

# A GENERAL FAMILY OF STOCHASTIC PROXIMAL GRADIENT METHODS FOR DEEP LEARNING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

We study the training of regularized neural networks where the regularizer can be non-smooth and non-convex. We propose a unified framework for stochastic proximal gradient descent, which we term PROXGEN, that allows for arbitrary positive preconditioners and lower semi-continuous regularizers. Our framework naturally encompasses standard stochastic proximal gradient methods *without* preconditioners as special cases. We present two important instances stemming from our approach: (i) the first proximal version of ADAM, one of the most popular adaptive SGD algorithm, and (ii) a revised version of PROXQUANT (Bai et al., 2019) that improves upon the original approach for quantization-specific regularizers by incorporating the effect of preconditioners when computing proximal mapping. We analyze the convergence of PROXGEN and show that the whole framework enjoys the same convergence rate as stochastic proximal gradient descent without preconditioners. We also empirically show the superiority of proximal methods compared to subgradient-based approaches via extensive experiments. Interestingly, our results indicate that proximal methods with non-convex regularizers are more effective than those with convex regularizers.

## 1 INTRODUCTION

We study the regularized training of neural networks, which can be formulated as the following (stochastic) optimization problem

$$\underset{\theta \in \Omega}{\text{minimize}} \quad F(\theta) := \overbrace{\mathbb{E}_{\xi \sim \mathbb{P}} [f(\theta; \xi)]}^{f(\theta)} + \mathcal{R}(\theta) \quad (1)$$

where  $\theta \in \mathbb{R}^p$  represents the network parameter,  $\xi$  is the random variable representing mini-batch data samples, and  $\mathcal{R}(\cdot)$  is a regularizer encouraging low-dimensional structural constraints on  $\theta$ .

For the *unregularized* case, i.e., when  $\mathcal{R}(\theta) = 0$ , stochastic gradient descent (SGD) has been a prevalent approach to solve the optimization problem stated in Eq. (1). At each iteration, SGD evaluates the gradient only on a randomly chosen subset of training samples (mini-batch). Vanilla SGD employs a uniform learning rate for all coordinates, and several adaptive variants have been proposed, which scale the learning rate for each coordinate by its gradient history. A prime example of such approaches is ADAGRAD (Duchi et al., 2011), which adjusts the learning rate by the sum of all the past squared gradients. However, the performance of ADAGRAD degrades in non-convex dense settings as the learning rates vanish too rapidly. To resolve this issue, exponential moving average (EMA) approaches such as RMSPROP (Tieleman & Hinton, 2012) and ADAM (Kingma & Ba, 2015) have been proposed and become popular. These scale down the gradients by square roots of exponential moving averages of squared past gradients to essentially limit the scope of the adaptation to only a few recent gradients. In terms of theory, convergence analyses of these unregularized SGD, whether adaptive or not, have been well studied both for convex (Kingma & Ba, 2015; Reddi et al., 2018) and non-convex (Chen et al., 2019b; Lei et al., 2019) loss  $f$  cases.

The technique of regularization is ubiquitous in machine learning as it can effectively prevent overfitting and yield better generalization. The  $\ell_1$ -regularized training for Lasso estimators/sparse Gaussian graphical model (GMRF) estimation (Tibshirani, 1996; Ravikumar et al., 2011) and  $\ell_2$

Table 1: Comparison among *stochastic* (or *online*) PGD for solving the problem in Eq. (1).

Algorithm	Non-convex Loss	Non-convex Regularizer	Preconditioner	Momentum	Convergence Guarantee
ADAGRAD (Duchi et al., 2011)			ADAGRAD		✓
Ghadimi et al. (2016)	✓		✓		✓
Wang et al. (2018)	✓			✓	✓
Pham et al. (2019)	✓			✓	✓
Davis & Drusvyatskiy (2019)	✓			✓	✓
Xu et al. (2019a)	✓	✓			✓
Xu et al. (2019b)	✓	✓	ADAGRAD		✓
Prox-SGD (Yang et al., 2020)	✓		✓	✓	
Davis et al. (2020)	✓	✓		✓	✓
PROXGEN (Ours)	✓	✓	✓	✓	✓

weight decay (Tychonoff, 1943) on parameters are prototypical examples. In the context of deep learning, important instances include network pruning (Wen et al., 2016; Louizos et al., 2018), which induces a sparse network structure, and network quantization (Yang et al., 2019; Courbariaux et al., 2015; Bai et al., 2019), which gives hard constraints so that parameters have only discrete values.

In many cases, the regularizer is non-smooth around some region (Consider  $\ell_1$  norm at zero). Therefore, instead of using the gradient, one employs the subgradient of the objective function  $F(\theta)$  in Eq. (1). Such a strategy, which is essentially adopted in modern machine learning libraries such as TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019), is problematic as it may slow down convergence and result in oscillations. A simple idea to tackle this issue is to bypass the non-smoothness of a regularizer via its proximal operator. This idea is the basis of proximal gradient descent (PGD) methods, which first update the parameter using the gradient of the loss function  $f(\theta)$  and then perform a proximal mapping of  $\mathcal{R}(\theta)$ .

In the *non-stochastic* case, the PGD with both convex and non-convex regularizers has been extensively studied in the literature (Reddi et al., 2016; Allen-Zhu, 2017; Wang et al., 2018; Pham et al., 2019; Chen et al., 2020). Another work, VMFB (Chouzenoux et al., 2014), analyzes the preconditioned gradient descent on convex regularized problems with non-convex loss but does not consider the first-order momentum. In contrast, PGD in the *stochastic* setting has been little explored. Duchi et al. (2011); Ghadimi et al. (2016) consider PGD to solve the stochastic objectives with convex regularizers. Recently, Xu et al. (2019b) studies non-convex and non-smooth regularized problems for DC (difference of convex) functions and Xu et al. (2019a); Davis et al. (2020) present non-asymptotic analysis for non-convex smooth loss and non-convex regularizers, which is the most general setting, but do not consider the preconditioner in the update rule.

