many of its modern variations. In this work, we provide a contrarian claim that, when training over-parameterized DNNs, the vanilla SGD can still compete well with, and oftentimes outperform, its more recent variations by simply using learning rates significantly larger than commonly used values. We provide some theoretical justifications to this claim, and also present computational evidence, across multiple tasks including image classification, speech recognition and natural language processing, in support of this claim.

1 INTRODUCTION

We are interested in minimizing a function $f : \mathbb{R}^d \to \mathbb{R}$ with an expectation form:

$$\min_{x \in \mathbb{R}^d} f(x) := \mathbb{E}_\xi[K(x, \xi)], \tag{1a}$$

where the subscript indicates expectation is computed on the random variable ξ. Specially, if the ξ probability distribution is clear and random variable ξ is uniformly distributed on N terms, the objective function f can be expressed into a finite-sum form:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{N} \sum_{i=1}^{N} f_i(x), \tag{1b}$$

where $f_i(x) : \mathbb{R}^d \to \mathbb{R}$ is the i-th component function. The optimization of Eqs.(1a) and (1b) is widely encountered in machine learning tasks (Goodfellow et al., 2016; Simonyan & Zisserman, 2014).

To solve this problem given in Eq.(1b), we can compute the gradient of objective function directly with a classic GD (Gradient Descent) algorithm. However, this method suffers from expensive gradient computation for extremely large N, and hence people apply stochastic gradient descent method (SGM) to address this issue. Incremental gradient descent (IGD) is a primary version of SGM, where its calculation of gradient proceeds on single component f_i at each iteration, instead of the whole.

As a special case of IGD, SGD (Robbins & Monro, 1951), a fundamental method to train neural networks, always updates parameters by the gradient computed on a mini-batch. There exists an intuitive view that SGD with constant step size (SGD-CS) potentially leads to faster convergence rate. Some studies (Solodov, 1998; Tseng, 1998) show that, under a so-called strong growth condition (SGC), SGD-CS converges to an optimal point faster than SGD with a diminishing step size. From this view, we would like to investigate whether the convergence can still be guaranteed in theory if we start with a larger learning rate. We would also like to know the practical impact from such an unusual choice of larger learning rate on the optimization results for DNN training. To address these questions, we need to discuss the convergence of SGD-CS applied to non-convex functions.

We also examine some specific examples to help understand how large learning rates can be while maintaining convergence in the context of IGD method.

Our main contributions are as follows. We first show the convergence of SGD-CS on non-convex functions under suitable conditions. We then construct linear regression problems with ℓ_2 regularization as illustrative examples to show that the size of convergent learning rates can be quite large. Our numerical results show that such large learning rates can actually deliver superior practical performance. Based on the above observations, we further introduce a modified SGD algorithm with a large initial learning rate (SGDL), and then provide extensive experiments on various popular tasks and models in computer vision, audio recognition and natural language processing. The results demonstrate that our SGDL method converges successfully and has a strong generalization performance, and outperform several popular adaptive and non-adaptive methods.

2 Related Work

There are many papers on stochastic optimization and we summarize typical ones which are most relevant with our work. The convergence of SGD with momentum for over-parameterized models is analyzed in the work of (Vaswani et al., 2018; Mai & Johansson, 2020). The work (Jastrzębski et al., 2017) investigates a large ratio of learning rate to batch size often leads to a wide endpoint. The work (Smith & Topin, 2019) shows a phenomenon called super-convergence which is in contrast to the results in (Bottou et al., 2018). Furthermore, the work (Li et al., 2020) assumes that momentum is also an important hyper-parameter for fine-tuning performance. The work (Li et al., 2019) demonstrates that a large learning rate on SGD with momentum can be used as a regularization technique to achieve better generalization in comparison to a small one. In this paper, we show a significantly larger learning rate can be applied than that used in (Li et al., 2019).

Adaptive gradient methods are widely in deep learning application. Popular solutions include AdaDelta (Zeiler, 2012), RMSProp(Hinton et al., 2012), Adam(Kingma & Ba, 2014) and so on. Unfortunately, it is believed that the adaptive methods may have a poor empirically performance. For example, Wilson et al. (2017) have observed that Adam hurts generalization performance in comparison to SGD with or without momentum.

More recently, several new learning rate schedules have proposed for SGD. The work of (Loshchilov & Hutter, 2016) introduce warm-restart technique, and the work of (Smith, 2017) proposes a strategy to cycle learning rate from maximum bound to minimum bound in a fixed step size. Several methods (Agarwal et al., 2017; Carmon et al., 2018) ensure to converge to local minima but these methods are not suitable for large problem due to expensive computation or slow convergence.

3 Preliminaries

3.1 SGD with Constant Step Size

In this section we introduce SGD-CS. For the problem defined in Eq.(1a), the iterative updates of SGD are given by

$$x_{k+1} = x_k - \alpha_k \nabla K(x_k, \xi_k), \alpha_k > 0, \tag{2a}$$

where x_{k+1} and x_k are the estimates at iterations $k+1$ and k respectively, ξ_k denotes the realization of the random variable ξ at iteration k, and α_k is the step size (a.k.a the learning rate) at iteration k. Assume that the gradient $\nabla_x K(x, \xi)$ is unbiased, i.e., $\mathbb{E}_\xi[\nabla_x K(x, \xi)] = \nabla f(x)$ holds for any x. For problem (1b), minibatch SGD updates are as follows:

$$x_{k+1} = x_k - \alpha_k \frac{1}{m} \sum_{j=1}^{m} \nabla f_{\xi_k^{(j)}}(x_k), \alpha_k > 0, \tag{2b}$$

where m is the fixed batch size, and $\xi_k^{(1)}, \ldots, \xi_k^{(m)}$ are samples uniformly generated from the set $\{1, \ldots, N\}$ at iteration k.

In order to converge to the optimal solution, the step size usually decreases closely to zero. Unfortunately, this technique leads to a slow convergence rate. For example, by using step size $\alpha_k = O(1/\sqrt{k})$, SGD converges to the ϵ-first order stationary points within $O(\epsilon^{-4})$ iterates for smooth and non-convex functions (Drori & Shamir, 2020). In the work of (Sa, 2017), SGD-CS is proposed to speed up the training process, but it does not guarantee to converge to a local optimal point in general. Even for strongly convex functions, SGD-CS will only converge to a neighborhood of the optimal solution no matter how sufficiently small a constant step size is employed (Bottou et al., 2018). Yet under the strong growth condition (Solodov, 1998; Tseng, 1998), SGD-CS converges to the optimal point. More particular descriptions about SGC are given in Appendix.A.1.

3.2 Assumptions

Our results are under the following assumptions with the notations in Appendix.A.2 :

(A.1) The objective function f has same structure as given in Eq.(1a).

(A.2) Each component function f_i in (1a) is twice continously differentiable, with the corresponding Hessian matrix H_i being Lipschitz continuous for constant L_i at the minimizer x^*. Without loss of generality, we suppose that

$$\|H_i(x) - H_i(x^*)\| \leq L\|x - x^*\|, \quad \text{for } i = 1, \ldots, N, \quad \text{where } L = \max_{i=1:N} L_i.$$

(A.3) As in (Zhang et al., 2000, Remark 5.2), we assume the point of interest x^* is a strong minimizer (i.e. $H(x^*) \succ 0$).

Based on the above technical assumptions, we focus on the convergence analysis of SGD-CS by discussing the convergence behavior precisely:

$$x_{k+1} = x_k - \alpha \frac{1}{m} \sum_{j=1}^{m} \nabla f_{\xi_k^{(j)}}(x_k). \tag{3}$$

Next, we introduce the notion of point attraction to help us understand the convergence behavior of the iteration (3).

Definition 1. *We say x^* is a point of attraction to Eq.(3) if there is an open ball $\mathcal{N}(x^*, \epsilon)$ of x^*, such that for any $x^0 \in \mathcal{N}$, $\{x_k\}$ generated by Eq.(3) all lie in $\mathcal{N}$ and converges to x^* in the ℓ^1 norm.*

The Ostrowski Theorem says a sufficient condition for x^* to be a point of attraction of the deterministic iteration $x_{k+1} = T(x_k)$ is that the spectral radius of $T'(x)$ is strictly less than one, which further implies the condition that there exists an open ball $\mathcal{N}$ such that once the initial $x_0 \in \mathcal{N}$, the iterates $\{x_k\}$ will converge to x^* with linear rate. We define the strong minimizer x^* as a point of strong attraction following the similar manner.

