Dual Averaging is Surprisingly Effective for Deep Learning Optimization

Anonymous authors
Paper under double-blind review

Abstract

First-order stochastic optimization methods are currently the most widely used class of methods for training deep neural networks. However, the choice of the optimizer has become an ad-hoc rule that can significantly affect the performance. For instance, SGD with momentum (SGD+M) is typically used in computer vision (CV) and Adam is used for training transformer models for Natural Language Processing (NLP). Using the wrong method can lead to significant performance degradation. Inspired by the dual averaging algorithm, we propose Modernized Dual Averaging (MDA), an optimizer that is able to perform as well as SGD+M in CV and as Adam in NLP. Our method is not adaptive and is significantly simpler than Adam. We show that MDA induces a decaying uncentered L_2-regularization compared to vanilla SGD+M and hypothesize that this may explain why it works on NLP problems where SGD+M fails.

1 Introduction

Stochastic first-order optimization methods have been extensively employed for training neural networks. It has been empirically observed that the choice of the optimization algorithm is crucial for obtaining a good accuracy score. For instance, stochastic variance-reduced methods perform poorly in computer vision (CV) (Defazio & Bottou, 2019). On the other hand, SGD with momentum (SGD+M) (Bottou, 1991; LeCun et al., 1998; Bottou & Bousquet, 2008) works particularly well on CV tasks and Adam (Kingma & Ba, 2014) out-performs other methods on natural language processing (NLP) tasks (Choi et al., 2019). In general, the choice of optimizer, as well as its hyper-parameters, must be included among the set of hyper-parameters that are searched over when tuning.

In this work we propose *Modernized Dual Averaging* (MDA), an optimizer that matches the performance of SGD+M on CV tasks and Adam on NLP tasks, providing the best result in both worlds. Dual averaging (Nesterov, 2009) and its variants have been heavily explored in the convex optimization setting. Our *modernized* version updates dual averaging with a number of changes that make it effective for non-convex problems. Compared to other methods, dual averaging has the advantage of accumulating new gradients with *non-vanishing* weights. Moreover, it has been very successful for regularized learning problems due to its ability to obtain desirable properties (e.g. a sparse solution in Lasso) faster than SGD (Xiao, 2010).

In this paper, we point out another advantage of dual averaging compared to SGD. As we show in Section 2, under the right parametrization, dual averaging is equivalent to SGD applied to the same objective function but with a *decaying ℓ_2-regularization*. This induced ℓ_2-regularization has two primary implications for neural network training. Firstly, from an optimization viewpoint, regularization smooths the optimization landscape, aiding optimization. From a learning viewpoint, ℓ_2-regularization (often referenced as weight decay) is crucial for generalization performance (Krogh & Hertz, 1992; Bos & Chug, 1996; Wei et al., 2019). Through an empirical investigation, we demonstrate that this implicit regularization effect is beneficial as MDA outperforms SGD+M in settings where the latter perform poorly.

CONTRIBUTIONS

This paper introduces MDA, an algorithm that matches the performance of the best first-order methods in a wide range of settings. More precisely, our contributions can be divided as follows:

– *Adapting dual averaging to neural network training*: We build on the subgradient method with double averaging (Nesterov & Shikhman, 2015) and adapt it to deep learning optimization. In particular, we specialize the method to the L_2-metric, modify the hyperparameters and design a proper scheduling of the parameters.

– *Theoretical analysis in the non-convex setting*: Leveraging a connection between SGD and dual averaging, we derive a convergence analysis for MDA in the non-convex and smooth optimization setting. This analysis is the first convergence proof for a dual averaging algorithm in the non-convex case.

– *MDA matches the performance of the best first-order methods*: We investigate the effectiveness of dual averaging in CV and NLP tasks. For supervised classification, we match the test accuracy of SGD+M on CIFAR-10 and ImageNet. For image-to-image tasks, we match the performance of Adam on MRI reconstruction on the fastMRI challenge problem. For NLP tasks, we match the performance of Adam on machine translation on IWSLT'14 De-En, language modeling on Wikitext-103 and masked language modeling on the concatenation of BookCorpus and English Wikipedia.

RELATED WORK

First-order methods in deep learning. While SGD and Adam are the most popular methods, a wide variety of optimization algorithms have been applied to the training of neural networks. Variants of SGD such as momentum methods and Nesterov's accelerated gradient improve the training performance (Sutskever et al., 2013). Adaptive methods as Adagrad (Duchi et al., 2011a), RMSprop (Hinton et al., 2012) and Adam have been shown to find solutions that generalize worse than those found by non-adaptive methods on several state-of-the-art deep learning models (Wilson et al., 2017). Berrada et al. (2018) adapted the Frank-Wolfe algorithm to design an optimization method that offers good generalization performance while requiring minimal hyper-parameter tuning compared to SGD.