All the aforementioned studies of the stochastic case, however, focus either on limited settings (e.g. Duchi et al. (2011) only covers the update rule of ADAGRAD) with convex regularizers only, or on pure vanilla gradient descent for non-convex regularizers. Hence, they cannot accommodate all advanced modern optimization algorithms with *preconditioners*, such as adaptive gradient methods. The only exception is PROX-SGD (Yang et al., 2020), with the caveat that PROX-SGD update rule is *not an exact* PGD. Moreover, the theory in Yang et al. (2020) only guarantees the convergence, *not how fast* Prox-SGD converges, and furthermore this analysis is performed without considering the preconditioners. Table 1 summarizes the previous studies and our work in terms of stochastic PGD.

In this paper, we propose an exact framework for stochastic proximal gradient methods with *arbitrary* positive preconditioners and lower semi-continuous (possibly non-convex) regularizers. With our framework, our goal is to provide theoretical and empirical understanding of stochastic proximal gradient methods. Our main contributions are summarized as follows:

- We propose the first general family of stochastic proximal gradient methods, which we term PROXGEN. We introduce two important instances stemming from our approach: (i) the first proximal version of ADAM (Kingma & Ba, 2015) and (ii) a revised version of PROXQUANT (Bai et al., 2019) that improves upon the original approach for quantization-specific regularizers by incorporating the effect of preconditioners when computing proximal mappings.

**Algorithm 1** PROXGEN: A General Stochastic Proximal Gradient Method

---

```

1: Input: Stepsize  $\alpha_t$ ,  $\{\rho_t\}_{t=1}^{t=T} \in [0, 1)$ , regularization parameter  $\lambda$ , and small constant  $0 < \delta \ll 1$ .
2: Initialize:  $\theta_1 \in \mathbb{R}^d$ ,  $m_0 = 0 \in \mathbb{R}^d$ , and  $C_0 = O \in \mathbb{R}^{d \times d}$ .
3: for  $t = 1, 2, \dots, T$  do
4:   Draw a minibatch sample  $\xi_t$  from  $\mathbb{P}$ 
5:    $g_t \leftarrow \nabla f(\theta_t; \xi_t)$  ▷ Stochastic gradient at time  $t$ 
6:    $m_t \leftarrow \rho_t m_{t-1} + (1 - \rho_t) g_t$  ▷ First-order momentum estimate
7:    $C_t \leftarrow$  Preconditioner construction
8:    $\theta_{t+1} \in \operatorname{argmin}_{\theta \in \Omega} \left\{ \langle m_t, \theta \rangle + \lambda \mathcal{R}(\theta) + \frac{1}{2\alpha_t} (\theta - \theta_t)^\top (C_t + \delta I) (\theta - \theta_t) \right\}$ 
9: end for
10: Output:  $\theta_{T+1}$ 

```

---

- We analyze the convergence of the general PROXGEN family and identify essential conditions for convergence. We show that PROXGEN enjoys the same convergence rate as vanilla SGD under mild conditions, and highlight the challenges in our theory and improvements upon previous work. Our convergence guarantee encompasses several existing approaches as special cases.
- In terms of practice, we demonstrate the superiority of proximal methods over subgradient-based methods with various non-convex regularizers which have not yet been studied in deep learning. Interestingly, our experiments indicate that proximal methods with non-convex regularizers are more effective than with convex regularizers for learning sparse deep models.

## 2 A GENERAL FAMILY OF STOCHASTIC PROXIMAL GRADIENT METHODS

In this section, we present PROXGEN, a general family of stochastic proximal gradient methods, and present both existing and novel instances as showcase examples in our family. Algorithm 1 describes the details of PROXGEN. The update rule on line 8 of Algorithm 1 can be written more compactly:

$$\begin{aligned}
 \theta_{t+1} &\in \operatorname{argmin}_{\theta \in \Omega} \left\{ \langle m_t, \theta \rangle + \lambda \mathcal{R}(\theta) + \frac{1}{2\alpha_t} (\theta - \theta_t)^\top (C_t + \delta I) (\theta - \theta_t) \right\} \\
 &= \operatorname{prox}_{\alpha_t \lambda \mathcal{R}(\cdot)}^{C_t + \delta I} \left( \theta_t - \alpha_t (C_t + \delta I)^{-1} m_t \right)
 \end{aligned} \tag{2}$$

where the proximal operator in Eq. (2) is defined as  $\operatorname{prox}_h^A(z) = \operatorname{argmin}_x \{h(x) + \frac{1}{2}\|x - z\|_A^2\}$ . In PROXGEN, we allow both the loss and the regularizer to be non-convex. Now, we introduce possible examples according to the proper combinations of preconditioners  $C_t$  and regularizers  $\mathcal{R}(\cdot)$ .

**Existing Instances of PROXGEN.** We briefly recover some known examples in PROXGEN family.

- ADAGRAD (Duchi et al., 2011) is the first key instance of adaptive gradient methods where  $C_t = (\sum_{\tau=1}^t g_\tau g_\tau^\top)^{1/2}$  and  $\mathcal{R}(\theta) = \|\theta\|_1$ . Any convex regularizer  $\mathcal{R}(\cdot)$  is allowed.
- The proximal Newton methods (Lee et al., 2012) employ the exact Hessian  $C_t = \nabla^2 f(\theta_t)$  and  $\mathcal{R}(\theta) = \|\theta\|_1$ . In addition, we can approximate the exact Hessian, which yield proximal Newton-type methods such as quasi-Newton approximation (Becker et al., 2019), L-BFGS approximation (Liu & Nocedal, 1989), and adding a multiple of the identity to the Hessian.

