Obtaining Adjustable Regularization for Free via Iterate Averaging

Jingfeng Wu [1] **Vladimir Braverman** [1] **Lin F. Yang** [2]

Abstract

Regularization for optimization is a crucial technique to avoid overfitting in machine learning. In order to obtain the best performance, we usually train a model by tuning the regularization parameters. It becomes costly, however, when a single round of training takes significant amount of time. Very recently, Neu & Rosasco (2018) show that if we run stochastic gradient descent (SGD) on linear regression problems, then by averaging the SGD iterates properly, we obtain a regularized solution. It left open whether the same phenomenon can be achieved for other optimization problems and algorithms. In this paper, we establish an averaging scheme that *provably* converts the iterates of SGD on an arbitrary strongly convex and smooth objective function to its regularized counterpart with an *adjustable* regularization parameter. Our approaches can be used for accelerated and preconditioned optimization methods as well. We further show that the same methods work empirically on more general optimization objectives including neural networks. In sum, we obtain *adjustable* regularization *for free* for a large class of optimization problems and resolve an open question raised by Neu & Rosasco (2018).

1. Introduction

Regularization for optimization is a key technique for avoiding over-fitting in machine learning and statistics (Grandvalet & Bengio, 2005; Krogh & Hertz, 1992; Tibshirani, 1996; Tikhonov & Arsenin, 1977). The effects of explicit regularization methods, i.e., an extra regularization term added to the vanilla objective, are well studied, e.g., ridge regression (Tikhonov & Arsenin, 1977), LASSO (Tibshirani, 1996) and entropy regularization (Grandvalet & Bengio, 2005). Despite the great benefits of adopting explicit regularization, it could cause a huge computational burden to search for the optimal hyperparameter associated with the extra regularization term, especially for large-scale machine learning problems (Devlin et al., 2018; He et al., 2016; Silver et al., 2017).

In another line of research, people recognize and utilize the implicit regularization caused by certain components in machine learning algorithms, e.g., initialization (He et al., 2015; Hu et al., 2020), batch normalization (Ioffe & Szegedy, 2015; Cai et al., 2018), iterate averaging (Bach & Moulines, 2013; Jain et al., 2018; Neu & Rosasco, 2018), and optimizer such as gradient descent (GD) (Gunasekar et al., 2018; Soudry et al., 2018; Suggala et al., 2018). The regularization effect usually happens along the process of training the model and/or requires little post-computation. A great deal of evidence indicates that such a implicit bias plays a crucial role for the generalization abilities in many modern machine learning models (Zhang et al., 2016; Zhu et al., 2018; Wilson et al., 2017; Soudry et al., 2018). However, the implicit regularization is often a fixed effect and lacks the flexibility to be adjusted. To fully utilize it, we need a thorough understanding about the mechanism of the implicit regularization.

Among all the efforts spent on understanding and utilizing the implicit regularization, the work on bridging iterate averaging with explicit regularization (Neu & Rosasco, 2018) is extraordinarily appealing. In particular, Neu & Rosasco (2018) show that for linear regression, one can achieve ℓ_2-regularization effect *for free* by simply taking geometrical averaging over the optimization path generated by stochastic gradient descent (SGD), which costs little additional computation. More interestingly, the regularization is *adjustable*, i.e., the solution biased by the regularizer in arbitrary strength can be obtained by iterate averaging using the corresponding weighting scheme. In a nutshell, this regularization approach has advantages over both the implicit regularization methods for being adjustable, and the explicit regularization methods for being cheap to tune.

Nevertheless, Neu & Rosasco (2018) only provide a method and its analysis for linear regression optimized by SGD. However linear regression itself is a rather restricted optimization objective. A nature question arises:

Can we obtain "free" and "adjustable" regularization for

[1]Johns Hopkins University, Baltimore, MD, USA [2]University of California, Los Angeles, CA, USA. Correspondence to: Jingfeng Wu <uuujf@jhu.edu>, Lin F. Yang <linyang@ee.ucla.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

broader objective functions and optimization methods?

In this work, we answer this question positively from the following aspects:

1. For linear regression, we analyze the regularization effects of averaging the optimization paths of SGD as well as preconditioned SGD, with adaptive learning rates. The averaged solutions achieve effects of ℓ_2-regularization and generalized ℓ_2-regularization respectively, in an adjustable manner. Similar results hold for kernel ridge regression as well.

2. We show that for Nesterov's accelerated stochastic gradient descent, the iterate averaged solution can also realize ℓ_2-regularization effect by a modified averaging scheme. This resolves an open question raised by Neu & Rosasco (2018).

3. Beside linear regression, we study the regularization effects of iterate averaging for strongly convex and smooth loss functions, hence establishing a provable approach for obtaining nearly *free* and *adjustable* regularization for a broad class of functions.

4. Empirical studies on both synthetic and real datasets verify our theory. Moreover, we test iterate averaging with modern *deep neural networks* on CIFAR-10 and CIFAR-100 datasets, and the proposed approaches *still* obtain effective and adjustable regularization effects with little additional computation, demonstrating the broad applicability of our methods.

Our analysis is motivated from continuous approximation based on differential equations. When the learning rate tends to zero, the discrete algorithmic iterates tends to be the continuous path of an ordinary differential equation (ODE), on which we can establish a continuous version of our theory. We then discretize the ODE and generalize the theory to that of finite step size. This technique is of independent interests since it can be applied to analyze other comprehensive optimization problems as well (Su et al., 2014; Hu et al., 2017a; Li et al., 2017; Yang et al., 2018; Shi et al., 2019). Our results, in addition to the linear regression result in (Neu & Rosasco, 2018), illustrate the promising application of iterate averaging to obtain *adjustable* regularization *for free*.