Dual averaging. Dual averaging is one of the most popular algorithms in convex optimization and presents two main advantages. In regularized learning problems, it is known to more efficiently obtain the desired regularization effects compared to other methods as SGD (Xiao, 2010). Moreover, dual averaging fits the distributed optimization setting (Duchi et al., 2011b; Tsianos et al., 2012; Hosseini et al., 2013; Shahrampour & Jadbabaie, 2013; Colin et al., 2016). Finally, this method seems to be effective in manifold identification (Lee & Wright, 2012; Duchi & Ruan, 2016). Our approach differs from these works as we study dual averaging in the non-convex optimization setting.

Convergence guarantees in non-convex optimization. While obtaining a convergence rate for SGD when the objective is smooth is standard (see e.g. Bottou et al. (2016)), it is more difficult to analyze other algorithms in this setting. Recently, Zou et al. (2018); Ward et al. (2019); Li & Orabona (2019) provided rates for the convergence of variants of Adagrad towards a stationary point. Défossez et al. (2020) builds on the techniques introduced in Ward et al. (2019) to derive a convergence rate for Adam. In the non-smooth weakly convex setting, Davis & Drusvyatskiy (2019) provides a convergence analysis for SGD and Zhang & He (2018) for Stochastic Mirror Descent. Our analysis for dual averaging builds upon the recent analysis of SGD+M by Defazio (2020).

Decaying regularization Methods that reduce the amount of regularization used over time have been explored in the convex case. Allen-Zhu & Hazan (2016) show that it's possible to use methods designed for strongly convex optimization to obtain optimal rates for other classes of convex functions by using polynomially decaying regularization with a restarting scheme. In Allen-Zhu (2018), it is shown that adding regularization centered around a sequence of points encountered during optimization, rather than the zero vector, results in better convergence in terms of gradient norm on convex problems.

2 Modernizing dual averaging

As we are primarily interested in neural network training, we focus in this work on the unconstrained stochastic minimization problem

$$\min_{x \in \mathbb{R}^n} \mathbb{E}_{\xi \sim P}[f(x, \xi)] := f(x). \tag{1}$$

We assume that ξ can be sampled from a fixed but unknown probability distribution P. Typically, $f(x, \xi)$ evaluates the loss of the decision rule parameterized by x on a data point ξ. Finally, $f \colon \mathbb{R}^n \to \mathbb{R}$ is a (potentially) non-convex function.

Dual averaging. To solve (1), we are interested in the dual averaging algorithm (Nesterov, 2009). In general, this scheme is based on a mirror map $\Phi \colon \mathbb{R}^n \to \mathbb{R}$ assumed to be strongly convex. An exhaustive list of popular mirror maps is present in Bregman (1967); Teboulle (1992); Eckstein (1993); Bauschke et al. (1997). In this paper, we focus on the particular choice

$$\Phi(x) := \frac{1}{2}\|x - x_0\|^2, \tag{2}$$

where $x_0 \in \mathbb{R}^n$ is the initial point of the algorithm.

Algorithm 1 (Stochastic) dual averaging

Input: initial point x_0, scaling parameter sequence $\{\beta_k\}_{k=1}^T$, step-size sequence $\{\lambda_k\}_{k=1}^T$, stopping time T.
for $k = 0 \ldots T$ **do**
    Sample $\xi_k \sim P$ and compute stochastic gradient $g_k = \nabla f(x_k, \xi_k)$
    $s_k = s_{k-1} + \lambda_k g_k$.   // Update the sum of gradients
    $x_{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \langle s_k, x \rangle + \frac{\beta_k}{2}\|x - x_0\|_2^2 \right\}$ // Update the iterate
end for
return $\hat{x}_T = \frac{1}{T+1} \sum_{k=0}^T x_k$.