Definition 2. *We say x^* is a point of strong attraction to Eq.(3) if there exists a neighborhood $\mathcal{N}$ of x^*, such that for any $x^0 \in N$, the sequence $\{x_k\}$ generated by Eq.(3) all lie in $\mathcal{N}$ and satisfy*

$$\mathbb{E}\|x^k - x^*\|^2 \leq \rho^k \cdot \|x^0 - x^*\|^2, \quad \textit{for some } \rho \in [0, 1).$$

4 Results for Non-Convex Functions

In this section, we rigorously show the necessary condition for the minimizer x^* being a point of attraction, and Theorem 1 provides a sufficient condition for the strong minimizer x^* to be a point of strong attraction with high probability. The detailed proofs are in Appendix.A.3.

4.1 Necessary Condition

Lemma 1. *Suppose that the assumptions (A.1) and (A.2) hold, then*

$$\|\nabla f_i(x)\| < L\epsilon^2 + \|H_i^*\|\epsilon \quad \textit{for } i = 1, \ldots, N,$$

provided that $x \in \mathcal{N}(x^*, \epsilon)$.

Theorem 1. *Suppose that the assumptions (A.1) and (A.2) hold. If the minimizer x^* is a point of attraction of Eq.(3), then the interpolation property is satisfied, i.e.,*

$$\nabla f_i(x^*) = 0 \quad \textit{for } i = 1, \ldots, N.$$

4.2 Sufficient Condition

4.2.1 Sufficient Condition for Points of Strong Attraction

Define the error function $e_i(x)$ as

$$e_i(x) = -H_i^* \cdot (x - x^*) + [\nabla f_i(x) - \nabla f_i(x^*)]. \tag{4}$$

By submiting $e_i(x)$ into Eq.(3), we have

$$\begin{aligned} x_{k+1} - x^* &= x_k - \alpha \frac{1}{m} \sum_{j=1}^{m} [H_{\xi_k^{(j)}}^* \cdot (x_k - x^*) + e_{\xi_k^{(j)}}(x_k)] - x^* \\ &= \frac{1}{m} \sum_{j=1}^{m} \left[(I - \alpha H_{\xi_k^{(j)}}^*)(x_k - x^*) - \alpha e_{\xi_k^{(j)}}(x_k) \right]. \end{aligned} \tag{5}$$

In addition, our proof for the sufficient condition indicates a bound on the error function.

Lemma 2. *Suppose that the assumptions (A.1) and (A.2) hold, and x^* is a local minimizer of f. If the interpolation property is satisfied, then the error function defined in Eq.(4) is bounded by:*

$$\|e_i(x)\| \leq L\|x - x^*\|^2. \tag{6}$$

Under the assumption that $\{x_k\}$ stay in the local neighborhood, we can show that interpolation property is a sufficient condition for x^* being a point of strong attraction of Eq.(3), provided that the step-size α is sufficiently small.

Theorem 2. *Suppose that the assumptions (A.1)-(A.3) hold, and $\{x_k\}$ generated by iteration (3) all lie in the neighborhood $\mathcal{N}(x^*, \epsilon)$, where ϵ and the step-size α are sufficiently small such that*

$$\rho \equiv \lambda_{\max}\left(I - 2\alpha H^* + \alpha^2 (H^*)^2 + \alpha^2 \Sigma^* + \alpha^2 L^2 \epsilon^2 I + 2\alpha L \epsilon I \right) < 1, \tag{7}$$

and

$$0 < \alpha \leq \min_{i=1,\ldots,N} \left\{ \frac{1}{\lambda_{\max}(H_i^*)}, \frac{2}{\lambda_{\max}(H^*)} \right\}.$$

If $\nabla f_i(x^*) = 0$ for $i = 1, \ldots, N$, then the minimizer x^* is a point of strong attraction of the iteration (3).

Remark 1. *In particular, the inequality in Eq.(7) can be satisfied by choosing ϵ and α such that*

$$0 < \alpha < \frac{1}{L} \lambda_{\min}(H^*), \tag{8a}$$

$$0 < \alpha \leq \min_{i=1,\ldots,N} \left\{ \frac{1}{\lambda_{\max}(H_i^*)}, \frac{1}{2\lambda_{\max}(H^*)} \right\}, \tag{8b}$$

$$0 < \alpha < 2\lambda_{\min} \left([(H^*)^2 + \Sigma^* + L^2 \epsilon^2 I]^{-1/2} (H^* - L\epsilon I) [(H^*)^2 + \Sigma^* + L^2 \epsilon^2 I]^{-1/2} \right). \tag{8c}$$

4.2.2 Staying in the Local Neighborhood

Now we show that $\{x_k\}$ generated from Eq.(3) all stay in the local neighborhood with high probability. The proof follows the idea in (Tan & Vershynin, 2019). In the remaining of this section, we abuse the notation slightly and denote X_k as a random vector in the k-th iteration of SGD-CS scheme, and x_k as the realization of X_k. Considering the setup of Theorem 2 without assuming that $\{x_k\}$ always lies in the neighborhood $\mathcal{N}(x^*, \epsilon)$, we have the following technical results.

Lemma 3. *Let $x_0 \in \mathcal{N}(x^*, \epsilon)$ such that $\|x_0 - x^*\| \leq \sqrt{\delta}\epsilon$ for some $\delta \in [0, 1)$. Define the stopping time $\tau = \min\{k : X_k \notin \mathcal{N}(x^*, \epsilon)\}$, then $P\{\tau = \infty\} \geq 1 - \delta$.*

Remark 2. *With the setup stated in Lemma 3, the sequence $\{x_k\}$ stay in the local neighborhood $\mathcal{N}(x^*, \epsilon)$ with probability at least $1 - \delta(1 + \rho^k)$.*

Indeed, we can show the result in Theorem 2 by relaxing the assumption into that the sequence generated by Eq.(3) stays in the local neighborhood within infinite iterates. Under the setup stated in the Lemma 3, we build the sufficient condition for points of strong attraction with the probabilistic guarantee.

Theorem 3. *There exists an event E which holds with probability at least $P(E) \geq 1 - \delta$ and the sequence $\{x_k\}$ generated by iteration (3) satisfies*

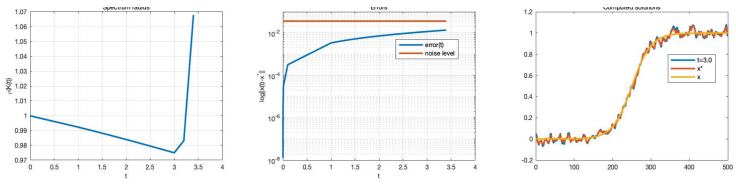
$$\mathbb{E}[\|X_k - x^*\|^2 1_E] \leq \rho^k \|x_0 - x^*\|^2 \tag{9}$$

Remark 3. *It's clear that x^* is a point of attraction if it is a point of strong attraction. Therefore, once the event that the iterates generated from SGD-CS stay in the local neighborhood happens, the interpolation property becomes a necessary and sufficient condition for the strong minimizer x^* to be a point of attraction to Eq.(3).*

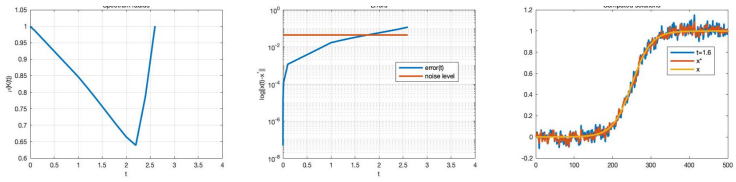
5 Case Studies on Large Learning Rates

This section shows Incremental Gradient Descent (IGD) with a constant step size actually converges into the global optimum point for some special problems, and then some numerical experiments are presented to illustrate this point. We present the results through an example as follows, with more examples appeared in Appendix.A.4.

$$f(x) = \frac{1}{2}\|Ax - b\|^2 + \frac{\mu}{2}\|D^{1/2}x\|^2. \tag{10}$$



(a) The first situation.



(b) The second situation.

Figure 1: Left: the range of spectral radius where IGD converges, it can be seen there are a wide range t for convergence. Middle: the Log error $\|x - \hat{x}\|_2$. The error line in blue grows slowly after an initial rapid rise. This implies that in the presence of noise, large t-values can still calculate x close to x^*. Right: calculated solution of a large radius, 3.0 for the first situation and 1.6 for the second one, the computed values of $x(t)$ do not significantly deviate from the noisy data $\hat{x}$ within the large t-value.