2. Preliminaries

Let $\{(x_i, y_i) \in \mathbb{R}^{d\times 1}\}_{i=1}^n$ be the training data and $w \in \mathbb{R}^d$ be the parameters to be optimized. The goal is to minimize a lower bounded loss function $L(w)$

$$\min_w L(w) := \frac{1}{n}\sum_{i=1}^n \ell(x_i, y_i, w). \qquad (\mathcal{P}_1)$$

One important example is linear regression under the square loss where $L(w) = \frac{1}{2n}\sum_{i=1}^n \left\|w^\top x_i - y_i\right\|_2^2$. The optimization problem often involves an explicit regularization term

$$\min_{\hat{w}} L(\hat{w}) + \lambda R(\hat{w}), \qquad (\mathcal{P}_2)$$

where $R(\hat{w})$ is a regularizer and λ is the associated hyperparameter. For example, the ℓ_2-regularizer is $R(\hat{w}) = \frac{1}{2}\left\|\hat{w}\right\|_2^2$. Given an iterative algorithm, e.g., SGD, an optimization path is generated by running the algorithm. With a little abuse of notations, we use $\{w_k\}_{k=0}^\infty$ and $\{\hat{w}_k\}_{k=0}^\infty$ to represent the optimization paths for the unregularized problem ($\mathcal{P}_1$) and the regularized problem ($\mathcal{P}_2$), respectively. Sometimes we write $\hat{w}_k$ with a script as $\hat{w}_{k,\lambda}$ to emphasize its dependence on the hyperparameter λ. We use η_k and γ_k to denote the learning rates for training the unregularized and regularized objectives respectively. For simplicity we always initialize the iterative algorithms from zero, i.e., $w_0 = \hat{w}_0 = 0$.

Iterate averaging The core idea in this work is a technique called *iterate averaging*. Given a series of parameters $\{w_k\}_{k=0}^\infty$, a *weighting scheme* $\{p_k\}_{k=0}^\infty$ is defined as a probability distribution associated to the series, i.e., $p_k \geq 0$, $\sum_{k=0}^\infty p_k = 1$. Its accumulation is denoted as $P_k = \sum_{i=0}^k p_i$, where $\lim_{k\to\infty} P_k = 1$. Since a weighting scheme and its accumulation identifies each other by $p_k = P_k - P_{k-1}$ for $k \geq 1$, we also call $\{P_k\}_{k=0}^\infty$ a weighting scheme. Then the iterate averaged parameters are

$$\tilde{w}_k = {P_k}^{-1}\textstyle\sum_{i=0}^k p_i w_i, \quad k \geq 0.$$

Various kinds of averaging schemes (for the SGD optimization path) have been studied before. Theoretically, arithmetic averaging is shown to bring better convergence (Bach & Moulines, 2013; Lakshminarayanan & Szepesvari, 2018); tail-averaging is analyzed by Jain et al. (2018); and Neu & Rosasco (2018) discuss geometrically averaging and its regularization effect for SGD and linear regression. Empirically, arithmetic averaging is also shown to be helpful for modern deep neural networks (Izmailov et al., 2018; Zhang et al., 2019; Granziol et al., 2020). Inspired by Neu & Rosasco (2018), in this work we explore in depth the regularization effect induced by iterate averaging for various kinds of optimization algorithms and loss functions.

Stochastic gradient descent The optimization problem ($\mathcal{P}_1$) is often solved by stochastic gradient descent (SGD): at every iteration, a mini-batch is sampled uniformly at random, and then the parameters are updated according to the gradient of the loss estimated using the mini-batch. For simplicity we let the batch size be 1. Then with learning rate $\eta_k > 0$, SGD takes the following update:

$$w_{k+1} = w_k - \eta_k \nabla \ell(x_k, y_k, w_k). \qquad (1)$$

Similarly, for the regularized problem ($\mathcal{P}_2$), with learning rate $\gamma_k > 0$, SGD takes update:

$$\hat{w}_{k+1} = \hat{w}_k - \gamma_k \left(\nabla \ell(x_k, y_k, \hat{w}_k) + \lambda \nabla R(\hat{w}_k) \right). \quad (2)$$

For linear regression problem and fixed learning rates, Neu & Rosasco (2018) discuss the geometrically averaging over the SGD iterates (1). They show that by doing so one obtains the solution of the ℓ_2-regularized problem ($\mathcal{P}_2$) where $R(\hat{w}) = \frac{1}{2}\|\hat{w}\|_2^2$ for arbitrary hyperparameter λ. In this work, we analyze a much broader class of algorithms and functions. In particular, we establish adjustable ℓ_2-regularization effect for (i) SGD with adaptive learning rate, (ii) kernel ridge regression (Mohri et al., 2018), and (iii) general strongly convex and smooth loss functions.

Preconditioned stochastic gradient descent We also study iterate averaging for preconditioned stochastic gradient descent (PSGD). Given a positive definite matrix Q as the preconditioning matrix and η_k as the learning rate, the PSGD takes following update to optimize problem ($\mathcal{P}_1$):

$$w_{k+1} = w_k - \eta_k Q^{-1} \nabla \ell(x_k, y_k, w_k), \quad (3)$$

Similarly, the regularized problem ($\mathcal{P}_2$) can be solved by PSGD with learning rate $\gamma_k > 0$ as:

$$\hat{w}_{k+1} = \hat{w}_k - \gamma_k Q^{-1} \left(\nabla \ell(x_k, y_k, \hat{w}_k) + \lambda \nabla R(\hat{w}_k) \right). \quad (4)$$

We remark that PSGD unifies several important algorithms as natural gradient descent and Newton's method at special cases where the curvature matrices can be replaced by constant matrices (Martens, 2014; Dennis Jr & Schnabel, 1996; Bottou & Bousquet, 2008).

For linear regression problems, we will show that geometrically averaging the PSGD iterates (3) leads to a solution biased by the *generalized ℓ_2-regularizer*, i.e., the solution of problem ($\mathcal{P}_2$) with $R(w) = \frac{1}{2} w^\top Q w = \frac{1}{2}\|w\|_Q^2$. The obtained regularization is adjustable, too.

Nesterov's accelerated stochastic gradient descent In problem ($\mathcal{P}_1$), suppose the loss function $L(w)$ is α-strongly convex. Let $\eta > 0$ be the learning rate and $\tau = \frac{1-\sqrt{\eta\alpha}}{1+\sqrt{\eta\alpha}}$, then the Nesterov's accelerated stochastic gradient descent (NSGD) takes update (Nesterov, 1983; Su et al., 2014; Yang et al., 2018):

$$\begin{aligned} w_{k+1} &= v_k - \eta \nabla \ell(x_k, y_k, v_k), \\ v_k &= w_k + \tau(w_k - w_{k-1}). \end{aligned} \quad (5)$$

Now we consider the regularized problem ($\mathcal{P}_2$) with the ℓ_2-regularizer, $R(\hat{w}) = \frac{1}{2}\|\hat{w}\|_2^2$. The objective function then becomes $(\alpha + \lambda)$-strongly convex. Let $\gamma > 0$ be the learning rate and $\hat{\tau} = \frac{1-\sqrt{\gamma(\alpha+\lambda)}}{1+\sqrt{\gamma(\alpha+\lambda)}}$, then the NSGD takes update:

$$\begin{aligned} \hat{w}_{k+1} &= \hat{v}_k - \gamma \left(\nabla \ell(x_k, y_k, \hat{v}_k) + \lambda \hat{v}_k \right), \\ \hat{v}_k &= \hat{w}_k + \hat{\tau}(\hat{w}_k - \hat{w}_{k-1}). \end{aligned} \quad (6)$$

