Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Recurrence of optimum for training weight and activation quantized networks

Ziang Long [a,*], Penghang Yin [b], Jack Xin [a]

[a] *University of California, Irvine, United States of America*
[b] *University at Albany, SUNY, United States of America*

ARTICLE INFO

Article history:
Received 19 January 2022
Received in revised form 15 July 2022
Accepted 26 July 2022
Available online 4 August 2022
Communicated by Radu Balan

Keywords:
Quantization
Neural networks
Discrete optimiztion
Straight-through estimator

ABSTRACT

Deep neural networks (DNNs) are quantized for efficient inference on resource-constrained platforms. However, training deep learning models with low-precision weights and activations involves a demanding optimization task, which calls for minimizing a stage-wise loss function subject to a discrete set-constraint. While numerous training methods have been proposed, existing studies for full quantization of DNNs are mostly empirical. From a theoretical point of view, we study practical techniques for overcoming the combinatorial nature of network quantization. Specifically, we investigate a simple yet powerful projected gradient-like algorithm for quantizing two-layer convolutional networks, by repeatedly moving one step at float weights in the negative direction of a heuristic *fake* gradient of the loss function (so-called coarse gradient) evaluated at quantized weights. For the first time, we prove that under mild conditions, the sequence of quantized weights recurrently visit the global optimum of the discrete minimization problem for training a fully quantized network. We also show numerical evidence of the recurrence phenomenon of weight evolution in training quantized deep networks.

1. Introduction

Deep neural networks (DNNs) have been profoundly transforming machine learning, in applications of computer vision, reinforcement learning, and natural language processing, and so on. While achieving human level or even super-human performances, DNNs typically have tremendous number of weights with high resource consumption at inference time, which poses a challenge for their deployment on mobile devices used in our daily lives. To address this challenge, research efforts have been made to the quantizing weights and activations of DNNs while maintaining their superior performance. Quantization methods train DNNs with the weights and activation values being constrained to low-precision arithmetic rather than the conventional floating-point representation in full-precision. [11,27,3,26,17,28], which offer the feasibility of

[*] Corresponding author.
E-mail address: zlong6@uci.edu (Z. Long).

https://doi.org/10.1016/j.acha.2022.07.006

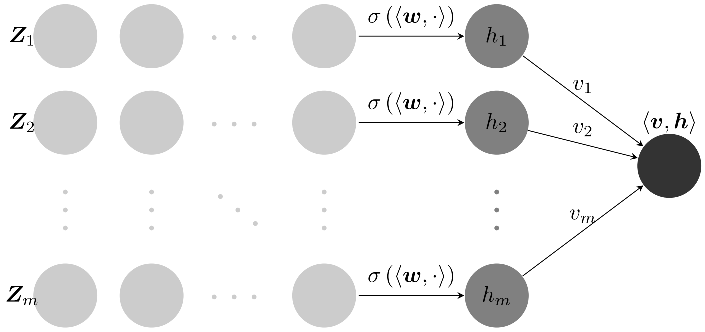
Z_1
Z_2
Z_m
$\sigma(\langle w, \cdot\rangle)$
$\sigma(\langle w, \cdot\rangle)$
$\sigma(\langle w, \cdot\rangle)$
h_1
h_2
h_m
v_1
v_2
v_m
$\langle v, h\rangle$

Fig. 1. One-hidden-layer neural network. The first linear layer resembles a convolutional layer with each $\boldsymbol{Z}_i$ being a patch of size n and $\boldsymbol{w}$ being the shared weights or filter. The second linear layer serves as the classifier.

running DNNs on edge devices with limited memory storage and battery power. For example, DNNs for ImageNet recognition [6] typically requires hundreds of megabytes storage and billions of floating-point operations at inference time. In contrast, the XNOR-Net [18] with binary weights and activations can achieve 58× faster convolutional operations and 32× memory savings, compared with the float counterpart.

Mathematically, training a fully quantized DNN requires solving a challenging optimization problem with a piecewise constant (and non-convex) training loss function and a discrete set-constraint. That is, one considers the following constrained population risk minimization problem:

$$\min_{\boldsymbol{w}} \ f(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\ell(\boldsymbol{w};\boldsymbol{x})] \quad \text{subject to} \quad \boldsymbol{w}\in\mathcal{Q}$$

where $p(\boldsymbol{x})$ is the probability distribution of data; $\ell(\boldsymbol{w};\boldsymbol{x})$ is the sample loss function for input $\boldsymbol{x}$; $\mathcal{Q}$ abstractly denotes the discrete set of quantized weights.

1.1. Problem setup

In this paper, we consider the training of a one-hidden-layer model that outputs the prediction for any input sample $\boldsymbol{Z}\in\mathbb{R}^{m\times n}$:

$$y(\boldsymbol{Z};\boldsymbol{w}) := \sum_{i=1}^{m} v_i \sigma\left(\boldsymbol{Z}_i^\top \boldsymbol{w}\right) = \boldsymbol{v}^\top\sigma\left(\boldsymbol{Z}\boldsymbol{w}\right) \tag{1}$$

where $\boldsymbol{Z}_i^\top$ denotes the i-th row vector of $\boldsymbol{Z}$; $\boldsymbol{w}\in\mathbb{R}^n$ is the trainable weights in the first linear layer, and $\boldsymbol{v}\in\mathbb{R}^m$ the weights in the second linear layer which are assumed to be known and fixed during the training process; the activation function $\sigma(x) = \mathbb{1}_{\{x>0\}}$ is *binary*, acting component-wise on the vector $\boldsymbol{Z}\boldsymbol{w}$. The label is generated according to $y^*_{\boldsymbol{Z}} := y(\boldsymbol{Z};\boldsymbol{w}^*)$ for some unknown true parameters $\boldsymbol{w}^*\in\mathbb{R}^n$ (see Fig. 1).

We fit the described model with quantized weights $\boldsymbol{w}\in\mathcal{Q}$ and binary activation function $\sigma(x) = \mathbb{1}_{\{x>0\}}$ on the i.i.d. Gaussian data $\{(\boldsymbol{Z}, y^*_{\boldsymbol{Z}})\}_{\boldsymbol{Z}\sim\mathcal{N}(\boldsymbol{0},\mathbf{I})}$. In this paper, we will focus on the cases of binary and ternary weights. In the binary case, every quantized weight in $\boldsymbol{w}$ is either α or $-\alpha$ for some universal real-valued constant $\alpha>0$, or equivalently, $\mathcal{Q} = \mathbb{R}_+\times\{\pm1\}^n$; this setup of binary weights is widely adopted in the literature; for example, [18]. Similarly in the ternary case, we take $\mathcal{Q} = \mathbb{R}_+\times\{0,\pm1\}^n$; see [14,25] for examples.

Furthermore, we use the squared loss to measure the discrepancy between the model output $y(\boldsymbol{Z};\boldsymbol{w})$ and the true label $y^*_{\boldsymbol{Z}}$:

$$\begin{aligned}\ell(\boldsymbol{w};\boldsymbol{Z}) &:= \frac{1}{2}\left(y(\boldsymbol{Z};\boldsymbol{w}) - y^*_{\boldsymbol{Z}}\right)^2 \\ &= \frac{1}{2}\left(\boldsymbol{v}^\top\sigma(\boldsymbol{Zw}) - \boldsymbol{v}^\top\sigma(\boldsymbol{Zw}^*)\right)^2,\end{aligned} \tag{2}$$

and cast the learning task as the following population loss minimization problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} f(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{Z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\ell(\boldsymbol{w};\boldsymbol{Z})\right] \quad \text{subject to} \quad \boldsymbol{w}\in\mathcal{Q} \tag{3}$$

where the sample loss function $\ell(\boldsymbol{w};\boldsymbol{Z})$ is given by (2). Hereby it is not hard to show that the gradient of $\ell(\boldsymbol{w};\boldsymbol{Z})$ w.r.t. $\boldsymbol{w}$ is

$$\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z}) = \boldsymbol{Z}^\top\left(\sigma'(\boldsymbol{Zw})\odot\boldsymbol{v}\right)\left(y(\boldsymbol{Z};\boldsymbol{w}) - y^*_{\boldsymbol{Z}}\right). \tag{4}$$

Note that σ' is zero a.e., which makes $\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z})$ zero a.e. and thus inapplicable to the training.

In this setting, we study the following iterative algorithm for training fully quantized networks

$$\begin{cases}\boldsymbol{y}^{t+1} = \boldsymbol{y}^t - \eta_t\,\mathbb{E}[\tilde{\nabla}_{\boldsymbol{w}}\ell(\boldsymbol{w}^t;\boldsymbol{x})] \\ \boldsymbol{w}^{t+1} = \mathrm{proj}_{\mathcal{Q}}(\boldsymbol{y}^{t+1}),\end{cases} \tag{QUANT}$$

where $\tilde{\nabla}_{\boldsymbol{w}}\ell$ denotes some heuristic modification of the vanished gradient $\nabla_{\boldsymbol{w}}\ell$ in (4) based on the so-called straight-through estimator (STE) [1,9], rendering a valid search direction; see section 2.1 for details. Following [24], we shall refer to this fake 'gradient' induced by STE as coarse gradient throughout this paper.

1.2. Related works

For the best possible performance under quantization, the pre-trained full-precision networks need to be re-trained. In the regime of weight quantization, the BinaryConnect scheme:

$$\begin{cases}\boldsymbol{y}^{t+1} = \boldsymbol{y}^t - \eta_t\,\mathbb{E}[\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}^t;\boldsymbol{x})] \\ \boldsymbol{w}^{t+1} = \mathrm{proj}_{\mathcal{Q}}(\boldsymbol{y}^{t+1})\end{cases} \tag{5}$$

was first proposed in [5] for training DNNs with binary (1-bit) weights. It is similar to QUANT, but simply uses the standard gradient $\nabla_{\boldsymbol{w}}\ell$ as the activation values were not quantized. The method was then extended to multi-bit weight quantization such as ternary weight networks [14]. On the theoretical side, [15] analyzed the convergence of BinaryConnect scheme for weight quantization, and proved that $\{\boldsymbol{w}^t\}$ converge to an error floor region of the optimal quantized weights under strong convexity and smoothness assumptions on f. Recently, [16] used an algorithm called "error feedback" for pruning networks [7,22]. It is basically the same as BinaryConnect, except that the weight quantization step $\mathrm{proj}_{\mathcal{Q}}$ is replaced with weight pruning/thresholding which can also be viewed as a projection. The authors showed the convergence to a neighborhood of optimal solution under strong convexity and smoothness assumptions whose radius is $O(\sqrt{d})$ with d being the number of model parameters. Moreover, it remains unclear whether the global optimum can actually be reached in this setting.

The idea of STE has been extensively used for efficiently handling discrete-valued functions arising in machine learning problems. A STE, used in the backward pass only, is a heuristic proxy that substitutes the a.e. zero derivative of discrete component composited in the loss function when computing the gradient under chain rule. Its applications include, but are not limited to, network quantization [10,3,27,4,11,21,2], neural architecture search [20], knowledge graphs [23], discrete latent representations [12]. For networks with binary activations (and real-valued weights), [24] showed that STE-based gradient (called coarse gradient) methods converge only when a proper STE like ReLU STE [3] is used. And they proved that the negative of the resulting coarse gradient points to a descent direction that reduces the training loss. For quantization of both weights and activations, [10,11,3,4,27] utilized QUANT scheme which is the combination of BinaryConnect and STE, and achieved state-of-the-art classification accuracies. Yet to our best knowledge, no convergence results of QUANT have been established to date.

1.3. Main contributions

In this paper, we examine the quantization of one-hidden-layer networks with binary activation and binary or ternary weights using the QUANT algorithm. Surprisingly, the sequence of quantized weights $\{\boldsymbol{w}^t\}$ generated by QUANT is generically divergent. Our key contributions are the *first* theoretical results on the dynamics of QUANT algorithm for learning fully quantized neural nets: (1) we prove the generic divergence if the teacher parameters are not in a quantized state, and give an explicit example of oscillatory divergence behavior (the sequence $\{\boldsymbol{w}^t\}$ has period 3 and jumps between sub-optimal quantized states; see Example 1). (2) We explicitly point out, in the ternary case, the n (out of $3^n - 1$) sub-optimal quantized states that $\{\boldsymbol{w}^t\}$ could visit infinite many times; see Remark 1 and Lemma 8. (3) We prove that $\{\boldsymbol{w}^t\}$ oscillates around the global optimum of quantization problem. Under conditions that teacher parameters and their quantized values are close enough (see Theorem 1), $\{\boldsymbol{w}^t\}$ visits the quantized teacher parameters (the optimum) infinitely often (*recurrence*). Compared with theoretical results for BinaryConnect [15,16], our analysis is more precise and in depth in order to overcome a biased gradient modification in QUANT based on straight-through estimator (STE) [9,1]. Our result is stronger in that the recurrence behavior at global minimum holds *without global convexity assumption of the loss function.*

Organization. In section 2, we introduce the concept of coarse gradient and present some useful preliminary results about the QUANT algorithm. In section 3, we summarize the main results regarding the recurrence behavior of QUANT algorithm. More technical details and sketch of proofs are presented in section 4.