Although the above examples enjoy good theoretical properties in convex settings, many of the modern practical optimization problems involve non-convex loss functions such as learning deep models. Moreover, it is known that non-convex regularizers yield better performance (also in terms of theory) than convex penalties in some applications (see Fu (1998); Park & Yoon (2011); Yang & Lozano (2017); Yun et al. (2019b) and references therein). Considering this motivation and recent advanced optimizers, we arrive at the following new examples.

**Novel Instances of PROXGEN.** Beyond the well-known methods above, PROXGEN naturally introduces proximal versions of standard SGD techniques developed for solving unregularized problems for deep learning. The following examples are just a few instances that have not been explored so far, and PROXGEN can cover a broader range of new examples depending on the combinations of preconditioners and regularizers.

- The *proximal version* of ADAM (Kingma & Ba, 2015) with  $\ell_q$  regularization is a possible example where  $C_t = \sqrt{\beta C_{t-1} + (1-\beta)g_t^2}$  with  $\beta \in [0, 1)$  and  $\mathcal{R}(\theta) = \|\theta\|_q$  for  $0 \leq q \leq 1$ . We mainly validate the superiority of our novel *proximal version* of ADAM to the usual subgradient-based counterpart empirically in Section 4.
- We can also consider the *proximal version* of KFAC (Martens & Grosse, 2015). For an  $L$ -layer neural network, KFAC approximates the Fisher information matrix with layer-wise block diagonal structure where  $l$ -th diagonal block  $C_{t,[l]}$  corresponds to Kronecker-factored approximation with respect to the parameters at  $l$ -th layer. The proximal version of KFAC, which corresponds to  $C_{t,[l]} = \mathbb{E}[\delta_l \delta_l^T] \otimes \mathbb{E}[a_{l-1} a_{l-1}^T]$  and  $\mathcal{R}(\theta) = \|\theta\|_q$  where  $\delta_l$  is the gradient with respect to the outputs of  $l$ -th layer and  $a_{l-1}$  is the activation of  $(l-1)$ -th layer, could be another example.

**Examples of Proximal Mappings for PROXGEN.** We provide update rules for PROXGEN with  $\ell_q$  regularization ( $0 \leq q \leq 1$ ) and diagonal preconditioners. Diagonal preconditioners are used by popular adaptive gradient methods such as ADAM. Specifically, we consider regularizer  $\mathcal{R}(\theta) = \lambda \sum_{j=1}^p |\theta_j|^q$  for  $\theta \in \mathbb{R}^p$  with diagonal preconditioner matrix  $C_t$ . Note that for  $C_t = I$  (i.e. vanilla gradient descent), it is known that closed-form solutions exist for proximal mappings for  $q \in \{0, \frac{1}{2}, \frac{2}{3}, 1\}$  (Cao et al., 2013). We denote the  $i$ -th coordinate of the vector  $\theta_t$  as  $\theta_{t,i}$  and the diagonal entry  $[C_t]_{ii}$  as  $C_{t,i}$ .

- **$\ell_1$  regularization.** The proximal mappings for the case of  $\ell_1$  regularization with preconditioner can be computed efficiently via soft-thresholding as

$$\hat{\theta}_{t,i} = \theta_{t,i} - \alpha_t \frac{m_{t,i}}{C_{t,i} + \delta}, \quad \theta_{t+1,i} = \text{sign}(\hat{\theta}_{t,i}) \left( |\hat{\theta}_{t,i}| - \frac{\alpha_t \lambda}{C_{t,i} + \delta} \right) \quad (3)$$

- **$\ell_0$  regularization.** In case of  $\ell_0$  regularization, we can compute the closed-form solutions via hard-thresholding as

$$\hat{\theta}_{t,i} = \theta_{t,i} - \alpha_t \frac{m_{t,i}}{C_{t,i} + \delta}, \quad \theta_{t+1,i} = \begin{cases} \hat{\theta}_{t,i}, & |\hat{\theta}_{t,i}| > \sqrt{\frac{2\alpha_t \lambda}{C_{t,i} + \delta}}, \\ 0, & |\hat{\theta}_{t,i}| < \sqrt{\frac{2\alpha_t \lambda}{C_{t,i} + \delta}}, \\ \{0, \hat{\theta}_{t,i}\}, & |\hat{\theta}_{t,i}| = \sqrt{\frac{2\alpha_t \lambda}{C_{t,i} + \delta}} \end{cases} \quad (4)$$

The closed-form proximal mappings for  $\ell_{1/2}$  and  $\ell_{2/3}$  regularization are provided in the Appendix.

**Revised PROXQUANT (Bai et al., 2019).** The recently proposed PROXQUANT proposes novel regularizations for network quantization. Especially for binary quantization, a W-shaped regularizer is defined as  $\mathcal{R}_{\text{bin}}(\theta) = \|\theta - \text{sign}(\theta)\|_1$  where  $\text{sign}(\theta)$  is applied on  $\theta$  in an element-wise manner. Using this regularizer, the main difference between PROXQUANT and our PROXGEN approach is shown in Table 2. Note that PROXQUANT (top in Table 2) does not consider the effect of preconditioners when computing proximal mappings. Therefore, we revise the proximal update in PROXQUANT by considering preconditioners in proximal mappings with PROXGEN (bottom in Table 2). Moreover, we also propose *generalized regularizers* motivated by  $\ell_q$  regularization for  $0 < q < 1$ :  $\mathcal{R}_{\text{bin}}^q(\theta) = \|\theta - \text{sign}(\theta)\|_q$ . In terms of theory, Bai et al. (2019) prove the convergence of PROXQUANT only for the *full-batch* gradient with *differentiable* regularizers, which is also guaranteed only for vanilla gradient descent. In contrast, using our *revised* PROXQUANT, we can completely bridge the gap in theory (via Theorem 1 in Section 3, which is stated for *stochastic* optimization), and we provide the *exact* update rule for solving problem in Eq. (1). We also investigate the empirical differences of PROXQUANT and our revised PROXQUANT in Section 4.