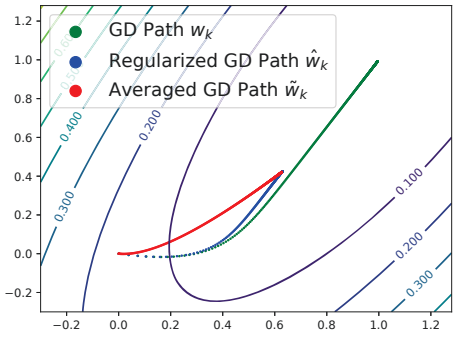


Figure 1. A 2-D demonstration of the effect of an averaged SGD path (Theorem 1). Green dots: the vanilla GD path w_k; blue dots: the regularized GD path $\hat{w}_k$; red dots: the averaged GD path $\tilde{w}_k$. The red dots converge to the blue ones.

It is proposed as an open question by Neu & Rosasco (2018) whether or not adjustable regularization can be obtained by averaging the NSGD optimization path. Our work offers an affirmative answer by showing that for linear regression, one can perform iterate averaging over the NSGD path to obtain the ℓ_2-regularized solution as well.

3. The free and adjustable regularization induced by iterate averaging

In this section, we show that adjustable regularization effects can be obtained for "free" via *iterate averaging* for: (i) different SGD schemes, e.g., linear regression or kernel ridge regression with adaptive learning rates; (ii) PSGD; (iii) NSGD; (iv) arbitrary strongly convex and smooth loss functions. Not limited to SGD with fixed learning rate and linear regression, our results manifest the broader potential of employing iterate averaging to obtain regularization that can be tuned with little computation overhead.

Our analysis is motivated from continuous differential equations, which is postponed to Section A of Supplementary Materials due to space limitation. In the following we present our results in discrete cases.

3.1. The effect of an averaged SGD path

We first introduce a generalized averaging scheme for the SGD algorithm. Unlike the method in (Neu & Rosasco, 2018), our approach works even with adaptive learning rates. Specifically, given a learning rate schedule and a regularization parameter λ, we compute a weighting scheme

for averaging a stored SGD path. Then the averaged solution converges to the regularized solution with hyperparameter λ. Theorem 1 formally justifies our method.

Theorem 1 (The effect of an averaged SGD path)**.** *Consider loss function $L(w) = \frac{1}{2n}\sum_{i=1}^{n}\left\|w^\top x_i - y_i\right\|_2^2$, and regularizer $R(w) = \frac{1}{2}\|w\|_2^2$. Let α and β be such that $L(w)$ is α-strongly convex[1] and β-smooth. Let $\{w_k\}_{k=0}^{\infty}$ and $\{\hat{w}_k\}_{k=0}^{\infty}$ be the SGD paths for the vanilla loss function $L(w)$ with learning rate η_k, and the regularized loss function $L(\hat{w}) + \lambda R(\hat{w})$ with learning rate γ_k, respectively. Suppose $1 - \lambda\gamma_k = \gamma_k/\eta_k$, $\eta_k \in (\eta, 1/\beta)$, $\eta > 0$ and $\gamma := \eta/(1+\lambda\eta)$. Let*

$$P_k := \textstyle\sum_{i=0}^{k} p_i = 1 - \Pi_{i=0}^{k}(\gamma_i/\eta_i).$$

Then for $\tilde{w}_k = P_k^{-1}\sum_{i=0}^{k} p_i w_i$ we have

1. $P_k \cdot \mathbb{E}[\tilde{w}_k] = \mathbb{E}[\hat{w}_k] - (1 - P_k)\cdot\mathbb{E}[w_k]$.

2. Both $\mathbb{E}[w_k]$ and $\mathbb{E}[\hat{w}_k]$ converge. Moreover, we have $\|\mathbb{E}[\hat{w}_k] - \mathbb{E}[\tilde{w}_k]\|_2 \le \mathcal{O}((1-\lambda\gamma)^k)$.

3. If the gradient noise $\epsilon_k = \nabla\ell(x_k, y_k, w) - \nabla L(w)$ has uniformly bounded variance $\mathbb{E}[\|\epsilon_k\|_2^2] \le \sigma^2$, then for k large enough, with probability at least $1-\delta$ we have[2]

$$\|P_k\tilde{w}_k - P_k\mathbb{E}[\tilde{w}_k]\|_2 \le \epsilon,$$

where $\epsilon = \frac{\sigma}{\gamma(\lambda+\alpha)(\lambda+\beta)^2}\cdot\sqrt{\frac{\lambda}{\delta\gamma(2-\lambda\gamma)}}$.

The proof is left in Supplementary Materials, Section C.1. A 2-D illustration for Theorem 1 is presented in Figure 1.

Theorem 1 guarantees the method of obtaining adjustable ℓ_2-regularization for free via iterate averaging. Specifically, we first collect an SGD path $\{w_k\}_{k=0}^{\infty}$ for $L(w)$ under a learning rate schedule η_k (it can be chosen in a broad range); then for a regularization parameter λ, we compute an averaging scheme $\{p_k\}_{k=0}^{\infty}$ that converts the collected SGD path to the regularized solution, $\hat{w}_\infty$. Note that the learning rate schedule γ_k is only for analysis and does not need to be known.

Specifically, when the learning rates are constants, i.e., $\eta_i = \eta$ and $\gamma_i = \gamma$, the first two conclusions in Theorem 1 recover the Proposition 1 and Proposition 2 in (Neu & Rosasco, 2018). Besides, the third claim in Theorem 1 characterizes the deviation of the averaged solution, which relies on the models, learning rates, and the regularization parameter, etc. And empirical studies in Section 4.2 do suggest that such a deviation is sufficiently small that it does not affect the induced regularization effect.

Remark. We emphasize that the method of Neu & Rosasco (2018) only applies to SGD with constant learning rate. Moreover, their theory only guarantees the averaged solution has *convergence in expectation*, which is not very useful since the averaged solution might not converge to the regularized solution almost surely, not even in probability (a.k.a. weak convergence) (see Section 4.2). Nevertheless, our theory carefully characterizes the deviation between the averaged solution and the regularized solution.

More interestingly, Theorem 1.1 shows that this method is also applicable to kernel ridge regression (in the dual space).