2. Preliminaries

We investigate the convergence behavior of QUANT described in Algorithm 1 for solving the quantization problem (3). In Algorithm 1, $\tilde{\nabla} f := \mathbb{E}[\tilde{\nabla}_{\boldsymbol{w}} \ell(\boldsymbol{w}^t; \boldsymbol{x})]$ stands for coarse gradient [24] in expectation specified in section 2.1 below, which side-steps the vanished gradient issue. Since the loss function $\ell(\boldsymbol{w}; \boldsymbol{Z})$ defined in (2) is scale-invariant, i.e., $\ell(\boldsymbol{Z}; \boldsymbol{w}) = \ell(\boldsymbol{Z}; \boldsymbol{w}/c)$ for any scalar $c > 0$, without loss of generality, we assume that $\|\boldsymbol{w}^*\| = 1$ is unit-normed.

Algorithm 1: QUANT algorithm for solving (3).

```
Input: number of iterations $T$, learning rate $\eta_t$, weight bits $b$;
Initialize: auxiliary real-valued weights $\boldsymbol{y}^0 \in \mathbb{R}^n$, iteration number $t = 1$;
while $t \leq T$ do
    $\boldsymbol{y}^t = \boldsymbol{y}^{t-1} - \eta_t \tilde{\nabla} f(\boldsymbol{w}^{t-1})$;
    $\boldsymbol{w}^t = \mathrm{proj}_{\mathcal{Q}}(\boldsymbol{y}^t)$ ;
    $t = t + 1$;
end
```

In addition, throughout this paper we make the following assumptions on the learning rate $\eta_t > 0$:

1. $\sum_{t=1}^{\infty} \eta_t = \infty$.
2. η_t is upper bounded by some positive constant η.

2.1. Coarse gradient

In this part, we specify the coarse gradient $\tilde{\nabla} f(\boldsymbol{w})$ in Algorithm 1. As shown in (4), the standard gradient of $\ell(\boldsymbol{w};\boldsymbol{Z})$ w.r.t. $\boldsymbol{w}$ is given by

$$\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z}) = \boldsymbol{Z}^\top \left(\sigma'(\boldsymbol{Z}\boldsymbol{w}) \odot \boldsymbol{v}\right) \left(y(\boldsymbol{Z};\boldsymbol{w}) - y^*_{\boldsymbol{Z}}\right).$$

The associated coarse gradient w.r.t. $\boldsymbol{w}$ associated with the sample $(\boldsymbol{Z}, y^*_{\boldsymbol{Z}})$ is given by replacing σ' with a surrogate derivative, known as straight-through estimator (STE) [1,24]. Here we consider the derivative of ReLU function $\mu(x) = \max\{x, 0\}$ which is a widely used STE for quantization, namely, we modify the original gradient $\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z})$ as follows:

$$\tilde{\nabla}_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z}) = \boldsymbol{Z}^\top \left(\mu'(\boldsymbol{Z}\boldsymbol{w}) \odot \boldsymbol{v}\right) \left(y(\boldsymbol{Z};\boldsymbol{w}) - y^*_{\boldsymbol{Z}}\right).$$

The coarse gradient induced by ReLU STE μ' is just the expectation of $\tilde{\nabla}_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z})$ over $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We evaluate the coarse gradient $\tilde{\nabla} f(\boldsymbol{w})$ used in Algorithm 1:

Lemma 1. *The expected coarse gradient of $\ell(\boldsymbol{w};\boldsymbol{Z})$ w.r.t. $\boldsymbol{w}$ is*

$$\begin{aligned}\tilde{\nabla} f(\boldsymbol{w}) :=& \mathbb{E}_{\boldsymbol{Z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\tilde{\nabla}_{\boldsymbol{w}}\ell(\boldsymbol{w};\boldsymbol{Z})] \\ =& \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} - \boldsymbol{w}^*\right).\end{aligned} \tag{6}$$

2.2. Characterization of optimal solutions

To study the convergence of Algorithm 1, we first obtain the closed-form expression of the objective function in (2), which only depends on the angle between quantized weight vector $\boldsymbol{w}$ and the true weight vector $\boldsymbol{w}^*$. This helps us find the expression of global minimum to (1).

Lemma 2. *Let $\boldsymbol{w} \neq \mathbf{0}$ be nonzero vector.*

- *the training loss in (3) is given by*

$$f(\boldsymbol{w}) = \frac{\|\boldsymbol{v}\|^2}{2\pi}\arccos\left(\frac{\boldsymbol{w}^\top\boldsymbol{w}^*}{\|\boldsymbol{w}\|}\right)$$

- *For any $\delta > 0$, $\boldsymbol{w} = \delta \cdot \mathrm{proj}_{\mathcal{Q}}(\boldsymbol{w}^*)$ is a global optimum of quantization problem (3).*

The above result can be easily derived from Lemma 1 of [24], so we omit the proof. Lemma 2 states that the optimal quantized weights are just the projection of $\boldsymbol{w}^*$ onto $\mathcal{Q}$, i.e., the direct quantization of teacher parameters $\boldsymbol{w}^*$. Note that the projection/quantization may not be unique, we refer to $\mathrm{proj}_{\mathcal{Q}}(\boldsymbol{y})$ as any choice of the projection of $\boldsymbol{y}$ onto $\mathcal{Q}$.

2.3. Weight quantization step

The following two lemmas give the closed-form formulas of the projection/quantization $\mathrm{proj}_{\mathcal{Q}}(\cdot)$ in Algorithm 1 in the binary and ternary cases, respectively.

Lemma 3 *(Binary Case). For any non-zero $\boldsymbol{y} \in \mathbb{R}^n$, the projection of $\boldsymbol{y}$ onto $\mathcal{Q} = \mathbb{R}_+ \times \{\pm 1\}^n$ is*

$$\operatorname{proj}_{\mathcal{Q}}(\boldsymbol{y}) = \frac{\|\boldsymbol{y}\|_1}{n} \widetilde{\operatorname{sign}}\,(\boldsymbol{y})\,,$$

where the sign function acts element-wise

$$\widetilde{\operatorname{sign}}\,(\boldsymbol{y})_i = \begin{cases} 1 & \textit{if } \ y_i \geq 0 \\ -1 & \textit{if } \ y_i < 0. \end{cases}$$

The above lemma is due to [18]. In the ternary case, [25] gives the following result:

Lemma 4 *(Ternary Case). For any non-zero $\boldsymbol{y} \in \mathbb{R}^n$, the projection of $\boldsymbol{y}$ on $\mathcal{Q} = \mathbb{R}_+ \times \{0, \pm 1\}^n$ is*

$$\operatorname{proj}_{\mathcal{Q}}(\boldsymbol{y}) = \frac{\left\|\boldsymbol{y}_{[j^*]}\right\|_1}{j^*} \operatorname{sign}\left(\boldsymbol{y}_{[j^*]}\right)$$

where $j^* = \arg\max_{1 \leq j \leq n} \frac{\left\|\boldsymbol{y}_{[j]}\right\|_1^2}{j}$, and $\boldsymbol{y}_{[j]} \in \mathbb{R}^n$ extracts the first j largest entries in magnitude of $\boldsymbol{y}$ and enforces 0 elsewhere. Here,

$$\operatorname{sign}\,(\boldsymbol{y})_i = \begin{cases} 1 & \textit{if } \ y_i > 0 \\ 0 & \textit{if } \ y_i = 0 \\ -1 & \textit{if } \ y_i < 0. \end{cases}$$

3. Main results

By Lemma 2, we assume, for the ease of presentation, that the iterates $\{\boldsymbol{w}^t\}$ are normalized, that is, we re-define $\boldsymbol{w}^t$ in Algorithm 1 by

$$\boldsymbol{w}^t = \widetilde{\operatorname{proj}}_{\mathcal{Q}}(\boldsymbol{y}^t) := \frac{\operatorname{proj}_{\mathcal{Q}}(\boldsymbol{y}^t)}{\|\operatorname{proj}_{\mathcal{Q}}(\boldsymbol{y}^t)\|}$$

Our results extend trivially to the original QUANT without normalization as the value of $f(\boldsymbol{w})$ does not depend on $\|\boldsymbol{w}\|$. Furthermore, we denote by $\widetilde{\operatorname{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*)$ the normalization of the quantization/projection of $\boldsymbol{w}^*$, $\operatorname{proj}_{\mathcal{Q}}(\boldsymbol{w}^*)$, which is a global minimum according to Lemma 2. Our main results show that the optimum $\widetilde{\operatorname{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*)$ is recurrent as long as $\boldsymbol{w}^*$ is close to its normalized quantization.

Theorem 1. *Consider the setup of quantization problem (3). Let $\mathcal{Q}$ be either $\mathbb{R}_+ \times \{\pm 1\}^n$ (binary case) or $\mathbb{R}_+ \times \{0, \pm 1\}^n$ (ternary case). There exists constant $\epsilon > 0$ that depends on the weight bit-width and dimension n only, such that for any $\boldsymbol{w}^*$ with*

$$0 < \left\|\boldsymbol{w}^* - \widetilde{\operatorname{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*)\right\| < \epsilon,$$

we have $\boldsymbol{w}^t = \widetilde{\operatorname{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*)$ for infinitely many t values, where $\{\boldsymbol{w}^t\}$ is the sequence generated by Algorithm 1 with any initialization.

Intuitively, ternary weights should work better than binary weights. The following remark confirms this intuition by showing that the number of points where $\boldsymbol{w}^t$ visits infinitely many times is limited.

Remark 1. In the ternary case, we can further prove that the sequence $\{\boldsymbol{w}^t\}$ generated by Algorithm 1 has at most n sub-sequential limits.

4. Proof sketch

On one hand, the binary case is rather simple. We show that part of the coordinates is stable while others have oscillating sign. We further prove that the set of oscillating coordinates is not empty as long as $\boldsymbol{w}^* \notin \mathcal{Q} = \mathbb{R}_+ \times \{\pm 1\}^n$ is not quantized.

On the other hand, the proof of the ternary case follows the following steps. Our first step shows the sequence $\boldsymbol{y}^t$ generated by Algorithm 1 is bounded away from the origin for all but finitely many t values. Then, our second step shows each coordinate of $\boldsymbol{y}^t$ is of the same sign of $\boldsymbol{w}^*$ for all but finitely many t values. This forces $\boldsymbol{y}^t$ to stay in the same orthant to which $\boldsymbol{w}^*$ belongs. As a matter of fact, an n-dimensional space has in total 2^n orthants, which means $\boldsymbol{y}^t$ can only stay in a small region near $\boldsymbol{w}^*$. After that, our third step furthermore cuts the orthant into $n!$ congruent cones and argues $\boldsymbol{y}^t$ must stay in the same cone where $\boldsymbol{w}^*$ is for all but finitely many t values. In the last step, we prove the ternary case of Theorem 1, which asserts that as long as the underlying true parameter $\boldsymbol{w}^*$ is close to quantized state $\mathcal{Q} = \mathbb{R}_+ \times \{0, \pm 1\}^n$, i.e., any vertex of the cone it belongs to, the optimum is guaranteed to be recurrent.

4.1. Binary weight

In view of Lemmas 2 and 3, we have that the normalized optimum of (3) is $\frac{1}{\sqrt{n}}\widetilde{\operatorname{sign}}\,(\boldsymbol{w}^*)$. The Lemma below shows that some coordinates of $\boldsymbol{w}^t$ generated by Algorithm 1 have oscillating signs.