Under some mild assumptions (c.f Example 3 in Appendix.A.4), the IGD with constant step size t converges when

$$t \in \left(0, \frac{2N}{\rho(\mu D + A^{\mathrm{T}} A)}\right).$$

where ρ is the spectral radius.

The problem in Eq.(10) can be extended into general nonlinear least squares:

$$f(x) = \frac{1}{2}\sum_{i=1}^{N} r_i^2(x) = \frac{1}{2}R(x)^T R(x).$$

A sufficient condition for the convergence of IGD is

$$t \in \left(0, \frac{1}{\lambda_{\max}(\mathbb{E}[H_i(x^*)])}\right),$$

where $\mathbb{E}[H_i(x^*)] \equiv \frac{1}{N}\sum_i H_i(x^*)$, and $H_i(\cdot)$ denotes the Hessian matrix for the i-th component function $r_i(\cdot)$.

We conduct a numerical simulation for the regularized least squares problem in Eq.(10). Let training data x be generated via $x = \frac{1}{1+e^z} + N(0, \sigma^2)$ where z is uniformly sampled from $(0, 10)$. The minimizer x^* does not necessarily represent the best solution available due to noise, so all approximate solutions within the same noise level should be equally good in fact. Fig.1 shows the behavior of spectral radius ρ and regression error. The first row shows the results when data dimension $(= 500)$ is larger than dataset size$(= 100)$, the second row shows when they both equal to 500.

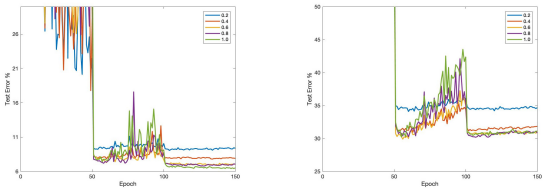
The experimental results demonstrate that with large step sizes, IGD still converge and have a decent performance, which meets our expectations. After the first sharp rise, the step size can be updated insensitive to the error. In our next section, we will implement vanilla SGD with a large initial learning rate and use this optimizer for large scale problems.

6 Experiments

We show the empirical results of different models to compare our SGDL with other popular optimization methods, including SGD, SGDM, Adam, and AdaDelta. We focus on the CIFAR10 and CIFAR100 image classification task (Krizhevsky et al., 2009), with the downsampled variant of ImageNet named as ImageNet32 (Chrabaszcz et al., 2017), the Speech Commands Dataset for audio recognition(Warden, 2018), and the language modeling task on Penn Treebank(Marcus et al., 1993).

6.1 CIFAR10 and CIFAR100

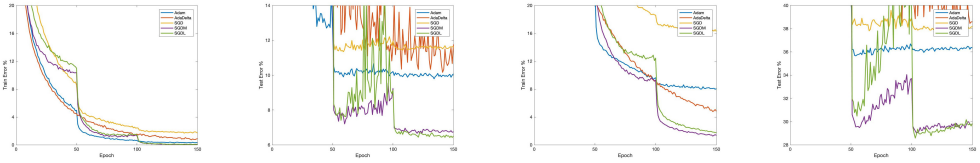
The first experiment is carried out on the CIFAR10 and CIFAR100 datasets, which are standard image collection with 10 and 100 classes respectively. After the preprocessing in Appendix.A.5.1, we train VGG (Simonyan & Zisserman, 2014), ResNet (He et al., 2016) and DenseNet(Huang et al., 2017) for up to 150 epochs and minibatch size to 256. Additional, for all optimizers without AdaDelta, we use an annealing strategy that the learning rate is lowered by 10 times at epoch 50 and 100. For ResNet experiments, we select a ResNet network with 56 layers and 120 layers respectively. For VGG, we use VGG16. For DenseNet, we use a DenseNet with 100 layers and growth rate $k = 12$. We first run a ResNet56 model on CIFAR10 and CIFAR100 with different learning rates. From the results show in Fig.2 we can see a small learning rate like 0.2, 0.4 has a relatively poor performance on the test set.



(a) CIFAR10. (b) CIFAR100.

Figure 2: Different learning rate in the ResNet56 model on CIFAR10 and CIFAR100.

We compare different optimization method including SGD,SGDM,Adam,AdaDelta, and shows the performance in Fig.3. Because of space constraints, more results can be found in Appendix.A.5.1. We can see that our method still works fairly well and has the same or even better performance in comparison to SGDM and other optimizers.

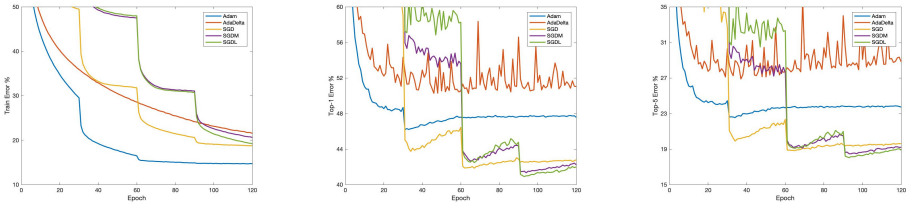


(a) Train and Test Error on CIFAR10. (b) Train and Test Error on CIFAR100.

Figure 3: Resnet56 performances on CIFAR10 and CIFAR100.

6.2 IMAGENET32

The second experiment is carried out on ImageNet32 dataset (Deng et al., 2009) with more than a million images in 1000 classes, out of which 50000 images are used as a testing set. Each image has 32×32 pixels. The objective is to train an image classifier. We apply a similar train strategy as on the CIFAR10 and CIFAR100 datasets in Section 6.1, except that the epoch budget is down to 120 and the learning rate is shrunk to one-tenth of its current value every 30 epochs.

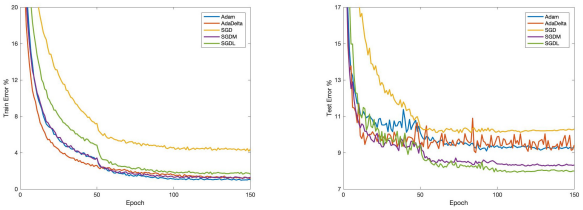


(a) Train Error. (b) Top-1 Error. (c) Top-5 Error.

Figure 4: Left: Train error for ResNet56 on ImageNet32; Middle: Top-1 error; Right: Top-5 error.

Our experiments is only focus on ResNet56 due to computational resource limitation. To adjust the difficulty on this dataset, we increase the model capacity for the previously used model by quadrupling the number of channels on each convolution layer. From the results on Fig.4, we find our method outperforms SGDM slightly about 0.5% for the best accuracy. More interestingly, SGD also has a good performance, while two adaptive methods both work just passably.

6.3 AUDIO RECOGNITION



(a) Train Error. (b) Test Error.

Figure 5: LeNet5 on Speech Commands Dataset.

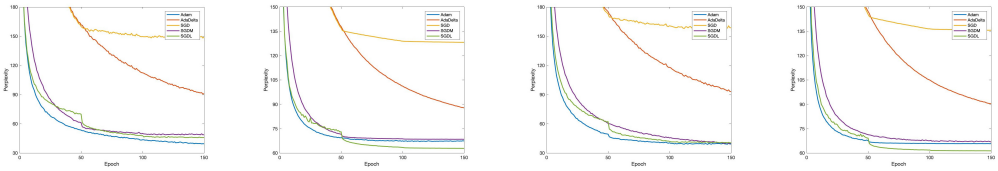
The third experiment is carried out on the Speech Commands Dataset, which consists of recordings from thousands of different people in uncontrolled recording conditions. Each sample is represented as a 16000-dimension vector. Each recording is one second in length. We divide the data into a training set and a testing set.

With the dataset, we train a 2-layered neural net with 20 channels, a 2D dropout layer and 2 full-connection layers with 1,000 hidden nodes. More specific hpyerparameter settings is in Ap-

pendix.A.5.2. From the results in Fig.5, we can see that SGDL has quite competitive effect and outperforms vanilla SGD exceedingly.

6.4 LSTM LANGUAGE MODEL

The fourth experiment is carried out on the standard language modeling task Penn Treebank (PTB) dataset. The learning rate is selected and optimized for several years and the current state-of-the-art results supported our perspective about large learning rate (Merity et al., 2017a). We focus to compare the effect of SGDL, SGDM and other optimizers on top of a state-of-the-art $\{2, 3\}$- LSTM training recipe with some training tricks (Merity et al., 2017b;a; Inan et al., 2016; Gal & Ghahramani, 2016). The details on LSTMs are in Appendix.A.5.3.