Table 2: PROXQUANT versus *revised* PROXQUANT

PROXQUANT	$\left\  \text{prox}_{\alpha_t \lambda \mathcal{R}(\cdot)} \left( \theta_t - \alpha_t (C_t + \delta I)^{-1} m_t \right) \right\ $
Revised PROXQUANT	$\left\  \text{prox}_{\alpha_t \lambda \mathcal{R}(\cdot)}^{C_t + \delta I} \left( \theta_t - \alpha_t (C_t + \delta I)^{-1} m_t \right) \right\ $

Using this regularizer, the main difference between PROXQUANT and our PROXGEN approach is shown in Table 2. Note that PROXQUANT (top in Table 2) does not consider the effect of preconditioners when computing proximal mappings. Therefore, we revise the proximal update in PROXQUANT by considering preconditioners in proximal mappings with PROXGEN (bottom in Table 2). Moreover, we also propose *generalized regularizers* motivated by  $\ell_q$  regularization for  $0 < q < 1$ :  $\mathcal{R}_{\text{bin}}^q(\theta) = \|\theta - \text{sign}(\theta)\|_q$ . In terms of theory, Bai et al. (2019) prove the convergence of PROXQUANT only for the *full-batch* gradient with *differentiable* regularizers, which is also guaranteed only for vanilla gradient descent. In contrast, using our *revised* PROXQUANT, we can completely bridge the gap in theory (via Theorem 1 in Section 3, which is stated for *stochastic* optimization), and we provide the *exact* update rule for solving problem in Eq. (1). We also investigate the empirical differences of PROXQUANT and our revised PROXQUANT in Section 4.

### 3 CONVERGENCE ANALYSIS

In this section, we provide convergence guarantees for the PROXGEN family. Our goal is to find an  $\epsilon$ -stationary point for the problem in Eq. (1) where  $\epsilon$  is the required precision. For notational convenience, we assume that the regularization parameter  $\lambda$  is incorporated into  $\mathcal{R}(\theta)$  in Eq. (1). To guarantee the convergence under this setting, we should deal with the subdifferential defined as:

**Definition 1** (Fréchet Subdifferential). Let  $\varphi$  be a real-valued function. The Fréchet subdifferential of  $\varphi$  at  $\bar{\theta}$  with  $|\varphi(\bar{\theta})| < \infty$  is defined by  $\hat{\partial}\varphi(\bar{\theta}) := \{\theta^* \in \Omega \mid \liminf_{\theta \rightarrow \bar{\theta}} \frac{\varphi(\theta) - \varphi(\bar{\theta}) - \langle \theta^*, \theta - \bar{\theta} \rangle}{\|\theta - \bar{\theta}\|} \geq 0\}$ .

To derive the convergence bound, we make the following mild conditions:

- (C-1) (*L-smoothness*) The loss function  $f$  is differentiable,  $L$ -smooth, and lower-bounded:  $\forall x, y, \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$  and  $f(x^*) > -\infty$  for the optimal solution  $x^*$ .
- (C-2) (*Bounded variance*) The stochastic gradient  $g_t = \nabla f(\theta_t; \xi)$  is unbiased and has the bounded variance:  $\mathbb{E}_\xi[\nabla f(\theta_t; \xi)] = \nabla f(\theta_t)$ ,  $\mathbb{E}_\xi[\|g_t - \nabla f(\theta_t)\|^2] \leq \sigma^2$ .
- (C-3) (i) final step-vector is finite, (ii) the stochastic gradient is bounded, and (iii) the momentum parameter should be exponentially decaying: (i)  $\|\theta_{t+1} - \theta_t\| \leq D$ , (ii)  $\|g_t\| \leq G$ , (iii)  $\rho_t = \rho_0 \mu^{t-1}$  with  $D, G > 0$  and  $\rho_0, \mu \in [0, 1)$ .
- (C-4) (*Sufficiently positive-definite*) The minimum eigenvalue of effective spectrums should be uniformly lower bounded over all time  $t$ :  $\forall t, \lambda_{\min}(\alpha_t(C_t + \delta I)^{-1}) \geq \gamma > 0$ .

(C-1) and (C-2) are standard in general non-convex optimization (Ghadimi & Lan, 2013; Ghadimi et al., 2016; Zaheer et al., 2018; Xu et al., 2019a). In addition, (C-3) is extensively studied in previous literature in the context of adaptive gradient methods (Kingma & Ba, 2015; Reddi et al., 2018; Chen et al., 2019a). Lastly, a similar condition to (C-4) is also considered in Chen et al. (2019a); Yun et al. (2019a), and it can be easily satisfied in practice. More discussion on (C-4) is provided later.

Since the loss function  $f$  is assumed to be differentiable as in (C-1), we have, at stationary points,  $\mathbf{0} \in \hat{\partial}F(\theta) = \nabla f(\theta) + \hat{\partial}\mathcal{R}(\theta)$ , so the convergence criterion is slightly different from that of general non-convex optimization. Hence, we use the following convergence criterion  $\mathbb{E}[\text{dist}(\mathbf{0}, \hat{\partial}F(\theta))] \leq \epsilon$  for an  $\epsilon$ -stationary point where  $\text{dist}(x, A)$  denotes the distance between a vector  $x$  and a set  $A$ . If no regularizer is considered ( $\mathcal{R} = 0$ ), this criterion boils down to the one usually used in non-convex optimization,  $\mathbb{E}[\|\nabla f(\theta)\|] \leq \epsilon$ . We are now ready to state our main theorem for general convergence.

