CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.SO-2025-0051

Vol. **x**, No. x, pp. 1-33
xx 2025

A Stochastic Three-Block Alternating Minimization Algorithm and its Application to Quantized Deep Neural Networks

Fengmiao Bian[1,*], Ren Liu[2] and Xiaoqun Zhang[2]

[1] *Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China.*
[2] *School of Mathematical Sciences, MOE-LSC and Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240, China.*

Received 29 May 2025; Accepted 15 October 2025

Abstract. Deep neural networks (DNNs) have made great progress in various fields. In particular, the quantized neural network is a promising technique for making DNNs compatible with resource-limited devices for memory and computation saving. In this paper, we mainly consider a non-convex minimization model with three blocks to train quantized DNNs and propose a novel stochastic three-block alternating minimization (STAM) algorithm to solve it. We develop a convergence theory for the STAM algorithm and obtain an ϵ-stationary point with an optimal convergence rate. Furthermore, we implement our STAM algorithm to train DNNs with relaxed binary weights. The experiments are carried out on three different network structures, namely VGG-11, VGG-16, and ResNet-18. These DNNs are trained using two different datasets, CIFAR-10 and CIFAR-100, respectively. We compare our STAM algorithm with state-of-the-art algorithms for training quantized neural networks. The test accuracy indicates the effectiveness of our model and algorithm for training relaxed binary quantization DNNs.

AMS subject classifications: 90C06, 90C26, 90C35, 90C90

Key words: Non-convex optimization, three-block splitting algorithm, quantized networks, ϵ-stationary point.

1 Introduction

1.1 Background

Deep neural networks have achieved great success in many practical fields, such as computer vision, speech recognition, and automatic driving [27, 45, 47]. This great success

*Corresponding author. *Email addresses:* mafmbian@ust.hk (F. Bian), xqzhang@sjtu.edu.cn (X. Zhang), liur0810@sjtu.edu.cn (R. Liu)

mainly depends on the flexibility of neural networks and their complex nonlinear structure. At present, most research makes neural networks more flexible by increasing the number of layers and/or the width of neural networks [43]. However, these also lead to a large increase in the number of parameters in neural networks. Since most parameters are floating point numbers, one requires a lot of storage space. For instance, popular models like AlexNet and VGG-16 occupy over 200 MB and 500 MB, respectively [20], making it challenging to deploy these networks on portable devices. To address this issue, research focuses on compressing DNNs while maintaining test accuracy. Pruning and weight quantization are common techniques employed for network compression [1, 2, 46]. In this paper, we focus on quantization by reducing the bit of the network weights. Much research work has achieved promising results, such as reducing fully connected layer full connection layer parameters to 5 bits, convolution layer parameters to 8 bits [20], and quantizing network pretraining parameters as 4-12 bits [15, 31]. Additionally, data-dependent algorithms proposed in [33] sequentially quantize each layer of the neural network, achieving successful training with relaxed 4-bit weights on MNIST and CIFAR-10 datasets. Binary quantization, also known as 1-bit quantization, is the most minimal form of quantization. For binary quantized DNNs, both weights and activations can be expressed as -1 (0) or 1, resulting in minimal memory consumption. Moreover, binary quantization enables lightweight bitwise and bit-counting operations instead of heavy matrix multiplications. This not only accelerates inference but also reduces memory usage and power consumption, making binary DNNs hardware-friendly. Notably, research works such as BNN [23] and Xnor-Net [39] have demonstrated the effectiveness of binary DNNs, with Xnor-Net achieving substantial memory savings and faster convolutional operations.

From another point of view, the binary quantization problem can be viewed as a non-convex optimization problem. Thus, many studies focus on optimized binary quantization which minimizes quantization error and improves the loss function. For example, in [13] the authors introduced BinaryConnect (BC), which modifies the projection stochastic gradient descent algorithm to train DNNs with binary weights. Experimental results demonstrated a significant improvement in accuracy. The BC method was further studied from a theoretical standpoint in [30], where the convergence of BC was established under strongly convex assumptions. In [28], the quantized neural network is regarded as an optimization problem with constraints, and the authors decoupled the continuous variables from the discrete constraints of the neural network based on alternating direction method of multipliers (ADMM). Additionally, a relaxed binary quantization algorithm called BinaryRelax (BR) [52] was proposed to better solve non-convex optimization problems with discrete constraints. The algorithm is a two-stage method with a pseudo-quantization weight constantly close to the quantization weight by increasing regularization parameters at the first stage, while in the second stage, the quantized weights are directly adopted. Further, the authors used the classical proximal stochastic gradient descent (PSGD) algorithm [12, 41] to train DNNs with binary quantized weights.

1.2 Problem formulation and motivation

Previous work mainly focused on algorithms for solving quantization weights, which directly minimized the loss function of quantization weights without considering any information about float weights. In this paper, we aim to explore new formulations by leveraging the relationship between quantized and floating-point weights. It is a natural idea that quantized parameters should approximate the full-precision parameter as closely as possible, expecting that the performance of the binary neural network model will be close to the full-precision one. Thus, we construct the following new model for training DNNs with quantized weights

$$\min_{W,\tilde{W}} \frac{\lambda}{2}\|W-\tilde{W}\|_F^2 + L_W(p,q) + \mathcal{I}_{\mathcal{Q}}(\tilde{W}). \tag{1.1}$$

Here W is the float parameters in the neural network, $\tilde{W}$ is the corresponding quantized parameters, $L_W(p,q)$ is the loss function of the neural network with p being the input data of the neural network and q being the corresponding label, and $\mathcal{I}_{\mathcal{Q}}$ denotes the indicator function of the quantized weights set $\mathcal{Q}$. In the model (1.1), we use the loss function $L_W(p,q)$ to find the floating point parameters with good generalization performance. Further, under the interaction of $\|W-\tilde{W}\|_F^2$ and $\mathcal{I}_{\mathcal{Q}}(\tilde{W})$ the quantized parameters would be close enough to the floating parameters. We would design a new stochastic three-block alternating minimization algorithm to solve the problem (1.1). The problem can be formulated in a more general form with a three-block composite structure

$$\min_{x,y} \Phi(x,y) := F(x) + G(y) + H(x,y), \tag{1.2}$$

where F, H, G are proper lower semi-continuous functions. Here we emphasize that the functions F, G, and H are not necessarily convex. In deep neural networks, the function $G(y)$ is generally a loss function which is a sum of many terms, such as

$$G(y) = \frac{1}{N}\sum_{i=1}^{N} G_i(y)$$

with N large. Remark also that G does not necessarily have a summation structure in this paper. Therefore, our model (1.2) encompasses a broader range of practical problems.

1.3 The proposed algorithm and related work

The main idea of our proposed algorithm is to minimize variables x and y alternately for solving the optimization problem (1.2). For y-direction, we consider linearizing the function $G+H$ and utilize stochastic gradient estimators instead of full gradient calculations. For x-direction, the corresponding composite problem is solved using the Douglas-Rachford splitting method. This allows us to decouple the two variables x and y, and each

subproblem involves only the computation of the proximal operator of a single function, thereby making the whole problem easy to solve and computationally straightforward.

Based on the above ideas, we propose a stochastic three-block alternating algorithm to solve the non-convex problem (1.2), see Algorithm 1. Throughout the paper, we assume that the gradient estimator $\tilde{\nabla}G(y)$ in Algorithm 1 is unbiased. More arguments of unbiased gradient estimators can be seen in the Section 2.1.

To address optimization problems of the form (1.2), numerous alternating minimization algorithms have been developed. Among them, the proximal alternating minimization (PAM) method [4] serves as a viable method. However, its subproblems often lack explicit solutions, requiring multiple inner iterations for each subproblem, which reduces the efficiency of PAM in practical applications. To overcome this drawback, the proximal alternating linearized minimization (PALM) method [7] replaces the subproblems in PAM with proximal linearized subproblems. When the proximal operators of functions F and G are easy to compute, PALM significantly improves the efficiency of solving subproblems compared to PAM. To further enhance the performance of PALM, momentum terms are introduced in [37], leading to the inertial proximal alternating linearized minimization (iPALM) algorithm. Due to the simplicity and ease of implementation of PALM, many variants and improvements based on PALM emerge such as those in [3, 10, 17, 35, 50, 55] and their references. When the function F in problem (1.2) has a finite-sum structure, the alternating structure-adaptive proximal gradient descent algorithm [36] addresses this case, and its global convergence to a critical point of the problem is established under the Kurdyka-Łojasiewicz (KL) inequality. For more general three-term problems, the classical three-operator splitting algorithm [14] is proposed to address such problems, which can be regarded as an extension of the Douglas-Rachford

Algorithm 1. A Stochastic Three-Block Alternating Minimization (STAM) Algorithm.

1: Choose the parameters $\gamma,\beta>0$ and an initial point x^0,y^0, and z^0.

2: For $t=0,\dots,T-1$ and set

$$y^{t+1}\in\arg\min_y\left\{\langle\tilde{\nabla}G(y^t)+\nabla_y H(x^t,y^t),y-y^t\rangle+\frac{\beta}{2}\|y-y^t\|^2\right\}, \tag{1.3a}$$

$$x^{t+1}\in\arg\min_x\left\{H(x,y^{t+1})+\frac{1}{2\gamma}\|x-z^t\|^2\right\}, \tag{1.3b}$$

$$u^{t+1}\in\arg\min_u\left\{F(u)+\frac{1}{2\gamma}\|2x^{t+1}-z^t-u\|^2\right\}, \tag{1.3c}$$

$$z^{t+1}=z^t+(u^{t+1}-x^{t+1}), \tag{1.3d}$$

where $\tilde{\nabla}G(y)$ is a gradient estimator of $\nabla G(y)$.

3: Output $\{y^{t+1},u^{t+1}\}$.

splitting algorithm [32]. Moreover, the non-convex Douglas-Rachford algorithm and its variants appear in [11,29], while the non-convex three-operator splitting algorithm is introduced in [6]. In [8], the authors propose a proximal alternating algorithm to solve three-block problems involving linear operator combinations.

All of the aforementioned algorithms are deterministic algorithms, which incur high computational costs and exhibit low efficiency for large-scale optimization problems. To address this issue, [51] first proposes a block stochastic gradient iteration algorithm by combining the simple stochastic gradient descent estimator with PALM when $H(x,y)$ has a finite-sum structure. To relax the assumptions on the objective function in [51] and improve the convergence rate of the stochastic PALM algorithm, [16] replaces the simple SGD estimator with the variance-reduced gradient estimators, leading to the development of the stochastic proximal alternating linearized minimization (SPRING) algorithm. Subsequently, [22] introduces an inertial variant of the stochastic PALM algorithm with a variance-reduced gradient estimator, referred to as SiPALM. When F has a finite-sum structure, for the general form of problem (1.2), the works [53,54] extend the three-operator splitting algorithm to the stochastic setting, incorporating unbiased stochastic gradient estimators. Additionally, [5] explores an extended version of the three-operator splitting algorithm, which combines unbiased gradient estimators and variance-reduced gradient estimators. In [34], the authors propose a mini-batch stochastic proximal algorithm for general stochastic problems, incorporating variance-reduced gradient estimators into the proximal algorithm to address finite-sum optimization problems. However, these works primarily focus on three-block problems that do not include cross terms $H(x,y)$. In contrast, our algorithm addresses problems that include cross terms $H(x,y)$. For such problems, [24] proposes a stochastic alternating structure-adaptive proximal (s-ASAP) gradient descent method. When the stochastic gradient is a variance-reduced estimator, the convergence of the sequence generated by the algorithm is established based on the KL inequality. Nevertheless, compared to the algorithm in [24], our algorithm does not require G to have a finite-sum structure. Moreover, while [24] considers variance-reduced stochastic gradient estimators, our algorithm leverages unbiased gradient estimators.

1.4 Contributions

In this paper, we propose a novel stochastic three-block alternating minimization algorithm to solve the large-scale problem (1.2), and explore its application in training binary deep neural networks. Specifically, our main contributions are elaborated as follows:

- Model. We present a new model (1.1) for training DNNs with quantized weights. Compared with the quantization models in the previous literature [13,52], our new model utilizes the information from floating-point parameters, allowing the generalization ability of floating-point parameters to be transferred to some extent to quantized parameters, as the loss function is continuous. This fact has also been validated through numerical experiments.