Proposition 1. *Let $\boldsymbol{w}^t$ be any infinite sequence generated by Algorithm 1. If $|w_j^*| < \frac{1}{\sqrt{n}}$, then there exist infinitely many t_1 and t_2 such that $w_j^{t_1} = \frac{1}{\sqrt{n}}$ and $w_j^{t_2} = -\frac{1}{\sqrt{n}}$.*

The above lemma clearly implies that $\boldsymbol{w}^t$ does not converge, as long as $\boldsymbol{w}^* \notin \mathcal{Q}$.

Corollary 1. *If $\boldsymbol{w}^* \notin \mathcal{Q}$, then any sequence $\{\boldsymbol{w}^t\}$ generated by Algorithm 1 does not converge.*

Since Algorithm 1 does not have a limit unless the weights in the network are already quantized, we ask a natural question: Can we guarantee the optimum to be visited infinitely many times? The general answer is no. We have the following example *demonstrating that the optimum may never be achieved.* We refer the proof of the following example to the appendix.

Example 1. Let $\boldsymbol{w}^* = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2}\sqrt{\frac{11}{3}}\right)$ so that the best the optimum $\widetilde{\operatorname{proj}}_{\mathcal{Q}}\boldsymbol{w}^* = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$. Let $\eta_t = \eta$, $\lambda = \frac{\eta\|\boldsymbol{v}\|^2}{6\sqrt{2\pi}}$ and

$$\begin{cases} y_1^0 \in (-\lambda, 0) \\ y_2^0 \in (0, \lambda) \\ y_3^0 \in (\lambda, 2\lambda) \\ y_4^0 \in (0, \infty) \end{cases}$$

the sequence $\{\boldsymbol{w}^t\}$ generated by Algorithm 1 with initialization $\boldsymbol{y}^0$ satisfies $\boldsymbol{w}^{t+3} = \boldsymbol{w}^t$ and $\boldsymbol{w}^t \neq \widetilde{\operatorname{proj}}_{\mathcal{Q}}\boldsymbol{w}^*$ for all t.

In the following, we give a sufficient condition for the optimum to be recurrent. The condition requires $\boldsymbol{w}^*$ to be close to $\mathcal{Q}$. The following result is for the binary case of Theorem 1.

Theorem 1 *(Binary Case). If the optimum* $\hat{\boldsymbol{w}} := \widetilde{\text{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*) = \frac{1}{\sqrt{n}}\widetilde{\text{sign}}\,(\boldsymbol{w}^*)$ *of (3) satisfies* $0 < \sum_{|w_j^*|<\frac{1}{\sqrt{n}}} |w_j^* - \hat{w}_j| < \frac{2}{\sqrt{n}}$ *then there exist infinitely many* t *values for any sequence* $\{\boldsymbol{w}^t\}$ *generated by Algorithm 1 such that* $\boldsymbol{w}^t = \widetilde{\text{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*)$.

4.2. Ternary weights

The first result shows that $\boldsymbol{w}^t$ generated by Algorithm 1 is generally divergent, and it converges only when the true parameters $\boldsymbol{w}^* \in \mathcal{Q} = \mathbb{R}_+ \times \{0, \pm 1\}^n$.

Proposition 2 *(Ternary Case). Let* $\{\boldsymbol{w}^t\}$ *be any sequence generated by Algorithm 1. If* $\boldsymbol{w}^* \notin \mathcal{Q} = \mathbb{R}_+ \times \{0, \pm 1\}^n$*, then* $\{\boldsymbol{w}^t\}$ *is not a convergent sequence.*

In what follows, we detail the proof of convergence behavior of Algorithm 1.

Our first step is to rule out an exceptional case that the direction of $\boldsymbol{y}^t$ changes significantly in only one iteration. As shown in Lemma 1, the coarse gradient is bounded by a constant depending only on the fixed weight vector $\boldsymbol{v}$. So it suffices to show that $\|\boldsymbol{y}^t\|$ is bounded away from zero for all but finitely many t values.

Lemma 5. *Let* $\{\boldsymbol{y}^t\}$ *be any auxiliary sequence generated by Algorithm 1. If* $\boldsymbol{w}^* \notin \mathcal{Q}$*, then* $\|\boldsymbol{y}^t\|_1$ *converges to infinity as* t *increases.*

Lemma 5 shows that for any positive constant $c > 0$, we have $\|\boldsymbol{y}^t\|_1 > c$ for all but finitely many t values.

Since Lemma 5 guarantees that the direction of $\boldsymbol{y}^t$ will not change significantly, we cut down the region that $\boldsymbol{y}^t$ can belong to in two steps. To describe our first cut down, we need the following definition to make our statement precise.

Definition 1. For any $\boldsymbol{x} \in \mathbb{R}^n$, we define the orthant of $\boldsymbol{x}$ as

$$\boldsymbol{O}(\boldsymbol{x}) := \{\boldsymbol{y} \in \mathbb{R}^n : \text{sign}\,(\boldsymbol{y}) = \text{sign}\,(\boldsymbol{x})\}\,,$$

where sign $(\cdot)$ acts coordinate-wise. Furthermore, we say $\boldsymbol{O}(\boldsymbol{x})$ is regular if any coordinate of $\boldsymbol{x}$ is not zero.

We state some basic properties of the defined orthant.

Proposition 3. *For any* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$*, the following statements are true:*

1. *Either* $\boldsymbol{O}(\boldsymbol{x}) = \boldsymbol{O}(\boldsymbol{y})$ *or* $\boldsymbol{O}(\boldsymbol{x}) \cap \boldsymbol{O}(\boldsymbol{y}) = \emptyset$.
2. $\boldsymbol{x} \in \boldsymbol{O}(\boldsymbol{x})$.
3. $\cup_{\boldsymbol{x}\in\mathbb{R}^n}\boldsymbol{O}(\boldsymbol{x}) = \mathbb{R}^n$.
4. *There are in total* 3^n *orthants.*
5. *There are in total* 2^n *regular orthants.*

Lemma 6. *Let* $\{\boldsymbol{y}^t\}$ *be any auxiliary sequence generated by Algorithm 1. If* $\boldsymbol{w}^* \notin \mathcal{Q}^n$*, then any subsequential limit of* $\tilde{\boldsymbol{y}}^t := \frac{\boldsymbol{y}^t}{\|\boldsymbol{y}^t\|}$ *belongs to the closure of* $\boldsymbol{O}(\boldsymbol{w}^*)$*. Furthermore, if* $\boldsymbol{O}(\boldsymbol{w}^*)$ *is regular, then* $\boldsymbol{y}^t$ *lies in* $\boldsymbol{O}(\boldsymbol{w}^*)$ *for all but finitely many* t *values.*

In our previous step, we have partitioned $\mathbb{R}^n$ into orthants and showed that $\boldsymbol{y}^t$ enter into a small neighborhood of the orthant where $\boldsymbol{w}^*$ stays. Now, we prove a stronger result based on the conclusion of our previous step. We would like to cut each orthant into several congruent cones which we shall define later and argue $\boldsymbol{y}^t$ will move and stay in close neighborhood of the cone where $\boldsymbol{w}^*$ stays. This step makes a stronger statement because we manage to shrink the size of the region where $\boldsymbol{y}^t$ can stay.

Definition 2. For any non-zero vector $\boldsymbol{x} \in \mathbb{R}^n$, we define the cone of $\boldsymbol{x}$ to be

$$\begin{aligned} Cone(\boldsymbol{x}) := \Big\{ & \boldsymbol{y} \in \boldsymbol{O}(\boldsymbol{x}) : \\ & \operatorname{sign}\left(|y_j| - |y_i|\right) = \operatorname{sign}\left(|x_j| - |x_i|\right) \text{ for } \forall i, j \in [n] \Big\}. \end{aligned}$$

Moreover, we say $Cone(\boldsymbol{x})$ is regular if $\boldsymbol{O}(\boldsymbol{x})$ is regular and any $|x_j| \neq |x_i|$ for all $j \neq i$.

Proposition 4. *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, the following statements are true:*

1. *Either $Cone(\boldsymbol{x}) = Cone(\boldsymbol{y})$ or $Cone(\boldsymbol{x}) \cap Cone(\boldsymbol{y}) = \emptyset$.*
2. *$\boldsymbol{x} \in Cone(\boldsymbol{x})$.*
3. *If $\boldsymbol{y} \in Cone(\boldsymbol{x})$, then $Cone(\boldsymbol{y}) = Cone(\boldsymbol{x})$.*
4. *$\cup_{\boldsymbol{y} \in \boldsymbol{O}(\boldsymbol{x})} Cone(\boldsymbol{y}) = \boldsymbol{O}(\boldsymbol{x})$.*
5. *Any regular orthant contains $n!$ regular cones.*

Lemma 7. *Let $\{\boldsymbol{y}^t\}$ be any auxiliary real-valued sequence generated by Algorithm 1. If $\boldsymbol{w}^* \notin \mathcal{Q}$, then any sub-sequential limit of $\tilde{\boldsymbol{y}}^t := \frac{\boldsymbol{y}^t}{\|\boldsymbol{y}^t\|}$ belongs to the closure of $Cone(\boldsymbol{w}^*)$. Moreover, if $Cone(\boldsymbol{w}^*)$ is regular, then $\boldsymbol{y}^t \in Cone(\boldsymbol{w}^*)$ for all but finitely many t values.*

The auxiliary weight vector $\boldsymbol{y}^t$ can only stay in a small region around $\boldsymbol{w}^*$ for large t values.

Definition 3. For any point $\boldsymbol{x} \in \mathbb{R}^n$, assume $(j_1, j_2, \cdots, j_n)$ is a permutation of $[n]$ such that

$$|x_{j_1}| \geq |x_{j_2}| \geq \cdots \geq |x_{j_n}|$$

We define the set of vertexes of $\boldsymbol{x}$ to be

$$\begin{aligned} & \Lambda(\boldsymbol{x}) := \\ & \left\{ \frac{1}{\sqrt{k}} \sum_{i=1}^{k} \operatorname{sign}\left(x_{j_i}\right) \boldsymbol{e}_{j_i} : x_{j_{k+1}} \neq x_{j_k} \text{ are nonzeros} \right\}. \end{aligned}$$

Below are some basic facts about connection between vertexes and cones.

Proposition 5. *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ let $k := |\Lambda(\boldsymbol{x})|$, the following statements are true:*

1. *$0 \leq k \leq n$.*
2. *$\Lambda(\boldsymbol{x})$ is empty if and only if $\boldsymbol{x} = \boldsymbol{0}$.*
3. *$\Lambda(\boldsymbol{x})$ is a subset of the boundary of $Cone(\boldsymbol{x})$.*
4. *$Cone(\boldsymbol{x}) = Cone(\boldsymbol{y})$ if and only if $\Lambda(\boldsymbol{x}) = \Lambda(\boldsymbol{y})$.*
5. *$\widetilde{\mathrm{proj}}_{\mathcal{Q}}(\boldsymbol{x}) \in \Lambda(\boldsymbol{x})$.*
6. *$\boldsymbol{y}$ lies in $Cone(\boldsymbol{x})$ if and only if there exists k positive numbers $\{\mu_{\boldsymbol{z}}(\boldsymbol{y}) : \boldsymbol{z} \in \Lambda(\boldsymbol{x})\}$ such that $\boldsymbol{y} = \sum_{\boldsymbol{z} \in \Lambda(\boldsymbol{x})} \mu_{\boldsymbol{z}}(\boldsymbol{y}) \boldsymbol{z}$.*
7. *$\boldsymbol{y}$ lies in the closure of $Cone(\boldsymbol{x})$ if and only if there exists k non-negative numbers $\{\mu_{\boldsymbol{z}}(\boldsymbol{y}) : \boldsymbol{z} \in \Lambda(\boldsymbol{x})\}$ such that $\boldsymbol{y} = \sum_{\boldsymbol{z} \in \Lambda(\boldsymbol{x})} \mu_{\boldsymbol{z}}(\boldsymbol{y}) \boldsymbol{z}$.*

Table 1
Validation Accuracy of LeNet-5 on MNIST and ResNet-20/VGG-11 on CIFAR-10.

	float	binary	ternary
LeNet-5	99.37	99.33	99.34
ResNet-20	92.33	89.42	90.86
VGG-11	92.15	89.47	90.91

8. $\cup_{\boldsymbol{x}\in\mathbb{R}^n}\Lambda(\boldsymbol{x}) = \{\boldsymbol{x}\in\mathcal{Q} : \|\boldsymbol{x}\| = 1\}$.