(a) Perplexity in 2-layer LSTM. (b) Perplexity in 3-layer LSTM.

Figure 6: LSTM on Penn Treebank Dataset.

From the result in Fig.6, we evaluate the performance by perplexity metric. We find that Adam is the fastest on the initial progress, but its final performance is worse than our SGDL. We also investigate AdaDelta has terrible performance, while SGDM has almost same curves with ADAM.

6.5 DISCUSSION

In Sections 6.1-6.4, we have observed that if vanilla SGD has the same learning rate as SGDM, it tends to a poor generalization performance in some tasks and this is one of the reasons why SGD is usually shelved for training models nowadays. Our experiments show that a larger initial learning rate increases the test accuracy on Fig.2. Further more, it can be seen that SGDL has an admirable performance in our experiments and outperforms SGDM in some tasks. For various deep neural networks such as ResNet,DenseNet,VGG, our SGDL works well, which is not affected by abundance layers or the existence of the residual mechanism. The experimental results have provided evidence that SGD with large learning rate is a good alternative to SGD with momentum in some practical tasks.

7 CONCLUSION

We provide a rigorous proof on the local convergence property for SGD-CS on smooth and non-convex function. Motivated by the numerical experiment in IGD, we recommend a vanilla SGD with large learning rate (SGDL) for training neural networks. Extensive evaluations have been carried out in deep learning tasks (CV,Speech,NLP) with popular neural network architectures. The results have demonstrated the smooth convergence and effective generalization performance of SGDL. For our future work, we will analyze the effect from more training strategies such as batch normalization about convergence. We will also explore the mutual effectiveness between the residual blocks and the convergence of object functions.

REFERENCES

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1195–1199, 2017.

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

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Volkan Cevher and Bang Cong Vu. On the linear convergence of the stochastic gradient method with constant step-size, 2018.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent, 2020.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pp. 1019–1027, 2016.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

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

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

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

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

Hao Li, Pratik Chaudhari, Hao Yang, Michael Lam, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Rethinking the hyperparameters for fine-tuning. *arXiv preprint arXiv:2002.11770*, 2020.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, pp. 11669–11680, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. In *International Conference on Machine Learning*, pp. 3325–3334, 2018.

Vien V Mai and Mikael Johansson. Convergence of a stochastic gradient method with momentum for nonsmooth nonconvex optimization. *arXiv preprint arXiv:2002.05466*, 2020.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017a.

Stephen Merity, Bryan McCann, and Richard Socher. Revisiting activation regularization for language rnns. *arXiv preprint arXiv:1708.01009*, 2017b.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Christopher De Sa. Syllabus for cs6787, lecture 2, 2017. URL `https://www.cs.cornell.edu/courses/cs6787/2017fa/`.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

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

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472. IEEE, 2017.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pp. 1100612. International Society for Optics and Photonics, 2019.

Mikhail V Solodov. Incremental gradient algorithms with stepsizes bounded away from zero. *Computational Optimization and Applications*, 11(1):23–35, 1998.

Yan Shuo Tan and Roman Vershynin. Phase retrieval via randomized kaczmarz: Theoretical guarantees. *Information and Inference: A Journal of the IMA*, 8(1):97–123, 2019.

Paul Tseng. An incremental gradient (-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

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

Yin Zhang, Richard Tapia, and Leticia Velazquez. On convergence of minimization methods: attraction, repulsion, and selection. *Journal of Optimization Theory and Applications*, 107(3):529–546, 2000.

A APPENDIX

A.1 STRONG GROWTH CONDITION

The strong growth condition (SGC) assumes that the norm of any noisy gradient $\nabla K(x,\xi)$ is uniformly bounded by a linear function of the average gradient $\mathbb{E}_\xi[\nabla K(x,\xi)]$:

$$\max_{\xi} \|\nabla K(x,\xi)\|^2 \leq B\|\mathbb{E}_\xi[\nabla K(x,\xi)]\|^2, \quad \text{for some } B > 0, \forall x \in \mathbb{R}^d,$$

and when $f(x)$ has a finite-sum structure, the SGC is equivalent to

$$\max_{1\leq i\leq N} \|\nabla f_i(x)\|^2 \leq B\|\nabla f(x)\|^2, \quad \text{for some } B > 0, \forall x \in \mathbb{R}^d.$$

Schmidt & Roux (2013) introduces SGD-CS attains *linear convergence rate* for *strongly convex* functions under the *strong growth condition* (SGC), and *sublinear convergence rate* for *convex* functions under the SGC, when f is defined from (1b). Cevher & Vu (2018) proposes the necessary condition for *linear convergence* of SGD-CS is the *growth condition* (GC) for f from (1a) , a weaker condition of SGC:

$$\mathbb{E}_{\xi_k}\|\nabla f_{\xi_k}(x_k)\|^2 \leq \frac{1}{1-\omega}\|\nabla f(x_k)\|^2, \quad \text{for some } \omega \in (0,1).$$

They also assume GC is the necessary and sufficient condition for linear convergence of SGD-CS for a restricted strongly convex f. For a general convex f, Ma et al. (2018) shows that SGD-CS attains linear convergence rate under the *interpolation* property, which can be viewed that the SGC is only satisfied at the minimizer x^*. For a finite sum f, the interpolation property means $\nabla f_i(x^*) = 0$ for $i = 1, \ldots, N$. However, the theoretical analysis of SGD-CS for non-convex functions is less as mature as in the convex case.

A.2 NOTATION

We denote the minimizer of object function f as x^* and use usual partial ordering for symmetric matrices: $A \succeq B$ means $A - B$ is positive semidefinite; similar for the relations $\preceq, \succ, \prec$. The norm $\|\cdot\|$ either denotes the Euclidean norm for vectors or Frobenius norm for matrices. The Hessian matrix of each component function $f_i(x)$ is denoted as $H_i(x)$. With slight abuse of notation, the Hessian matrix H_i at x^* is denoted as H_i^*. We also denote the mean and variance of $\{H_1^*, \ldots, H_N^*\}$ as $H^* \equiv \frac{1}{N}\sum_{i=1}^N H_i^*$ and $\Sigma^* \equiv \frac{1}{N}\sum_{i=1}^N (H_i^*)^2 - (H^*)^2$, and define $a \wedge b = \min(a,b)$. $X \xrightarrow{\mathbb{P}} Y$ means random variable X converges to random variable Y in probability, while $X \xrightarrow{L^1} Y$ means X converges to random variable Y in L^1 norm.

A.3 PROOFS FOR SECTION 4

Proof of Lemma 1. For $i = 1, \ldots, N$, by Lipschitz continuity of H_i at x^*, for any $y \in \mathcal{N}(x^*, \epsilon)$,

$$\|H_i(y) - H_i^*\| \leq L\|y - x^*\| < L\epsilon,$$

and therefore $\|H_i(y)\| \leq \|H_i(y) - H_i^*\| + \|H_i^*\| < L\epsilon + \|H_i^*\|$. For fixed $x \in \mathcal{N}(x^*, \epsilon)$, define $g_i(t) = \nabla f_i(x^* + t(x - x^*))$, and therefore $g_i'(t) = (x - x^*)^{\mathrm{T}} H_i(x^* + t(x - x^*))$. By the

fundamental theorem of calculus,

$$\begin{aligned}\|\nabla f_i(x) - \nabla f_i(x^*)\| = \|g_i(1) - g_i(0)\| &= \left\| \int_0^1 g_i'(t)\,\mathrm{d}t \right\| \\ &\le \int_0^1 \|g_i'(t)\|\,\mathrm{d}t = \int_0^1 \|(x-x^*)^{\mathrm{T}} H_i(x^* + t(x-x^*))\| \\ &\le \int_0^1 \|x-x^*\| \|H_i(x^* + t(x-x^*))\|\,\mathrm{d}t \\ &< \int_0^1 \|x-x^*\|(L\epsilon + \|H_i^*\|)\,\mathrm{d}t \\ &< L\epsilon^2 + \|H_i^*\|\epsilon.\end{aligned}$$