- Algorithm and convergence. We propose a new stochastic alternating minimization algorithm for solving the problem (1.2) with three-block structures. Convergence analysis is established for our Algorithm 1 under the condition that the stochastic gradient estimator $\widetilde{\nabla} G$ satisfies the expected smoothness (ES) inequality, which is the weakest assumption regarding unbiased gradient estimators. This makes our algorithm applicable to more large-scale non-convex optimization problems. Furthermore, in comparison to other methods employed for solving binary neural networks [13,52], we also obtain the convergence rate of our algorithm.
- Experiments. We apply our Algorithm 1 to train VGG-11 [45], VGG-16 [45] and ResNet-18 [21] DNNs with relaxed binary weights on two standard datasets: CIFAR-10 [26] and CIFAR-100 [26], respectively. The experimental results show the effectiveness of our algorithm. In particular, for the DNN with VGG-11 structure and the CIFAR-10 data set, our test accuracy is far better than the existing quantization DNN methods.

The rest of this paper is organized as follows. In Section 2, we first review the unbiased gradient estimator and then obtain the convergence rate of Algorithm 1 for non-convex problems. In Section 3, we present experimental results of the STAM algorithm, demonstrating its efficiency compared to other algorithms for quantized DNNs. Finally, Section 4 concludes the paper with some remarks on our algorithm.

2 Gradient estimator and convergence

In this section, we first give the definition of unbiased gradient estimators and some corresponding sampling methods. Then we recall the important expected smoothness assumption on gradient estimators proposed by [25]. Finally, we establish the convergence and convergence rate of Algorithm 1 based on the ES assumption.

2.1 Unbiased gradient estimator

In our theoretical analysis and experiments, we always assume that the stochastic gradient estimator $\widetilde{\nabla} G(y)$ is unbiased, and its definition is given in the following.

Definition 2.1. *The stochastic gradient estimator $\widetilde{\nabla} G(y)$ is unbiased if $\mathbb{E}[\widetilde{\nabla} G(y)] = \nabla G(y)$.*

Remark 2.1. In many applications of deep learning, function G generally has the following finite-sum structure:

$$G(y) = \frac{1}{N} \sum_{i=1}^{N} G_i(y) \tag{2.1}$$

such as the empirical risk in supervised machine learning. In this setting, the source of stochasticity comes from the way of sampling from the sum, and the unbiased stochastic

gradient can be written in the following unified form:

$$\widetilde{\nabla} G(y) = \frac{1}{N}\sum_{i=1}^{N} v_i \nabla G_i(y), \tag{2.2}$$

where $(v_1,\ldots,v_N)$ is a sampling vector drawn from certain distribution $\mathcal{D}$. The random variable v_i has different forms for different sampling methods. Here we give a representative sampling distribution:

- b-nice sampling without replacement. This is a well-known method in deep learning, and we also use this sampling method in our experiments. We generate a random subset $S \subset \{1,2,\ldots,N\}$ by uniformly choosing from all subsets of size b, where $b \in [1,N]$ is an integer. Then we define $v_i = 1_{i\in S}/p_i$ with $1_{i\in S} = 1$ for $i \in S$ and 0 otherwise, and $p_i = b/N$ for all i.

There are many other sampling methods that make stochastic gradients unbiased such as sampling with replacement and independent sampling without replacement. We refer to [25] for more details.

2.2 Second moment of stochastic gradient

In this subsection, we review the so-called expected smoothness assumption on the second moment of stochastic gradient proposed recently by [25], related to the work [19,40] for stochastic gradient descent in the convex setting. This ES assumption can be applied to non-convex problems and is essential for our convergence analysis. Given the functions $G(x)$ and $H(x,y)$, from now on we use the notation $G^{\inf} = \inf_{x\in\mathbb{R}^n} G(x)$ and $H^{\inf} = \inf_{\{x,y\}\in\mathbb{R}^n} H(x,y)$. In the following, we give the ES assumption.

Assumption 2.1 (Expected Smoothness)**.** The second moment of the stochastic gradient $\widetilde{\nabla} G(y)$ satisfies

$$\mathbb{E}\left[\|\widetilde{\nabla} G(y)\|^2\right] \le 2A_0\big(G(y) - G^{\inf}\big) + B_0\|\nabla G(y)\|^2 + C_0 \tag{2.3}$$

for some $A_0, B_0, C_0 \ge 0$ and for all $y \in \mathbb{R}^n$.

Remark 2.2. When analyzing the convergence of stochastic gradient descent, various assumptions on the second moment of unbiased gradient estimators have been proposed. [25] have shown that ES is the weakest among all these assumptions, including such as bounded variance (BV) [18], maximal strong growth (M-SG) [48], expected strong growth (E-SG) [49], relaxed growth (RG) [9], and gradient confusion (GC) [42]. We refer to [25, Section 3] for more details. Hence, for many practical applications, the expected smoothness assumption can be more easily verified than other assumptions.

In our convergence analysis, we require that $\widetilde{\nabla} G(y) + \nabla_y H(x,y)$ satisfies the ES assumption for the general unbiased gradient estimators $\widetilde{\nabla} G$. In particular, we would

prove that when G has the -sum structure (2.1), the ES assumption of $\widetilde{\nabla} G(y)+\nabla_y H(x,y)$ is automatically satisfied under certain natural conditions on the functions G_i and H in the following lemma. For its proof, one can see Appendix A.

Lemma 2.1. *Let G be a function with the form* (2.1) *and let each G_i be bounded from below by $G_i^{\inf}$ and be L_1^i smooth. Let H satisfy*

$$\|\nabla_y H(x,y_1)-\nabla_y H(x,y_2)\|\leq L_3^*\|y_1-y_2\|,\quad \forall y_1,y_2,x\in\mathbb{R}^n.$$

Suppose that $\mathbb{E}[v_i^2]$ is finite for all i and that G and H is bounded from below by $G^{\inf}$ and $H^{\inf}$, respectively. Define

$$\Delta^{\inf}=\frac{1}{N}\sum_{i=1}^{N}\big((G+H)^{\inf}-G_i^{\inf}-H^{\inf}\big).$$

Then $\Delta^{\inf}\geq 0$ and the following ES inequality:

$$\mathbb{E}\big[\|\widetilde{\nabla} G(y)+\nabla_y H(x,y)\|^2\big]\leq 2A\big(G(y)+H(x,y)-(G+H)^{\inf}\big)+C \tag{2.4}$$

holds, where $A=(4\max_i L_1^i\mathbb{E}[v_i^2]+L_3^*)/2$ and $C=2A\Delta^{\inf}$.

2.3 Convergence analysis

We now establish the convergence rate of the STAM Algorithm 1. In analyzing STAM Algorithm 1, we need the following mild assumptions on the non-convex functions G and H.

Assumption 2.2. Functions G and H satisfy

(a1) G is bounded from below by $G^{\inf}$, and G has a Lipschitz continuous gradient, i.e. there exists a constant $L_1>0$ such that

$$\|\nabla G(y_1)-\nabla G(y_2)\|\leq L_1\|y_1-y_2\|,\quad \forall y_1,y_2\in\mathbb{R}^n.$$

(a2) There exist $L_2^*,L_4^*>0$ such that

$$\|\nabla_x H(x_1,y)-\nabla_x H(x_2,y)\|\leq L_2^*\|x_1-x_2\|,\quad \forall x_1,x_2,y\in\mathbb{R}^n, \tag{2.5}$$
$$\|\nabla_x H(x,y_1)-\nabla_x H(x,y_2)\|\leq L_4^*\|y_1-y_2\|,\quad \forall y_1,y_2,x\in\mathbb{R}^n, \tag{2.6}$$

and there exist $L_3^*,L_5^*>0$ such that

$$\|\nabla_y H(x,y_1)-\nabla_y H(x,y_2)\|\leq L_3^*\|y_1-y_2\|,\quad \forall y_1,y_2,x\in\mathbb{R}^n, \tag{2.7}$$
$$\|\nabla_y H(x_1,y)-\nabla_y H(x_2,y)\|\leq L_5^*\|x_1-x_2\|,\quad \forall x_1,x_2,y\in\mathbb{R}^n. \tag{2.8}$$

Next, we denote $L:=L_3^*+L_1$. Let $l\in\mathbb{R}$ be such that $H(\cdot,y)+l\|\cdot\|^2/2$ is convex for all $y\in\mathbb{R}^n$. Note that such l always exists by (2.5). Particularly, one can always take $l=L_2^*$. If the stochastic gradient $\widetilde{\nabla}G$ satisfies the ES inequality (2.3), then the following lemma shows that $\widetilde{\nabla}G(y)+\nabla_y H(x,y)$ also satisfies the corresponding ES assumption. This result is essential for analyzing the convergence of STAM Algorithm 1.

Lemma 2.2. *Suppose that the stochastic gradient $\widetilde{\nabla}G$ satisfies Assumption* 2.1 *and G,H satisfy Assumption* 2.2*. Then we have*

$$\mathbb{E}\big[\|\widetilde{\nabla}G(y)+\nabla_y H(x,y)\|^2\big]\le 2A[G(y)+H(x,y)-(G+H)^{\inf}]+C, \tag{2.9}$$

where

$$\begin{aligned}&A=\max(2A_0+2B_0L_1,2L_3^*),\\ &C=2A[(G+H)^{\inf}-G^{\inf}-H^{\inf}]+2C_0.\end{aligned}$$

We refer the proof of Lemma 2.2 to Appendix A. Compared to Lemmas 2.1, 2.2 does not assume that G has a summation structure, but rather that $\widetilde{\nabla}G$ satisfies the ES assumption. For G being a finite-sum as in (2.1), from Lemma 2.1 we know that $\widetilde{\nabla}G(y)+\nabla_y H(x,y)$ automatically satisfied the ES assumption.

The following lemma plays an important role in establishing the convergence of Algorithm 1 when applied to solving the non-convex problem (1.2).

Lemma 2.3. *Let $\{(y^t,x^t,u^t,z^t)\}$ be a sequence generated from the STAM Algorithm* 1*. Suppose that the stochastic gradient $\widetilde{\nabla}G$ is unbiased and satisfies Assumption* 2.1*, and that G,H satisfy Assumption* 2.2*. Suppose that the parameters $\beta>0$ and $\gamma>0$ are chosen such that*

$$\mathcal{K}_1:=\frac{1-\big(10(l+L_2^*)+4\big)\gamma-5(L_2^*)^2\gamma^2}{4\gamma}>0. \tag{2.10}$$

Then

$$\sum_{t=0}^{T-1}\omega_t\eta_t+\frac{\beta\mathcal{K}_1}{\mathcal{K}_2}\sum_{t=0}^{T-1}\omega_t\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\le\beta\omega_{-1}\delta_0+\beta\omega_{-1}\mathcal{M}_0'+\frac{(L+M)C}{2\beta}\sum_{t=0}^{T-1}\omega_t, \tag{2.11}$$

where

$$\eta_t=\mathbb{E}\big\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\big\|^2,\quad \omega_t=\frac{\omega_{-1}}{\big(1+(L+M)A/\beta^2\big)^{t+1}}$$

for $\omega_{-1}>0$ arbitrary, $\mathcal{M}_0'=\mathbb{E}[\mathcal{M}_0-\inf_{t\ge0}\mathcal{M}_t]$ with

$$\begin{aligned}&\mathcal{M}_t:=\mathcal{M}(x^t,u^t,z^t)=F(u^t)+\frac{1}{2\gamma}\|2x^t-u^t-z^t\|^2-\frac{1}{2\gamma}\|x^t-z^t\|^2-\frac{1}{\gamma}\|u^t-x^t\|^2,\\ &\delta_t=\mathbb{E}\big[G(y^t)+H(x^t,y^t)-(G+H)^{\inf}\big],\\ &\mathcal{K}_2:=\frac{5(1+\gamma L_2^*)^2}{4\gamma^2},\end{aligned}$$

and $M=2\mathcal{K}_3$ with $\mathcal{K}_3:=L_4^*(1+5\gamma L_4^*)+5\mathcal{K}_1(L_4^*)^2/\mathcal{K}_2$.

Remark 2.3. Notice that $\lim_{\gamma\to 0^+}\mathcal{K}_1=+\infty$. Therefore, for any given $l\in\mathbb{R}$ and $L_2^*\geq 0$, the condition (2.10) will be satisfied when $\gamma>0$ is sufficiently small. Moreover, using the quadratic formula, we easily obtain the following computable threshold

$$0<\gamma<\frac{\sqrt{(10L_2^*+10l+4)^2+20(L_2^*)^2}-(10L_2^*+10l+4)}{10(L_2^*)^2}$$

such that (2.10) holds. On the other hand, the choice of γ in Lemma 2.3 implies that $5(L_2^*\gamma+1)^2+10l\gamma+4\gamma-6<0$, thus we have $\gamma<6/(10l+4)<1/l$.

The proof of Lemma 2.3 can be seen in Appendix A. Lemma 2.3 provides a bound on a weighted sum of gradients about y and x. A similar idea of weighting different iterations has also been used in the analysis of stochastic gradient descent in [25, 38, 44].