Theorem 1.1. *Let $K \in \mathbb{R}^{n\times n}$ be a kernel, $K(i,j) = \phi(x_i)^\top\phi(x_j)$, where $\phi : \mathbb{R}^d \to \mathcal{H}$ is the kernel map. Consider kernel ridge regression*

$$\min_{\alpha\in\mathbb{R}^n} L(\alpha, \lambda) := \frac{1}{2}\|y - K\alpha\|_2^2 + \frac{\lambda}{2}\alpha^\top K\alpha$$

where $y = (y_1, \ldots, y_n)^T$ is the labels and $\alpha \in \mathbb{R}^n$ is the dual parameter. Let $\{\alpha_k\}_{k=0}^{\infty}$ and $\{\hat{\alpha}_k\}_{k=0}^{\infty}$ be the GD paths for the loss $L(\alpha, \lambda)$ with learning rate η_k, and the loss $L(\hat{\alpha}, \hat{\lambda})$ with generalized learning rate γ_k, respectively. Suppose $\hat{\lambda} > \lambda$, $\gamma_k = \eta_k\left(I + (\hat{\lambda} - \lambda)\eta_k K\right)^{-1}$. Let

$$P_k := \textstyle\sum_{i=0}^{k} p_i = 1 - \Pi_{i=0}^{k}\left(\gamma_i/\eta_i\right).$$

Then for $\tilde{\alpha}_k = P_k^{-1}\sum_{i=0}^{k} p_i\alpha_i$ we have

1. $P_k\tilde{\alpha}_k = \hat{\alpha}_k - (1 - P_k)\alpha_k$.

2. Both α_k and $\hat{\alpha}_k$ converge provided suitable learning rates. Moreover, we have $\|\hat{\alpha}_k - \tilde{\alpha}_k\|_2 \le \mathcal{O}(C^k)$ where $C \in (0,1)$ is a constant decided by K, $\hat{\lambda} - \lambda$ and η_k.

3.2. The effect of an averaged PSGD path

In practice, we usually need many different regularizers. And one important class of them is the *generalized ℓ_2-regularizers*, i.e., $R(w) := \frac{1}{2}w^\top Qw$ for some positive definite matrix Q. But it is painful to adjust its regularization parameter λ by re-training the model. Luckily, we show that the solution biased by such a regularizer can also be obtained for "free" by averaging the optimization path of PSGD. Our result is formally presented in the next theorem.

Theorem 2 (The effect of an averaged PSGD path)**.** *Consider loss function $L(w) = \frac{1}{2n}\sum_{i=1}^{n}\left\|w^\top x_i - y_i\right\|_2^2$, and regularizer $R(w) = \frac{1}{2}w^\top Qw$, where Q is a positive definite matrix. Let α and β be such that $\alpha Q \preceq \Sigma =$*

[1] The strong convexity assumption does not limit the application of our method. For a convex but not strongly convex loss $L(w)$, we can instead collect an optimization path of $L(w) + \lambda_0\|w\|_2^2$ for some small λ_0, which is then strongly convex, and then we apply Theorem 1 to obtain the regularized solutions for a different λ. Similar arguments apply to the theorems afterwards as well.

[2] In this high probability result, the confidence parameter δ appears in a polynomial order, $\frac{1}{\sqrt{\delta}}$. However this is only due to the assumption of bounded variance of the noise and an application of Chebyshev's inequality. It is straightforward to obtain a logarithm dependence on δ by assuming the sub-Gaussianity of the noise and applying Hoeffding's inequality. Similar arguments apply to the theorems afterwards as well.

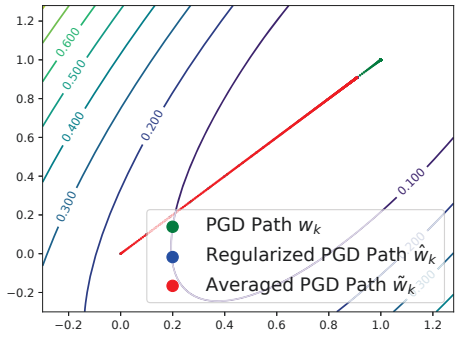


Figure 2. A 2-D demonstration of the effect of an averaged PSGD path (Theorem 2). The Hessian is used as the preconditioning matrix. Green dots: the vanilla PGD path w_t; blue dots: the regularized PGD path $\hat{w}_t$; red dots: the averaged PGD path $\tilde{w}_t$. The red dots converge to the blue ones.

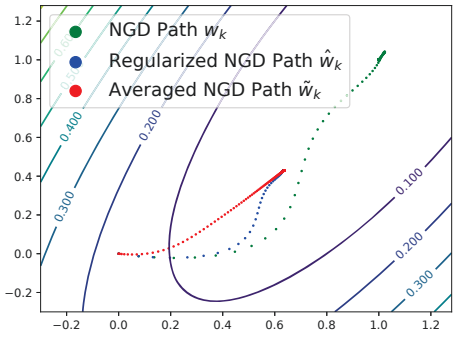


Figure 3. A 2-D demonstration of the effect of an averaged NSGD path (Theorem 3). Green dots: the vanilla NGD path w_t; blue dots: the regularized NGD path $\hat{w}_t$; red dots: the averaged NGD path $\tilde{w}_t$. The red dots converge to the blue ones.

$n^{-1}\sum_{i=1}^n x_i x_i^\top \preceq \beta Q$. *With Q as the preconditioning matrix, let $\{w_k\}_{k=0}^\infty$ and $\{\hat{w}_k\}_{k=0}^\infty$ be the PSGD paths for the vanilla loss function $L(w)$ with learning rate η_k, and the regularized loss function $L(\hat{w}) + \lambda R(\hat{w})$ with learning rate γ_k, respectively. Suppose $1 - \lambda\gamma_k = \gamma_k/\eta_k$, $\eta_k \in (\eta, 1/\beta)$, $\eta > 0$ and $\gamma := \eta/(1+\lambda\eta)$. Let*

$$P_k := \sum_{i=0}^k p_i = 1 - \Pi_{i=0}^k(\gamma_i/\eta_i).$$

Then for $\tilde{w}_k = P_k^{-1}\sum_{i=0}^k p_i w_i$ we have

1. $P_k \cdot \mathbb{E}[\tilde{w}_k] = \mathbb{E}[\hat{w}_k] - (1-P_k)\cdot\mathbb{E}[w_k]$.

2. Both $\mathbb{E}[w_k]$ and $\mathbb{E}[\hat{w}_k]$ converge. Moreover, we have $\|\mathbb{E}[\hat{w}_k] - \mathbb{E}[\tilde{w}_k]\|_2 \le \mathcal{O}((1-\lambda\gamma)^k)$.