Lemma 8. *Let* $\{\boldsymbol{w}^t\}$ *be the sequence generated by Algorithm 1. If* $\boldsymbol{w}^* \notin \mathcal{Q} = \mathbb{R}_+ \times \{0, \pm 1\}^n$*, then* $\boldsymbol{w}^t \in \Lambda(\boldsymbol{w}^*)$ *for all but finitely many* t *values.*

The following result is the ternary case of Theorem 1 stated in section 3.

Theorem 1 *(Ternary Case). Let* $\{\boldsymbol{z}_j\}_{j=1}^k = \Lambda(\boldsymbol{w}^*)$ *where* $\boldsymbol{z}_1 = \widetilde{\mathrm{proj}}_{\mathcal{Q}}\boldsymbol{w}^*$ *is the optimum and* $\boldsymbol{w}^* = \sum_{j=1}^k \lambda_j \boldsymbol{z}_j$*. If* $0 < \sum_{j=2}^k \lambda_j < 1$*, we have* $\boldsymbol{w}^t = \widetilde{\mathrm{proj}}_{\mathcal{Q}}\boldsymbol{w}^*$ *for infinitely many* t *values, where* $\boldsymbol{w}^t$ *is any infinite sequence generated by Algorithm 1 with any initialization.*

Intuitively, the parameter λ_j in Theorem 1 stands for the proportion of time that $\{\boldsymbol{w}^t\}$ stays at $\boldsymbol{z}_j$. For instance, if $\lambda_j \approx 1$, then most of $\{\boldsymbol{w}^t\}$ stay at $\boldsymbol{z}_j$ so that the oscillation has a longer ‘period’ and is harder to observe. On the contrary, if all λ_j’s are almost the same then $\{\boldsymbol{w}^t\}$ behaves like uniform distribution and oscillation becomes more obvious. Beside λ_j’s, a smaller learning rate can render $\boldsymbol{y}^t$ moves slower which can also slow down the oscillation. Although there are ways to stabilize the training process, both our theorem and the experiments in the next section suggests the oscillation behavior is inevitable.

5. Experiments

In this section, we implement QUANT algorithm on both synthetic data and MNIST/CIFAR image data. Our goals are (1) to validate our theoretical findings and (2) to show the appearance of the oscillation behavior in more complicated setups. **With that said, we emphasize that we did not extensively tune the hyper-parameters or use ad-hoc tricks to achieve the best possible validation accuracy.** More comprehensive experimental results for QUANT-based approaches can be found in, for examples, [3,4,11,27]. Here we report the validation accuracies on MNIST and CIFAR-10 for fully quantized networks in Table 1. For both synthetic and image data sets, we observed the oscillation behavior.

5.1. Synthetic data

We take $m = 4$, $n = 8$ in (1) and construct $\boldsymbol{v} \sim N(\boldsymbol{0}, \boldsymbol{I}_m)$ and $\boldsymbol{w}^* \sim N(\boldsymbol{0}, \boldsymbol{I}_n)$ be random vectors. For each run, we fix $\boldsymbol{v}$ and $\boldsymbol{w}^*$ and train the neural network (1) by algorithm (1) for 200 iterations with a learning rate being 0.1. Fig. 2 show the evolution of binary/ternary weight of $\boldsymbol{w}^t$ in the last 100 iterations. Each block of size 8 × 100 corresponds to the evolution of $\boldsymbol{w}^t$ during the 100 iterations. The (quantized) global minimum $\mathrm{proj}_{\mathcal{Q}}\boldsymbol{w}^*$ for each run is shown on the right side of the corresponding subplot in Fig. 2.

5.2. MNIST

We train LeNet-5 with binary/ternary weights and 4-bit activations using QUANT algorithm. For deep networks, the (quantized) global optimum is generally unknown, we instead show the oscillating behavior around local optimum. Note that Fig. 4 shows the training loss no longer drop significantly during the last

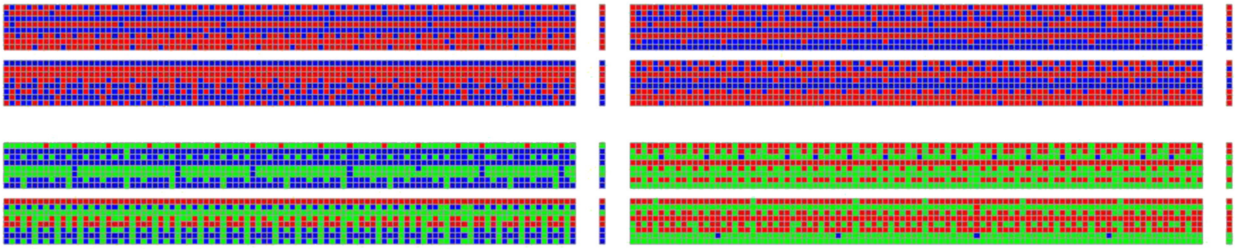

Fig. 2. Evolution of Weight signs of synthetic network described in (1). Each of the 8 large blocks is a colored display of weight sign values via 8 × 100 matrix (i.e., 8 filter weight signs evolved over the last 100 iterations). The bars to the right of blocks are the corresponding optima. **Top two rows**: Binary weight signs, red /blue for 1/−1. **Bottom two rows**: Ternary weight signs, red/green/blue for 1/0/−1.

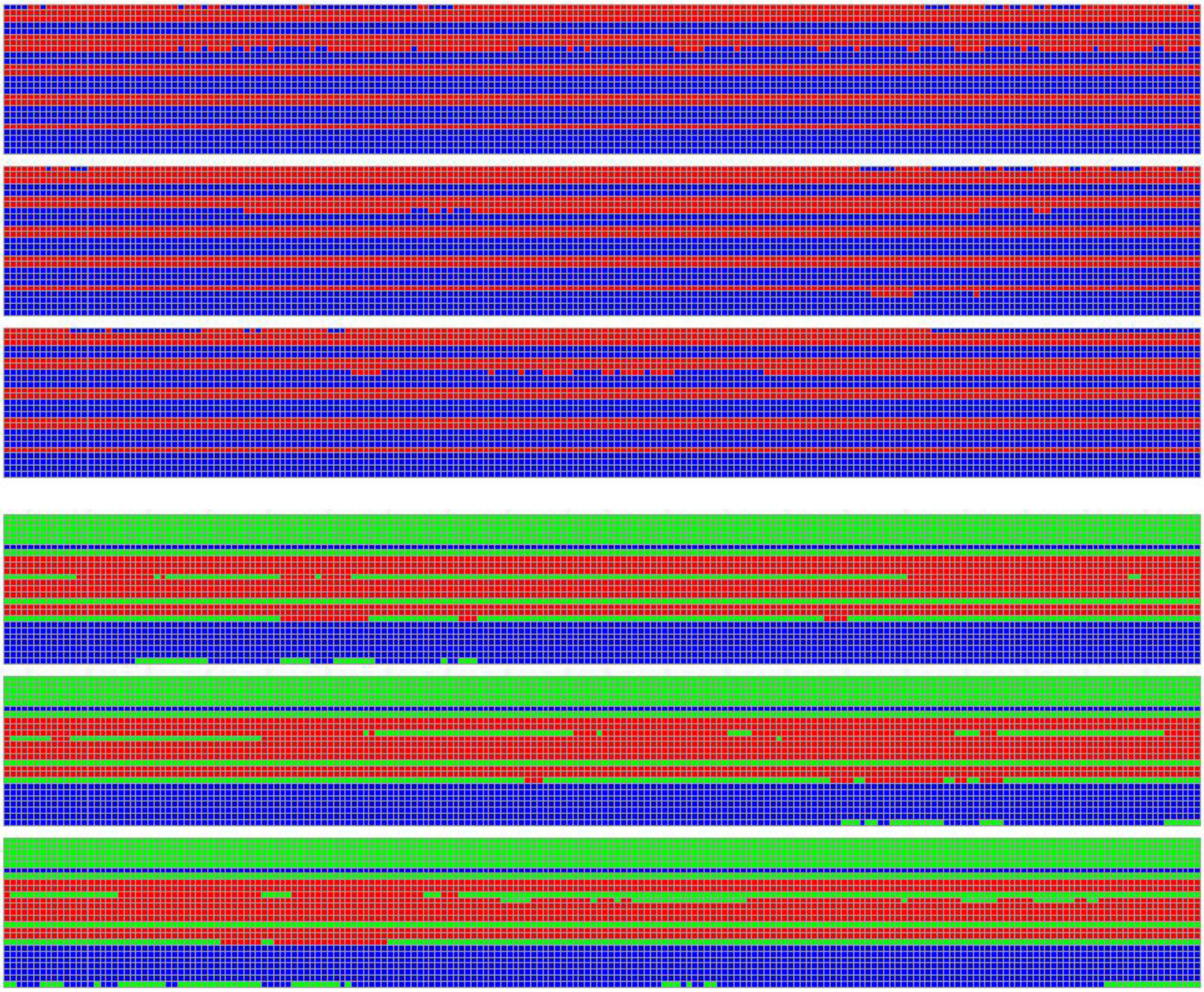

Fig. 3. Evolution of signs of weight filters in the last training epoch (or 600 iterations) of LeNet-5. Each of the six 25 × 200 blocks corresponds to evolution of the 5 × 5 convolutional filter over 200 iterations. **Top three rows**: Binary weights over the last 600 iterations of training, red/blue for sign values 1/−1. **Bottom three rows**: Ternary weights over the last 600 iterations of training, red/green/blue for sign values 1/0/−1.

30 epochs (50 in total). This suggests the network parameters have reached a local valley. However, Fig. 3 shows the iterating sequence of model parameters still have oscillating signs towards the end of training.

Fig. 3 shows the evolution of the quantized weights of one convolutional filter in the first convolutional layer during the last 600 iterations. To visualize the weights, each quantized filter is reshaped into a 25-

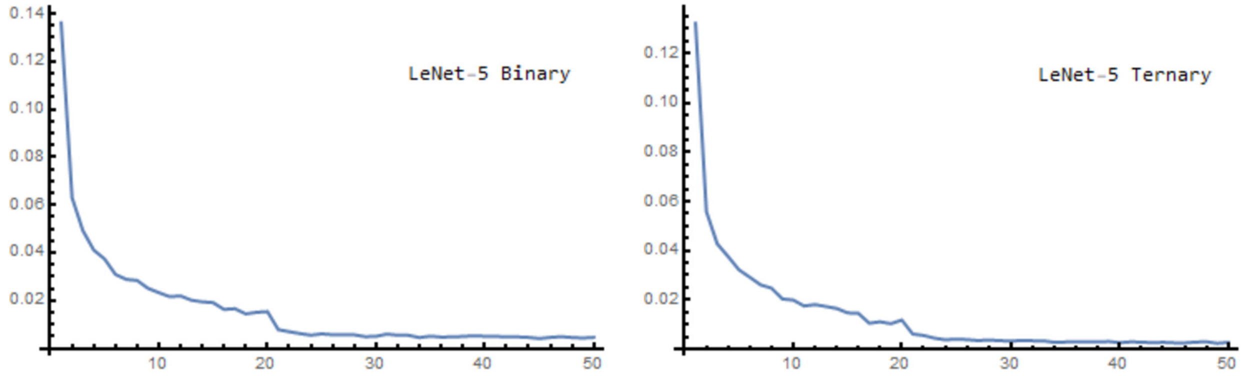

Fig. 4. LeNet-5 Training Loss v.s. Epoch. **Left**: Binary weights. **Bottom**: Ternary weights.

dimensional column vector. Each block (3 in a group) of size 25 × 200 corresponds to the evolution of the one filter during 200 iterations. As we can see from these two figures, a proportion of the weights do not converge to a limit but rather have oscillating signs.

5.3. CIFAR-10

We repeat the experiments on CIFAR-10 [13] with ResNet-20/VGG-11. We train ResNet-20 [8]/VGG-11 [19] with binary/ternary weights and 4-bits activation using QUANT for 200 epochs. We refer to the appendix for some figures that show similar oscillation behavior. Towards the end of training, although there has been no noticeable decay of training loss, we can see a clearer pattern of the oscillating signs of the weights.

6. Concluding remarks

We studied the convergence behavior of widely used QUANT algorithm [10,3,4,27] for the quantization of one-hidden-layer networks. We showed that the sequence of quantized weights $\{\boldsymbol{w}^t\}$ generated by QUANT is generically divergent if the teacher parameters are not in a quantized state, and constructed an explicit example of oscillatory divergence behavior. Under conditions that teacher parameters and their quantized values are close enough, we proved the recurrence of QUANT algorithm at the global minimum.