Theorem 2.1. *Let $\{(y^t,x^t,u^t,z^t)\}$ be a sequence generated from STAM Algorithm* 1. *Suppose that the stochastic gradient $\widetilde{\nabla}G$ is unbiased and satisfies Assumption* 2.1, *and that G,H satisfy Assumption* 2.2. *Suppose that the parameters $\gamma,\beta>0$ are chosen such that* (2.10) *and*

$$\left[(L_2^*)^2+(L_5^*)^2\right]\gamma^2\leq 1,\quad \frac{\beta\mathcal{K}_1}{\mathcal{K}_2}\geq 2$$

hold. Then we have the following estimate:

$$\begin{aligned}&\min_{0\leq t\leq T-1}\left[\mathbb{E}\left\|\nabla G(y^t)+\nabla_y H(u^t,y^t)\right\|^2+\mathbb{E}\,\mathrm{dist}^2\left(0,\nabla_x H(u^t,y^t)+\partial F(u^t)\right)\right]\\ \leq\,&2\frac{\left(1+(L+M)A/\beta^2\right)^T}{T}\beta(\delta_0+\mathcal{M}_0')+\frac{(L+M)C}{\beta},\end{aligned}$$

where the constants $\delta_0,\mathcal{M}_0'$ and M are defined as in Lemma 2.3.

Although the bound of Theorem 2.1 shows possible exponential growth, by carefully controlling the parameters we could obtain an ϵ-stationary point. More precisely, we have the following convergence rate when using our STAM Algorithm 1 to find an ϵ-stationary point of the non-convex optimization problem (1.2).

Theorem 2.2 (Convergence Rate)**.** *Let $\{(y^t,x^t,u^t,z^t)\}$ be a sequence generated from STAM Algorithm* 1. *Suppose that the stochastic gradient $\widetilde{\nabla}G$ is unbiased and satisfies Assumption* 2.1, *and that G,H satisfy Assumption* 2.2. *Suppose that the parameter $\gamma>0$ is chosen such that* (2.10) *and*

$$\left[(L_2^*)^2+(L_5^*)^2\right]\gamma^2\leq 1$$

hold. Given $\epsilon>0$, choose the parameter

$$\beta>\max\left\{\sqrt{(L+M)AT},\frac{2\mathcal{K}_2}{\mathcal{K}_1},\frac{2(L+M)C}{\epsilon^2}\right\}.$$

If

$$T \geq \frac{12(L+M)(\delta_0+M_0')}{\epsilon^2}\max\left\{\frac{2C}{\epsilon^2},\frac{12(\delta_0+M_0')A}{\epsilon^2},\frac{2\mathcal{K}_2}{\mathcal{K}_1(L+M)}\right\}=\mathcal{O}(\epsilon^{-4}),$$

then there exists $0\leq t_0\leq T-1$ *such that*

$$\begin{aligned}&\mathbb{E}\left\|\nabla G(y^{t_0})+\nabla_y H(u^{t_0},y^{t_0})\right\|\leq\epsilon,\\&\mathbb{E}\,\mathrm{dist}\big(0,\nabla_x H(u^{t_0},y^{t_0})+\partial F(u^{t_0})\big)\leq\epsilon.\end{aligned}$$

For the clarity of the presentation, we refer to the proof of Theorems 2.1 and 2.2 to Appendix B.

3 Experiments

In this section, we give the numerical experiments of training DNNs with relaxed binary weights using Algorithm 1. Our experiments are mainly performed on three different network structures, VGG-11 [45], VGG-16 [45], and ResNet-18 [21]. These DNNs are trained using two different data sets, CIFAR-10 [26] and CIFAR-100 [26], respectively. All experiments are implemented in the PyTorch platform with Python 3.6. The experiments are run on a remote desktop with a Tesla V100 GPU and 32 GB of memory. In these experiments, we compare our algorithm with BC, BR, and PSGD algorithms.

3.1 Algorithms

In all experiments, we compare our Algorithm 1 with the BinaryConnect [13], BinaryRelax(BR) [52] and proximal stochastic gradient descent (PSGD) [12, 41] algorithms. The BC algorithm has become one of the most important algorithms for training quantized DNNs (such as Xnor-net). The BinaryRelax algorithm is a relaxed two-stage algorithm proposed in [52] and has been shown to be effective for training quantized DNNs. BC [13] and PSGD [12, 41] trained DNNs by minimizing the following problem:

$$\min_{\tilde{W}} L(\tilde{W})+\mathcal{I}_Q(\tilde{W}), \tag{3.1}$$

where the $L(\cdot)$ is the loss function of DNN and $\mathcal{I}_Q$ is the indicator function of the quantized set Q defined as

$$Q=\prod_{i=1}^{N}Q_i \tag{3.2}$$

with

$$Q_i=\left\{s_iB_i\,|\,B_i\in\{-1,+1\}^{n_i},s_i\in\mathbb{R}^+\right\},$$

where N is the number of layers in the network, n_i is the dimension of the vectorized filter in the i-th layer, and s_i is the magnitude that can be calculated precisely (see [39]). To keep the paper self-contained, here we recall the calculation details of s_i.

The projection of floating-point weights onto the quantized set Q is to solve the following optimization problem:

$$\tilde{W}^* \in \arg\min_{\tilde{W}\in Q} \|\tilde{W}-U\|^2 =: \mathrm{Proj}_Q(U). \tag{3.3}$$

According to the definition of Q, the projection problem (3.3) can be reformulated as

$$\begin{aligned} &(s_i^*, Z^*) = \arg\min_{s_i, Z} \|s_i \cdot Z - U_i\|^2 \\ &\text{s.t.} \quad Z \in \{-1,+1\}^n, \end{aligned} \tag{3.4}$$

where U_i denotes the weights of the i-th layer. It has been shown in [39] that the closed (exact) solution of (3.4) can be obtained as follows:

$$s_i^* = \frac{\|vec(U_i)\|_1}{n}, \quad Z_{i,j} = \begin{cases} 1, & \text{if} \quad U_{ij} \geq 0, \\ -1, & \text{otherwise.} \end{cases} \tag{3.5}$$

When PSGD and BC algorithms are applied to problem (3.1), the specific form is respectively given as

$$\begin{aligned} &\text{(PSGD)} \quad \begin{cases} U^{t+1} \in U^t - \gamma \tilde{\nabla} L(U^t), \\ \tilde{W}^{t+1} \in \mathrm{Proj}_Q(U^{t+1}), \end{cases} \\ &\text{(BC)} \quad \begin{cases} U^{t+1} \in U^t - \gamma \tilde{\nabla} L(\tilde{W}^t), \\ \tilde{W}^{t+1} \in \mathrm{Proj}_Q(U^{t+1}). \end{cases} \end{aligned} \tag{3.6}$$

Notice that the BinaryRelax [52] is a two-stage algorithm. In the first stage, the BR algorithm minimizes the following problem:

$$\min_{\tilde{W}} \frac{\lambda}{2} \mathrm{dist}(\tilde{W}, Q)^2 + L(\tilde{W}),$$

where λ is the regularization parameter, $L(\cdot)$ is the loss function of DNN. In the second stage, the BR algorithm solves the problem (3.1) as BC and PSGD. The specific BR algorithm is given as

$$\text{(BR)} \quad \begin{cases} \tilde{W}^{t+1} = \begin{cases} \dfrac{\lambda_t \mathrm{Proj}_Q(U^{t+1}) + U^{t+1}}{\lambda_t + 1} (\lambda_t = \rho \lambda_{t-1}, \rho > 1), & \text{if} \quad t < K, \\ \mathrm{Proj}_Q(U^{t+1}), & \text{if} \quad t \geq K, \end{cases} \\ U^{t+1} \in U^t - \gamma^t \tilde{\nabla} L(\tilde{W}^t). \end{cases} \tag{3.7}$$

When our Algorithm 1 is applied to train DNN with quantized weights, we solve the model

$$\min_{W,\tilde{W}} \frac{\lambda}{2} \|W - \tilde{W}\|_F^2 + L_W(p,q) + \mathcal{I}_{\mathcal{Q}}(\tilde{W}),$$

where $L(\cdot)$ is the loss function of DNN and $\mathcal{I}_Q$ is the indicator function of the quantitative set as in (3.2). Let

$$G(W)=L_W(p,q),\quad H(\tilde{W},W)=\frac{\lambda}{2}\|W-\tilde{W}\|_F^2,\quad F(\tilde{W})=\mathcal{I}_Q(\tilde{W}).$$

Then, we can obtain the following closed solution for the first subproblem:

$$W^{t+1}=\frac{(\beta-\lambda)W^t+\lambda\tilde{W}^t-\tilde{\nabla}L(W^t)}{\beta}.$$

For the second subproblem (1.3b), we can calculate

$$\tilde{W}^{t+1}=\frac{\gamma\lambda W^{t+1}+Z^t}{\lambda\gamma+1}.$$

Finally, the third subproblem becomes

$$U^{t+1}\in\arg\min_U \mathcal{I}_Q(U)+\frac{1}{2\gamma}\|U-2\tilde{W}^{t+1}+Z^t\|^2,$$

and then the closed solution of this subproblem is

$$U^{t+1}\in\mathrm{Proj}_Q(2\tilde{W}^{t+1}-Z^t).$$

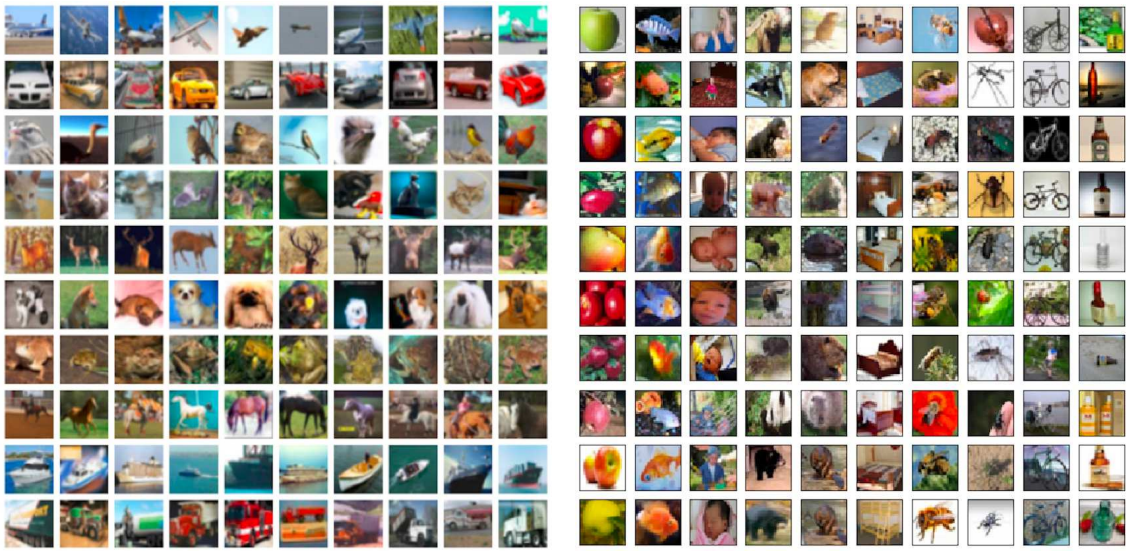
Thus, the specific algorithm is presented as follows:

$$\text{(STAM)}\quad\begin{cases}W^{t+1}=\dfrac{(\beta-\lambda)W^t+\lambda\tilde{W}^t-\tilde{\nabla}L(W^t)}{\beta},\\ \tilde{W}^{t+1}=\dfrac{\gamma\lambda W^{t+1}+Z^t}{\gamma\lambda+1},\\ U^{t+1}\in\mathrm{Proj}_Q(2\tilde{W}^{t+1}-Z^t),\\ Z^{t+1}=Z^t+(U^{t+1}-\tilde{W}^{t+1}).\end{cases}\tag{3.8}$$

Based on (3.8), we know that $\|Z^{t+1}-Z^t\|=\|U^{t+1}-\tilde{W}^{t+1}\|$. From the proof of Theorem 2.2 and (B.4), we can deduce that when $T>\mathcal{O}(\epsilon^{-4})$, $\|U^T-\tilde{W}^T\|<\epsilon$. Since U^T directly represents the quantized weights, we use U^T as the quantized weight to calculate test accuracy in all experiments. For PSGD, BC, and BR, we use $\tilde{W}^T$ to calculate the test accuracy.

3.2 CIFAR-10 dataset

In this subsection, we train DNNs on the CIFAR-10 dataset [26] using the four different algorithms presented in subsection 3.1. The CIFAR-10 dataset consists of 10 categories of 32×32 color images containing a total of 60,000 images, with each category containing



(a) CIFAR-10 (b) CIFAR-100

Figure 1: Sampled images from CIFAR-10 and CIFAR-100 training datasets.