**Theorem 1.** Let  $\theta_a$  denote an iterate uniformly randomly chosen from  $\{\theta_1, \dots, \theta_T\}$ . Under the conditions (C-1), (C-2), (C-3), (C-4) with the initial stepsize  $\alpha_0 \leq \frac{\delta}{3L}$  and non-increasing stepsize  $\alpha_t$ , PROXGEN, Algorithm 1, is guaranteed to yield  $\mathbb{E}_a[\text{dist}(\mathbf{0}, \hat{\partial}F(\theta_a))^2] \leq \frac{Q_1\sigma^2}{T} \sum_{t=0}^{T-1} \frac{1}{b_t} + \frac{Q_2\Delta}{T} + \frac{Q_3}{T}$  where  $\Delta = f(\theta) - f(\theta^*)$  with optimal point  $\theta^*$ , and  $b_t$  is the minibatch size at time  $t$ . The constants  $\{Q_i\}_{i=1}^3$  on the right-hand side depend on the constants  $\{\alpha_0, \delta, L, D, G, \rho_0, \mu, \gamma\}$ , but not on  $T$ .

From Theorem 1, the appropriate minibatch size is important to ensure a good convergence. Various settings for the minibatch size could be employed for convergence guarantee (for example,  $b_t = t$ ), but considering practical cases, we provide the following important corollary for constant minibatch.

**Corollary 1** (Constant Mini-batch). Under the same assumptions as in Theorem 1 with sample size  $n$  and constant minibatch size  $b_t = b = \Theta(T) = \Theta(\sqrt{n})$ , we have  $\mathbb{E}[\text{dist}(\mathbf{0}, \hat{\partial}F(\theta_a))^2] \leq \mathcal{O}(1/T)$  and the total complexity is  $\mathcal{O}(1/\epsilon^4)$  in order to have  $\mathbb{E}[\text{dist}(\mathbf{0}, \hat{\partial}F(\theta_a))] \leq \epsilon$ .

Here we make several remarks on our results and relationship with prior work.

- **Improvements upon Prior Work.** The most challenging parts in our analysis compared to previous study (Xu et al., 2019a) (which is only for vanilla SGD) is that we should handle the momentum  $m_t$  and non-trivial preconditioner  $C_t$ . Due to the existence of  $m_t$ , it is highly non-trivial to bound the term  $\|m_t - \nabla f(\theta_t)\|_2$  without suitable assumptions whereas  $\|g_t - \nabla f(\theta_t)\|_2$  in Xu et al. (2019a) can be easily bounded using (C-2). The second challenge is to deal with quadratic approximation term  $(\theta - \theta_t)^\top (C_t + \delta I)(\theta - \theta_t)$  in Algorithm 1 which is not problematic in Xu et al. (2019a) due to trivial  $C_t = I$ . We can successfully bypass those difficulties using mild conditions (C-3) and (C-4) respectively and also allow non-increasing stepsize.

- **On Condition (C-4).** (C-4) is easily satisfied both theoretically and empirically. Most of the popular optimization algorithms for deep learning such as ADAGRAD, ADAM, and KFAC satisfies this condition (see Appendix D). In order to investigate whether this condition could be satisfied in real problems, we train ResNet-34 on CIFAR-10 dataset. In Figure 1, we can see the minimum eigenvalue of  $\alpha_t(C_t + \delta I)^{-1}$  tends to increase, so the condition (C-4) is also satisfied empirically.

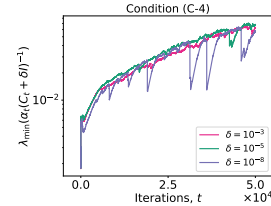


Figure 1: Empirical results for condition (C-4).



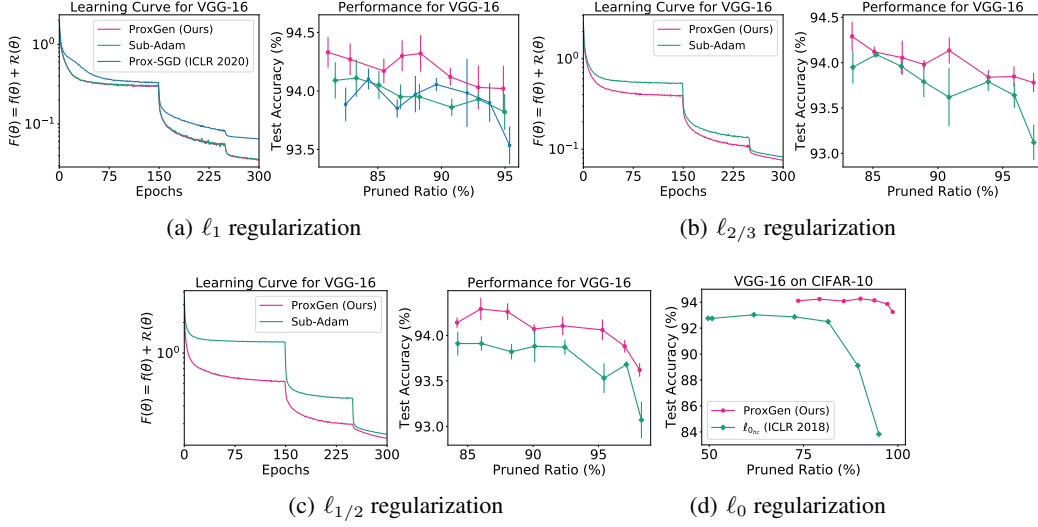


Figure 2: Comparison for sparse VGG-16 on CIFAR-10 dataset.