Acknowledgment

This work was partially supported by NSF grants IIS-1632935, DMS-1854434, DMS-1924548, DMS-1924935, and DMS-1952644.

Appendix A

Lemma 1. *The expected coarse gradient of $\ell(\boldsymbol{w};\boldsymbol{Z})$ w.r.t. $\boldsymbol{w}$ is*

$$\tilde{\nabla} f(\boldsymbol{w}) = \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} - \boldsymbol{w}^*\right). \tag{6}$$

Proof or Lemma 1. [24] gives

$$\tilde{\nabla} f(\boldsymbol{w}) = \frac{\|\boldsymbol{v}^*\|^2}{\sqrt{2\pi}}\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} - \cos\left(\frac{\theta}{2}\right)\frac{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}+\boldsymbol{w}^*}{\left\|\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}+\boldsymbol{w}^*\right\|}\right).$$

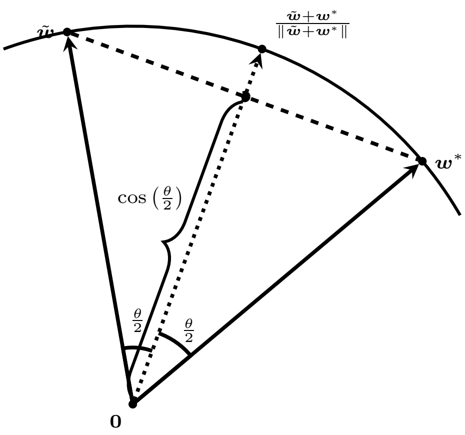
$\tilde{\boldsymbol{w}}$
$\frac{\tilde{\boldsymbol{w}}+\boldsymbol{w}^*}{\|\tilde{\boldsymbol{w}}+\boldsymbol{w}^*\|}$
$\boldsymbol{w}^*$
$\cos\left(\frac{\theta}{2}\right)$
$\frac{\theta}{2}$
$\frac{\theta}{2}$
$\mathbf{0}$

Fig. 5. 2-dim section of $\mathbb{R}^n$ spanned by $\tilde{\boldsymbol{w}}$ and $\boldsymbol{w}^*$.

Let $\tilde{\boldsymbol{w}} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$, we can easily see from Fig. 5 that the coarse gradient can be further simplified as (6). □

Proposition 1. *Let $\boldsymbol{w}^t$ be any infinite sequence generated by Algorithm 1. If $|w_j^*| < \frac{1}{\sqrt{n}}$, then there exist infinitely many t_1 and t_2 values such that $w_j^{t_1} = \frac{1}{\sqrt{n}}$ and $w_j^{t_2} = -\frac{1}{\sqrt{n}}$.*

Proof of Lemma 1. For notational simplicity, since $\left\|w_j^*\right\| < \frac{1}{\sqrt{n}}$, we have

$$\alpha := \frac{1}{\sqrt{n}} - w_j^* > 0 \quad \text{and} \quad \beta := \frac{1}{\sqrt{n}} + w_j^* > 0.$$

Using Lemma 3 in Algorithm 1, we see that

$$\begin{aligned} y_j^{t+1} &= y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}(w_j^* - w_j^t) \\ &= y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^* - \frac{1}{\sqrt{n}} \widetilde{\operatorname{sign}}\left(y_j^t\right) \right), \end{aligned}$$

and thus

$$y_j^{t+1} = \begin{cases} y_j^t - \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \alpha & \text{if } y_j^t \geq 0 \\ y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \beta & \text{if } y_j^t < 0 \end{cases}$$

Since y_j^t is bounded for each fixed $t \geq 0$ and $j \in [n]$, our desired result follows from our assumptions on learning rate η_t. □

Corollary 1. *If $\boldsymbol{w}^* \notin \mathcal{Q}$, any sequence $\{\boldsymbol{w}^t\}$ generated by Algorithm 1 does not converge.*

Proof of Corollary 1. Since $\boldsymbol{w}^* \notin \tilde{\mathcal{Q}}_1^n$, we know there must exist some $j \in [n]$ such that $|w_j^*| < \frac{1}{\sqrt{n}}$ and Proposition 1 gives our desired result. □

Example 1. Let $\boldsymbol{w}^* = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2}\sqrt{\frac{11}{3}}\right)$ so that the best the optimum $\widetilde{\operatorname{proj}}_{\mathcal{Q}} \boldsymbol{w}^* = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$. Let $\eta_t = \eta$, $\lambda = \frac{\eta \|\boldsymbol{v}\|^2}{6\sqrt{2\pi}}$ and

$$\begin{cases} y_1^0 \in (-\lambda, 0) \\ y_2^0 \in (0, \lambda) \\ y_3^0 \in (\lambda, 2\lambda) \\ y_4^0 \in (0, \infty) \end{cases}$$

the sequence $\{\boldsymbol{w}^t\}$ generated by Algorithm 1 with initialization $\boldsymbol{y}^0$ satisfies $\boldsymbol{w}^{t+3} = \boldsymbol{w}^t$ and $\boldsymbol{w}^t \neq \widetilde{\text{proj}}_{\mathcal{Q}} \boldsymbol{w}^*$ for all t.

Proof of Example 1. In order to show the periodicity, it suffices to show $w_j^{t+3} = w_j^t$. Note that $\tilde{\partial}_{w_4} f(\boldsymbol{w}) < 0$ we have $y_4^t > 0$ for all t since $w_4^0 > 0$. It follows that $w_4^t = w_4^0 = \frac{1}{2}$. Next, we would like to show the periodicity of w_j^t for $j \in [3]$. Note that

$$y_j^{t+1} = \begin{cases} y_j^t + \dfrac{\eta \|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^* + \dfrac{1}{2} \right) & \text{if } y_j^t < 0 \\ y_j^t + \dfrac{\eta \|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(-w_j^* + \dfrac{1}{2} \right) & \text{if } y_j^t \geq 0 \end{cases}$$

we choose $\boldsymbol{w}_j^* = \frac{1}{6}$ so that with

$$\lambda = \frac{\eta \|\boldsymbol{v}\|^2}{6\sqrt{2\pi}}$$

we have

$$y_j^{t+1} = \begin{cases} y_j^t + 2\lambda & \text{if } y_j^t < 0 \\ y_j^t - \lambda & \text{if } y_j^t \geq 0 \end{cases}$$

Hence, we have

$$\boldsymbol{w}^t = \begin{cases} \left(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right) & \text{if } t \equiv 0 (\text{mod } 3) \\ \left(\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right) & \text{if } t \equiv 1 (\text{mod } 3) \\ \left(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2} \right) & \text{if } t \equiv 2 (\text{mod } 3) \end{cases} \quad \square$$

Theorem 1 *(Binary Case). If the optimum $\hat{\boldsymbol{w}} := \widetilde{\text{proj}}_{\mathcal{Q}_1^n} \boldsymbol{w}^*$ of (3) satisfies*

$$\sum_{|w_j^*| < \frac{1}{\sqrt{n}}} |w_j^* - \hat{w}_j| < \frac{2}{\sqrt{n}}$$

then there exists infinitely many t values for any sequence $\{\boldsymbol{w}^t\}$ generated by Algorithm 1 such that $\boldsymbol{w}^t = \widetilde{\text{proj}}_{\mathcal{Q}}(\boldsymbol{w}^*)$.

Proof of Theorem 1 on $b = 1$. Without loss of generality, we can assume $w_j^* \geq 0$ for all $j \in [n]$ so that $\hat{w}_j = \frac{1}{\sqrt{n}}$ for all j.

Firstly, if $w_j^* > \frac{1}{\sqrt{n}}$, we know

$$y_j^{t+1} = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^* - w_j^t\right) \geq w_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^* - \frac{1}{\sqrt{n}}\right),$$

so that

$$y_j^t \geq y_j^0 + \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(\sum_{s=0}^{t-1} \eta_s\right) \left(w_j^* - \frac{1}{\sqrt{n}}\right)$$

where the right hand side goes to infinity and thus $w_j^t = \hat{w}_j$ for all but finitely many t values.

Secondly, if $w_j^* = \frac{1}{\sqrt{n}}$, we know when $w_j^t < 0$:

$$y_j^{t+1} = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^* - w_j^t\right) = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \frac{2}{\sqrt{n}}$$

holds so that there must exist some t such that $y_j^t > 0$. Once $y_j^t > 0$ we have $w_j^* = w_j^t$ so that $y_j^{t+1} = y_j^t$ and hence $w_j^t = \hat{w}_j$ for all but finitely many t values.

Third, if $w_j^* < \frac{1}{\sqrt{n}}$, we have $y_j^t \cdot \tilde{\partial}_j f(\boldsymbol{w}^t) > 0$ so that y_j^t is increasing when $y_j^t < 0$ and decreasing when $y_k^t > 0$. This tells us y_j^t is bounded uniformly in t. Furthermore,

$$y_j^t = y_j^0 + \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \Bigg[\left(\sum_{s=0}^{t-1} \mathbb{1}_{\left\{w_j^s > 0\right\}} \eta_s\right) \left(w_j^* - \frac{1}{\sqrt{n}}\right) + \left(\sum_{s=0}^{t-1} \mathbb{1}_{\left\{w_j^s < 0\right\}} \eta_s\right) \left(w_j^* + \frac{1}{\sqrt{n}}\right) \Bigg].$$

For notation simplicity, we let

$$\alpha_j = \frac{1}{\sqrt{n}} - w_j^* > 0 \quad \text{and} \quad \beta_j = w_j^* + \frac{1}{\sqrt{n}} > 0,$$
$$a_j^t = \frac{1}{t} \sum_{s=0}^{t-1} \mathbb{1}_{\left\{w_j^s > 0\right\}} \eta_s \quad \text{and} \quad b_j^t = \frac{1}{t} \sum_{s=0}^{t-1} \mathbb{1}_{\left\{w_j^s < 0\right\}} \eta_s.$$

Now, we have

$$\frac{y_j^t - y_j^0}{t} = \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(-\alpha_j a_j^t + \beta_j b_j^t\right).$$

Since y_j^t is bounded for all $w_j^* < \frac{1}{\sqrt{n}}$, we let $t \to \infty$ so that left hand side vanishes and

$$\lim_{t \to \infty} \frac{b_j^t}{a_j^t + b_j^t} = \frac{\alpha_j}{\alpha_j + \beta_j}.$$

By assumption, we have

$$\lim_{t \to \infty} \sum_{j=1}^{n} \frac{b_j^t}{a_j^t + b_j^t} = \sum_{j=1}^{n} \frac{\alpha_j}{\alpha_j + \beta_j} < 1.$$

Hence, we know

$$\lim_{t\to\infty}\sum_{s=0}^{t-1}\mathbb{1}_{\{\boldsymbol{w}^s=\hat{\boldsymbol{w}}^*\}}\eta_s \geq \lim_{t\to\infty}\left[\left(1-\sum_{j=1}^{n}\frac{b_j^t}{a_j^t+b_j^t}\right)\sum_{s=0}^{t-1}\eta_s\right]=\infty,$$

where we used the assumption $\sum_{t=0}^{\infty}\eta_t=\infty$. Now, the desired result follows. □

Proposition 2 *(Ternary Case). Let $\boldsymbol{w}^t$ be any sequence generated by Algorithm 1. If $\boldsymbol{w}^* \notin \mathcal{Q}$, then $\{\boldsymbol{w}^t\}$ is not a converging sequence.*

Proof of Proposition 2. We prove by contradiction. Observe that $\mathcal{Q}\cap\mathcal{S}^{n-1}$ is a finite set, we know $\boldsymbol{w}^t$ converges to $\boldsymbol{w}^\infty$ is equivalent to $\boldsymbol{w}^t=\boldsymbol{w}^\infty$ for all but finitely many t values. Assume $\boldsymbol{w}^t=\boldsymbol{w}^\infty$ for all but finitely many t values, we know there exists some $T\geq 0$ such that $\boldsymbol{w}^t=\boldsymbol{w}^\infty$ for all $t\geq T$. Thus,

$$\begin{aligned}\boldsymbol{y}^{T+t} &= \boldsymbol{y}^T-\sum_{s=0}^{t-1}\eta_{T+s}\tilde{\nabla}f\left(\boldsymbol{w}^{T+s}\right)\\ &= \boldsymbol{y}^T-\left(\sum_{s=0}^{t-1}\eta_{T+s}\right)\tilde{\nabla}f\left(\boldsymbol{w}^\infty\right)\\ &= \boldsymbol{y}^t+\left(\sum_{s=0}^{t-1}\eta_{T+s}\right)\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(\boldsymbol{w}^*-\boldsymbol{w}^\infty\right).\end{aligned}$$