3. If the noise $\epsilon_k = Q^{-1}(\nabla\ell(x_k, y_k, w) - \nabla L(w))$ has uniform bounded variance $\mathbb{E}[\|\epsilon_k\|_2^2] \le \sigma^2$, then for k large enough, with probability at least $1-\delta$ we have

$$\|P_k\tilde{w}_k - P_k\mathbb{E}[\tilde{w}_k]\|_2 \le \epsilon,$$

where $\epsilon = \frac{\sigma\|Q\|_2}{\gamma(\lambda+\alpha)(\lambda+\beta)^2}\sqrt{\frac{\lambda}{\delta\gamma(2-\lambda\gamma)}}$.

The proof is left in Supplementary Materials, Section C.3. A 2-D illustration for Theorem 2 is presented in Figure 2.

The importance of Theorem 2 is two-folds. On the one hand, averaging the PSGD path has an effect as the generalized ℓ_2-regularizer. And as before, this induced regularization is both adjustable and costless. The considered PSGD algorithm applies to natural gradient descent and Newton's method in certain circumstances where the curvature matrices can be replaced by constant matrices (Martens, 2014; Dennis Jr & Schnabel, 1996; Bottou & Bousquet, 2008). On the other hand, to obtain a desired type of generalized ℓ_2-regularization effect, we should store and average a PSGD path with the corresponding preconditioning matrix as indicated in Theorem 2, instead of using a SGD path.

3.3. The effect of an averaged NSGD path

In this part, we show how to obtain adjustable regularization effect by applying averaging schemes on the NSGD path.

Theorem 3 (The effect of an averaged NSGD path)**.** *Consider loss function $L(w) = \frac{1}{2n}\sum_{i=1}^n \left\|w^\top x_i - y_i\right\|_2^2$, and regularizer $R(w) = \frac{1}{2}\|w\|_2^2$. Let α and β be such that $L(w)$ is α-strongly convex and β-smooth. Let $\{w_k\}_{k=0}^\infty$ and $\{\hat{w}_k\}_{k=0}^\infty$ be the NSGD paths for the vanilla loss function $L(w)$ with learning rate η, and the regularized loss function $L(\hat{w}) + \lambda R(\hat{w})$ with learning rate γ, respectively. Suppose $1 - \lambda\gamma = \gamma/\eta$, $\eta \in (0, 1/\beta)$. Let*

$$P_k := \sum_{i=0}^k p_k = 1 - \frac{\gamma}{\eta}\left(\frac{1-\sqrt{\gamma(\alpha+\lambda)}}{1-\sqrt{\eta\alpha}}\right)^{k-1}.$$

Then for $\tilde{w}_k = P_k^{-1}\sum_{i=0}^k p_i w_i$ we have

1. $P_k \cdot \mathbb{E}[\tilde{w}_k] = \mathbb{E}[\hat{w}_k] - (1-P_k)\cdot\mathbb{E}[w_k]$.

2. $\mathbb{E}[w_k]$ and $\mathbb{E}[\hat{w}_k]$ converge. And $\|\mathbb{E}[\hat{w}_k] - \mathbb{E}[\tilde{w}_k]\|_2 \le \mathcal{O}\left(C^k\right)$, where $C = \frac{1-\sqrt{\gamma(\alpha+\lambda)}}{1-\sqrt{\eta\alpha}} \in (0,1)$.

3. If the gradient noise $\epsilon_k = \nabla\ell(x_k, y_k, w) - \nabla L(w)$ has uniformly bounded variance $\mathbb{E}[\|\epsilon_k\|_2^2] \le \sigma^2$, then for k large enough, with probability at least $1-\delta$ we have

$$\|P_k\tilde{w}_k - P_k\mathbb{E}[\tilde{w}_k]\|_2 \le \epsilon,$$

where ϵ depends on $\sigma, \alpha, \beta, \eta, \gamma$.

The proof and the exact value of ϵ are given in Supplementary Materials, Section C.4. A 2-D illustration for Theorem 3 is presented in Figure 3.

Theorem 3 affirmatively answers an open question raised by Neu & Rosasco (2018): there exists an averaging scheme for NSGD to achieve ℓ_2-regularization in arbitrary strength. In addition to the results for averaging SGD, Theorem 3 provides us wider choices of applicable optimizers for obtaining adjustable ℓ_2-regularization effect by iterate averaging.

3.4. The effect of an averaged GD path for strongly convex and smooth loss functions

In this section, we show that the iterate averaging methods work for not only simple optimization objectives like least square, but also a much broader set of loss functions. In fact, we show that any strongly convex and smooth function admits an iterate averaging scheme, which brings ℓ_2-regularization effect in a tunable manner. More formally, in the problems ($\mathcal{P}_1$) and ($\mathcal{P}_2$), let $L(w)$ be α-strongly convex and β-smooth, and $R(w) := \frac{1}{2}\|w\|_2^2$ be the ℓ_2-regularizer. For the sake of representation, we focus on gradient descent (GD) with constant learning rate applied on the loss functions. Similar arguments can also be applied for SGD, PSGD and NSGD. The GD takes update

$$w_{k+1} = w_k - \eta \nabla L(w_k),$$
$$\hat{w}_{k+1,\lambda} = \hat{w}_{k,\lambda} - \gamma(\nabla L(\hat{w}_{k,\lambda}) + \lambda \hat{w}_{k,\lambda}),$$

for optimizing problems ($\mathcal{P}_1$) and ($\mathcal{P}_2$), respectively. Let $b = -\nabla L(w_0) = -\nabla L(0)$. Let us denote two iterations

$$u_{k+1} - u_k = -\eta(\alpha u_k - b), \quad v_{k+1} - v_k = -\eta(\beta v_k - b),$$

where $u_0 = v_0 = 0$. Consider an averaging scheme $P_k = \sum_{i=0}^k p_i = 1 - (\gamma/\eta)^{k+1}$. Let $\tilde{u}_k = {P_k}^{-1} \sum_{i=0}^k p_i u_i$, $\tilde{v}_k = {P_k}^{-1} \sum_{i=0}^k p_i v_i$, and $\tilde{w}_k = P_k^{-1} \sum_{i=0}^k p_i w_i$. Then the next theorem characterizes the regularization effect of a averaged GD path for general strongly convex and smooth loss functions.