Dual averaging generates a sequence of iterates $\{x_k, s_k\}_{k=0}^T$ as detailed in Algorithm 1. At time step k of the algorithm, the algorithm receives g_k and updates the sum of the weighted gradients s_k. Lastly, it updates the next iterate x_{k+1} according to a proximal step. Intuitively, x_{k+1} is chosen to minimize an averaged first-order approximation to the function f, while the regularization term $\beta_k \Phi(x)$ prevents the sequence $\{x_k\}_{k=0}^T$ from oscillating too wildly. The sequence $\{\beta_k\}_{k=1}^T$ is chosen to be non-decreasing in order to counter-balance the growing influence of $\langle s_k, x \rangle$. We remark that the update in Algorithm 1 can be rewritten as:

$$x_{k+1} = -s_k/\beta_k. \tag{3}$$

In the convex setting, Nesterov (2009) chooses $\beta_{k+1} = \beta_k + 1/\beta_k$ and $\lambda_k = 1$ and shows convergence of the average iterate $\hat{x}_T$ to the optimal solution at a rate of $O(1/\sqrt{T})$. That sequence of β values grows proportionally to the square-root of k, resulting in a method which an effective step size that decays at a rate $O(1/\sqrt{T})$. This rate is typical of decaying step size sequences used in first order stochastic optimization methods when no strong-convexity is present.

Connection with SGD. With our choice of mirror map (2), stochastic mirror descent (SMD) is equivalent to SGD whose update is

$$x_{k+1} = x_k - \eta_k g_k. \tag{4}$$

Dual averaging and SMD share similarities. While in constrained optimization the two algorithms are different Juditsky et al. (2019), they yield the same update in the unconstrained case when $\lambda_k = \eta_k$ and $\beta_k = 1$. In this paper, we propose another viewpoint on the relationship between the two algorithms.

Proposition 2.1. *Let $f \colon \mathbb{R}^n \to \mathbb{R}$ be a function and let $T > 0$. Let $\{h^{(k)}\}_{k=0}^T$ be a sequence of functions such that $h^{(k)} \colon \mathbb{R}^n \to \mathbb{R}$ and*

$$h^{(k)}(x) = f(x) + \frac{\alpha_k}{2}\|x - x_0\|^2, \tag{5}$$

where $\{\alpha_k\}_{k=0}^T$ is a sequence in $\mathbb{R}$. Then, for $k \in \{1, \dots, T\}$, the update of dual averaging at iteration k for the minimization problem on f is equivalent to the one of SGD for the minimization problem on $h^{(k)}$ when

$$\eta_k = \frac{\lambda_k}{\beta_k} \quad and \quad \alpha_k = \frac{\beta_k - \beta_{k-1}}{\lambda_k}. \tag{6}$$

Proof of Proposition 2.1. We start by deriving the SGD update for $h^{(k)}$.

$$x_{k+1} = x_k - \eta_k g_k - \eta_k \alpha_k x_k. \tag{7}$$

We now rewrite the update of dual averaging. By evaluating (3) at iterations k and $k-1$, we obtain:

$$\begin{cases} x_{k+1} = -s_k/\beta_k \\ x_k = -s_{k-1}/\beta_{k-1} \end{cases} \implies x_{k+1} = x_k - \frac{\lambda_k}{\beta_k}g_k - \left(1 - \frac{\beta_{k-1}}{\beta_k}\right)x_k. \tag{8}$$

By comparing (7) and (8), we obtain (6). $\qquad\qquad\square$

Proposition 2.1 shows that dual averaging implicitly induces a time-varying L_2-regularization to an SGD update.

Modernized dual averaging. The *modernized dual averaging* (MDA) algorithm, our adaptation of dual averaging for deep learning optimization, is given in Algorithm 2.

Algorithm 2 Modernized dual averaging (MDA)

Input: $x_0 \in \mathbb{R}^n$ initial point, $\eta_k > 0$ stepsize sequence, c_k momentum parameter sequence, $T > 0$ stopping time.
Initialize $s_{-1} = 0$.
for $k = 0 \dots T-1$ **do**
    Set the scaling coefficient $\beta_k = \sqrt{k+1}$ and the stepsize $\lambda_k = \eta_k\sqrt{k+1}$.
    Sample $\xi_k \sim P$ and compute stochastic gradient $g_k = \nabla f(x_k, \xi_k)$.
    $s_k = s_{k-1} + \lambda_k g_k$ // Update the sum of gradients
    $z_{k+1} = x_0 - s_k/\beta_k$ // Update the dual averaging iterate
    $x_{k+1} = (1 - c_{k+1})x_k + c_{k+1}z_{k+1}$ // Update the averaged iterate
end for
return x_T.