Now, we have

$$\left\langle\boldsymbol{y}^{T+t},\boldsymbol{w}^\infty\right\rangle=\left\langle\boldsymbol{y}^T,\boldsymbol{w}^\infty\right\rangle+\left(\sum_{s=0}^{t-1}\eta_{T+s}\right)\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left\langle\boldsymbol{w}^*-\boldsymbol{w}^\infty,\boldsymbol{w}^\infty\right\rangle$$

where

$$\left\langle\boldsymbol{w}^*-\boldsymbol{w}^\infty,\boldsymbol{w}^\infty\right\rangle=\left\langle\boldsymbol{w}^*,\boldsymbol{w}^\infty\right\rangle-1<0.$$

Note that $\sum_{s=0}^{\infty}\eta_{T+s}=\infty$, there exists some $T_1(T)$, such that for all $t>T_1(T)$

$$\left\langle\boldsymbol{y}^t,\boldsymbol{w}^\infty\right\rangle<0.$$

This contradicts Lemma 4 and our desired result follows. □

Lemma 5. *Let $\{\boldsymbol{y}^t\}$ be any auxiliary sequence generated by Algorithm 1. If $\boldsymbol{w}^*\notin\mathcal{Q}$, then $\|\boldsymbol{y}^t\|_1$ converges to infinity as t increases.*

Proof of Lemma 5. $\mathcal{Q}\cap\mathcal{S}^{n-1}$ is a compact set because it is finite. Also, since $\mathcal{Q}$ is symmetric, $\boldsymbol{w}^*\notin\mathcal{Q}$ also implies $-\boldsymbol{w}^*\notin\mathcal{Q}$. It follows that

$$\alpha:=\inf_{\boldsymbol{w}\in\mathcal{Q}\cap\mathcal{S}^{n-1}}\theta\left(\boldsymbol{w}^*,\boldsymbol{w}\right)\in(0,\pi).$$

Hence, for any $\boldsymbol{w}\in\mathcal{Q}\cap\mathcal{S}^{n-1}$ we have

$$\left\langle-\tilde{\nabla}f(\boldsymbol{w}),\boldsymbol{w}^*\right\rangle=\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left\langle\boldsymbol{w}^*-\boldsymbol{w},\boldsymbol{w}^*\right\rangle\geq\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(1-\cos\alpha\right).$$

Now, we know

$$\begin{aligned}\left\langle \boldsymbol{y}^T, \boldsymbol{w}^* \right\rangle &= \left\langle \boldsymbol{y}^0, \boldsymbol{w}^* \right\rangle + \sum_{t=0}^{T-1} \eta_t \left\langle -\tilde{\nabla} f\left(\boldsymbol{w}^t\right), \boldsymbol{w}^* \right\rangle \\ &\geq \left\langle \boldsymbol{y}^0, \boldsymbol{w}^* \right\rangle + \left(\sum_{t=0}^{T-1} \eta_t\right) \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \cdot (1-\cos\alpha).\end{aligned}$$

Let $T \to \infty$, we see that $\lim_{t\to\infty} \|\boldsymbol{y}^t\| = \infty$ which is equivalent to $\lim_{t\to\infty} \|\boldsymbol{y}^t\|_1 = \infty$. □

Lemma 9. *Let $\boldsymbol{w} = \mathrm{proj}_{\mathcal{Q}}(\boldsymbol{y})$, then $|y_j| < \frac{1}{5n}\|\boldsymbol{y}\|_1$ implies $w_j = 0$.*

Proof of Lemma 9. Without loss of generality, we assume $y_i \geq 0$ for all $i \in [n]$ and $y_j < \frac{1}{5n}\|\boldsymbol{y}\|_1$ for a fixed $j \in [n]$. Let $\delta = \frac{1}{5n}\|\boldsymbol{y}\|_1$ and

$$j_\delta := |\{i \in [n] : |y_i| \geq \delta\}|$$

we know $j_\delta \geq 1$ by the principle of drawer. Now, with

$$j^* = \arg\max \frac{\left\|\boldsymbol{y}_{[j]}\right\|_1^2}{j}$$

for any $1 \leq k \leq n - j_\delta$

$$\begin{aligned}&\frac{\left\|\boldsymbol{y}_{[j^*]}\right\|_1^2}{j^*} - \frac{\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1^2}{j_\delta+k} \\ \geq&\frac{\left\|\boldsymbol{y}_{[j_\delta]}\right\|_1^2}{j_\delta} - \frac{\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1^2}{j_\delta+k} \\ =&\frac{(j_\delta+k)\left\|\boldsymbol{y}_{[j_\delta]}\right\|_1^2 - j_\delta\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1^2}{j_\delta(j_\delta+k)},\end{aligned}$$

where the numerator is

$$\begin{aligned}&k\left\|\boldsymbol{y}_{[j_\delta]}\right\|_1^2 - j_\delta\left(\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1^2 - \left\|\boldsymbol{y}_{[j_\delta]}\right\|_1^2\right) \\ \geq&k\left[\left\|\boldsymbol{y}_{[j_\delta]}\right\|_1^2 - j_\delta\delta\left(\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1 + \left\|\boldsymbol{y}_{[j_\delta]}\right\|_1\right)\right].\end{aligned}$$

With $\tau = \frac{\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1}{n\delta}$, we have

$$\begin{aligned}&k\left[\left\|\boldsymbol{y}_{[j_\delta]}\right\|_1^2 - j_\delta\delta\left(\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1 + \left\|\boldsymbol{y}_{[j_\delta]}\right\|_1\right)\right] \\ \geq&k\left[\left(\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1 - k\delta\right)^2 - 2n\delta\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1\right] \\ =&k(n\delta)^2\left(\tau^2 - 4\tau + 1\right).\end{aligned}$$

Note that

$$\tau = \frac{\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1}{n\delta} \geq \frac{\|\boldsymbol{y}\|_1 - n\delta}{n\delta} \geq 4,$$

we conclude that

$$\frac{\left\|\boldsymbol{y}_{[j^*]}\right\|_1^2}{j^*} > \frac{\left\|\boldsymbol{y}_{[j_\delta+k]}\right\|_1^2}{j_\delta + k}$$

and hence $j^* \leq j_\delta$. Now, Lemma 4 gives $w_j = 0$. □

Lemma 10. *Let $\{\boldsymbol{w}^t\}$ and $\{\boldsymbol{y}^t\}$ be the sequence and the auxiliary sequence generated by Algorithm 1. Assume $\boldsymbol{w}^* \notin \mathcal{Q}$, the following statements hold.*

- *If $w_j^* = 0$, then y_j^t is bounded and $w_j^t = 0$ for all but finitely many t values.*
- *If $w_j^* \neq 0$, then $\operatorname{sign}\left(y_j^t\right) = \operatorname{sign}\left(w_j^*\right)$ for all but finitely many t values.*

Proof of Lemma 10. On the one hand, we consider the case $w_j^* = 0$, so that

$$y_j^{t+1} = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^* - w_j^t\right) = y_j^t - \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} w_j^t.$$

Note that Lemma 4 shows y_j^t and w_j^t are of the same sign if $w_j^t \neq 0$, we know y_j^t is bounded by $C_j := \max\left\{|y_j^0|, \eta\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\right\}$. Moreover Lemma 5 shows $\|\boldsymbol{y}^t\|_1 > 5nC_j$ for all but finitely many t values. Finally, we see from Lemma 9 that $w_j^t = 0$ for all but finitely many t values.

On the other hand, consider the case $w_j^* \neq 0$. Without loss of generality, we can assume $w_j^* > 0$. Note that whenever $y_j^t \leq 0$, we also have $w_j^t \leq 0$ so that

$$y_j^{t+1} = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^* - w_j^t\right) \geq y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} w_j^*.$$

From the above inequality, we see that y_j^t is increasing where the increment is bounded from below by $\eta_t\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}w_j^* > 0$ where $\sum \eta_t = \infty$, so that there must exist some $T_j > 0$ such that $y_j^{T_j} > 0$. With Lemma 5, we can without loss of generality assume that $\|\boldsymbol{y}^t\|_1 \geq 5n\eta\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}$ for all $t \geq T_j$. For ease of notation, we let $\delta = \eta\frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}$ so that $\|\boldsymbol{y}^t\|_1 \geq 5n\delta$ for all $t \geq T_j$. We shall next prove that $y_j^t \geq 0$ for all $t \geq T_j$. We prove by induction, assume $y_j^t > 0$ for some $t > T_j$ and show $y_j^{t+1} > 0$.

1. If $y_j^t > \delta$,

$$y_j^{t+1} = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^* - w_j^t\right) \geq y_j^t - \delta > 0.$$

2. If $0 < y_j^t \leq \delta$, since $\|\boldsymbol{y}^t\|_1 \geq 5n\delta$, Lemma 9 shows $w_j^t = 0$ so that

$$y_j^{t+1} = y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^* - w_j^t\right) = y_j^t + \frac{\eta_t\delta}{\eta} w_j^* > y_j^t > 0.$$

Combining the above two cases, we get our desired result. □

Lemma 6. *Let $\{\boldsymbol{y}^t\}$ be any auxiliary sequence generated by Algorithm 1. If $\boldsymbol{w}^* \notin \mathcal{Q}$, then any sub-sequential limit of $\tilde{\boldsymbol{y}}^t := \frac{\boldsymbol{y}^t}{\|\boldsymbol{y}^t\|}$ belongs to the closure of $\boldsymbol{O}(\boldsymbol{w}^*)$. Furthermore, if $\boldsymbol{O}(\boldsymbol{w}^*)$ is regular, then $\boldsymbol{y}^t$ lies in $\boldsymbol{O}(\boldsymbol{w}^*)$ for all but finitely many t values.*

Proof of Lemma 6. By Lemma 10, we see that $\operatorname{sign}\left(y_j^t\right) = \operatorname{sign}\left(w_j^*\right)$ for all $\boldsymbol{w}_j^* \neq 0$. We only need to prove $w_j^* = 0$ implies $\lim_{t\to\infty} \tilde{y}_j^t = 0$. Indeed, by Lemma 10, we know that y_j^t is bounded by C_j while Lemma 5 tells us $\|\boldsymbol{y}^t\|$ goes to infinity. Thus, $\lim_{t\to\infty} \tilde{y}_j^t = \frac{y_j^t}{\|\boldsymbol{y}\|} = 0$. □

Lemma 11. *Let $\{\boldsymbol{w}^*\}$ and $\{\boldsymbol{y}^t\}$ be any sequence and auxiliary sequence generated by Algorithm 1. Assuming that $\boldsymbol{w}^* \notin \mathcal{Q}_2^n$, we have the following fact.*

1. *If $|w_j^*| > |w_i^*|$, then $|y_j^t| > |y_i^t|$ for all but finitely many t values.*
2. *If $|w_j^*| = |w_i^*|$, then $\left||y_j^t| - |y_i^t|\right|$ is bounded and $|w_j^t| = |w_i^t|$ for all but finitely many t values.*

Proof of Lemma 11. Without loss of generality, we can assume $w_1^* \geq w_2^* \geq \cdots \geq w_n^* \geq 0$.