Theorem 4 (The effect of an averaged GD path for strongly convex and smooth loss functions)**.** *Without loss of generality, assume the unique minimum w_* of $L(w)$ satisfies $w_* > w_0 = 0$ entry-wisely. Suppose $1/(2\beta - \alpha) < \eta < 1/\beta$, $0 < \gamma < \eta/(\eta(\beta - \alpha) + 1)$. Then for hyperparameters*

$$\lambda_1 = 1/\gamma - 1/\eta + \beta - \alpha, \quad \lambda_2 = 1/\gamma - 1/\eta + \alpha - \beta,$$

we have

1. $\hat{w}_{k,\lambda_1} + (1 - P_k)(\tilde{v}_k - v_k) \le \tilde{w}_k \le \hat{w}_{k,\lambda_2} + (1 - P_k)(\tilde{u}_k - u_k)$, where the "$\le$" is defined entry-wisely.

2. $u_k, \tilde{u}_k, v_k, \tilde{v}_k, \hat{w}_{k,\lambda_1}, \hat{w}_{k,\lambda_2}$ converge. Moreover let $m = (\hat{w}_{\infty,\lambda_2} + \hat{w}_{\infty,\lambda_1})/2$, $d = (\hat{w}_{\infty,\lambda_2} - \hat{w}_{\infty,\lambda_1})/2$ and $C = \max\{(1 - \gamma(\alpha + \lambda_1), (1 - \gamma(\alpha + \lambda_2), \frac{\gamma}{\eta}\} \in (0, 1)$, then $\|\tilde{w}_k - m\|_2 \le \|d\|_2 + \mathcal{O}(C^k)$.

The proof is left in Supplementary Materials, Section C.5 .

According to Theorem 4, for strongly convex and smooth objectives, the averaged GD path $\{\tilde{w}_k\}_{k=0}^\infty$ lies in the area between two regularized GD paths, $\{\hat{w}_{k,\lambda_1}\}_{k=0}^\infty$ and $\{\hat{w}_{k,\lambda_2}\}_{k=0}^\infty$. Furthermore, $\tilde{w}_k$ converges to a hyper cube whose diagonal vertices are defined by $\hat{w}_{\infty,\lambda_1}$ and $\hat{w}_{\infty,\lambda_2}$. In this way for this class of loss functions, averaging the GD path has an "approximate" ℓ_2-regularization effect that is in between two ℓ_2-regularizers with hyperparameters as λ_1 and λ_2 respectively. In addition, λ_1 and λ_2 can be adjusted through changing the weighting scheme. Finally, we note that $\|d\|_2 = \mathcal{O}(\beta - \alpha)$, thus when the objective is quadratic, we have $\alpha = \beta$ and $d = 0$ and thus the "approximate" ℓ_2-regularization effect becomes the exact ℓ_2-regularization by Theorem 4.

We therefore conjecture that, generally, *for arbitrary loss functions and iterative optimizers, an iterate averaging scheme admits a specific yet unknown regularization effect.* Indeed, our experiments in the next section empirically verifies such an effect by performing iterate averaging on deep neural networks, which are highly comprehensive.

4. Experiments

In this section we present our empirical studies. The detailed setups are explained in Supplementary Materials, Section D. The code is available at `https://github.com/uuujf/IterAvg`.

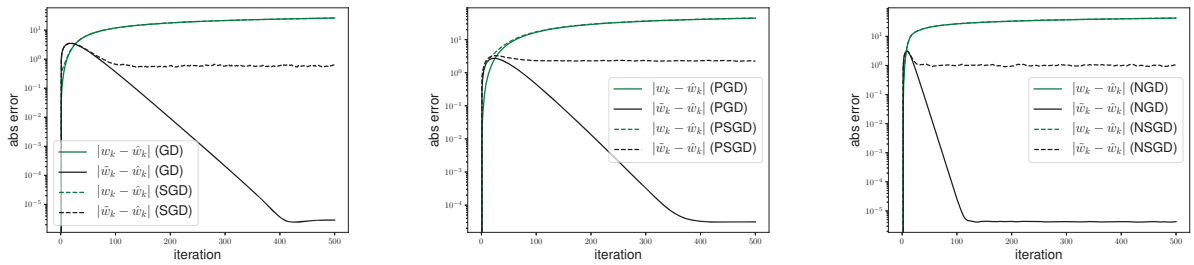
4.1. Two dimensional demonstration

We first introduce a two dimensional toy example to demonstrate the regularization effect of iterate averaging. The vanilla loss function is quadratic with a unique minimum at $(1, 1)$, as shown in Figure 1∼3. For the purpose of demonstration we only run deterministic algorithms with constant learning rates. We plot the trajectories of the concerned optimizers for learning the vanilla loss function and the regularized loss function, as well as the averaged solutions. All of the optimizers start iterations from zero.

In Figure 1, the green and the blue dots represent the GD paths for optimizing the vanilla/regularized loss functions respectively, while the red dots are the path of iterate averaged solutions. We observe that the red dots do converge to the blue ones, indicating the averaged solution has the same effect of an ℓ_2-regularizer, as suggested by Theorem 1. Similarly the phenomenon holds for averaging the NGD path, as indicated in Figure 3. In Figure 2, the preconditioning matrix is set to be the Hessian. And as predicted by Theorem 2, the averaged solution converges to the solution biased by a generalized ℓ_2-regularizer.

4.2. Real data verification

We then present experiments on the MNIST dataset.



(a) GD and SGD (b) PGD and PSGD (c) NGD and PNGD

Figure 4. Linear regression on MNIST dataset. X-axis: iteration; y-axis: the absolute approximation error in logarithmic scale. Green lines represent $\|w_t - \hat{w}_t\|_1$ and black lines represent $\|\tilde{w}_t - \hat{w}_t\|_1$, where w_t, $\hat{w}_t$ and $\tilde{w}_t$ are the unregularized path, the regularized path and the iterate averaged path, respectively. Solid lines and dashed lines are the results obtained by running deterministic and stochastic algorithms respectively. For deterministic algorithms, the error between $\tilde{w}_t$ and $\hat{w}_t$ converges to zero. For stochastic algorithms, the error between $\tilde{w}_t$ and $\hat{w}_t$ remains small.

Linear regression Firstly, we study linear regression under quadratic loss functions and the regularization effects caused by averaging the optimization paths of (S)GD, P(S)GD and N(S)GD. The learning rates are set to be constant. For P(S)GD, we set the preconditioning matrix as the Hessian, which is known as the Newton's method.