□

Proof of Theorem 1. Being a point of attraction, the sequence $\{x_k\}$ satisfies the Cauchy criteria:

$$\lim_{N\to\infty} \sup_{m,n\ge N} \mathbb{E}\|x_m - x_n\| = 0.$$

Then $\lim_{k\to\infty} \mathbb{E}\|x_{k+1} - x_k\| = 0$. Substituting x_{k+1} with the iteration (3),

$$\begin{aligned}0 &= \lim_{k\to\infty} \mathbb{E}\alpha \left\| \frac{1}{m} \sum_{j=1}^m \nabla f_{\xi_k^{(j)}}(x_k) \right\| \\ &= \alpha \lim_{k\to\infty} \mathbb{E}\left\{ \mathbb{E}\left[\left\| \frac{1}{m} \sum_{j=1}^m \nabla f_{\xi_k^{(j)}}(x_k) \right\| \middle| x_k \right] \right\} \\ &= \alpha \lim_{k\to\infty} \mathbb{E} \sum_{\ell_{1:m}\in\{1,\dots,N\}^m} \frac{1}{N^m} \cdot \left\| \frac{1}{m} \sum_{j=1}^m \nabla f_{\ell_j}(x_k) \right\|.\end{aligned}$$

Since $\|\frac{1}{m}\sum_{j=1}^m \nabla f_{\ell_j}(x_k)\| \ge 0$ for $\ell_{1:m} \in \{1,\dots,N\}^m$ and all iterates k,

$$\lim_{k\to\infty} \mathbb{E}\left\| \frac{1}{m} \sum_{j=1}^m \nabla f_{\ell_j}(x_k) \right\| = 0, \quad \text{for any } \ell_{1:m} \in \{1,\dots,N\}^m.$$

In particular, when $\ell_1 = \cdots = \ell_m = i$ for fixed $i = 1,\dots,N$,

$$\lim_{k\to\infty} \mathbb{E}\|\nabla f_i(x_k)\| = 0. \tag{12}$$

For fixed $i = 1,\dots,N$, the hyphothesis of the bounded convergence theorem (Durrett, 2019, Theorem 1.5.3) is satisfied:

- $\|\nabla f_i(x)\| \le L\epsilon^2 + \|H_i^*\|\epsilon$ for any $x \in \mathcal{N}(x^*, \epsilon)$, following the result in Lemma (1).
- $\|\nabla f_i(x_k)\| \xrightarrow{\mathbb{P}} \|\nabla f_i(x^*)\|$: $x_k \xrightarrow{L^1} x^*$ implies $\|\nabla f_i(x_k)\| \xrightarrow{L^1} \|\nabla f_i(x^*)\|$. By comparison theorem for convergence, $\|\nabla f_i(x_k)\| \xrightarrow{\mathbb{P}} \|\nabla f(x^*)\|$.

Applying the bounded convergence theorem on (12) over the region $\mathcal{N}(x^*, \epsilon)$ gives

$$\mathbb{E}\lim_{k\to\infty} \|\nabla f_i(x_k)\| = 0 \implies \mathbb{E}\|\nabla f_i(x^*)\| = \|\nabla f_i(x^*)\| = 0, \quad \text{for } i = 1,\dots,N,$$

which implies the desired result. □

Proof of Lemma 2. By the interpolation property and definition of $e_i(x)$ for $i = 1, \dots, N$,

$$\begin{aligned} e_i(x) &= -H_i^* \cdot (x - x^*) + \nabla f_i(x) \\ &= -H_i^* \cdot (x - x^*) + \nabla f_i(x^* + (x - x^*)) \\ &= -H_i^* \cdot (x - x^*) + [\nabla f_i(x^*) + H_i(x^* + t(x - x^*)) \cdot (x - x^*)] \quad \text{for some } t \in [0, 1] \qquad (13a) \\ &= [H_i(x^* + t(x - x^*)) - H_i^*] \cdot (x - x^*), \qquad (13b) \end{aligned}$$

where (13a) is by the Taylor expansion on $\nabla f_i(x)$, and (13b) is because $\nabla f_i(x^*) = 0$. By Cauchy-schwarz inequality,

$$\begin{aligned} \|e_i(x)\| &\le \|H_i(x^* + t(x - x^*)) - H_i^*\| \|x - x^*\| \\ &\le L\|t(x - x^*)\| \|x - x^*\| \qquad (13c) \\ &\le L\|x - x^*\|^2, \end{aligned}$$

where (13c) is by Lipschitz continuity of H_i at x^*. □

Proof of Theorem 2. Since the step size $\alpha \le \min_{i \in \{1,\dots,N\}} \frac{1}{\lambda_{\max}(H_i^*)}$, the term

$$0 \preceq I - \alpha H_i^* \preceq I, \quad \text{for } i = 1, \dots, N.$$

Substituting (5) into the term $\|x_{k+1} - x^*\|^2$ gives

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \left\| \frac{1}{m} \sum_{j=1}^{m} \left[(I - \alpha H_{\xi_k^{(j)}}^*)(x_k - x^*) - \alpha e_{\xi_k^{(j)}}(x_k) \right] \right\|^2 \\ &\le \max_{j \in \{1,\dots,m\}} (x_k - x^*)^{\mathrm{T}} (I - \alpha H_{\xi_k^{(j)}}^*)^2 (x_k - x^*) \\ &\qquad + \alpha^2 \|e_{\xi_k^{(j)}}(x_k)\|^2 - 2\alpha (x_k - x^*)^{\mathrm{T}} (I - \alpha H_{\xi_k^{(j)}}^*) e_{\xi_k^{(j)}}(x_k) \qquad (14a) \\ &\le \max_{j \in \{1,\dots,m\}} (x_k - x^*)^{\mathrm{T}} (I - \alpha H_{\xi_k^{(j)}}^*)^2 (x_k - x^*) \\ &\qquad + \alpha^2 \|e_{\xi_k^{(j)}}(x_k)\|^2 + 2\alpha \|x_k - x^*\| \|I - \alpha H_{\xi_k^{(j)}}^*\| \|e_{\xi_k^{(j)}}(x_k)\| \qquad (14b) \\ &\le \max_{j \in \{1,\dots,m\}} (x_k - x^*)^{\mathrm{T}} (I - \alpha H_{\xi_k^{(j)}}^*)^2 (x_k - x^*) \\ &\qquad + \alpha^2 L^2 \|x_k - x^*\|^4 + 2\alpha L \|I - \alpha H_{\xi^{(k)}}^*\| \|x_k - x^*\|^3 \qquad (14c) \\ &= \max_{j \in \{1,\dots,m\}} (x_k - x^*)^{\mathrm{T}} \\ &\qquad \left((I - \alpha H_{\xi_k^{(j)}}^*)^2 + \alpha^2 L^2 \|x_k - x^*\|^2 + 2\alpha L \|x_k - x^*\| \|I - \alpha H_{\xi_k^{(j)}}^*\| \right) (x_k - x^*) \\ &\le \max_{j \in \{1,\dots,m\}} (x_k - x^*)^{\mathrm{T}} \left((I - \alpha H_{\xi_k^{(j)}}^*)^2 + \alpha^2 L^2 \epsilon^2 + 2\alpha L \epsilon \right) (x_k - x^*) \qquad (14d) \end{aligned}$$

where (14a) follows from the inequality $\|\frac{1}{D} \sum_{i=1}^{d} a_i\|^2 \le \max_{i \in \{1,\dots,d\}} \|a_i\|^2$; (14b) follows from the Cauchy-Schwarz inequality on the last term; (14c) follows from Lemma (2); (14d) is because that the sequence $\{x_k\} \in \mathcal{N}(x^*, \epsilon)$, and that $\|I - \alpha H_{\xi^{(k)}}^*\| \le 1$.