For the first statement, we only need to show that $w_j^* > w_{j+1}^*$ implies $y_j^t > y_{j+1}^t$ for all but finitely many t values. Note that whenever $y_j^t < y_{j+1}^t$, then Lemma 4 implies $w_j^t \leq w_{j+1}^t$, hence

$$\begin{aligned}
& y_j^{t+1} - y_{j+1}^{t+1} \\
&= \left(y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^* - w_j^t\right)\right) - \left(y_{j+1}^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_{j+1}^* - w_{j+1}^t\right)\right) \\
&= \left(y_j^t - y_{j+1}^t\right) + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left[\left(w_j^* - w_{j+1}^*\right) + \left(w_{j+1}^t - w_j^t\right)\right] \\
&\geq \left(y_j^t - y_{j+1}^t\right) + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^* - w_{j+1}^*\right).
\end{aligned}$$

Now that we know $y_j^t - y_{j+1}^t$ is increasing as long as it is negative and $\sum \eta_t = \infty$. Therefore, we conclude that there exist infinitely many t values such that $y_j^t - y_{j+1}^t > 0$. We can therefore assume $y_j^T - y_{j+1}^T > 0$, where T is the constant in Lemma 5 such that $\|\boldsymbol{y}^t\|_1 \geq 5n\sqrt{2\epsilon}$ for all $t \geq T$ where we set $\epsilon = \frac{\eta\|\boldsymbol{v}^*\|^2}{\sqrt{2\pi n}}$. Next, we would like to show $y_j^t - y_{j+1}^t > 0$ for all $t \geq T$ by induction.

Next, assuming $y_j^t - y_{j+1}^t > 0$, we want to show $y_j^{t+1} - y_{j+1}^{t+1} > 0$.

On the one hand, if $y_j^t - y_{j+1}^t \geq \epsilon$, we have

$$\begin{aligned}
& y_j^{t+1} - y_{j+1}^{t+1} \\
&= \left(y_j^t - y_{j+1}^t\right) + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left[\left(w_j^* - w_{j+1}^*\right) + \left(w_{j+1}^t - w_j^t\right)\right] \\
&> \left(y_j^t - y_{j+1}^t\right) + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_{j+1}^t - w_j^t\right) \\
&\geq \left(y_j^t - y_{j+1}^t\right) - \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \frac{1}{\sqrt{n}} \geq \left(y_j^t - y_{j+1}^t\right) - \epsilon \geq 0.
\end{aligned}$$

On the other hand, if $y_j^t - y_{j+1}^t < \epsilon$, we still have

$$y_j^{t+1} - y_{j+1}^{t+1} > \left(y_j^t - y_{j+1}^t\right) - \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}\left(w_j^t - w_{j+1}^t\right),$$

so that it suffices to show $w_j^t = w_{j+1}^t$. From Lemma 4, we see that with

$$j^* = \arg\max_{j\in[n]} \frac{\left\|\boldsymbol{y}_{[j]}^t\right\|_1^2}{j}, \tag{7}$$

we only need to show $j \neq j^*$. We prove by contradiction, assuming $j = j^*$ so that $w_j^t > 0$ and $w_{j+1}^t = 0$. Lemma 9 shows $y_j^t \geq \frac{1}{5n}\left\|\boldsymbol{y}^t\right\|_1$. Also, (7) gives

$$\frac{\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1^2}{j-1} \leq \frac{\left\|\boldsymbol{y}_{[j]}^t\right\|_1^2}{j} = \frac{\left(\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1 + y_j^t\right)^2}{j}. \tag{8}$$

Simplifying the above inequality, we get

$$\left(\frac{\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1}{y_j^t}\right)^2 - 2(j-1)\left(\frac{\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1}{y_j^t}\right) - (j-1) \leq 0.$$

Left hand side is a quadratic function of $\left(\frac{\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1}{y_j^t}\right)$, we know

$$\frac{\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1}{y_j^t} \leq j-1+\sqrt{j(j-1)} \leq n-1+\sqrt{n(n-1)} < 2n. \tag{9}$$

We write equation (8) in a different way and get

$$j \geq \frac{\left(\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1 + y_j^t\right)^2}{y_j^t\left(2\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1 + y_j^t\right)}. \tag{10}$$

Now, we use $j = j^*$ again, to get

$$\frac{\left\|\boldsymbol{y}_{[j]}^t\right\|_1^2}{j} \leq \frac{\left\|\boldsymbol{y}_{[j+1]}^t\right\|_1^2}{j+1}. \tag{11}$$

Rewriting the above inequality, we get

$$j \leq \frac{\left(\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1 + y_j^t\right)^2}{y_{j+1}^t\left(2\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1 + 2y_j^t + y_{j+1}^t\right)}. \tag{12}$$

Combining (10) and (12), we get

$$\left(y_j^t - y_{j+1}^t\right)^2 - \left(2\left\|\boldsymbol{y}_{[j-1]}^t\right\|_1 + 4y_j^t\right)\left(y_j^t - y_{j+1}^t\right) + 2\left(y_j^t\right)^2 \leq 0.$$

Solving the above inequality, we get

$$
\begin{aligned}
y_j^t - y_{j+1}^t \geq & \left\| \boldsymbol{y}_{[j-1]}^t \right\|_1 + 2y_j^t \\
& - \sqrt{\left\| \boldsymbol{y}_{[j-1]}^t \right\|_1^2 + 4 \left\| \boldsymbol{y}_{[j-1]}^t \right\|_1 y_j^t + 2(y_j^t)^2}.
\end{aligned} \tag{13}
$$

Combining (9) and (13), we get

$$
y_j^t - y_{j+1}^t \geq (y_j^t)^2 \left(2n + 2 - \sqrt{4n^2 + 4n + 2} \right) > \frac{(y_j^t)^2}{2}. \tag{14}
$$

Recalling that $y_j^t \geq \frac{1}{5n} \|\boldsymbol{y}^t\|_1 \geq \sqrt{2\epsilon}$, we have

$$
y_j^t - y_{j+1}^t > \frac{1}{2} \left(\frac{\|\boldsymbol{y}^t\|_1}{5n} \right)^2 > \epsilon.
$$

This contradiction shows $j \neq j^*$, and hence $w_j^t = w_{j+1}^t$ and it follows that $y_j^{t+1} > y_{j+1}^{t+1}$. Now, we have proved our first statement.

For the second statement, since $w_j^* = w_i^*$, we have

$$
\begin{aligned}
& y_j^{t+1} - y_i^{t+1} \\
= & \left(y_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^* - w_j^t \right) \right) - \left(y_i^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_i^* - w_i^t \right) \right) \\
= & y_j^t - y_i^t - \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left(w_j^t - w_i^t \right) = y_j^t - y_i^t - 2\frac{\eta_t \epsilon}{\eta} \left(w_j^t - w_i^t \right).
\end{aligned}
$$

Hence, we know that $|y_j^t - y_i^t|$ is bounded by

$$
C_{i,j} := \max \left\{ |y_j^0 - y_i^0|, \frac{\eta \|\boldsymbol{v}^*\|^2}{\sqrt{2\pi}} \right\}.
$$

Without loss of generality, we can assume $j < i$ and $\min \left\{ y_j^t, y_i^t \right\} \geq 0$ by Lemma 10. Recalling (14), we have $w_j^t \neq w_i^t$ implying that

$$
|y_j^t - y_i^t| > \frac{\max \left\{ y_j^t, y_i^t \right\}^2}{2} \geq \frac{1}{2} \left(\frac{\|\boldsymbol{y}^t\|}{5n} \right)^2
$$

where the right hand side goes to infinity. This contradicts the boundedness of $|y_j^t - y_i^t|$ if there are infinitely many t values such that $w_j^t \neq w_i^t$. □

Lemma 7. *Let $\{\boldsymbol{y}^t\}$ be any auxiliary sequence generated by Algorithm 1. If $\boldsymbol{w}^* \notin \mathcal{Q}$, then any sub-sequential limit of $\tilde{\boldsymbol{y}}^t := \frac{\boldsymbol{y}^t}{\|\boldsymbol{y}^t\|}$ belongs to the closure of Cone($\boldsymbol{w}^*$). Moreover, if Cone($\boldsymbol{w}^*$) is regular, then $\boldsymbol{y}^t \in$ Cone($\boldsymbol{w}^*$) for all but finitely many t values.*

Proof of Lemma 7. Note that we already have Lemma 6, we only need to show for any sub-sequential limit $\boldsymbol{y}$ of $\tilde{\boldsymbol{y}}^t$, we have $\operatorname{sign}(|y_j| - |y_i|) = \operatorname{sign}\left(|w_j^*| - |w_i^*|\right)$. The first statement of Lemma 11 tells us that it is true for all $\operatorname{sign}\left(|w_j^*| - |w_i^*|\right) \neq 0$. Thus, it suffices to show that $|w_j^*| = |w_i^*|$ implies $|y_j| = |y_i|$.

Note that the second statement of Lemma 11 says that $||y_j| - |y_i||$ is bounded by $C_{i,j}$, while Lemma 5 gives $\lim_{t\to\infty} \|\boldsymbol{y}^t\| = \infty$, we see that

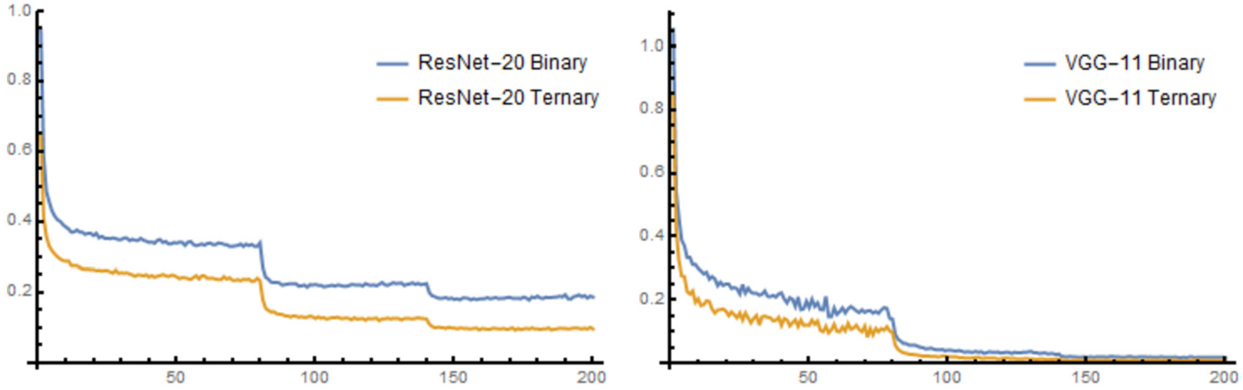

Fig. 6. Training Loss of CIFAR-10. **Left:** Binary/Ternary weight ResNet-20. **Right:** Binary/Ternary weight VGG-11.

$$|\tilde{y}_j| = \lim_{k\to\infty} \frac{|y_j^{t_k}|}{\|\boldsymbol{y}^{t_k}\|} = \lim_{k\to\infty} \frac{|y_i^{t_k}|}{\|\boldsymbol{y}^{t_k}\|} = |\tilde{y}_i|. \quad \square$$

Lemma 8. *Let $\{\boldsymbol{w}^t\}$ be the sequence generated by Algorithm 1. If $\boldsymbol{w}^* \notin \mathcal{Q}$, then $\boldsymbol{w}^t \in \Lambda(\boldsymbol{w}^*)$ for all but finitely many t values.*

Proof of Lemma 8. First, by Proposition 4, $\boldsymbol{y}^t \in Cone(\boldsymbol{w}^*)$ implies $\widetilde{\text{proj}}_{\mathcal{Q}}(\boldsymbol{y}^t) \in \Lambda(\boldsymbol{w}^*)$.

Second, let $\tilde{\partial} Cone(\boldsymbol{w}^*) = \overline{Cone(\boldsymbol{w}^*)} - Cone(\boldsymbol{w}^*)$. Now, a non-zero $\boldsymbol{y}^t \in \tilde{\partial} Cone(\boldsymbol{w}^*)$ implies $Cone(\boldsymbol{y}^t) \subset \tilde{\partial} Cone(\boldsymbol{w}^*)$ so that we also have $\widetilde{\text{proj}}_{\mathcal{Q}}(\boldsymbol{y}^t) \in \Lambda(\boldsymbol{y}^t) \subset \Lambda(\boldsymbol{w}^*)$.

Third, by compactness of $\overline{Cone(\boldsymbol{w}^*)} \cap \mathcal{S}^{n-1}$, we know there exists some $\epsilon > 0$ such that $\tilde{\boldsymbol{y}}^t := \frac{\boldsymbol{y}^t}{\|\boldsymbol{y}^t\|}$ lies in ϵ-neighborhood of $Cone(\boldsymbol{w}^*) \cap \mathcal{S}^{n-1}$ implying $\widetilde{\text{proj}}_{\mathcal{Q}}(\boldsymbol{y}^t) \in \Lambda(\boldsymbol{w}^*)$.