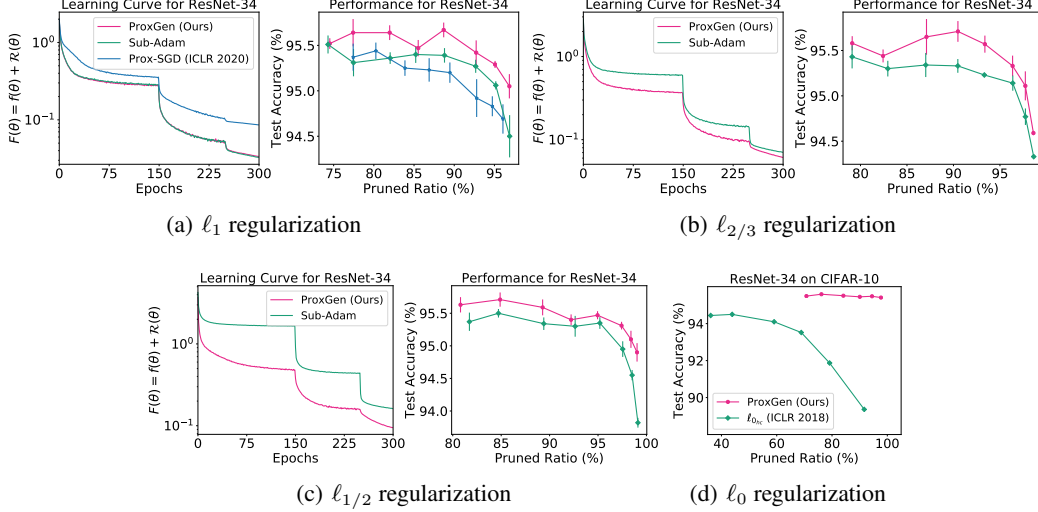


Figure 3: Comparison for sparse ResNet-34 on CIFAR-10 dataset.

- **Implications of Condition (C-4) on Theory.** Our analysis relies on (C-4), the lower bound for the minimum eigenvalue of  $\Gamma_t := \alpha_t(C_t + \delta I)^{-1}$ . This means that Theorem 1 guarantees  $\mathbb{E}_a[\text{dist}(\mathbf{0}, \hat{\partial}F(\theta_a)^2) \leq \mathcal{O}(1/\sqrt{T})$  (in case of  $b = \Theta(\sqrt{n})$  as in Corollary 1) for *any* change of basis of  $\Gamma_t$ , so in that sense, we provide a worst-case analysis and there is room for more optimistic bounds.
- **Connections to Second-order Methods.** Our analysis can provide guarantees for *positive* second-order preconditioners as long as (C-4) is satisfied (The empirical Fisher information (Martens & Grosse, 2015) is one example). Although second-order solvers generally enjoy very fast convergence under strongly convex loss (Lee et al., 2012; Zhang et al., 2019), it can be understood that our theory guarantees *at least a sublinear rate for such second-order curvatures* with less stringent conditions.

## 4 EXPERIMENTS

We consider two important tasks for regularized training in deep learning communities: (i) training sparse neural networks and (ii) network quantization. Throughout our experiments, we consider ADAM as a representative of PROXGEN where  $m_t = \rho_t m_{t-1} + (1 - \rho_t)g_t$  with constant decaying parameter  $\rho_t = 0.9$  and  $C_t = \sqrt{\beta}C_{t-1} + (1 - \beta)g_t^2$  with  $\beta = 0.999$  in Algorithm 1. The details on other hyperparameter/experiment settings are provided in the Appendix.

Table 3: Comparison for binary neural networks. The best performance in mean value is highlighted.

Test Error (%)						
Baselines				PROXGEN (Ours)		
Model	Full Precision (32-bit)	BinaryConnect Courbariaux et al. (2015)	PROXQUANT Bai et al. (2019)	Revised ProxQuant $\ell_1$	Revised ProxQuant $\ell_{2/3}$	Revised ProxQuant $\ell_{1/2}$
ResNet-20	8.06	9.54 $\pm$ 0.03	<b>9.35</b> $\pm$ 0.13	9.50 $\pm$ 0.12	9.72 $\pm$ 0.06	9.78 $\pm$ 0.18
ResNet-32	7.25	8.61 $\pm$ 0.27	8.53 $\pm$ 0.15	8.29 $\pm$ 0.07	<b>8.22</b> $\pm$ 0.05	8.43 $\pm$ 0.15
ResNet-44	6.96	8.23 $\pm$ 0.23	7.95 $\pm$ 0.05	<b>7.68</b> $\pm$ 0.07	7.91 $\pm$ 0.08	7.90 $\pm$ 0.13
ResNet-56	6.54	7.97 $\pm$ 0.22	7.70 $\pm$ 0.06	<b>7.52</b> $\pm$ 0.18	7.60 $\pm$ 0.09	7.61 $\pm$ 0.12

**Training Sparse Neural Networks.** Motivated by the lottery ticket hypothesis (Frankle & Carbin, 2019), we consider training VGG-16 (Simonyan & Zisserman, 2014) and ResNet-34 (He et al., 2016) on CIFAR-10 dataset using sparsity encouraging regularizers. Toward this, we consider the following objective function with  $\ell_q$  regularization:  $F(\theta) := \mathbb{E}_{\xi \sim \mathbb{P}}[f(\theta; \xi)] + \lambda \sum_{j=1}^p |\theta_j|^q$  where  $0 \leq q \leq 1$ . We train the network parameters with the closed-form proximal mappings introduced in Section 2.

We compare PROXGEN with subgradient methods and also include PROX-SGD (Yang et al., 2020) as a baseline especially for  $\ell_1$  regularization since PROX-SGD considers only convex regularizers. In PROX-SGD, the hand-crafted fine-tuned scheduling on  $\alpha_t$  and  $\rho_t$  is essential for fast convergence and good performance, but in our experiments we use standard settings  $\rho_t = 0.9$  with step-decay learning rate scheduling for fair comparisons. For  $\ell_0$  regularization, the problem in Eq. (1) cannot be optimized in a subgradient manner, so we compare PROXGEN with another popular baseline,  $\ell_{0_{hc}}$  (Louizos et al., 2018) which approximates the  $\ell_0$ -norm via hard-concrete distributions.

Figures 2 and 3 illustrate the results for VGG-16 and ResNet-34 respectively. In terms of convergence, PROXGEN shows faster convergence than PROX-SGD for  $\ell_1$  case, but there is no difference between PROXGEN and subgradient methods. However, there are notable differences in convergence for non-convex regularizers  $\ell_{1/2}$  and  $\ell_{2/3}$ , which get bigger as  $q$  decreases. We believe this might be because the  $\ell_q$ -norm derivative,  $q/|\theta|^{1-q}$ , is very large for non-zero tiny  $\theta$  for  $q \in (0, 1)$ . Meanwhile,  $\partial|\theta|/\partial\theta$  is merely the sign value regardless of size of  $\theta$ , so the large gradient of  $|\theta|^q$  may hinder convergence. The learning curves in Figure 2-(b,c) and 3-(b,c) empirically corroborate this phenomenon.