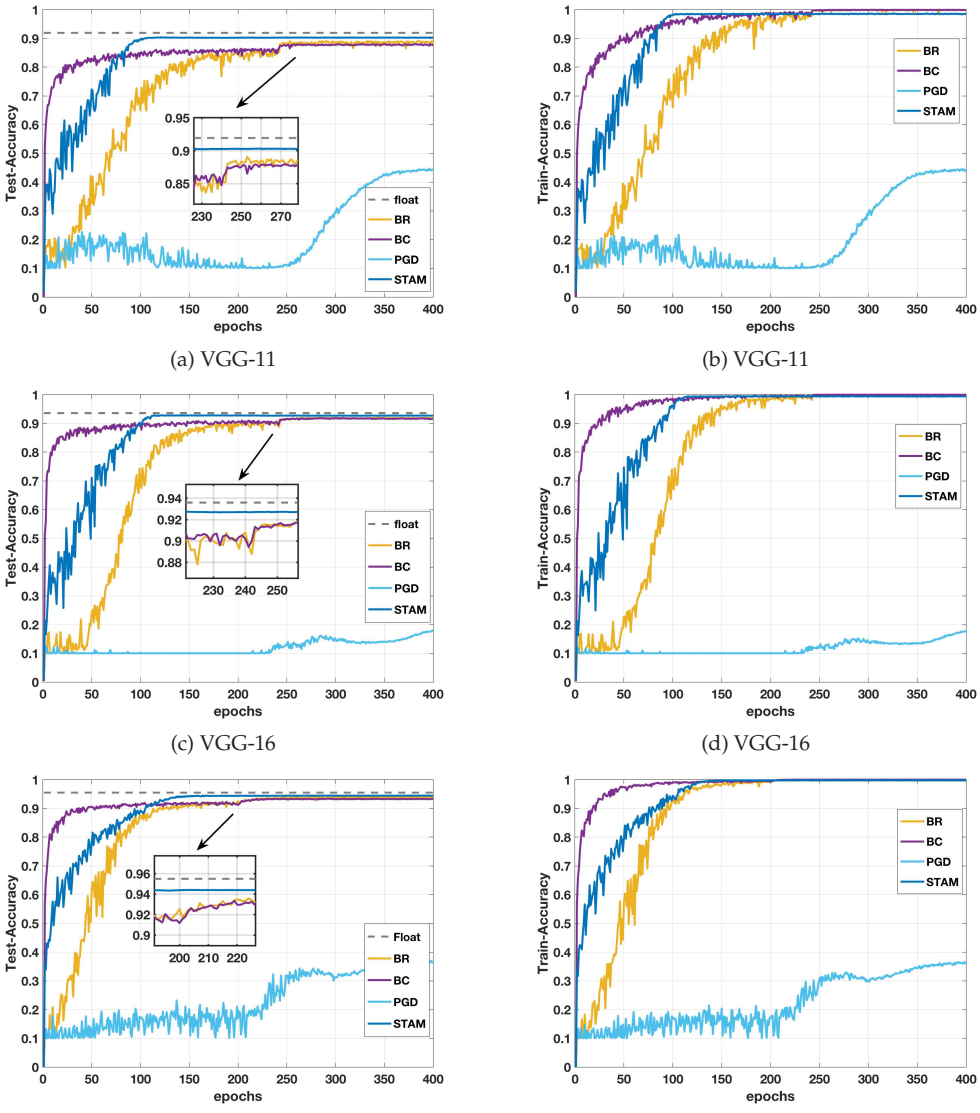
6,000 images (see Fig. 1(a)). A set of 50000 images is used as the training set and 10000 as the test set. In this experiment, we use 400 epochs to train DNNs with quantized weights for all algorithms. In the process of training DNN, we set the batch size to 128. In the BR algorithm, we set the parameter $K = 250$ to start the second phase. We finally compare the accuracy of all algorithms on the test and train sets of CIFAR-10.

We show the train and test accuracy in Fig. 2 and give the best test accuracy of all methods in Table 1. From Fig. 2, we can see that when training the first 80 epochs, our algorithm is slower than BC and is faster than BR. However, after about 100 epochs have been trained, our algorithm tends to be stable and the test accuracy reaches the optimal for all DNNs. This is because we set the step size γ to be relatively large at the beginning, and then decrease γ to the minimum value 1e-2 to ensure convergence. This treatment is similar to the gradual decay of the learning rate in BC and BR algorithms. This experimental result is consistent with our convergence analysis. In Table 1, we present the best test accuracy for the different methods. We can see that our algorithm, STAM has the best test accuracy, especially for VGG-11 DNN, and STAM is far superior to any other algorithm.

At the same time, for VGG-11 DNN, we also present the change of test accuracy with the running time in Fig. 4(a). For all algorithms, we compare the running time on a computer with an Intel 8700k CPU and an NVIDIA gtx 1080ti GPU. In Table 2, we show the average time of each epoch for each algorithm to train VGG-11. From Table 2, we can see that the average running time per epoch of our algorithm is slightly higher than that of other algorithms. Even so, we can see from Fig. 4, that our algorithm can achieve the best test accuracy in a very short running time compared with other algorithms.

Table 1: The best test accuracy of different methods for CIFAR-10 dataset.

DNN	Float	PSGD	BC	BR	STAM
VGG-11	91.93	45.13	88.12	89.11	**90.37**
VGG-16	93.59	18.84	91.86	92.01	**92.88**
ResNet-18	95.49	37.27	93.43	93.91	**94.55**



(a) VGG-11 (b) VGG-11

(c) VGG-16 (d) VGG-16

(e) ResNet-18 (f) ResNet-18

Figure 2: Test accuracy and training accuracy for CIFAR-10 dataset.

Table 2: The average time (second/epoch) for different methods.

	BC	BR	STAM
CIFAR-10	5.2874	5.6831	6.8525
CIFAR-100	5.6260	5.7406	6.6307

3.3 CIFAR-100 dataset

In this subsection, we perform the test on the CIFAR-100 dataset [26] containing 100 classes, each of which contains 600 images (see Fig. 1(b)). These 600 images are divided into 500 training images and 100 test images, respectively.

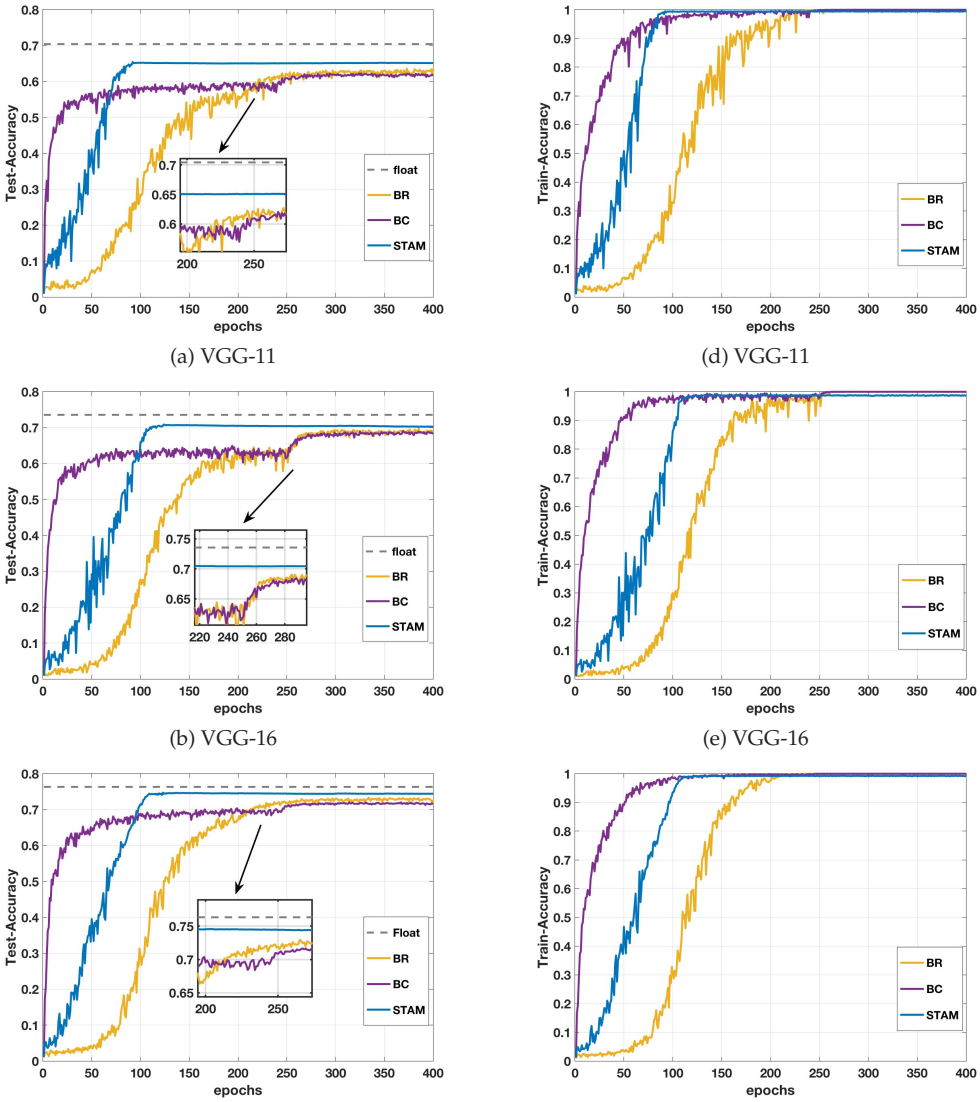
For the CIFAR-100 dataset, the same three neural networks, VGG-11, VGG-16, and ResNet-18 are investigated. The total epochs number is set as 400 and the batch size is set to 128 for all the neural networks. In the BR algorithm, we set the parameter $K=250$ to start the second phase. The parameters of BC and BR algorithms are set as described in their papers [13] and [52] respectively. For a fair comparison, in all the algorithms, we do not use pretraining. The compared results of the train and test accuracy are shown in Fig. 3. It can be seen from Fig. 3 that the test accuracy of our algorithm is always higher than that of the other two algorithms after 100 epochs. The best test accuracy for different methods is presented in Table 3. We note that the results for BC and BR presented here are lower than those with pretraining techniques employed in [52]. However, our best test accuracy is still far higher than the best accuracy of BR and BC from [52]. In particular, for VGG-11, the test accuracy of our algorithm is 2% higher than that of the other two algorithms. We also give the results of running time in Fig. 4(b) and Table 2. Our algorithm can reach the best test accuracy in about 10 minutes, and the test accuracy is far better than the other two algorithms.

Table 3: The best test accuracy of different methods for CIFAR-100 dataset.

DNN	Float	BC	BR	STAM
VGG-11	70.43	62.37	63.70	**65.43**
VGG-16	73.55	69.01	69.73	**70.87**
ResNet-18	76.32	72.04	73.94	**74.72**

4 Conclusion

Binary quantized neural networks effectively reduce storage requirements and computational complexity, enabling deployment on resource-constrained portable devices. In this paper, we propose a novel model for training binary quantized neural networks. Compared to existing methods for training binary quantized neural networks, we leverage the information from floating-point parameters to achieve better-quantized weights and improve generalization performance. We formulate a composite optimization prob-



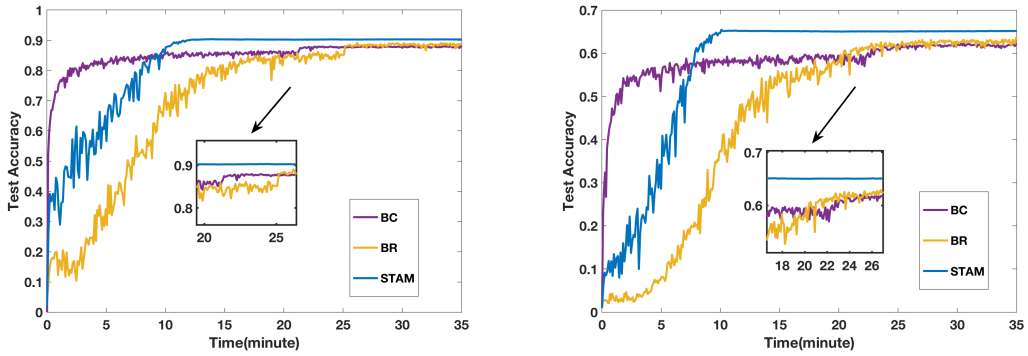
(a) VGG-11 (d) VGG-11

(b) VGG-16 (e) VGG-16

(c) ResNet-18 (f) ResNet-18

Figure 3: Test accuracy and train accuracy for CIFAR-100 dataset.

lem with a cross-term and introduce a stochastic three-block alternating minimization algorithm that fully exploits the cross-structure. Based on the expected smoothness assumption, which is known to be very weak for practical applications, we analyze the convergence and convergence rate of the proposed algorithm. Additionally, we employ STAM to train relaxed binary quantized deep neural networks on CIFAR-10 and CIFAR-100 datasets, using three different network architectures (VGG-11, VGG-16, and ResNet-18). Compared with state-of-the-art methods for training quantized neural networks, our model and algorithm demonstrate the effectiveness in training relaxed binary quantized



(a) CIFAR-10 (b) CIFAR-100

Figure 4: Test accuracy and running time for CIFAR-10 and CIFAR-100 datasets.