Therefore, taking conditional expectation both sides given x_k implies

$$\begin{aligned} \mathbb{E}[\|x_{k+1} - x^*\|^2 \mid x_k] \le \max_{j \in \{1,\dots,m\}} \Bigg[&(x_k - x^*)^{\mathrm{T}} \\ &\left(\mathbb{E}_{\xi_k^{(j)}}[(I - \alpha H_{\xi^{(k)}}^*)^2] + \alpha^2 L^2 \epsilon^2 + 2\alpha L \epsilon \right) \qquad (15) \\ &(x_k - x^*) \Bigg] \end{aligned}$$

The expectation term $\mathbb{E}_{\xi_k^{(j)}}\{(I-\alpha H^*_{\xi^{(k)}})^2\}$ can be simplified as follows:

$$\begin{aligned}
\mathbb{E}_{\xi_k^{(j)}}[(I-\alpha H^*_{\xi_k^{(j)}})^2] &= \mathbb{E}_{\xi_k^{(j)}}\left[I-2\alpha H^*_{\xi_k^{(j)}}+\alpha^2(H^*_{\xi_k^{(j)}})^2\right] \\
&= I-2\alpha H^*+\alpha^2\mathbb{E}_{\xi_k^{(j)}}[(H^*_{\xi_k^{(j)}})^2] \\
&= I-2\alpha H^*+\alpha^2\frac{1}{N}\sum_{i=1}^{N}(H_i^*)^2 \\
&= I-2\alpha H^*+\alpha^2(H^*)^2+\alpha^2\left(\frac{1}{N}\sum_{i=1}^{N}(H_i^*)^2-(H^*)^2\right) \\
&= I-2\alpha H^*+\alpha^2(H^*)^2+\alpha^2\Sigma^*.
\end{aligned} \tag{16}$$

Combining (15) and (16) gives an upper bound on $\mathbb{E}[\|x_{k+1}-x^*\|^2 \mid x_k]$:

$$\mathbb{E}[\|x_{k+1}-x^*\|^2 \mid x_k] \le (x_k-x^*)^{\mathrm{T}}\left(I-2\alpha H^*+\alpha^2(H^*)^2+\alpha^2\Sigma^*+\alpha^2L^2\epsilon^2I+2\alpha L\epsilon I\right)(x_k-x^*)$$

Since α and ϵ are sufficiently small such that

$$0 \le \rho \equiv \lambda_{\max}\left(I-2\alpha H^*+\alpha^2(H^*)^2+\alpha^2\Sigma^*+\alpha^2L^2\epsilon^2I+2\alpha L\epsilon I\right) < 1,$$

the conditional expectation is further upper bounded by

$$\mathbb{E}[\|x_{k+1}-x^*\|^2 \mid x_k] \le \rho\|x_k-x^*\|^2.$$

Therefore,

$$\mathbb{E}[\|x_{k+1}-x^*\|^2] = \mathbb{E}\left\{\mathbb{E}[\|x_{k+1}-x^*\|^2 \mid x_k]\right\} \le \rho\mathbb{E}[\|x_k-x^*\|^2]. \tag{17}$$

Furtheremore,

$$\mathbb{E}[\|x^k-x^*\|^2] \le \rho\mathbb{E}[\|x^{k-1}-x^*\|^2] \le \cdots \le \rho^k\|x^0-x^*\|.$$

□

Proof of Remark 1. By (8a),

$$H^* \succ L\epsilon I. \tag{18}$$

By (8b),

$$I-2\alpha H^* \succeq 0. \tag{19}$$

Since $H^*-L\epsilon I \succ 0$ and by (8c),

$$\alpha((H^*)^2+\Sigma^*)+\alpha L^2\epsilon^2 I \prec 2H^*-2L\epsilon I. \tag{20}$$

Combining (19) and (20) gives

$$0 \preceq I-2\alpha H^*+\alpha^2(H^*)^2+\alpha^2\Sigma^*+\alpha^2L^2\epsilon^2I+2\alpha L\epsilon I \prec I.$$

□

Proof of Lemma 3. Let $\mathcal{F}_k$ denote the σ-algebra generated by the first k SGD-CS random vectors $\xi_1^{(1:m)},\ldots,\xi_k^{(1:m)}$. Construct $Z_k=\frac{\|X_{\tau\wedge k}-x^*\|^2}{\rho^{\tau\wedge k}}$. Firstly we show that Z_k is a supermatingale:

$$\begin{aligned}
\mathbb{E}[Z_{k+1} \mid \mathcal{F}_k] &= \mathbb{E}\left[\frac{\|X_{\tau\wedge(k+1)}-x^*\|^2}{\rho^{\tau\wedge(k+1)}}1_{\tau\le k}\middle|\mathcal{F}_k\right]+\mathbb{E}\left[\frac{\|X_{\tau\wedge(k+1)}-x^*\|^2}{\rho^{\tau\wedge(k+1)}}1_{\tau>k}\middle|\mathcal{F}_k\right] \\
&= \mathbb{E}\left[\frac{\|X_{\tau\wedge k}-x^*\|^2}{\rho^{\tau\wedge k}}1_{\tau\le k}\middle|\mathcal{F}_k\right]+\mathbb{E}\left[\frac{\|X_{k+1}-x^*\|^2}{\rho^{k+1}}1_{\tau>k}\middle|\mathcal{F}_k\right] \\
&\overset{\text{(i)}}{=} Z_k1_{\tau\le k}+\frac{1}{\rho^{k+1}}\mathbb{E}\left[\|X_{k+1}-x^*\|^21_{\tau>k}\middle|\mathcal{F}_k\right] \\
&\overset{\text{(ii)}}{\le} Z_k1_{\tau\le k}+\frac{1}{\rho^{k+1}}\rho\mathbb{E}\left[\|X_k-x^*\|^21_{\tau>k}\middle|\mathcal{F}_k\right] \\
&= Z_k1_{\tau\le k}+Z_k1_{\tau>k}=Z_k,
\end{aligned}$$

where (i) is because that $\frac{\|X_{\tau\wedge k}-x^*\|^2}{\rho^{\tau\wedge k}}1_{\tau\le k}$ is measurable with respect to $\mathcal{F}_k$; (ii) is by applying (17) in Theorem (2). As a result,

$$Z_0 \ge \mathbb{E}[Z_k \mid \mathcal{F}_0] \ge \mathbb{E}\left[\frac{\|X_{\tau\wedge k}-x^*\|^2}{\rho^{\tau\wedge k}}1_{k\ge\tau}\middle|\mathcal{F}_0\right] \ge \mathbb{E}\left[\frac{\|X_{\tau}-x^*\|^2}{\rho^{\tau}}1_{k\ge\tau}\middle|\mathcal{F}_0\right].$$

By definition of the stopping time, $\|X_\tau - x^*\|^2 \ge \epsilon^2$; and the term $Z_0 := \|x_0 - x^*\|^2 \le \delta\epsilon^2$ since $\|x_0 - x^*\| \le \sqrt{\delta}\epsilon$. Substituting these two relations into the inequality above gives

$$\delta\epsilon^2 \ge \mathbb{E}\left[\frac{\epsilon^2}{\rho^\tau}1_{k\ge\tau} \mid \mathcal{F}_0\right] \implies \delta \ge \mathbb{E}\left[\frac{1_{k\ge\tau}}{\rho^\tau}\middle|\mathcal{F}_0\right] \ge \mathbb{E}\left[1_{k\ge\tau}|\mathcal{F}_0\right] \ge P\{k\ge\tau\},\ \forall k.$$

Or after a rearrangement, $\delta \ge P(\tau < \infty)$, which implies $P(\tau = \infty) \ge 1 - \delta$.

Furthermore,

$$\mathbb{E}[\|X_k - x^*\|^2 1_{\tau=\infty}] = \mathbb{E}[\|X_k - x^*\|^2 \mid \tau = \infty]P(\tau = \infty) \ge (1-\delta)\mathbb{E}[\|X_k - x^*\|^2 \mid \tau = \infty]. \tag{21}$$

Apply Theorem (2) to bound $\mathbb{E}[\|X_k - x^*\|^2 1_{\tau=\infty}]$:

$$\mathbb{E}[\|X_k - x^*\|^2 1_{\tau=\infty}] \le \rho^k \|x_0 - x^*\|^2. \tag{22}$$

We now estimate $\mathbb{E}[\|X_k - x^*\|^2 \mid \tau = \infty]$ by utilizing (21) and (22):

$$\mathbb{E}[\|X_k - x^*\|^2 \mid \tau = \infty] \le \frac{\rho^k}{1-\delta}\|x_0 - x^*\|^2.$$

□

Proof of Theorem 3. Define the event $E = \{\tau = \infty\}$. It suffices to show the relation (9). By direct calculation,

$$\begin{aligned}
\mathbb{E}[\|X_{k+1} - x^*\|_2^2 1_{\tau>k+1} \mid X_k = x_k] &\le \mathbb{E}[\|X_{k+1} - x^*\|_2^2 1_{\tau>k} \mid X_k = x_k] \\
&= \mathbb{E}[\|X_{k+1} - x^*\|_2^2 1_{\tau>k} \mid X_k = x_k, \mathcal{F}_k] \\
&= \mathbb{E}[\|X_{k+1} - x^*\|_2^2 \mid X_k = x_k, \mathcal{F}_k] 1_{\tau>k} \\
&\le \rho\|x_k - x^*\|^2 1_{\tau>k}
\end{aligned}$$

where the last inequality follows from (17). As a result,

$$\begin{aligned}
\mathbb{E}[\|X_{k+1} - x^*\|_2^2 1_{\tau>k+1}] &= \mathbb{E}\left\{\mathbb{E}[\|X_{k+1} - x^*\|_2^2 1_{\tau>k+1} \mid X_k]\right\} \\
&\le \rho\mathbb{E}[\|X_k - x^*\|^2 1_{\tau>k}]
\end{aligned}$$