In terms of performance, we can see that PROXGEN consistently achieves better performance than baselines for both VGG-16 and ResNet-34 with similar or even better sparsity level. Importantly, PROXGEN with  $\ell_0$  outperforms  $\ell_{0_{hc}}$  baseline by a great margin. This might be due to the design of  $\ell_{0_{hc}}$ , which approximates  $\|\theta\|_0 = \sum_{j=1}^p \mathbb{I}\{\theta_j \neq 0\}$  with binary mask  $z_j$  parameterized by learnable probability  $\pi_j$  for each coordinate. Thus, the number of parameters to be optimized is doubled, which might make optimization harder. In contrast, PROXGEN does not introduce additional parameters.

More results for MCP (Zhang et al., 2010) and SCAD (Fan & Li, 2001) regularizers are in Appendix.

**Training Binary Neural Networks.** In the second set of experiments, we consider the network quantization constraining the parameters to some set of discrete values which is a key approach for model compression. We evaluate our revised PROXQUANT in Table 2 with extended regularization  $\mathcal{R}_{\text{bin}}^q$  in Section 2. We consider the following objective function with quantization-specific regularizers:  $F(\theta) := \mathbb{E}_{\xi \sim \mathbb{P}}[f(\theta; \xi)] + \lambda \sum_{j=1}^p |\theta_j - \text{sign}(\theta_j)|^q$  where  $0 \leq q \leq 1$ . For comparisons, we quantize ResNet on CIFAR-10 dataset and follow the same experiment settings as in PROXQUANT.

Table 3 presents the results. For all  $q$  values, revised PROXQUANT consistently outperforms the baselines except for ResNet-20, which implies PROXGEN may work better for larger networks. As such, our generalized regularizers  $\mathcal{R}_{\text{bin}}^q$  contribute to one of the state-of-the-art optimization-based methods in network quantization. Notably, revised PROXQUANT  $\ell_1$  greatly outperforms PROXQUANT baseline while these two approaches differ only in update rules (see Table 2). Hence, we can conclude that revised PROXQUANT based on PROXGEN provides an *exact* proximal update and also yields more generalizable solutions. In our experience, revised PROXQUANT  $\ell_0$  shows little degradation in performance, so we do not include this result. However, revised PROXQUANT  $\ell_0$  shows superiority to baselines for language modeling, whose preliminary results are in Appendix.

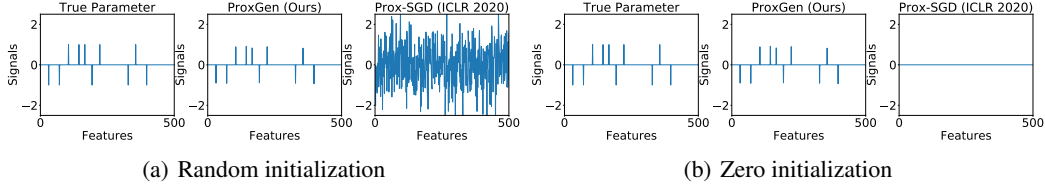


Figure 4: Lasso simulations with different initialization schemes.

## 5 A CLOSER LOOK INTO PROX-SGD (YANG ET AL., 2020) VS. PROXGEN

Prox-SGD (Yang et al., 2020) is the approach closest to our PROXGEN method. However, PROX-SGD is *not an exact* proximal approach and is significantly different from PROXGEN. PROXGEN’s update rule involves directly solving the quadratic subproblem (Eq. (2)). In contrast, PROX-SGD’s update rule consists of two stages: (i) solving the quadratic subproblem *without* learning rate, then (ii) updating the parameters with the computed direction (i.e.  $\hat{\theta}_t - \theta_t$ ) by the learning rate  $\alpha_t$  (Eq. (5)).

$$\hat{\theta}_t = \underset{\theta \in \Omega}{\operatorname{argmin}} \left\{ \langle m_t, \theta \rangle + \lambda \mathcal{R}(\theta) + \frac{1}{2}(\theta - \theta_t)^\top (C_t + \delta I)(\theta - \theta_t) \right\}, \theta_{t+1} = \theta_t + \alpha_t(\hat{\theta}_t - \theta_t) \quad (5)$$

To clearly see the differences between both approaches, we conduct two studies.

**Study 1: Lasso Support Recovery.** For this task, the two-stage update scheme of PROX-SGD might have some potential issues. For example, for  $\ell_1$ -regularized problems, the updated parameter  $\theta_{t+1}$  (Eq. (5)) *might not achieve exact zero* (while  $\hat{\theta}_t$  can) whereas  $\theta_{t+1}$  for PROXGEN (Eq. (2)) can attain exact zero value according to the update rule (Eq. (3)) in Section 2. Another potential caveat is that PROX-SGD might *overestimate* the sparsity level. In view of the above, we run Lasso simulations with different two initialization schemes: (i) random initialization and (ii) zero initialization. For random initialization, it can be seen in Figure 4-(a) that PROX-SGD could not achieve exact zero value, which corroborates our first observation. More interestingly, for zero initialization, we can see in Figure 4-(b) that the estimates using PROX-SGD are exactly zeros for all coordinates, which supports our second observation. This might be because  $\hat{\theta}_t$  (Eq. (5)) is always zero since the quadratic subproblem does not consider the learning rate, which might overestimate the sparsity level. Hence, the subsequent iterate  $\theta_{t+1}$  would be always zero since we initialize the parameters with zeros. On the other hand, PROXGEN correctly recovers the support in both cases.