Our theories predict that averaging the (S)GD and N(S)GD paths leads to the solutions biased by ℓ_2-regularizers (Theorem 1, 3), while averaging the P(S)GD path introduces an effect of the generalized ℓ_2-regularization (Theorem 2). To verify the predictions, we generate the paths of the averaged solutions $\tilde{w}_k$ and the regularized solutions $\hat{w}_k$, and then compute the approximation errors between them. The results are plotted in Figure 4.

In Figure 4 (a), the solid lines clearly indicate that the averaged solution converges to the regularized solution when running GD, which also corresponds to the convergence in expectation in SGD cases, as predicted by Theorem 1. For SGD, however, the dashed lines in Figure 4 (a) show that there is a small error between the averaged solution and the regularized solution. The error exists since the convergence of the averaged solution does not hold in probability. Luckily, the error would not grow large as the deviation of the averaged solution is controllable by Theorem 1. Hence by comparing the dashed green and black lines, we see that averaging the SGD path still leads to an effect of ℓ_2-regularization ignoring a tolerable error.

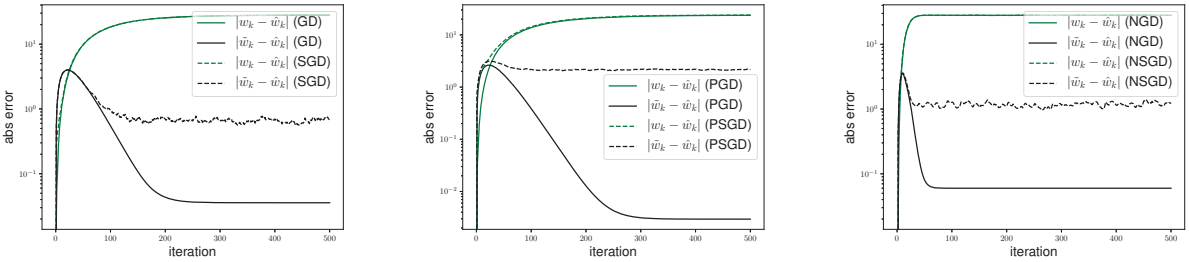
Figure 4 (b) shows the results for PGD and PSGD. Again, averaging the PGD path causes a perfect generalized ℓ_2-regularization effect, and there is a small gap for averaging the optimization path with noise. These support Theorem 2.

The results related to NGD and NSGD are shown in Figure 4 (c). Again, for the deterministic algorithm, the solid lines manifest the convergence between the averaged solution and the regularized solution, verifying our Theorem 3. And the dashed lines once more suggest the stochastic algorithm causes a tolerable approximation error.

Logistic regression Next we set the loss function $L(w)$ to be the logistic regression objective with a small ℓ_2-regularizer, which is then strongly convex and smooth, as required by Theorem 4. We firstly generate the unregularized paths and perform iterate averaging over them. Next, since it is impossible to visualize a high dimensional cubic with vertices decided by Theorem 4, instead we set $\lambda = 1/\gamma - 1/\eta$, and add an extra regularization term with this particular hyperparameter to obtain the regularized paths. Lastly we measure the errors between the averaged solutions and the regularized solutions to verify the effect of iterate averaging applied on strongly convex and smooth loss functions. The learning rates are set to be constant. The approximation errors are plotted in Figure 5.

In Figure 5 (a), the solid black line measures the error between the averaged GD path and the regularized GD path, and indeed the error is bounded and small as predicted by Theorem 4; the dashed black line is the result obtained by running SGD, which suggests that the approximation error, though increases a little due to randomness, is still small.

For completeness, we also test P(S)GD and N(S)GD with results shown in Figure 5 (b) and (c). For P(S)GD, we use the Hessian in linear regression experiments as the preconditioning matrix (since the Hessian of the logistic loss varies during training). Figure 5 (b) and (c) show that the averaged solutions approximately achieve the generalized/vanilla ℓ_2-regularization effects respectively. And for the stochastic optimization paths, the approximation errors between the averaged paths and the regularized paths increase by a small amount due to the randomness of the algorithms.



(a) GD and SGD (b) PGD and PSGD (c) NGD and NSGD

Figure 5. Logistic regression on MNIST dataset. X-axis: iteration; y-axis: the absolute approximation error in logarithmic scale. Green lines represent $\|w_t - \hat{w}_t\|_1$ and black lines represent $\|\tilde{w}_t - \hat{w}_t\|_1$, where w_t, $\hat{w}_t$ and $\tilde{w}_t$ are the unregularized path, the regularized path and the iterate averaged path, respectively. Solid lines and dashed lines are the results obtained by running deterministic and stochastic algorithms respectively. For both deterministic and stochastic algorithms, we see that the error between $\tilde{w}_t$ and $\hat{w}_t$ has a small upper bound. Moreover, the error bounds for the paths generated by stochastic algorithms are relatively bigger.

Table 1. CIFAR-10 and CIFAR-100 experiments

Dataset	CIFAR-10		CIFAR-100
Model	VGG-16	ResNet-18	ResNet-18
Accuracy after training (%)	92.54 ±0.22	94.54 ±0.04	75.62 ±0.16
Accuracy after averaging (%)	**93.18** ±0.06	**94.72** ±0.04	**76.24** ±0.05
Time of training	$\sim$ 4.5h	$\sim$ 8.3h	$\sim$ 8.3h
Time of averaging[3]	$\sim$ 47s	$\sim$ 56s	$\sim$ 58s

4.3. Application in deep neural networks

Lastly, we study the benefits of using iterate averaging in modern deep neural networks.

We train VGG-16 (Simonyan & Zisserman, 2014) and ResNet-18 (He et al., 2016) on CIFAR-10 and CIFAR-100 datasets, with standard tricks including batch normalization, data augmentation, learning rate decay and weight decay. All experiments are repeated three times to obtain the mean and deviation. The running times are measured by performing the experiments using a single GPU K80. The models are trained for 300 epochs using SGD. We perform "epoch averaging" using the 240 checkpoints saved from the 61st to the 300th epoch. The first 60 epochs are skipped since the models in the early phase are extremely unstable. After averaging the parameters, we apply a trick proposed by Izmailov et al. (2018) to handle the batch normalization statistics which are not trained by SGD. Specifically, we make a forward pass on the training data to compute the activation statistics for the batch normalization layers. For the choice of averaging scheme, we test standard geometric distribution with success probability $p \in \{0.9999, 0.999, 0.99, 0.9\}$.

[3]The time of averaging contains the time of IO and fixing BN, which takes the major overhead. For example, in CIFAR-10 and VGG-16 experiments, IO takes $\sim$ 22s, fixing BN takes $\sim$ 18s, while performing averaging and evaluation take merely $\sim$ 7s.