MDA differs from dual averaging in the following fundamental ways. Firstly, it maintains an iterate x_{k+1} obtained as a weighted average of the previous average x_k and the current dual averaging iterate z_{k+1}. It has been recently noticed that this averaging step can be interpreted as introducing momentum in the algorithm (Sebbouh et al., 2020; Tao et al., 2018) (more details in Appendix A). For this reason, we will refer to c_k as the momentum parameter. While dual averaging with double averaging has already been introduced in Nesterov & Shikhman (2015), it is our choices of parameters that make it suitable for non-convex deep learning objectives. In particular, our choice of β_k and λ_k, motivated by a careful analysis of our theoretical convergence rate bounds, result in the following adaptive regularization when viewed as regularized SGD with momentum:

$$\alpha_k = \frac{\sqrt{k+2} - \sqrt{k+1}}{\eta_k\sqrt{k+2}} \approx \frac{1}{k+2} \tag{9}$$

In practice, a schedule of the momentum parameter (in our case c_k) and learning rate (η_k) must also be chosen to get the best performance out of the method. We found that the schedules used for SGD or Adam can be adapted for MDA with small modifications. For CV tasks, we found it was most effective to use modifications of existing stage-wise schemes where instead of a sudden decrease at

the end of each stage, the learning rate decreases linearly to the next stages value, over the course of a few epochs. For NLP problems, a warmup stage is necessary following the same schedule typically used for Adam. Linear decay, rather than inverse-sqrt schedules, were the most effective post-warmup.

For the momentum parameter, in each case the initial value can be chosen to match the momentum β used for other methods, with the mapping $c_k = 1 - \beta$. Our theory suggests that when the learning rate is decreased, c_k should be increased proportionally (up to a maximum of 1) so we used this rule in our experiments, however it doesn't make a large difference.

3 CONVERGENCE ANALYSIS

Our analysis requires the following assumptions. We assume that f has Lipschitz gradients but in not necessarily convex. Similarly to Robbins & Monro (1951), we assume unbiasedness of the gradient estimate and boundedness of the variance.

Assumption 1 (Stochastic gradient oracle). *We make the two following assumptions.*
(A1) Unbiased oracle: $\mathbb{E}_{\xi \sim P}[\nabla f(x, \xi)] = \nabla f(x)$.
(A2) Bounded second moment: $\mathbb{E}_{\xi \sim P}[\|\nabla f(x, \xi)\|_2^2] \leq \sigma^2$.

Assumption 2 (Boundedness of the domain). *Let* $x_0 \in \mathbb{R}^n$. *Then, we assume that there exists* $R > 0$ *such that* $R^2 = \sup_{x \in \mathbb{R}^d} \|x - x_0\|_2^2 < \infty$.

Theorem 3.1. *Let* f *be a Lipschitz-smooth function with minimum* f^*. *Let Assumption 1 and Assumption 2 for* (1) *hold. Let* $x_0 \in \mathbb{R}^n$ *be the initial point of MDA. Assume that we run MDA for* T *iterations. Let* $z_1, \ldots, z_T$ *and* $x_1, \ldots, x_T$ *be the points returned by MDA and set* $\lambda_k = \eta_k \sqrt{k+1}$, $\beta_k = \sqrt{k+1}$ *and* $c_k = c$ *where* $\eta_k = 1/\sqrt{T}$ *and* $c \in (0, 1]$. *Assume that* $T \geq L^2/c^2$. *Then, we have:*

$$
\frac{1}{2T} \sum_{k=0}^{T} (\|\nabla f(x_k)\|_2^2 + \|\nabla f(z_k)\|_2^2)
$$
$$
\leq \frac{2((f(x_0) - f^*) - \mathbb{E}[f(z_{T+1}) - f^*])}{\sqrt{T}}
$$
$$
+ 2 \left(\frac{1}{c} - 1 \right) \frac{(L+1)(f(x_0) - f^*) - (L + \alpha_T)\mathbb{E}[f(x_T) - f^*]}{T} \tag{10}
$$
$$
+ 2 \left[\left(\frac{L}{\sqrt{T}} + \frac{\log(T+1)}{T} \right) \sigma^2 + \left(\frac{L \log(T)}{T} + \frac{2 \log(T)}{\sqrt{T}} \right) R^2 \right],
$$

where f^* *is the value of* f *at a stationary point and* $\alpha_T = \sqrt{T} \left(1 - \frac{\sqrt{T+1}}{\sqrt{T+2}} \right)$.

Theorem 3.1 informs us that the convergence rate of MDA to a stationary point is of $O(1/\sqrt{T})$ and is similar to the one obtained with SGD. A proof of this statement can be found in Appendix B.