**Study 2: DenseNet-201 on CIFAR-100 Dataset.** To validate the superiority of PROXGEN upon PROX-SGD, we revisit the largest experiments in Yang et al. (2020). We train DenseNet-201 architecture on CIFAR-100 dataset with  $\ell_1$  regularization since PROX-SGD only consider convex regularizers. For both methods, we use the same hyperparameter settings for fair comparison. Figure 5 illustrates the training learning curves, and it can be seen that our PROXGEN achieves faster convergence as well as lower objective values. For our experience, the learning curves show the similar dynamics for different  $\lambda$  values.

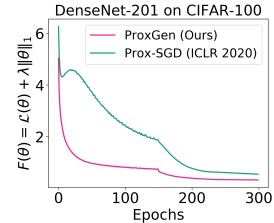


Figure 5: Learning curve.

**Comparison of Theoretical Contributions.** Yang et al. (2020) guarantees the convergence of PROX-SGD, but *not how fast* it converges. Moreover, this is proved *without* considering preconditioners. In contrast, our analysis for the PROXGEN framework appropriately incorporates the first-order momentum and arbitrary positive preconditioner with detailed non-asymptotic convergence.

## 6 CONCLUSION

In this work, we proposed PROXGEN, the first general family of stochastic proximal gradient methods. Within our framework, we presented novel examples of proximal versions of standard SGD approaches, including a proximal version of ADAM. We analyzed the convergence of the whole PROXGEN family and showed that PROXGEN can encompass the results of several previous studies. We also demonstrated that PROXGEN empirically outperforms subgradient-based methods for popular deep learning problems. As future work, we plan to study efficient approximations of proximal mappings for structured regularizers such as  $\ell_1/\ell_q$  norms with preconditioners.



## REFERENCES

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.
- Zeyuan Allen-Zhu. Natasha: Faster non-convex stochastic optimization via strongly non-convex parameter. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 89–97. JMLR. org, 2017.
- Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=HyzMyhCck7>.
- Stephen Becker, Jalal Fadili, and Peter Ochs. On quasi-newton forward-backward splitting: Proximal calculus and convergence. *SIAM Journal on Optimization*, 29(4):2445–2481, 2019.
- Wenfei Cao, Jian Sun, and Zongben Xu. Fast image deconvolution using closed-form thresholding formulas of  $l_q$  ( $q=12, 23$ ) regularization. *Journal of visual communication and image representation*, 24(1):31–41, 2013.
- Tianyi Chen, Tianyu Ding, Bo Ji, Guanyi Wang, Yixin Shi, Sheng Yi, Xiao Tu, and Zhihui Zhu. Orthant based proximal stochastic gradient method for  $\ell_1$ -regularized optimization. *arXiv preprint arXiv:2004.03639*, 2020.
- Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019a. URL <https://openreview.net/forum?id=H1x-x309tm>.
- Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019b.
- Emilie Chouzenoux, Jean-Christophe Pesquet, and Audrey Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, 2014.
- Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pp. 3123–3131, 2015.
- Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.
- Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research (JMLR)*, 2011.
- Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.
- Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=rJl-b3RcF7>.
- Wenjiang J Fu. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416, 1998.

- Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.
- 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.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation (ICLR)*, 2015.
- Jason D Lee, Yuekai Sun, and Michael Saunders. Proximal newton-type methods for convex optimization. In *Advances in Neural Information Processing Systems*, pp. 827–835, 2012.
- Yunwen Lei, Ting Hu, Guiying Li, and Ke Tang. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.
- Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.
- Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through  $l_0$  regularization. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=H1Y8hhg0b>.
- James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417, 2015.
- Cheolwoo Park and Young Joo Yoon. Bridge regression: adaptivity and group selection. *Journal of Statistical Planning and Inference*, 141(11):3506–3519, 2011.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.
- Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. Proxsarah: An efficient algorithmic framework for stochastic composite nonconvex optimization. *arXiv preprint arXiv:1902.05679*, 2019.
- P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing  $\ell_1$ -penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.
- Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems*, pp. 1145–1153, 2016.
- Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=ryQu7f-RZ>.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

- Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.
- A. N. Tychonoff. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39(5):195–198, 1943.
- Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.
- Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pp. 2074–2082, 2016.
- Yi Xu, Rong Jin, and Tianbao Yang. Non-asymptotic analysis of stochastic methods for non-smooth non-convex regularized problems. In *Advances in Neural Information Processing Systems*, pp. 2626–2636, 2019a.
- Yi Xu, Qi Qi, Qihang Lin, Rong Jin, and Tianbao Yang. Stochastic optimization for DC functions and non-smooth non-convex regularizers with non-asymptotic convergence. In *International conference on machine learning*, 2019b.
- Eunho Yang and Aurélie C Lozano. Sparse+ group-sparse dirty models: Statistical guarantees without unreasonable conditions and a case for non-convexity. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3911–3920. JMLR. org, 2017.
- Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7308–7316, 2019.
- Yang Yang, Yaxiong Yuan, Avraam Chatzimichailidis, Ruud JG van Sloun, Lei Lei, and Symeon Chatzinotas. Proxsgd: Training structured neural networks under regularization and constraints. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=HygpthEtvvr>.
- Jihun Yun, Aurelie C. Lozano, and Eunho Yang. Stochastic gradient methods with block diagonal matrix adaptation. *arXiv preprint arXiv:1905.10757*, 2019a.
- Jihun Yun, Peng Zheng, Eunho Yang, Aurelie Lozano, and Aleksandr Aravkin. Trimming the  $\ell_1$  regularizer: Statistical analysis, optimization, and applications to deep learning. In *International Conference on Machine Learning*, pp. 7242–7251, 2019b.
- Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, pp. 9793–9803, 2018.
- Cun-Hui Zhang et al. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.
- Guodong Zhang, James Martens, and Roger B Grosse. Fast convergence of natural gradient descent for over-parameterized neural networks. In *Advances in Neural Information Processing Systems*, pp. 8080–8091, 2019.