DNNs. Furthermore, our STAM algorithm 1 should also be applied to other practical problems, such as sparse nonnegative matrix factorization, blind image deblurring, and sparse principal component analysis. We will further explore these applications in future work.

Appendix A. Proofs of Lemmas 2.1-2.3

Firstly, we recall the following lemma whose proof can be found in [25].

Lemma A.1. *Let the function f be bounded from below by an infimum $f^{\inf}\in\mathbb{R}$, differentiable, and ∇f is L-Lipschitz. Then for all $x\in\mathbb{R}^d$ we have*

$$\|\nabla f(x)\|^2\leq 2L\big(f(x)-f^{\inf}\big).$$

Proof of Lemma 2.1. According to the definition of $\nabla G(\cdot)$ and using the convexity of the squared norm $\|\cdot\|^2$, we have

$$\begin{aligned}
&\mathbb{E}\big[\|\widetilde{\nabla}G(y)+\nabla_y H(x,y)\|^2\big]\\
=&\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N v_i\nabla G_i(y)+\nabla_y H(x,y)\right\|^2\right]\\
\leq&\mathbb{E}\left[2\left\|\frac{1}{N}\sum_{i=1}^N v_i\nabla G_i(y)\right\|^2+2\|\nabla_y H(x,y)\|^2\right]\\
\leq&2\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N v_i\nabla G_i(y)\right\|^2+2\mathbb{E}\|\nabla_y H(x,y)\|^2\\
\leq&\frac{2}{N}\sum_{i=1}^N\big[\mathbb{E}\big[v_i^2\big]\,\|\nabla G_i(y)\|^2+\|\nabla_y H(x,y)\|^2\big].
\end{aligned}$$

Furthermore, from Lemma A.1 we have

$$\|\nabla G_i(y)\|^2 \le 2L_1^i\big(G_i(y)-G_i^{\inf}\big),$$
$$\|\nabla_y H(x,y)\|^2 \le 2L_3^*\big(H(x,y)-H^{\inf}\big),$$

where $G_i^{\inf}=\inf_y G_i(y)$ and $H^{\inf}=\inf_{x,y}H(x,y)$. Then we get

$$\begin{aligned}
&\mathbb{E}\big[\|\widetilde{\nabla}G(y)+\nabla_y H(x,y)\|^2\big]\\
\le&\frac{4}{N}\sum_{i=1}^N\Big[\mathbb{E}\big[v_i^2\big]L_1^i\big(G_i(y)-G_i^{\inf}\big)+L_3^*\big(H(x,y)H^{\inf}\big)\Big]\\
\le&\frac{4\max_i\big(L_1^i\mathbb{E}\big[v_i^2\big]\big)+L_3^*}{N}\sum_{i=1}^N\big(G_i(y)+H(x,y)-G_i^{\inf}-H^{\inf}\big)\\
\le&\frac{4\max_i\big(L_1^i\mathbb{E}\big[v_i^2\big]\big)+L_3^*}{N}\sum_{i=1}^N\big(G_i(y)+H(x,y)-(G+H)^{\inf}+(G+H)^{\inf}-G_i^{\inf}-H^{\inf}\big)\\
\le&\Big(4\max_i\big(L_1^i\mathbb{E}[v_i^2]\big)+L_3^*\Big)[G(y)+H(x,y)-(G+H)^{\inf}]\\
&+\frac{4\max_i\big(L_1^i\mathbb{E}\big[v_i^2\big]\big)+L_3^*}{N}\sum_{i=1}^N\big((G+H)^{\inf}-G_i^{\inf}-H^{\inf}\big).
\end{aligned}$$

Note that

$$G(y)+H(x,y)=\frac{1}{N}\sum_{i=1}^N\big(G_i(y)+H(x,y)\big)\ge\frac{1}{N}\sum_{i=1}^N\big(G_i^{\inf}+H^{\inf}\big),$$

then $(1/N)\sum_{i=1}^N(G_i^{\inf}+H^{\inf})$ is a lower bound of $G(y)+H(x,y)$, and then

$$\Delta^{\inf}:=\frac{1}{N}\sum_{i=1}^N\big((G+H)^{\inf}-G_i^{\inf}-H^{\inf}\big)=(G+H)^{\inf}-\frac{1}{N}\sum_{i=1}^N\big(G_i^{\inf}+H^{\inf}\big)\ge 0.$$

The proof is complete. □

Proof of Lemma 2.2. By Cauchy inequality and ES inequality, we have

$$\begin{aligned}
&\mathbb{E}\big[\|\widetilde{\nabla}G(y)+\nabla_y H(x,y)\|^2\big]\\
\le&2\mathbb{E}\|\widetilde{\nabla}G(y)\|^2+2\|\nabla_y H(x,y)\|^2\\
\le&4A_0\big(G(y)-G^{\inf}\big)+2B_0\|\nabla G(y)\|^2+2C_0+2\|\nabla_y H(x,y)\|^2.
\end{aligned}$$

It follows from Lemma A.1 that

$$\|\nabla G(y)\|^2\le 2L_1\big(G(y)-G^{\inf}\big),$$
$$\|\nabla_y H(x,y)\|^2\le 2L_3^*\big(H(x,y)-H^{\inf}\big).$$

Thus, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\widetilde{\nabla}G(y)+\nabla_y H(x,y)\|^2\right]\\
\leq&(4A_0+4B_0L_1)\left(G(y)-G^{\inf}\right)+4L_3^*(H(x,y)-H^{\inf})+2C_0\\
\leq&2A[G(y)+H(x,y)-(G+H)^{\inf}]+2A[(G+H)^{\inf}-G^{\inf}-H^{\inf}]+2C_0\\
=:&2A[G(y)+H(x,y)-(G+H)^{\inf}]+C,
\end{aligned}$$

where

$$\begin{aligned}
A:=&\max(2A_0+2B_0L_1,2L_3^*),\\
C:=&2A[(G+H)^{\inf}-G^{\inf}-H^{\inf}]+2C_0.
\end{aligned}$$

Thus, we obtain the conclusion. □

Next, we begin to prove Lemma 2.3. Before proving Lemma 2.3, we show an estimate for the gradient of the objective function in a model (1.2) with respect to y-variable.

Lemma A.2. *Let Assumptions* 2.1 *and* 2.2 *be satisfied. Let $\beta,\gamma>0$ be the parameters in Algorithm* 1 *and $A,C\geq 0$ be the constants in Lemma* 2.2*. Then, for any $M>0$, we have*

$$\begin{aligned}
&\frac{1}{\beta}\eta_t+\frac{M}{2}\mathbb{E}\|y^{t+1}-y^t\|^2\\
\leq&\left(1+\frac{(L+M)A}{\beta^2}\right)\delta_t-\delta_{t+1}+\frac{(L+M)C}{2\beta^2}+\mathbb{E}[H(x^{t+1},y^{t+1})-H(x^t,y^{t+1})],
\end{aligned}\tag{A.1}$$

where

$$\begin{aligned}
\eta_t=&\mathbb{E}\left\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\right\|^2,\\
\delta_t=&\mathbb{E}[G(y^t)+H(x^t,y^t)-(G+H)^{\inf}].
\end{aligned}$$

Proof. By the optimality condition of the first subproblem (1.3a), we have

$$y^{t+1}-y^t=\frac{-\widetilde{\nabla}G(y^t)-\nabla_y H(x^t,y^t)}{\beta}.\tag{A.2}$$

Using the L-smoothness of $G+H$ with respect to the variable y in Assumption 2.2, we get

$$\begin{aligned}
&G(y^{t+1})+H(x^t,y^{t+1})\\
\leq&G(y^t)+H(x^t,y^t)+\left\langle\nabla G(y^t)+\nabla_y H(x^t,y^t),y^{t+1}-y^t\right\rangle+\frac{L}{2}\|y^{t+1}-y^t\|^2,
\end{aligned}$$

where $L:=L_3^*+L_1$. For any $M>0$, adding up $(M/2)\|y^{t+1}-y^t\|^2$ on the both sides yields that

$$G(y^{t+1})+H(x^t,y^{t+1})+\frac{M}{2}\|y^{t+1}-y^t\|^2$$

$$
\begin{aligned}
&\leq G(y^t)+H(x^t,y^t)+\left\langle \nabla G(y^t)+\nabla_y H(x^t,y^t), y^{t+1}-y^t\right\rangle+\frac{L+M}{2}\|y^{t+1}-y^t\|^2\\
&\leq G(y^t)+H(x^t,y^t)+\left\langle \nabla G(y^t)+\nabla_y H(x^t,y^t), \frac{-\widetilde{\nabla} G(y^t)-\nabla_y H(x^t,y^t)}{\beta}\right\rangle\\
&\quad+\frac{L+M}{2}\left\|\frac{-\widetilde{\nabla} G(y^t)-\nabla_y H(x^t,y^t)}{\beta}\right\|^2\\
&\leq G(y^t)+H(x^t,y^t)-\frac{1}{\beta}\left\langle \nabla G(y^t)+\nabla_y H(x^t,y^t), \widetilde{\nabla} G(y^t)+\nabla_y H(x^t,y^t)\right\rangle\\
&\quad+\frac{L+M}{2\beta^2}\|\widetilde{\nabla} G(y^t)+\nabla_y H(x^t,y^t)\|^2, \qquad \text{(A.3)}
\end{aligned}
$$

where the second inequality follows from (A.2). Taking expectations in (A.3) conditional on y^t, we obtain

$$
\begin{aligned}
&E\left[G(y^{t+1})+H(x^t,y^{t+1})+\frac{M}{2}\|y^{t+1}-y^t\|^2\,|\,y^t\right]\\
&\leq G(y^t)+H(x^t,y^t)-\frac{1}{\beta}\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\|^2\\
&\quad+\frac{L+M}{2\beta^2}\mathbb{E}[\|\widetilde{\nabla} G(y^t)+\nabla_y H(x^t,y^t)\|^2].
\end{aligned}
$$

From Lemma 2.2, we have

$$
\begin{aligned}
&\mathbb{E}\left[G(y^{t+1})+H(x^t,y^{t+1})+\frac{M}{2}\|y^{t+1}-y^t\|^2\,|\,y^t\right]\\
&\leq G(y^t)+H(x^t,y^t)-\frac{1}{\beta}\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\|^2\\
&\quad+\frac{(L+M)A}{\beta^2}[G(y^t)+H(x^t,y^t)-(G+H)^{\inf}]+\frac{(L+M)C}{2\beta^2},
\end{aligned}
$$

where $(G+H)^{\inf}=\min_{x,y} G(y)+H(x,y)$. Subtracting $(G+H)^{\inf}$ from both sides gives

$$
\begin{aligned}
&\mathbb{E}\left[G(y^{t+1})+H(x^t,y^{t+1})+\frac{M}{2}\|y^{t+1}-y^t\|^2-(G+H)^{\inf}\,|\,y^t\right]\\
&\leq\left(1+\frac{(L+M)A}{\beta^2}\right)[G(y^t)+H(x^t,y^t)-(G+H)^{\inf}]+\frac{(L+M)C}{2\beta^2}\\
&\quad-\frac{1}{\beta}\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\|^2.
\end{aligned}
$$

Taking expectation again and applying the tower property, we obtain

$$
\mathbb{E}\left[G(y^{t+1})+H(x^t,y^{t+1})+\frac{M}{2}\|y^{t+1}-y^t\|^2-(G+H)^{\inf}\right]
$$

$$\leq \left(1+\frac{(L+M)A}{\beta^2}\right)\mathbb{E}[G(y^t)+H(x^t,y^t)-(G+H)^{\inf}]+\frac{(L+M)C}{2\beta^2}$$
$$-\frac{1}{\beta}\mathbb{E}\left\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\right\|^2.$$

Adding up $H(x^{t+1},y^{t+1})$ on the both sides and rearranging,

$$\begin{aligned}&\mathbb{E}\left[G(y^{t+1})+H(x^{t+1},y^{t+1})-(G+H)^{\inf}\right]\\&\quad+\frac{1}{\beta}\mathbb{E}\left\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\right\|^2+\frac{M}{2}\mathbb{E}\|y^{t+1}-y^t\|^2\\&\leq\left(1+\frac{(L+M)A}{\beta^2}\right)\mathbb{E}[G(y^t)+H(x^t,y^t)-(G+H)^{\inf}]\\&\quad+\frac{(L+M)C}{2\beta^2}+\mathbb{E}[H(x^{t+1},y^{t+1})-H(x^t,y^{t+1})].\end{aligned}$$

Setting

$$\delta_{t+1}=\mathbb{E}[G(y^{t+1})+H(x^{t+1},y^{t+1})-(G+H)^{\inf}],$$
$$\eta_t=\mathbb{E}\left\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\right\|^2,$$

we have

$$\begin{aligned}&\frac{1}{\beta}\eta_t+\frac{M}{2}\mathbb{E}\|y^{t+1}-y^t\|^2\\&\leq\left(1+\frac{(L+M)A}{\beta^2}\right)\delta_t-\delta_{t+1}+\frac{(L+M)C}{2\beta^2}+\mathbb{E}[H(x^{t+1},y^{t+1})-H(x^t,y^{t+1})].\end{aligned}$$

The proof is complete. □

We next show an estimate for the gradient of the objective function in a model (1.2) with respect to x-variable.