The results are shown in Table 1. We see that (i) averaging the SGD path does improve performance since it introduce an implicit regularization by our understanding; (ii) obtaining such regularization by iterate averaging is computationally cheap. It only takes *a few seconds* to test a hyperparameter of the averaging scheme. In contrast, several hours are required to test a hyperparameter for traditional explicit regularization since it requires re-training the model. Finally, we emphasize that the space cost of our method is also affordable. In fact, in our experiments, we perform epoch-wise averaging instead of iterate-wise averaging, thus we only need to store a few hundreds of the checkpoints.

5. Discussion

ℓ_1-regularization Notice that all our results obtain ℓ_2-type regularization effects. A natural follow-up question would be whether or not there is an averaging scheme that acts as an ℓ_1-regularizer. However, we here provide some evidence that this question is relatively hard. As illustrated in Figure 6, even for simple quadratic loss, the ℓ_1-regularized solutions could lie outside of the convex hull of a SGD path. Therefore, any averaging scheme with positive weights fails to obtain such ℓ_1-regularized solutions.

Infinite width neural network Recent works suggest that a sufficient wide neural network trained by SGD behaves like a quadratic model, i.e., the neural tangent kernel (NTK) (Jacot et al., 2018; Arora et al., 2019; Cao & Gu, 2019). Nonetheless, the NTK approximation fails when there is an explicit ℓ_2-regularizer (Wei et al., 2019). Since our results hold for kernel ridge regression, we conjecture that iterate averaging could be a potential approach to achieve ℓ_2-regularization for the NTK regime. We leave

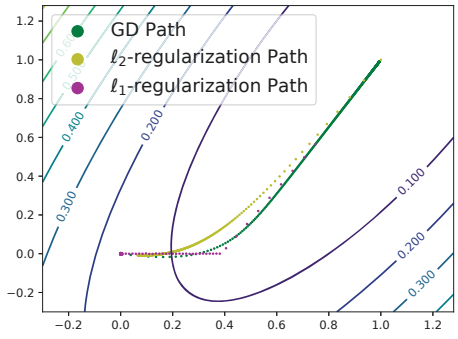


Figure 6. A 2-D demonstration of the ℓ_1-regularization path. Green dots: the vanilla GD path w_t; yellow dots: the ℓ_2-regularization path $\hat{w}_{\lambda,\ell_2}$; purple dots: the ℓ_1-regularization path $\hat{w}_{\lambda,\ell_1}$. There exist ℓ_1-regularized solutions outside of the convex hull of the GD path, while all of the ℓ_2-regularized solutions are inside of that.

further investigation of this issue in future works.

6. Conclusions

In this work, we establish averaging schemes for various optimization methods and objective functions to obtain adjustable ℓ_2-type regularization effects, i.e., SGD with preconditioning and adaptive learning rate schedules, Nesterov's accelerated stochastic gradient descent, and strongly convex and smooth objective functions. Particularly, we resolve an open question in (Neu & Rosasco, 2018). The method of achieving regularization by iterate averaging requires little computation. It is further shown experimentally that iterate averaging even benefits practical deep learning models. Our theoretical and empirical results demonstrate the potential of adopting iterate averaging to obtain adjustable regularization for free in a much broader class of optimization methods and objective functions.

Acknowledgement

This research is supported in part by NSF CAREER grant 1652257, ONR Award N00014-18-1-2364 and the Lifelong Learning Machines program from DARPA/MTO.

References

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019.

Bach, F. and Moulines, E. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Advances in neural information processing systems*, pp. 773–781, 2013.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

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

Cai, Y., Li, Q., and Shen, Z. A quantitative analysis of the effect of batch normalization on gradient descent. *arXiv preprint arXiv:1810.00122*, 2018.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *arXiv preprint arXiv:1905.13210*, 2019.

Clark, D. S. Short proof of a discrete gronwall inequality. *Discrete applied mathematics*, 16(3):279–281, 1987.

Dennis Jr, J. E. and Schnabel, R. B. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16. Siam, 1996.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pp. 529–536, 2005.

Granziol, D., Wan, X., and Roberts, S. Iterate averaging helps: An alternative perspective in deep learning. *arXiv preprint arXiv:2003.01247*, 2020.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. 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.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. doi: 10.1109/cvpr.2016.90. URL http://dx.doi.org/10.1109/CVPR.2016.90.

Hu, W., Li, C. J., Li, L., and Liu, J.-G. On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv preprint arXiv:1705.07562*, 2017a.

Hu, W., Li, C. J., and Su, W. On the global convergence of a randomly perturbed dissipative nonlinear oscillator. *arXiv preprint arXiv:1712.05733*, 2017b.

Hu, W., Xiao, L., and Pennington, J. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*, 2020.

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

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Jain, P., Kakade, S., Kidambi, R., Netrapalli, P., and Sidford, A. Parallelizing stochastic gradient descent for least squares regression: mini-batching, averaging, and model misspecification. *Journal of Machine Learning Research*, 18, 2018.

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

Lakshminarayanan, C. and Szepesvari, C. Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *International Conference on Artificial Intelligence and Statistics*, pp. 1347–1355, 2018.

Li, Q., Tai, C., et al. Stochastic modified equations and adaptive stochastic gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2101–2110. JMLR. org, 2017.

Martens, J. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Nesterov, Y. E. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

Neu, G. and Rosasco, L. Iterate averaging as regularization for stochastic gradient descent. *arXiv preprint arXiv:1802.08009*, 2018.

Shi, B., Du, S. S., Su, W., and Jordan, M. I. Acceleration via symplectic discretization of high-resolution differential equations. In *Advances in Neural Information Processing Systems*, pp. 5745–5753, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

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

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Su, W., Boyd, S., and Candes, E. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pp. 2510–2518, 2014.

Suggala, A., Prasad, A., and Ravikumar, P. K. Connecting optimization and regularization paths. In *Advances in Neural Information Processing Systems*, pp. 10608–10619, 2018.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Tikhonov, A. N. and Arsenin, V. Y. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. In *Advances in Neural Information Processing Systems*, pp. 9709–9721, 2019.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

Yang, L., Arora, R., Zhao, T., et al. The physical systems behind optimization algorithms. In *Advances in Neural Information Processing Systems*, pp. 4372–4381, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhang, M., Lucas, J., Ba, J., and Hinton, G. E. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pp. 9593–9604, 2019.

Zhou, X. On the fenchel duality between strong convexity and lipschitz continuous gradient. *arXiv preprint arXiv:1803.06573*, 2018.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.