4 NUMERICAL EXPERIMENTS

We investigate the numerical performance of MDA on a wide range of learning tasks, including image classification, MRI reconstruction, neural machine translation (NMT) and language modeling. We performed a comparison against both SGD+M and Adam. Depending on the task, one of these two methods is considered the current state-of-the-art for the problem. For each task, to enable a fair comparison, we perform a grid-search over step-sizes, weight decay and learning rate schedule to obtain the best result for each method. For our CV experiments we use the torchvision package, and for our NLP experiments we use fairseq (Ott et al., 2019). We now briefly explain each of the learning tasks and present our results. Details of our experimental setup and experiments on Wikitext-103 can be respectively found in Appendix C and Appendix D.

4.1 IMAGE CLASSIFICATION

We run our image classification experiments on the CIFAR-10 and ImageNet datasets. The CIFAR-10 dataset consists of 50k training images and 10k testing images. The ILSVRC 2012 ImageNet
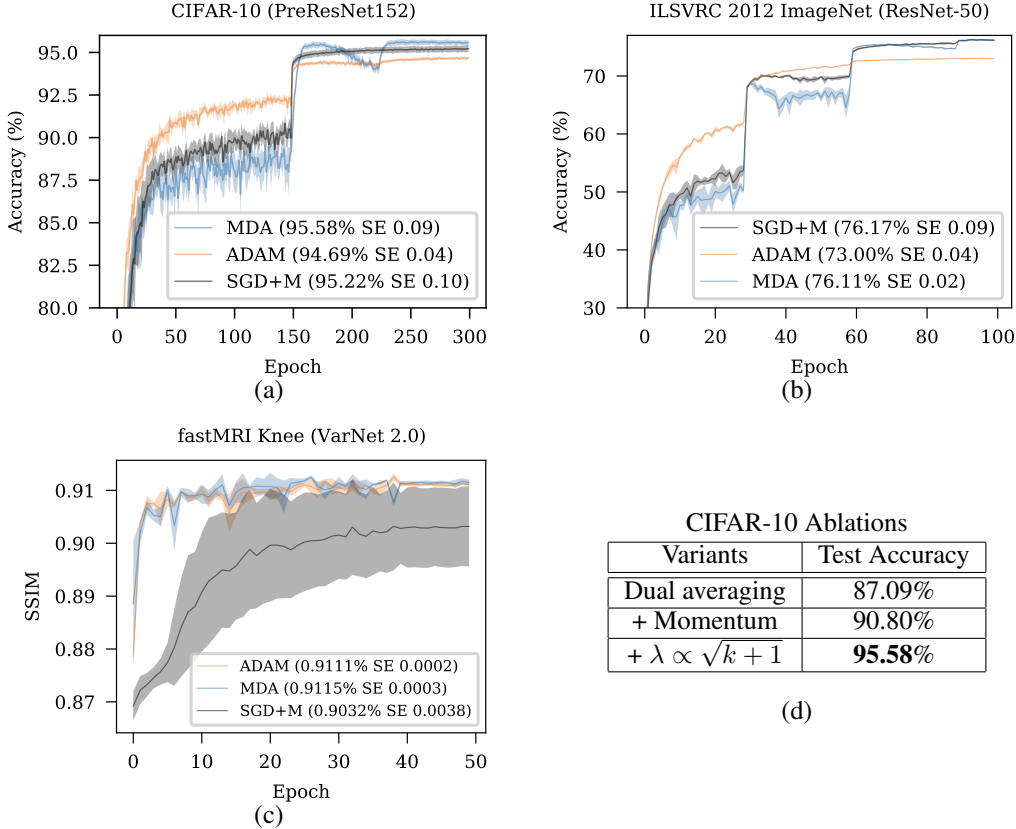
Figure 1: (a): Test accuracy of SGD with momentum and MDA on CIFAR-10 and (b) on ImageNet. (c): Test SSIM of Adam and MDA on fastMRI Knee dataset. MDA matches the performance of the best first-order methods in these computer vision tasks. (d): Ablation study for MDA on CIFAR-10.

dataset has 1.2M training images and 50k validation images. We train a pre-activation ResNet-152 on CIFAR-10 and a ResNet-50 model for ImageNet. Both architectures are commonly used base-lines for these problems. We follow the settings described in He et al. (2016) for training. Figure 1 (a) represents the accuracy obtained on CIFAR-10. MDA achieves a slightly better accuracy compared to SGD with momentum (by 0.36%). This is an interesting result as SGD with momentum serves as first-order benchmark on CIFAR-10. We speculate this difference is due to the beneficial properties of the decaying regularization that MDA contains. Figure 1 (b) represents the accuracy obtained on ImageNet. In this case the difference between MDA and SGD+M is within the standard errors, with Adam trailing far behind.