Lemma A.3. *Let Assumption* 2.2 *be satisfied. Suppose that the parameter $\gamma>0$ is chosen such that*

$$\mathcal{K}_1:=-\frac{-3+5\gamma l+2\gamma}{2\gamma}-\frac{5(1+\gamma L_2^*)^2}{4\gamma}>0.$$

Then

$$\begin{aligned}&\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\\&\leq\frac{\mathcal{K}_2}{\mathcal{K}_1}\left(\mathbb{E}[H(x^t,y^{t+1})-H(x^{t+1},y^{t+1})]+\mathbb{E}[\mathcal{M}_t-\mathcal{M}_{t+1}]+\mathcal{K}_3\mathbb{E}\|y^{t+1}-y^t\|^2\right),\end{aligned}$$

where

$$\mathcal{M}_t:=\mathcal{M}(x^t,u^t,z^t)=F(u^t)+\frac{1}{2\gamma}\|2x^t-u^t-z^t\|^2-\frac{1}{2\gamma}\|x^t-z^t\|^2-\frac{1}{\gamma}\|u^t-x^t\|^2,$$
$$\mathcal{K}_2:=\frac{5(1+\gamma L_2^*)^2}{4\gamma^2},\quad \mathcal{K}_3:=L_4^*(1+5\gamma L_4^*)+\frac{5\mathcal{K}_1(L_4^*)^2}{\mathcal{K}_2}.$$

Proof. Since x^{t+1} is the minimizer of subproblem (1.3b), by the strongly convex of $H(\cdot,y^{t+1})+\|\cdot-z^t\|^2/(2\gamma)$ we have

$$\begin{aligned}&H(x^{t+1},y^{t+1})+\frac{1}{2\gamma}\|z^t-x^{t+1}\|^2\\ \leq& H(x^t,y^{t+1})+\frac{1}{2\gamma}\|z^t-x^t\|^2-\frac{1}{2}\left(\frac{1}{\gamma}-l\right)\|x^{t+1}-x^t\|^2.\end{aligned}\tag{A.4}$$

On the other hand, using the fact that u^{t+1} is the minimizer of subproblem (1.3c), we get

$$\begin{aligned}&F(u^{t+1})+\frac{1}{2\gamma}\|2x^{t+1}-u^{t+1}-z^t\|^2\\ \leq& F(u^t)+\frac{1}{2\gamma}\|2x^{t+1}-u^t-z^t\|^2.\end{aligned}\tag{A.5}$$

Summing up (A.4) and (A.5), we have

$$\begin{aligned}&H(x^{t+1},y^{t+1})+F(u^{t+1})+\frac{1}{2\gamma}\|2x^{t+1}-u^{t+1}-z^t\|^2+\frac{1}{2\gamma}\|z^t-x^{t+1}\|^2\\ \leq& H(x^t,y^{t+1})+F(u^t)+\frac{1}{2\gamma}\|2x^{t+1}-u^t-z^t\|^2+\frac{1}{2\gamma}\|z^t-x^t\|^2\\ &-\frac{1}{2}\left(\frac{1}{\gamma}-l\right)\|x^{t+1}-x^t\|^2.\end{aligned}\tag{A.6}$$

Notice that we can rewrite

$$\begin{aligned}&\|2x^{t+1}-u^{t+1}-z^t\|^2-\|2x^{t+1}-u^{t+1}-z^{t+1}\|^2\\ =&2\langle 2x^{t+1}-u^{t+1}-(u^{t+1}-x^{t+1})-z^t,z^{t+1}-z^t\rangle+\|z^{t+1}-z^t\|^2\\ =&-4\|z^{t+1}-z^t\|^2+2\langle x^{t+1}-z^t,z^{t+1}-z^t\rangle+\|z^{t+1}-z^t\|^2\\ =&-2\|z^{t+1}-z^t\|^2-\|x^{t+1}-z^{t+1}\|^2+\|x^{t+1}-z^t\|^2,\end{aligned}\tag{A.7}$$

where the subproblem (1.3d) is used in the second and third equalities, and the fact that $2\langle a,b\rangle=-(\|a-b\|^2-\|a\|^2-\|b\|^2)$ is used in the last equality. Combining (A.7) with (A.6) and using the subproblem (1.3d), we obtain

$$\begin{aligned}&H(x^{t+1},y^{t+1})+F(u^{t+1})+\frac{1}{2\gamma}\|2x^{t+1}-u^{t+1}-z^{t+1}\|^2\\ &-\frac{1}{2\gamma}\|x^{t+1}-z^{t+1}\|^2-\frac{1}{\gamma}\|u^{t+1}-x^{t+1}\|^2\\ \leq& H(x^t,y^{t+1})+F(u^t)+\frac{1}{2\gamma}\|2x^{t+1}-u^t-z^t\|^2+\frac{1}{2\gamma}\|z^t-x^t\|^2\\ &-\frac{1}{\gamma}\|x^{t+1}-z^t\|^2-\frac{1}{2}\left(\frac{1}{\gamma}-l\right)\|x^{t+1}-x^t\|^2.\end{aligned}\tag{A.8}$$

Using the basic equality $\|2a-b-c\|^2-2\|a-c\|^2=2\|a-b\|^2-\|b-c\|^2$ twice, we get

$$\begin{aligned}&\|2x^{t+1}-u^t-z^t\|^2+\|z^t-x^t\|^2\\&=2\|x^{t+1}-u^t\|^2+2\|x^{t+1}-z^t\|^2+\|2x^t-u^t-z^t\|^2-\|x^t-z^t\|^2-2\|x^t-u^t\|^2.\end{aligned}\tag{A.9}$$

Substituting (A.9) into (A.8) yields that

$$\begin{aligned}&H(x^{t+1},y^{t+1})+F(u^{t+1})+\frac{1}{2\gamma}\|2x^{t+1}-u^{t+1}-z^{t+1}\|^2\\&\quad-\frac{1}{2\gamma}\|x^{t+1}-z^{t+1}\|^2-\frac{1}{\gamma}\|u^{t+1}-x^{t+1}\|^2\\&\le H(x^t,y^{t+1})+F(u^t)+\frac{1}{2\gamma}\|2x^t-u^t-z^t\|^2-\frac{1}{2\gamma}\|x^t-z^t\|^2-\frac{1}{\gamma}\|x^t-u^t\|^2\\&\quad+\frac{1}{\gamma}\|x^{t+1}-u^t\|^2-\frac{1}{2}\left(\frac{1}{\gamma}-l\right)\|x^{t+1}-x^t\|^2.\end{aligned}\tag{A.10}$$

Furthermore, the optimal condition of (1.3b) is given as

$$0=\nabla_x H(x^{t+1},y^{t+1})+\frac{1}{\gamma}(x^{t+1}-z^t),\tag{A.11}$$

this implies that

$$\frac{1}{\gamma}(z^t-x^{t+1})+lx^{t+1}=\nabla_x\left[H(\cdot,y^{t+1})+\frac{l}{2}\|\cdot\|^2\right](x^{t+1}).$$

Thus, we also have

$$\frac{1}{\gamma}(z^{t-1}-x^t)+lx^t+\nabla_x H(x^t,y^{t+1})-\nabla_x H(x^t,y^t)=\nabla_x\left[H(\cdot,y^{t+1})+\frac{l}{2}\|\cdot\|^2\right](x^t).$$

Since the function $H(\cdot,y^{t+1})+l\|\cdot\|^2/2$ is a convex function, we get

$$\left\langle\frac{1}{\gamma}(z^t-x^{t+1})+lx^{t+1}-\frac{1}{\gamma}(z^{t-1}-x^t)-lx^t-\left[\nabla_x H(x^t,y^{t+1})-\nabla_x H(x^t,y^t)\right],x^{t+1}-x^t\right\rangle\ge 0.$$

Using the Cauchy inequality, we get

$$\begin{aligned}&\frac{1}{\gamma}\langle z^t-z^{t-1},x^{t+1}-x^t\rangle+\frac{1}{2}\left\|\nabla_x H(x^t,y^{t+1})-\nabla_x H(x^t,y^t)\right\|^2+\frac{1}{2}\|x^{t+1}-x^t\|^2\\&\ge\left(\frac{1}{\gamma}-l\right)\|x^{t+1}-x^t\|^2,\end{aligned}$$

and hence

$$\begin{aligned}
&\frac{1}{\gamma}\langle z^t-z^{t-1},x^{t+1}-x^t\rangle\\
\geq&\left(\frac{1}{\gamma}-l-\frac{1}{2}\right)\|x^{t+1}-x^t\|^2-\frac{1}{2}\left\|\nabla_x H(x^t,y^{t+1})-\nabla_x H(x^t,y^t)\right\|^2\\
\geq&\left(\frac{1}{\gamma}-l-\frac{1}{2}\right)\|x^{t+1}-x^t\|^2-\frac{L_4^*}{2}\|y^{t+1}-y^t\|^2,
\end{aligned}$$

where the Lipschitz condition of $\nabla_x H(x,y)$ is used in the last inequality. This is also equivalent to

$$\langle z^t-z^{t-1},x^{t+1}-x^t\rangle\geq\left(1-\gamma l-\frac{\gamma}{2}\right)\|x^{t+1}-x^t\|^2-\frac{\gamma L_4^*}{2}\|y^{t+1}-y^t\|^2.$$

Therefore, we can obtain the estimate on $\|x^{t+1}-u^t\|^2$ as follows:

$$\begin{aligned}
\|x^{t+1}-u^t\|^2&=\|x^{t+1}-x^t+x^t-u^t\|^2\\
&=\|x^{t+1}-x^t\|^2-2\langle x^{t+1}-x^t,z^t-z^{t-1}\rangle+\|z^t-z^{t-1}\|^2\\
&\leq(-1+2\gamma l+\gamma)\|x^{t+1}-x^t\|^2+\gamma L_4^*\|y^{t+1}-y^t\|^2+\|z^t-z^{t-1}\|^2. \qquad \text{(A.12)}
\end{aligned}$$

Denote

$$\Phi(x^t,u^t,z^t;y^{t+1}):=H(x^t,y^{t+1})+\mathcal{M}(x^t,u^t,z^t),$$

where

$$\mathcal{M}_t:=\mathcal{M}(x^t,u^t,z^t)=F(u^t)+\frac{1}{2\gamma}\|2x^t-u^t-z^t\|^2-\frac{1}{2\gamma}\|x^t-z^t\|^2-\frac{1}{\gamma}\|u^t-x^t\|^2.$$

Then substituting (A.12) into (A.10), we have

$$\begin{aligned}
&\Phi(x^{t+1},u^{t+1},z^{t+1};y^{t+1})-\Phi(x^t,u^t,z^t;y^{t+1})\\
\leq&\frac{-3+5\gamma l+2\gamma}{2\gamma}\|x^{t+1}-x^t\|^2+L_4^*\|y^{t+1}-y^t\|^2+\frac{1}{\gamma}\|z^t-z^{t-1}\|^2. \qquad \text{(A.13)}
\end{aligned}$$

Furthermore, according to the optimal condition (A.11), we have

$$0=\nabla_x H(x^{t+1},y^{t+1})-\nabla_x H(x^t,y^t)+\frac{1}{\gamma}(x^{t+1}-z^t)-\frac{1}{\gamma}(x^t-z^{t-1}). \qquad \text{(A.14)}$$

Rewrite (A.14) as

$$\frac{1}{\gamma}(z^t-z^{t-1})=\nabla_x H(x^{t+1},y^{t+1})-\nabla_x H(x^t,y^t)+\frac{1}{\gamma}(x^{t+1}-x^t).$$

Using the Lipschitz continuity of $\nabla_x H(x,y)$ in Assumption 2.2, we have

$$\|z^t-z^{t-1}\|\leq(1+\gamma L_2^*)\|x^{t+1}-x^t\|+\gamma L_4^*\|y^{t+1}-y^t\|.$$