Inductively, $\mathbb{E}[\|X_k - x^*\|^2 1_{\tau>k}] \le \rho^k\|x_0 - x^*\|^2$. Therefore,

$$\mathbb{E}[\|X_k - x^*\|^2 1_E] = \mathbb{E}[\|X_k - x^*\|^2 1_{\tau=\infty}] \le \mathbb{E}[\|X_k - x^*\|^2 1_{\tau>k}] \le \rho^k\|x_0 - x^*\|^2,$$

which completes the proof. □

Proof of Remark 2. By Markov's inequality,

$$P(\|X_k - x^*\|^2 > \epsilon^2 \mid \tau = \infty) \le \frac{\mathbb{E}[\|X_k - x^*\|^2 \mid \tau = \infty]}{\epsilon^2} \le \frac{\rho^k}{(1-\delta)\epsilon^2}\|x_0 - x^*\|^2.$$

It follows that

$$\begin{aligned}
P(\|X_k - x^*\|^2 \le \epsilon^2) &\ge P(\|X_k - x^*\|^2 \le \epsilon^2 \mid \tau = \infty)P(\tau = \infty) \\
&\ge \left[1 - \frac{\rho^k}{(1-\delta)\epsilon^2}\|x_0 - x^*\|^2\right](1-\delta) \\
&= 1 - \delta - \frac{\rho^k}{\epsilon^2}\|x_0 - x^*\|^2 \\
&\ge 1 - \delta(1 + \rho^k).
\end{aligned}$$

□

A.4 EXAMPLES

Example 1. ***(Quadratic Functions)*** *Consider minimizing for the objective function $f(x) = \sum_{i=1}^{N} x_i^2 \equiv x^{\mathrm{T}}x$, where $x \in \mathbb{R}^{N\times 1}$, and $f_i = x_i^2, i \in [N]$. Then the gradient of each component f_i is given by*

$$\nabla f_i(x) = 2E_i \cdot x,$$

where $E_i \in \mathbb{R}^{N\times N}$ is a all-zero matrix except $E_{i,i} = 1$. Then the end-to-end N runs of IGD update with step size t can be expressed as a matrix compact form:

$$x^{new} = K_N(t)x^{old},$$

where

$$K_N(t) = \prod_{i=1}^{N}\left(I - 2tE_i\right) = \mathrm{diag}(1-2t, 1-2t, \ldots, 1-2t). \tag{23}$$

Applying the basic knowledge in linear algegra, $\rho(K_N(t)) = |1-2t|$. As long as $t \in (0,1)$, the IGD will converge from any initial point.

Example 2. ***(Standard Least Squares Problem)*** *Consider the standard un-determined least squares problem*

$$f(x) = \frac{1}{2}\|Ax-b\|^2, \quad A \in \mathbb{R}^{m\times N}, N < n.$$

Similarly, the end-to-end N runs of IGD update with step size t forms the linear system

$$x^{new} = K_N(t)x^{old} + tc(t),$$

with

$$K_0(t) \equiv I, \quad K_j(t) = \prod_{i=1}^{j}\left(I - ta_ia_i^{\mathrm{T}}\right), \quad j = 1, 2, \ldots, N,$$

and $c(t) = \sum_{j=1}^{N} b_jK_{j-1}(t)a_j$. The sufficient condition for convergence is $\lambda_{\max}(K_N(t)) < 1$, and the necessary condition is $\lambda_{\max}(K_N(t)) \le 1$. In this situation, we can assert that $\lambda_{\max}(K_N(t)) \ge 1$: we can pick $x_0 \in \mathbb{R}^n \setminus \{0\}$ so that $Ax_0 = 0$, which implies $(K_N(t))x_0 = x_0$. This means that 1 is an eigenvalue of $K_N(t)$, i.e., $\lambda_{\max}(K_N(t)) \ge 1$. This example also suggests that regularization helps with the convergence in optimization.

Example 3. ***(Regularized Least Squares Problem)*** *Consider the regularized least squares objective function*

$$f(x) = \frac{1}{2}\|Ax-b\|^2 + \frac{\mu}{2}\|D^{1/2}x\|^2.$$

It can be shown that the end-to-end N iteration of IGD with step size t can be expressed as a compact matrix form:

$$x^{new} = K_N(t)x^{old} + tc(t), \tag{24a}$$

$$K_0(t) \equiv I, \quad K_j(t) = \prod_{i=1}^{j}\left(I - ta_ia_i^{\mathrm{T}} - t\frac{\mu}{m}D\right), \quad j = 1, \ldots, N, \tag{24b}$$

$$c(t) = \sum_{j=1}^{m} b_jK_{j-1}(t)a_j. \tag{24c}$$

The necessary and sufficient conditions for the convergence of IGD suffice to characterize the condition $\rho(t) \equiv \rho(K_N(t)) < 1$. We need to make use of the famous conjecture about matrix AM-GM inequality:

For any positive semi-definite matrix $P_1, P_2, \ldots, P_n \in \mathbb{R}^{n\times n}$,

$$\frac{1}{n!}\sum_{\sigma=(\sigma_1,\ldots,\sigma_n)\in\Gamma} P_{\sigma_1}P_{\sigma_2}\cdots P_{\sigma_n} \preceq \left(\frac{1}{n}\sum_i A_i\right)^n.$$

We make a reasonable assumption that the products within $K_N(t)$ are commutative, then applying this conjecture gives

$$K_N(t) \preceq \left[\frac{1}{N}\sum_{i=1}^{N}\left(I - ta_i a_i^{\mathrm{T}} - t\frac{\mu}{N}D\right)\right]^N = \left[I - \frac{t}{N}[\mu D + A^{\mathrm{T}}A]\right]^N$$

Hence, a sufficient condition for the convergence would be:

$$\rho\left(\left[I - \frac{t}{N}[\mu D + A^{\mathrm{T}}A]\right]^N\right) < 1.$$

Or equivalently, we need to pick t such that, for any eigenvalue λ of the matrix $\mu D + A^{\mathrm{T}}A$, we have

$$\left(1 - \frac{t\lambda}{N}\right)^N < 1 \Longleftrightarrow t \in \left(0, \frac{2N}{\rho(\mu D + A^{\mathrm{T}}A)}\right).$$

Example 4. ***(General Nonlinear Least Squares Problem)*** *Consider the general nonlinear least squares problem:*

$$f(x) = \frac{1}{2}\sum_{i=1}^{N} r_i^2(x) = \frac{1}{2}R(x)^T R(x).$$

By assuming that $r_i(x)$ are twice continuously differentiable for all $i \in [N]$, the end-to-end N iteration of IGD with step size t can be safely approximated as

$$x^{new} = K_N(t)x^{old} + tc(t) + o(\|x^0 - x^*\|^2), \tag{25a}$$

$$K_0(t) \equiv I, \quad K_j(t) = \prod_{i=1}^{j}\left(I - tH_i(x^*)\right), \quad j = 1, \ldots, N, \tag{25b}$$

$$c(t) = \sum_{j=1}^{m} K_{j-1}(t)H_j(x^*)x^*. \tag{25c}$$

It is reasonable to assume that $H_i(x^*)$ is full rank for all i, since adding small regularization terms can resolve the rank deficiency issue. Applying the conjecture again, we imply that

$$K_N(t) \preceq \left[I - \frac{t}{N}\sum_i H_i(x^*)\right]^N$$

Therefore, a sufficient condition for the convergence of IGD would be

$$t \in \left(0, \frac{1}{\lambda_{\max}(\mathbb{E}[H_i(x^*)])}\right), \quad \textit{where } \mathbb{E}[H_i(x^*)] \equiv \frac{1}{N}\sum_i H_i(x^*).$$

A.5 Supplement to Exeperiements

A.5.1 Image classification

We normalize data and then augment them by horizontal flips and random crops from the image padded by 4 pixels, filling missing pixels with reflections of the original image. We adopt Kaiming initialization(He et al., 2015), BN (Ioffe & Szegedy, 2015) but without dropout. The models are trained for up to 150 epochs and minibatch size to 256. We select the best learning rate from Table.1 in Appendix.A.5.1 for SGD(M), and for SGDL, the learning rate is more than 10 times larger than the best SGDM lr, in our paper, we select it from $\{1.0, 1.1, 1.2, 1.3\}$, more specific parameter settings are as follows. Additional, for all optimizers without AdaDelta, we use an annealing strategy to divide learning rate by 10 on 50 and 100 epochs.