4.2 MRI RECONSTRUCTION

For our MRI reconstruction task we used the fastMRI knee dataset (Zbontar et al., 2018). It consists of more than 10k training examples from approximately 1.5k fully sampled knee MRIs. The fastMRI challenge is a kind of image-to-image prediction task where the model must predict a MRI image from raw data represented as "k-space" image. We trained the VarNet 2.0 model introduced by Sriram et al. (2020), which is currently the state-of-the-art for this task. We used 12 cascades, batch-size 8, a 4x acceleration factor, 16 center lines and the masking procedure described in Defazio (2019). Figure 1 (c) shows the SSIM scores obtained for each method. We observe that MDA performs slightly better than Adam, although the difference is within the standard error. SGD performs particularly badly for this task, no mater what tuning of learning rate schedule is tried. The visual difference between reconstructions given by the best SGD trained model, versus the best model from the other two methods is readily apparent when compared at the pixel level (Figure 2).

Ground truth   MDA (SSIM 0.48844

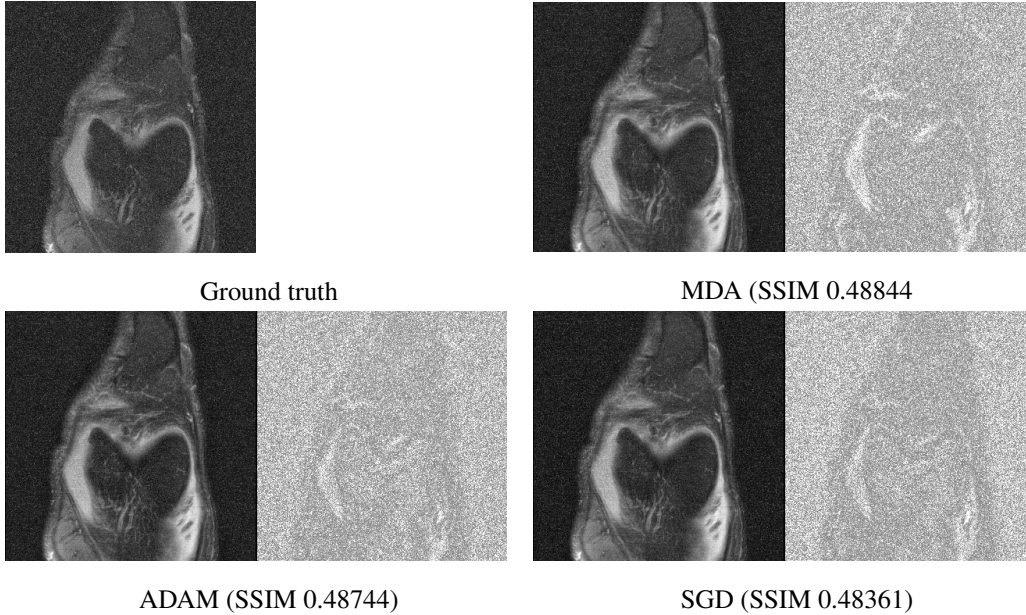ADAM (SSIM 0.48744)   SGD (SSIM 0.48361)

Figure 2: Reconstruction images for an illustrative knee slice for the same model trained with each of the 3 methods, using the best model for the seeds for each. The difference image between the ground-truth and the noise is shown on the right.
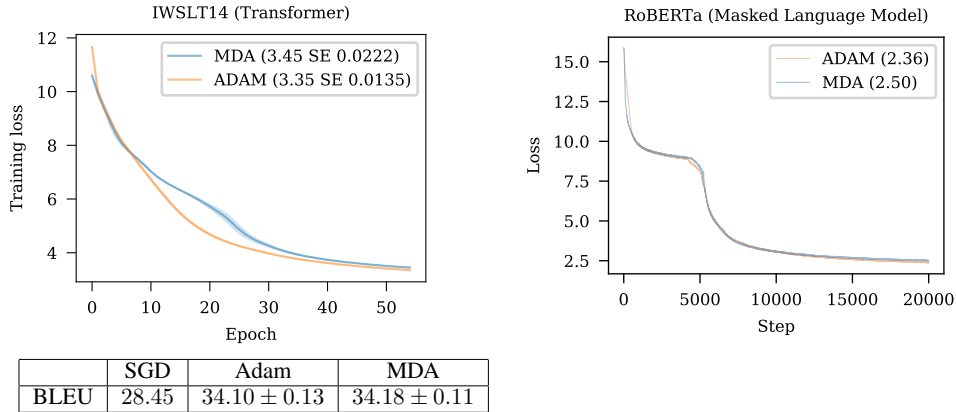