This together with the Cauchy inequality gives

$$\|z^t-z^{t-1}\|^2\leq\frac{5}{4}(1+\gamma L_2^*)^2\|x^{t+1}-x^t\|^2+5(\gamma L_4^*)^2\|y^{t+1}-y^t\|^2. \tag{A.15}$$

Combining this with (A.13), we have

$$\begin{aligned}&\Phi(x^{t+1},u^{t+1},z^{t+1};y^{t+1})-\Phi(x^t,u^t,z^t;y^{t+1})\\ \leq&\left[\frac{-3+5\gamma l+2\gamma}{2\gamma}+\frac{5(1+\gamma L_2^*)^2}{4\gamma}\right]\|x^{t+1}-x^t\|^2+L_4^*(1+5\gamma L_4^*)\|y^{t+1}-y^t\|^2\\ =:&-\mathcal{K}_1\|x^{t+1}-x^t\|^2+L_4^*(1+5\gamma L_4^*)\|y^{t+1}-y^t\|^2,\end{aligned}$$

where

$$\mathcal{K}_1=-\left[\frac{-3+5\gamma l+2\gamma}{2\gamma}+\frac{5(1+\gamma L_2^*)^2}{4\gamma}\right].$$

It is clear that $\mathcal{K}_1>0$ when γ is less than computable thresholding. Thus, we have

$$\begin{aligned}\mathcal{K}_1\|x^{t+1}-x^t\|^2\leq&\Phi(x^t,u^t,z^t;y^{t+1})-\Phi(x^{t+1},u^{t+1},z^{t+1};y^{t+1})\\ &+L_4^*(1+5\gamma L_4^*)\|y^{t+1}-y^t\|^2.\end{aligned} \tag{A.16}$$

Denote $\mathcal{K}_2:=5(1+\gamma L_2^*)^2/(4\gamma^2)$. From (A.15) and (A.16), we have

$$\begin{aligned}\frac{\mathcal{K}_1}{\mathcal{K}_2}\left(\frac{1}{\gamma}\|z^t-z^{t-1}\|\right)^2\leq&\mathcal{K}_1\|x^{t+1}-x^t\|^2+\frac{5\mathcal{K}_1(L_4^*)^2}{\mathcal{K}_2}\|y^{t+1}-y^t\|^2,\\ \leq&\Phi(x^t,u^t,z^t;y^{t+1})-\Phi(x^{t+1},u^{t+1},z^{t+1};y^{t+1})\\ &+\left(L_4^*(1+5\gamma L_4^*)+\frac{5\mathcal{K}_1(L_4^*)^2}{\mathcal{K}_2}\right)\|y^{t+1}-y^t\|^2,\\ =:&\Phi(x^t,u^t,z^t;y^{t+1})-\Phi(x^{t+1},u^{t+1},z^{t+1};y^{t+1})+\mathcal{K}_3\|y^{t+1}-y^t\|^2,\end{aligned}$$

where

$$\mathcal{K}_3=L_4^*(1+5\gamma L_4^*)+\frac{5\mathcal{K}_1(L_4^*)^2}{\mathcal{K}_2}.$$

By the definition of $\Phi(x^t,u^t,z^t;y^{t+1})$, we have

$$\frac{\mathcal{K}_1}{\mathcal{K}_2}\left(\frac{1}{\gamma}\|z^t-z^{t-1}\|\right)^2\leq H(x^t,y^{t+1})-H(x^{t+1},y^{t+1})+\mathcal{M}_t-\mathcal{M}_{t+1}+\mathcal{K}_3\|y^{t+1}-y^t\|^2.$$

Taking the expectation on both sides, we get

$$\begin{aligned}&\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\\ \leq&\frac{\mathcal{K}_2}{\mathcal{K}_1}\Big(\mathbb{E}\big[H(x^t,y^{t+1})-H(x^{t+1},y^{t+1})\big]+\mathbb{E}\big[\mathcal{M}_t-\mathcal{M}_{t+1}\big]+\mathcal{K}_3\mathbb{E}\|y^{t+1}-y^t\|^2\Big).\end{aligned} \tag{A.17}$$

The proof is complete. □

Finally, combining Lemmas A.2 and A.3, we give the proof of Lemma 2.3.

Proof of Lemma 2.3. Summing (A.1) and (A.17), we have

$$\begin{aligned}
&\frac{1}{\beta}\eta_t+\frac{M}{2}\mathbb{E}\|y^{t+1}-y^t\|^2+\frac{\mathcal{K}_1}{\mathcal{K}_2}\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\\
&\leq\left(1+\frac{(L+M)A}{\beta^2}\right)\delta_t-\delta_{t+1}+\frac{(L+M)C}{2\beta^2}+\mathbb{E}[H(x^{t+1},y^{t+1})-H(x^t,y^{t+1})]\\
&\quad+\mathbb{E}[H(x^t,y^{t+1})-H(x^{t+1},y^{t+1})]+\mathcal{K}_3\mathbb{E}\|y^{t+1}-y^t\|^2+\mathbb{E}[\mathcal{M}_t-\mathcal{M}_{t+1}].
\end{aligned}$$

Taking $M=2\mathcal{K}_3$, we get

$$\begin{aligned}
&\frac{1}{\beta}\eta_t+\frac{\mathcal{K}_1}{\mathcal{K}_2}\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\\
&\leq\left(1+\frac{(L+M)A}{\beta^2}\right)\delta_t-\delta_{t+1}+\frac{(L+M)C}{2\beta^2}+\mathbb{E}[\mathcal{M}_t-\mathcal{M}_{t+1}]\\
&=\left(1+\frac{(L+M)A}{\beta^2}\right)\delta_t-\delta_{t+1}+\frac{(L+M)C}{2\beta^2}+\mathbb{E}\left[\mathcal{M}_t-\inf_{t\geq0}\mathcal{M}_t\right]-\mathbb{E}\left[\mathcal{M}_{t+1}-\inf_{t\geq0}\mathcal{M}_t\right]\\
&=\left(1+\frac{(L+M)A}{\beta^2}\right)\delta_t-\delta_{t+1}+\mathcal{M}_t'-\mathcal{M}_{t+1}'+\frac{(L+M)C}{2\beta^2},
\end{aligned}$$

where $\mathcal{M}_t'=\mathbb{E}[\mathcal{M}_t-\inf_{t\geq0}\mathcal{M}_t]$. For $\omega_{-1}>0$, define

$$\omega_t=\frac{\omega_{t-1}}{\left(1+(L+M)A/\beta^2\right)}.$$

Clearly that $\{\omega_t\}_{t\geq-1}$ is a decreasing and positive sequence. Multiplying $\beta\omega_t$ on both sides, we get

$$\begin{aligned}
&\omega_t\eta_t+\frac{\beta\mathcal{K}_1}{\mathcal{K}_2}\omega_t\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\\
&\leq\beta\left(1+\frac{(L+M)A}{\beta^2}\right)\omega_t\delta_t-\beta\omega_t\delta_{t+1}+\frac{(L+M)C}{2\beta}\omega_t+\beta\omega_{t-1}\mathcal{M}_t'-\beta\omega_t\mathcal{M}_{t+1}'\\
&\leq\beta\omega_{t-1}\delta_t-\beta\omega_t\delta_{t+1}+\frac{(L+M)C}{2\beta}\omega_t+\beta\omega_{t-1}\mathcal{M}_t'-\beta\omega_t\mathcal{M}_{t+1}'.
\end{aligned}$$

Summing up both sides from $t=0$ to $t=T-1$ we have

$$\begin{aligned}
&\sum_{t=0}^{T-1}\omega_t\eta_t+\frac{\beta\mathcal{K}_1}{\mathcal{K}_2}\sum_{t=0}^{T-1}\omega_t\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\\
&\leq\beta\omega_{-1}\delta_0-\beta\omega_{T-1}\delta_T+\frac{(L+M)C}{2\beta}\sum_{t=0}^{T-1}\omega_t+\beta\omega_{-1}\mathcal{M}_0'-\beta\omega_{T-1}\mathcal{M}_T'
\end{aligned}$$

$$\leq \beta\omega_{-1}\delta_0+\beta\omega_{-1}\mathcal{M}_0'+\frac{(L+M)C}{2\beta}\sum_{t=0}^{T-1}\omega_t.$$

This completes the proof of lemma. □

Appendix B. Proofs of Theorems 2.1 and 2.2

Proof of Theorem 2.1. We will make use of Lemma 2.3 to complete the proof. Define $W_T=\sum_{t=0}^{T-1}\omega_t$. Dividing W_T on the both sides of (2.11), we get

$$\begin{aligned}&\min_{0\leq t\leq T-1}\left(\eta_t+\frac{\beta\mathcal{K}_1}{\mathcal{K}_2}\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\right)\\&\leq\frac{1}{W_T}\left(\sum_{t=0}^{T-1}\omega_t\eta_t+\frac{\beta\mathcal{K}_1}{\mathcal{K}_2}\sum_{t=0}^{T-1}\omega_t\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\right)\\&\leq\frac{\omega_{-1}}{W_T}\beta\delta_0+\frac{\omega_{-1}}{W_T}\beta\mathcal{M}_0'+\frac{(L+M)C}{2\beta}.\end{aligned}\tag{B.1}$$

It is easy to see that

$$W_T=\sum_{t=0}^{T-1}\omega_t\geq\sum_{t=0}^{T-1}\min_{0\leq i\leq T-1}\omega_i=T\omega_{T-1}=\frac{T\omega_{-1}}{\left(1+(L+M)A/\beta^2\right)^T}.$$

Using this in (B.1) and the fact that $\beta\mathcal{K}_1/\mathcal{K}_2\geq 2$, we have

$$\begin{aligned}&\min_{0\leq t\leq T-1}\left[\eta_t+2\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\right]\\&\leq\frac{\left(1+(L+M)A/\beta^2\right)^T}{T}(\beta\delta_0+\beta\mathcal{M}_0')+\frac{(L+M)C}{2\beta},\end{aligned}\tag{B.2}$$

where $\eta_t=\mathbb{E}\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\|^2$. By the Lipschitz continuity of $\nabla_y H(\cdot,y^t)$ and Cauchy inequality, we have

$$\begin{aligned}&\left\|\nabla G(y^t)+\nabla_y H(u^t,y^t)\right\|^2\\&\leq 2\left\|\nabla G(y^t)+\nabla_y H(x^t,y^t)\right\|^2+2(L_5^*)^2\|z^t-z^{t-1}\|^2.\end{aligned}\tag{B.3}$$

On the other hand, note that the optimal condition of the subproblem (1.3c) is $0\in\partial F(u^t)+(1/\gamma)(u^t-2x^t+z^{t-1})$. Combined this with (A.11) and (1.3d), we obtain

$$\frac{1}{\gamma}(z^{t-1}-z^t)\in\nabla_x H(x^t,y^t)+\partial F(u^t).$$

Hence,

$$\frac{1}{\gamma}(z^{t-1}-z^t)+\big(\nabla_x H(u^t,y^t)-\nabla_x H(x^t,y^t)\big)\in\nabla_x H(u^t,y^t)+\partial F(u^t).$$

This implies that

$$\mathrm{dist}^2\big(0,\nabla_x H(u^t,y^t)+\partial F(u^t)\big)\le 2\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2+2(L_2^*)^2\|z^t-z^{t-1}\|^2.$$

Combined this with (B.3), we get

$$\begin{aligned}
&\mathbb{E}\big\|\nabla G(y^t)+\nabla_y H(u^t,y^t)\big\|^2+\mathbb{E}\,\mathrm{dist}^2\big(0,\nabla_x H(u^t,y^t)+\partial F(u^t)\big)\\
\le&\,2\eta_t+2\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|^2}{\gamma}\right)^2+2\big((L_2^*)^2+(L_5^*)^2\big)\mathbb{E}\|z^t-z^{t-1}\|^2\\
\le&\,2\eta_t+4\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2
\end{aligned}$$

due to $(L_2^*)^2+(L_5^*)^2\le 1/\gamma^2$. Thus, it follows from (B.2) that

$$\begin{aligned}
&\min_{0\le t\le T-1}\Big[\mathbb{E}\big\|\nabla G(y^t)+\nabla_y H(u^t,y^t)\big\|^2+\mathbb{E}\,\mathrm{dist}\big(0,\nabla_x H(u^t,y^t)+\partial F(u^t)\big)\Big]\\
\le&\,2\min_{0\le t\le T-1}\left[\eta_t+2\mathbb{E}\left(\frac{\|z^t-z^{t-1}\|}{\gamma}\right)^2\right]\\
\le&\,2\frac{\big(1+(L+M)A/\beta^2\big)^T}{T}\beta(\delta_0+\mathcal{M}_0')+\frac{(L+M)C}{\beta}.
\end{aligned}\tag{B.4}$$