The figures show results for ResNet110,VGG,DenseNet on CIFAR10 and CIFAR100. For each figure, the specific learning rate is given for SGDL. On **CIFAR10**, SGDL learning rate is 1.0 for ResNet 56, 1.1 for DenseNet and 1.2 for ResNet110 and VGG16, while on **CIFAR100**, learning rate is 1.0 for VGG16 and 1.3 for ResNet 56, ResNet110 and DenseNet on SGDL. For SGDM, the learning rate is 0.1 for most experiments, while it is 0.05 for VGG16 on CIFAR10. Finally, 0.001 is the most suitable learning rate for ADAM. On **ImageNet**, we use 1.1, 0.1, 0.001 as learning rate for SGDL, SGDM, and ADAM.

Table 1: Popular optimizers in the paper on CIFAR10, CIFAR100, ImageNet32 and Speech Commands Dataset. η denotes momentum, γ is the weight decay coefficient,β is from Kingma & Ba (2014) and ρ is from Zeiler (2012).

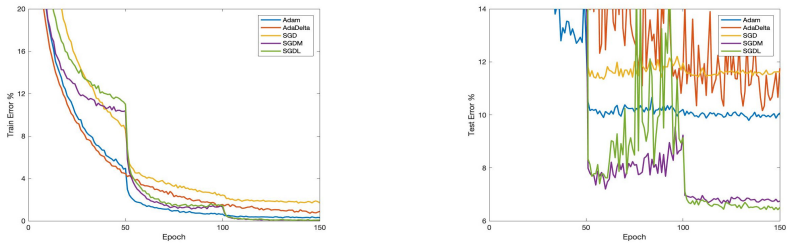
Optimizer	Learning rate	Other Parameter Tuning
Adam	$\{0.0001, 0.001, 0.01\}$	$\beta = (0.9, 0.999)$
AdaDelta	1.0	$\rho = 0.9$
SGD	$\{0.001, 0.01, 0.05, 0.1\}$	$\gamma = 0.005$
SGDM	$\{0.001, 0.01, 0.05, 0.1\}$	$\eta = 0.9, \gamma = 0.005$
SGDL	$\geq 10 * \mathrm{lr}(\mathrm{SGDM})$	$\gamma = 0.005$

A.5.2 SPEECH COMMANDS DATASET

On Speech Commands Dataset, Similarly, the learning rate is 0.001 for Adam and 1.0 for AdaDelta as above. We choose 0.01 from $\{0.001, 0.01, 0.05, 0.1\}$ as the best for SGDM and 0.15 for SGDL 15 times larger than SGDM.

A.5.3 DETAILS ON LSTM

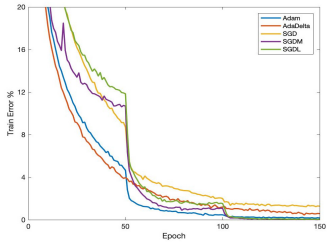
For LSTMs, there are 1150 units in the hidden layer, an embedding of 400 and a batch size of 20. We select the best learning rate 1 from $\{0.001, 0.01, 0.1, 1\}$ for SGD(M) and the learning rate is same as above sections for Adam and Adadelta. For SGDL, we use 30 as an initial learning rate from the author's advise(Merity et al., 2017a) on $\{2, 3\}$-Layer LSTM for 150 epochs with the same annealing mechanism similarly as above. To train all models, we carry out gradient clipping with maximum norm 0.25. More specifically, for those dropout values, we use $(0.4, 0.3, 0.4, 0.1, 0.5)$ on word vector, output between LSTM layers, output of final LSTM layer, embedded dropout and DropConnect respectively.



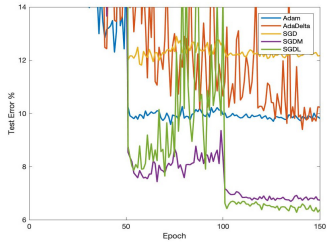
(a) Train Error for Resnet56.

(b) Test Error for Resnet56.

Figure 7: Resnet56 on CIFAR-10.

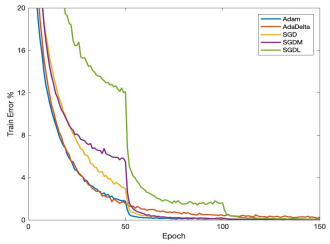


(a) Train Error for Resnet110.

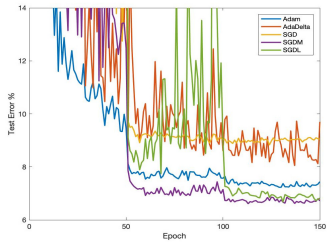


(b) Test Error for Resnet110.

Figure 8: Resnet110 on CIFAR-10.

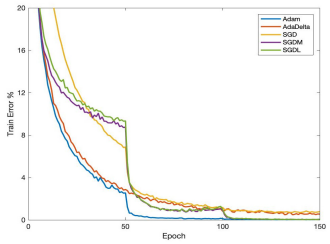


(a) Train Error for VGG16.

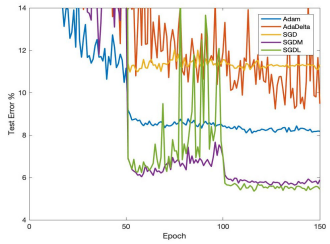


(b) Test Error for VGG16.

Figure 9: VGG16 on CIFAR-10.

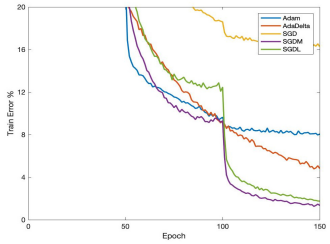


(a) Train Error for DenseNet.

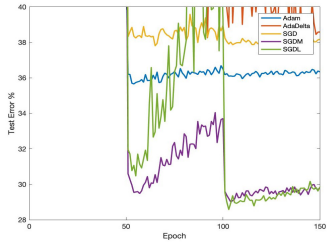


(b) Test Error for DenseNet.

Figure 10: DenseNet on CIFAR10.

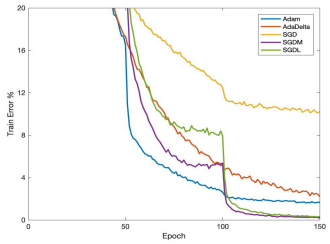


(a) Train Error for Resnet56.

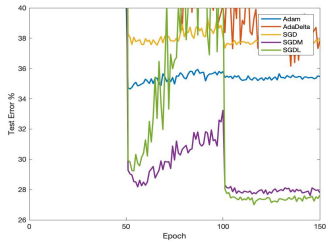


(b) Test Error for Resnet56.

Figure 11: Resnet56 on CIFAR100.

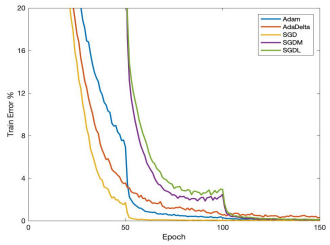


(a) Train Error for Resnet110.

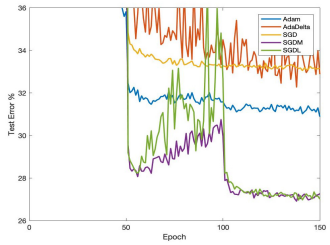


(b) Test Error for Resnet110.

Figure 12: Resnet110 on CIFAR100.

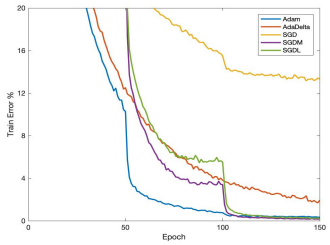


(a) Train Error for Resnet110.

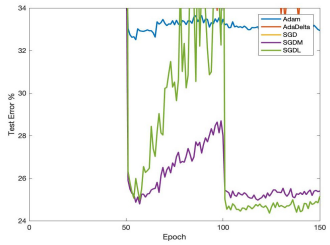


(b) Test Error for Resnet110.

Figure 13: VGG16 on CIFAR100.



(a) Train Error for DenseNet.



(b) Test Error for DenseNet.

Figure 14: DenseNet on CIFAR100.