	SGD	Adam	MDA
BLEU	28.45	34.10 ± 0.13	34.18 ± 0.11

Figure 3: Left: performance of Adam and MDA in NMT on IWSLT'14. The plot shows the training loss convergence, while the table provides the BLEU score on the test set. Right: performance of ADAM and MDA on RoBERTa training.

4.3 NEURAL MACHINE TRANSLATION (NMT)

We run our machine translation task on the IWSLT'14 German-to-English (De-En) dataset (approximately 160k sentence pairs) (Cettolo et al., 2014). We use a Transformer architecture and follow the settings reported in Ott et al. (2019), using the pre-normalization described in Wang et al. (2019). The length penalty is set to 0.6 and the beam size is set to 5. For NMT, BLEU score is used (Papineni et al., 2002). We report the results of the best checkpoints with respect to the BLEU score averaged over 20 seeds. We report tokenized case-insensitive BLEU. Figure 3 reports the training loss and the BLEU score on the test set of SGD, MDA and Adam on IWSLT'14. SGD as reported in (Yao et al., 2020) performs worse than the other methods. While Adam and MDA match in terms of training loss, MDA outperforms Adam (by 0.20) on the test set. Despite containing no adaptivity, the MDA method is as capable as Adam for this task.

4.4 MASKED LANGUAGE MODELING

Our largest comparison was on the task of masked language modeling. Pretraining using masked language models has quickly become a standard approach within the natural language processing community (Devlin et al., 2019), so it serves as a large-scale, realistic task that is representative of optimization in modern NLP. We used the RoBERTa variant of the BERT model (Liu et al., 2019), as implemented in fairseq, training on the concatenation of BookCorpus (Zhu et al., 2015) and English Wikipedia. Due to the high computational costs of training we present the results of 1 run for each method rather than an average over seeds. We performed hyper-parameter search only on the learning rate. Figure 3 shows the training loss for Adam and MDA; SGD fails on this task. MDA's learning curve is virtually identical to Adam. The "elbow" shape of the graph is due to the training reaching the end of the first epoch around step 4000. On validation data, MDA achieves a perplexity of 5.3 broadly comparable to the 4.95 value of Adam. As we are using hyper-parameters tuned for Adam, we believe this small gap can be further closed with additional tuning.

4.5 ABLATION STUDY

As our approach builds upon regular dual averaging, we performed an ablation study on CIFAR-10 to assess the improvement from our changes. We ran each method with a sweep of learning rates, both with flat learning rate schedules and the standard stage-wise scheme. The results are shown in Figure 1 (d). Regular dual averaging performs extremely poorly on this task, which may explain why dual averaging variants have seen no use that we are aware of for deep neural network optimization. The best hyper-parameter combination was LR 1 with the flat LR scheme. We report the results based on the last-iterate, rather than a random iterate (required by the theory), since such post-hoc sampling performs poorly for non-convex tasks. The addition of momentum in the form of iterate averaging within the method brings the test accuracy up by 3.7%, and allows for the use of a larger learning rate of 2.5. The largest increase is from the use of an increasing lambda sequence, which improves performance by a further 4.78%.

5 TIPS FOR USAGE

When applying the MDA algorithm to a new problem, we found the following guidelines to be useful:

- MDA may be slower than other methods at the early iterations. It is important to run the method to convergence when comparing against other methods.

- The amount of weight decay required when using MDA is often significantly lower than for SGD+M or Adam. We recommend trying a sweep with a maximum of the default for SGD+M or Adam, and a minimum of zero.

- Learning rates for MDA are much larger than SGD+M or Adam, due to their different parameterizations. When comparing to SGD+M with learning rate η and momentum β, a value comparable to $\eta/(1-\beta)$ is a good starting point. On NLP problems, learning rates as large at 15.0 are sometimes effective.

6 CONCLUSION

Based on our experiments, the MDA algorithm appears to be a good choice for a general purpose optimization algorithm for the non-convex problems encountered in deep learning. It avoids the sometimes suboptimal test performance of Adam, while converging on problems where SGD+M fails to work well. Unlike Adam which has no general convergence theory for non-convex problems under the standard hyper-parameter settings, we have proven convergence of MDA under realistic hyper-parameter settings for non-convex problems. It remains an open question why MDA is able to provide the best result in SGD and Adam worlds.