The proof of theorem is complete. □

Proof of Theorem 2.2. According to the fact that $1+x\le e^x$ $(x\ge 0)$ and choosing $\beta>0$ such that $\beta\ge\sqrt{(L+M)AT}$, we have

$$\left(1+\frac{(L+M)A}{\beta^2}\right)^T\le\exp\left(\frac{(L+M)AT}{\beta^2}\right)\le\exp(1)\le 3.\tag{B.5}$$

It follows from Theorem 2.1 that (B.4) holds if $\beta\ge 2\mathcal{K}_2/\mathcal{K}_1$. Thus, by (B.5) and (B.4) we obtain

$$\begin{aligned}
&\min_{0\le t\le T-1}\Big[\mathbb{E}\big\|\nabla G(y^t)+\nabla_y H(u^t,y^t)\big\|^2+\mathbb{E}\,\mathrm{dist}^2\big(0,\nabla_x H(u^t,y^t)+\partial F(u^t)\big)\Big]\\
\le&\,\frac{(L+M)C}{\beta}+\frac{6\beta}{T}(\delta_0+\mathcal{M}_0').
\end{aligned}\tag{B.6}$$

To make the right-hand side of (B.6) less than ϵ^2, we could require that

$$\frac{(L+M)C}{\beta}\le\frac{\epsilon^2}{2},\quad\frac{6\beta}{T}(\delta_0+\mathcal{M}_0')\le\frac{\epsilon^2}{2}.$$

Then we have

$$\frac{(L+M)C}{\beta} \leq \frac{\epsilon^2}{2} \;\Rightarrow\; \beta \geq \frac{2C(L+M)}{\epsilon^2},$$

and

$$\frac{6\beta}{T}(\delta_0+\mathcal{M}_0') \leq \frac{\epsilon^2}{2} \;\Rightarrow\; T \geq \frac{12\beta(\delta_0+M_0')}{\epsilon^2}. \tag{B.7}$$

Substituting three requirements on β into (B.7) yields that

$$\begin{aligned} T &\geq \frac{12\sqrt{(L+M)AT}(\delta_0+M_0')}{\epsilon^2}, \\ T &\geq \frac{24\mathcal{K}_2(\delta_0+M_0')}{\mathcal{K}_1\epsilon^2}, \\ T &\geq \frac{24C(L+M)(\delta_0+M_0')}{\epsilon^4}. \end{aligned} \tag{B.8}$$

By simplifying the form (B.8), we get

$$T \geq \frac{12(L+M)(\delta_0+M_0')}{\epsilon^2}\max\left\{\frac{2C}{\epsilon^2}, \frac{12(\delta_0+M_0')A}{\epsilon^2}, \frac{2\mathcal{K}_2}{\mathcal{K}_1(L+M)}\right\}.$$

Thus, the desired conclusion is shown. □

Acknowledgments

The authors would like to thank the two anonymous referees very much for their careful reading and valuable comments, which greatly improved the quality of this manuscript.

X. Zhang was supported by the National Natural Science Foundation of China (Grant No. 12090024) and by the Shanghai Municipal Science and Technology Major Project 2021SHZDZX0102. F. Bian is also partly supported by an outstanding PhD graduates development scholarship from the Shanghai Jiao Tong University.

We thank the Student Innovation Center at Shanghai Jiao Tong University for providing us the computing services.

References

[1] A. Aghasi, A. Abdi, N. Nguyen, and J. Romberg, *Net-Trim: Convex pruning of deep neural networks with performance guarantee*, in: Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 3178–3187, 2017.

[2] A. Aghasi, A. Abdi, and J. Romberg, *Fast convex pruning of deep neural networks*, SIAM J. Math. Data Sci., 2(1):158–188, 2020.

[3] M. Ahookhosh, L. T. K. Hien, N. Gillis and P. Patrinos, *A block inertial Bregman proximal algorithm for nonsmooth nonconvex problems with application to symmetric nonnegative matrix tri-factorization*, J. Optim. Theory Appl., 190:234–258, 2021.

[4] H. Attouch, J. Bolte, P. Redont and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality*, Math. Oper. Res., 35:438–457, 2010.
[5] F. Bian, J. Liu, X. Zhang, H. Gao, and J.-F. Cai, *Flash proton radiation therapy via a stochastic three-operator splitting method*, Inverse Problems, 41(2):025007, 2025.
[6] F. Bian and X. Zhang, *A three-operator splitting algorithm for nonconvex sparsity regularization*, SIAM J. Sci. Comput., 43(4):A2809–A2839, 2021.
[7] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146:459–494, 2014.
[8] R. Boţ, E. Csetnek, and D. Nguyen, *A proximal minimization algorithm for structured nonconvex and nonsmooth problems*, SIAM J. Optim., 29(2):1300–1328, 2019.
[9] L. Bottou, F. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev., 60(2):223–311, 2018.
[10] M. T. Chao, F. F. Nong, and M. Y. Zhao, *An inertial alternating minimization with Bregman distance for a class of nonconvex and nonsmooth problems*, J. Appl. Math. Comput., 69:1559–1581, 2023.
[11] C.-S. Chuang, H. J. He, and Z. Y. Zhang, *A unified Douglas-Rachford algorithm for generalized DC programming*, J. Glob. Optim., 82:331–349, 2022.
[12] P. Combettes and J. Pesquet, *Stochastic approximations and perturbations in forward-backward splitting for monotone operators*, Pure Appl. Funct. Anal., 1:13–37, 2016.
[13] M. Courbariaux, Y. Bengio, and J. David, *BinaryConnect: Training deep neural networks with binary weights during propagations*, in: Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 3123–3131, 2015.
[14] D. Davis and W. Yin, *A three-operator splitting scheme and its optimization applications*, Set-Valued Var. Anal., 25:829–858, 2017.
[15] T. Dettmers, *8-bit approximations for parallelism in deep learning*, in: International Conference on Learning Representations, 2016. URL: `http://arxiv.org/abs/1511.04561`.
[16] D. Driggs, J. Tang, J. Liang, M. Davies, and C. Schonlieb, *Spring: A fast stochastic proximal alternating method for non-smooth non-convex optimization*, SIAM J. Imaging Sci., 14(4):1932–1970, 2021.
[17] X. Gao, X. J. Cai, and D. R. Han, *A Gauss-Seidel type inertial proximal alternating linearized minimization for a class of nonconvex optimization problems*, J. Glob. Optim., 76:863–887, 2020.
[18] S. Ghadimi and G. Lan, *Stochastic first- and zeroth-order methods for nonconvex stochastic programming*, SIAM J. Optim., 23(4):2341–2368, 2013.
[19] R. Gower, P. Richtrik, and F. Bach, *Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching*, Math. Program., 188:135–192, 2021.
[20] S. Han, H. Mao, and W. Dally, *Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding*, in: International Conference on Learning Representations, 2016. URL: `http://arxiv.org/abs/1510.00149`.
[21] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 770–778, 2016.
[22] J. Hertrich and G. Steidl, *Inertial stochastic PALM and applications in machine learning*, Sampl. Theory Signal Process. Data Anal., 20:4, 2022.
[23] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, *Binarized neural networks*, in: Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 4114–4122, 2016.
[24] Z. Jia, W. Zhang, X. J. Cai, and D. R. Han, *Stochastic alternating structure-adapted proximal gra-*

dient descent method with variance reduction for nonconvex nonsmooth optimization, Math. Comput., 93(348):1677–1714, 2024.

[25] A. Khaled and P. Richtárik, *Better theory for SGD in the nonconvex world*, Transact. Mach. Learn. Res., 2835–8856, 2023.

[26] A. Krizhevsky, *Learning multiple layers of features from tiny images*, Technical Report TR-2009, University of Toronto, 2009.

[27] A. Krizhevsky, I. Sutskever, and G. Hinton, *ImageNet classification with deep convolutional neural networks*, in: Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 1097–1105, 2012.

[28] C. Leng, Z. Dou, H. Li, S. Zhu, and R. Jin, *Extremely low bit neural network: Squeeze the last bit out with ADMM*, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, AAAI Press, 3466–3473, 2018.

[29] G. Li and T. K. Pong, *Douglas-Rachford splitting for nonconvex optimization with application to nonconvex feasibility problems*, Math. Program., 159:371–401, 2016.

[30] H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein, *Training quantized nets: A deeper understanding*, in: Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 5813–5823, 2017.

[31] Z. Lin, M. Courbariaux, R. Memisevic, and Y. Bengio, *Neural networks with few multiplications*, in: International Conference on Learning Representations, 2016. URL: `https://arxiv.org/abs/1510.03009`.

[32] P. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16(6):964–979, 1979.

[33] E. Lybrand and R. Saab, *A greedy algorithm for quantizing neural networks*, J. Mach. Learn. Res., 22:156, 2021.

[34] M. R. Metel and A. Takeda, *Stochastic proximal methods for non-smooth non-convex constrained sparse optimization*, J. Mach. Learn. Res., 22:115, 2021.

[35] M. C. Mukkamala, P. Ochs, T. Pock and S. Sabach, *Convex-concave backtracking for inertial Bregman proximal gradient algorithms in nonconvex optimization*, SIAM J. Math. Data Sci., 2:658–682, 2020.

[36] M. Nikolova and P. Tan, *Alternating structure-adapted proximal gradient descent for nonconvex nonsmooth block-regularized problems*, SIAM J. Optim., 29(3):2053–2078, 2019.

[37] T. Pock and S. Sabach, *Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems*, SIAM J. Imaging Sci., 9:1756–1787, 2017.

[38] A. Rakhlin, O. Shamir, and K. Sridharan, *Making gradient descent optimal for strongly convex stochastic optimization*, in: Proceedings of the 29th International Conference on Machine Learning, PMLR, 1571–1578, 2012.

[39] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, *XNOR-Net: ImageNet classification using binary convolutional neural networks*, in: Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9908, Springer, 525–542, 2016.

[40] P. Richtarik and M. Takac, *Stochastic reformulations of linear systems: Algorithms and convergence theory*, SIAM J. Matrix Anal. Appl., 41(2):487–524, 2020.

[41] L. Rosasco, S. Villa, and B. C. Vu, *Convergence of stochastic proximal gradient algorithm*, Appl. Math. Optim., 82:891–917, 2020.

[42] K. Sankararaman, S. De, Z. Xu, W. Huang, and T. Goldstein, *The impact of neural network overparameterization on gradient confusion and stochastic gradient descent*, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 119:8469–8479, 2020.

[43] J. Schmidhuber, *Deep learning in neural networks: An overview*, Neural Netw., 61:85–117, 2015.

[44] O. Shamir and T. Zhang, *Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes*, in: Proceedings of the 30th International Conference on Machine Learning, PMLR, 28(1):71–79, 2013.
[45] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, in: International Conference on Learning Representations, 2015. URL: `https://arxiv.org/abs/1409.1556`.
[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, J. Mach. Learn. Res., 15:1929–1958, 2014.
[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 1–9, 2015.
[48] P. Tseng, *An incremental gradient(-projection) method with momentum term and adaptive stepsize rule*, SIAM J. Optim., 8(2):506–531, 1998.
[49] S. Vaswani, F. Bach, and M. Schmidt, *Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron*, in: Proceedings of Machine Learning Research, PMLR, 89:1195–1204, 2019.
[50] Q. X. Wang and D. R. Han, *A generalized inertial proximal alternating linearized minimization method for nonconvex nonsmooth problems*, Appl. Numer. Math., 189:66–87, 2023.
[51] Y. Xu and W. Yin, *Block stochastic gradient iteration for convex and nonconvex optimization*, SIAM J. Optim., 25(3):1686–1716, 2015.
[52] P. Yin, S. Zhang, J. Lyu, S. Osher, Y. Qi, and J. Xin, *Binaryrelax: A relaxation approach for training deep neural networks with quantized weights*, SIAM J. Imaging Sci., 11(4):2205–2223, 2018.
[53] A. Yurtsever, A. Gu, and S. Sra, *Three operator splitting with subgradients, stochastic gradients, and adaptive learning rates*, in: Proceedings of the 35th International Conference on Neural Information Processing Systems, Curran Associates, Inc., 19743–19756, 2021.
[54] A. Yurtsever, V. Mangalick, and S. Sra, *Three operator splitting with a nonconvex loss function*, in: Proceedings of the 38th International Conference on Machine Learning, PMLR, 139:12267–12277, 2021.
[55] J. Zhao, Q. L. Dong, Th. R. Michael and F. H. Wang, *Two-step inertial Bregman alternating minimization algorithm for nonconvex and nonsmooth problems*, J. Glob. Optim., 84:941–966, 2022.