REFERENCES

Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 1157–1167. Curran Associates, Inc., 2018.

Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1614–1622. Curran Associates, Inc., 2016.

Heinz H Bauschke, Jonathan M Borwein, et al. Legendre functions and the method of random bregman projections. *Journal of convex analysis*, 4(1):27–67, 1997.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Deep frank-wolfe for neural network optimization. *arXiv preprint arXiv:1811.07591*, 2018.

Siegfried Bos and E Chug. Using weight decay to optimize the generalization ability of a perceptron. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pp. 241–246. IEEE, 1996.

L Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes 91, Nimes, France*, 1991.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pp. 161–168, 2008.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning, 2016.

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT 2014)*, 2014.

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

Igor Colin, Aurélien Bellet, Joseph Salmon, and Stéphan Clémençon. Gossip dual averaging for decentralized optimization of pairwise functions. *arXiv preprint arXiv:1606.02421*, 2016.

Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

Aaron Defazio. Offset sampling improves deep learning based accelerated mri reconstructions by exploiting symmetry, 2019.

Aaron Defazio. Understanding the role of momentum in non-convex optimization: Practical insights from a lyapunov analysis. *arXiv preprint*, 2020.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, pp. 1755–1765, 2019.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. On the convergence of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

John Duchi and Feng Ruan. Asymptotic optimality in stochastic optimization. *arXiv preprint arXiv:1612.05612*, 2016.

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

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3): 592–606, 2011b.

Jonathan Eckstein. Nonlinear proximal point algorithms using bregman functions, with applications to convex programming. *Mathematics of Operations Research*, 18(1):202–226, 1993.

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, 2012.

Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi. Online distributed optimization via dual averaging. In *52nd IEEE Conference on Decision and Control*, pp. 1484–1489. IEEE, 2013.

Anatoli Juditsky, Joon Kwon, and Éric Moulines. Unifying mirror descent and dual averaging. *arXiv preprint arXiv:1910.13742*, 2019.

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

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pp. 950–957, 1992.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Sangkyun Lee and Stephen J Wright. Manifold identification in dual averaging for regularized stochastic online learning. *The Journal of Machine Learning Research*, 13(1):1705–1744, 2012.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 983–992, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Yu Nesterov and Vladimir Shikhman. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3):917–940, 2015.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2013.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

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

Othmane Sebbouh, Robert M Gower, and Aaron Defazio. On the convergence of the stochastic heavy ball method. *arXiv preprint arXiv:2006.07867*, 2020.

Shahin Shahrampour and Ali Jadbabaie. Exponentially fast parameter estimation in networks using distributed dual averaging. In *52nd IEEE Conference on Decision and Control*, pp. 6196–6201. IEEE, 2013.

Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron Defazio, C Lawrence Zitnick, Nafissa Yakubova, Florian Knoll, and Patricia Johnson. End-to-end variational networks for accelerated mri reconstruction. *arXiv preprint arXiv:2004.06688*, 2020.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013.

Wei Tao, Zhisong Pan, Gaowei Wu, and Qing Tao. Primal averaging: A new gradient evaluation step to attain the optimal individual convergence. *IEEE transactions on cybernetics*, 50(2):835–845, 2018.

Marc Teboulle. Entropic proximal mappings with applications to nonlinear programming. *Mathematics of Operations Research*, 17(3):670–690, 1992.

Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Push-sum distributed dual averaging for convex optimization. In *2012 ieee 51st ieee conference on decision and control (cdc)*, pp. 5453–5458. IEEE, 2012.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pp. 6677–6686, 2019.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. In *Advances in Neural Information Processing Systems*, pp. 9712–9724, 2019.

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.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Zhewei Yao, Amir Gholami, Sheng Shen, Kurt Keutzer, and Michael W Mahoney. Adahessian: An adaptive second order optimizer for machine learning. *arXiv preprint arXiv:2006.00719*, 2020.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Matthew J Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J Geras, Joe Katsnelson, Hersh Chandarana, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.

Siqi Zhang and Niao He. On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1806.04781*, 2018.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pp. 19–27, 2015.

Fangyu Zou, Li Shen, Zequn Jie, Ju Sun, and Wei Liu. Weighted adagrad with unified momentum. *arXiv preprint arXiv:1808.03408*, 2018.