Finally, Lemma 7 suggests $\tilde{\boldsymbol{y}}^t$ lies in ϵ-neighborhood of $Cone(\boldsymbol{w}^*)$ for all but finitely many t values. We get our desired result. $\square$

Theorem 1 *(Ternary Case). Let $\{\boldsymbol{z}_j\}_{j=1}^k = \Lambda(\boldsymbol{w}^*)$ where $\boldsymbol{z}_1 = \widetilde{\text{proj}}_{\mathcal{Q}}\boldsymbol{w}^*$ is the optimum and $\boldsymbol{w}^* = \sum_{j=1}^k \lambda_j \boldsymbol{z}_j$. If $0 < \sum_{j=2}^k \lambda_j < 1$, we have $\boldsymbol{w}^t = \widetilde{\text{proj}}_{\mathcal{Q}}\boldsymbol{w}^*$ for infinite many t values, where $\boldsymbol{w}^t$ is any infinite sequence generated by Algorithm 1 with any initialization.*

Proof of Theorem 1 (Ternary Case). Note that Lemma 7 suggests $\tilde{\boldsymbol{y}}^t = \frac{\boldsymbol{y}^t}{\|\boldsymbol{y}^t\|}$ lies in ϵ-neighborhood of $Cone(\boldsymbol{w}^*)$ for all but finitely many t values. Let $\Lambda(\boldsymbol{w}^*) = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k\}$ and define μ_j^t be the constants such that

$$\boldsymbol{y}^t = \sum_{j=1}^k \mu_j^t \boldsymbol{z}_j$$

which is determined uniquely by $\boldsymbol{y}^t$.

Let $\boldsymbol{w}^t = \boldsymbol{z}_{j_t}$, we know from Algorithm 1 that

$$\boldsymbol{y}^{t+1} - \boldsymbol{y}^t = \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}}(\boldsymbol{w}^* - \boldsymbol{z}_{j_t}).$$

Thus

$$\sum_{j=2}^k \mu_j^{t+1} = \sum_{j=2}^k \mu_j^t + \eta_t \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left[\left(\sum_{j=2}^k \lambda_j \right) - 1 \right].$$

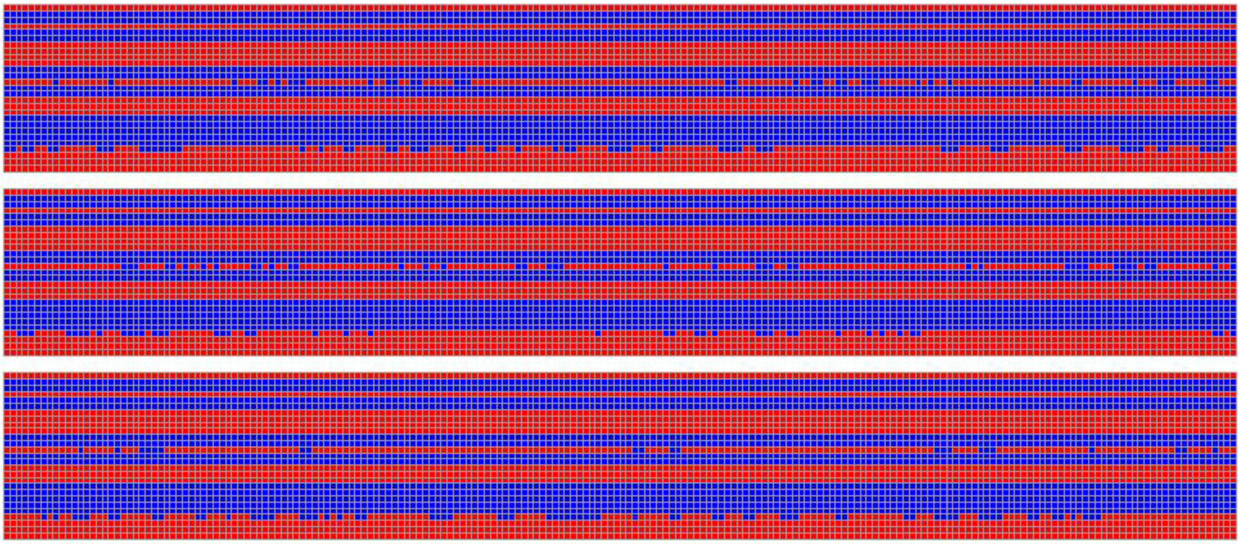

Fig. 7. Evolution of signs of weight filters in the last training epoch (or 600 iterations) of ResNet-20. Each of the three 27 × 200 blocks corresponds to evolution of the 3 × 3 × 3 convolutional filter over 200 iterations. Binary weights over the last 600 iterations of training, red/blue for sign values 1/−1.

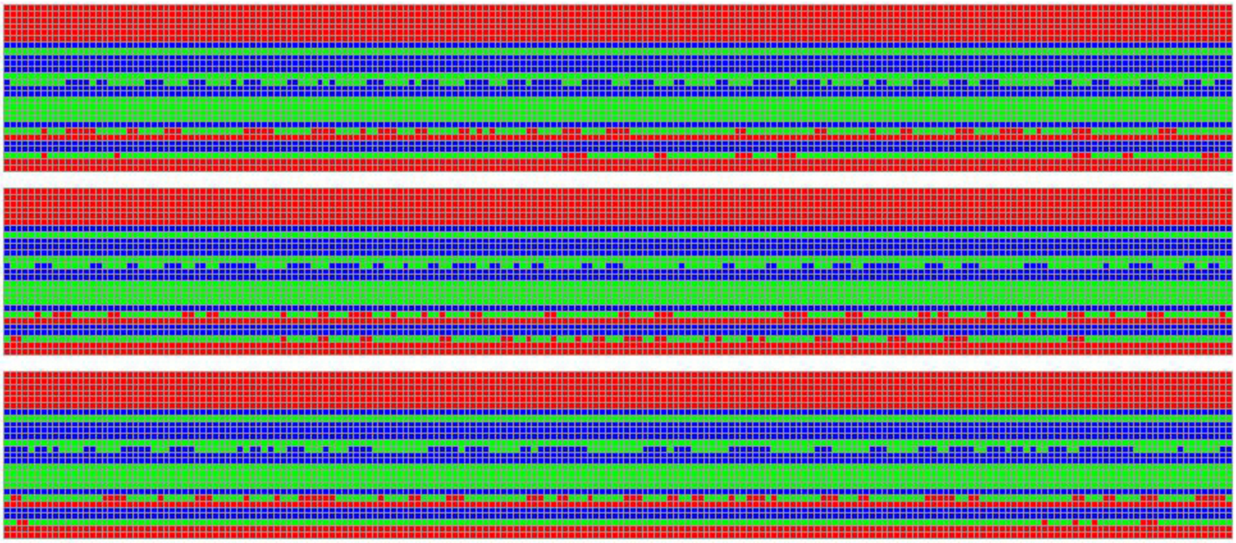

Fig. 8. Evolution of signs of weight filters in the last training epoch (or 600 iterations) of ResNet-20. Each of the three 27 × 200 blocks corresponds to evolution of the 3 × 3 × 3 convolutional filter over 200 iterations. Ternary weights over the last 600 iterations of training, red/green/blue for sign values 1/0/−1.

It follows that

$$\sum_{j=2}^{k} \mu_j^t = \text{Constant} + \left(\sum_{s=0}^{t-1} \eta_s\right) \frac{\|\boldsymbol{v}\|^2}{2\sqrt{2\pi}} \left[\left(\sum_{j=2}^{k} \lambda_j\right) - 1\right] < 0,$$

for large t's. Now we see that when t is large enough, $\tilde{\boldsymbol{y}}^t$ is bounded away from $Cone(\boldsymbol{w}^*)$ which contradicts Lemma 7 and our desired result follows (see Figs. 6–10). □

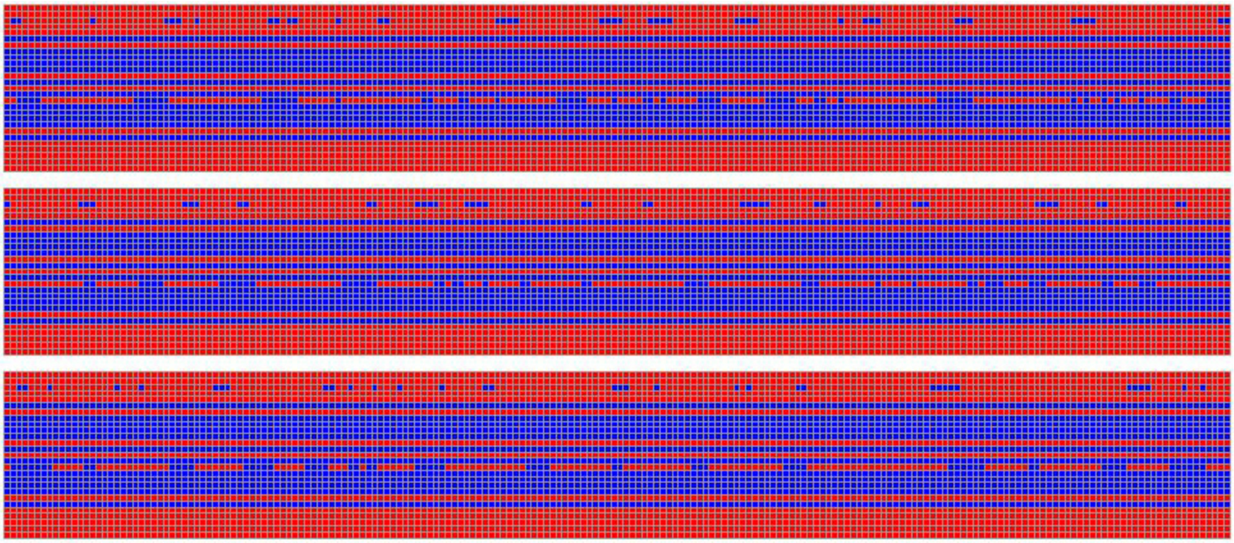

Fig. 9. Evolution of signs of weight filters in the last training epoch (or 600 iterations) of VGG-11. Each of the three 27×200 blocks corresponds to evolution of the $3 \times 3 \times 3$ convolutional filter over 200 iterations. Binary weights over the last 600 iterations of training, red/blue for sign values $1/-1$.

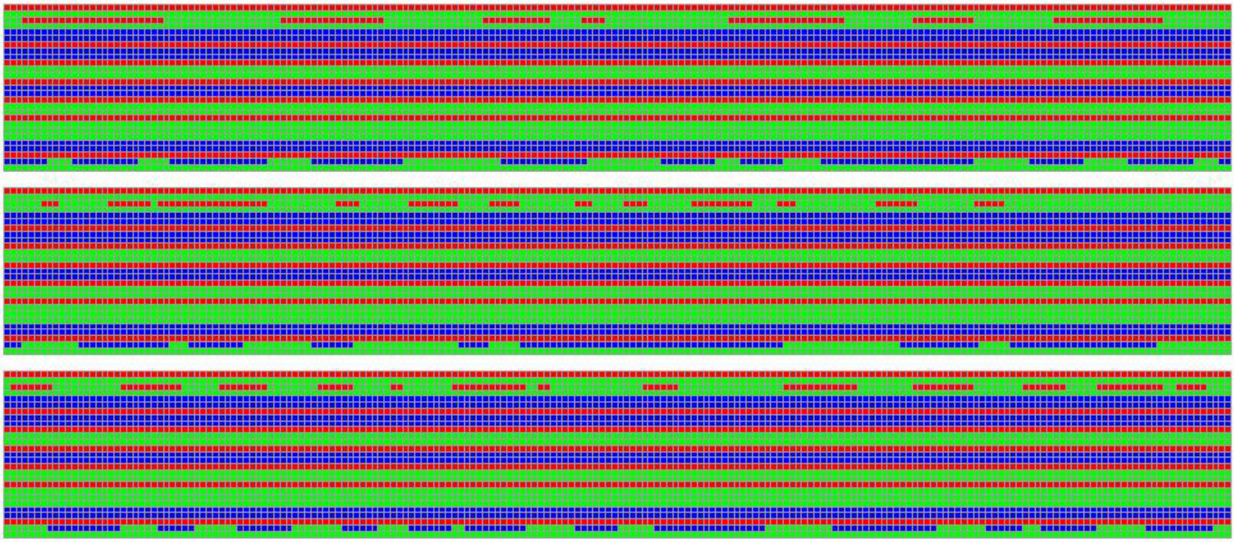

Fig. 10. Evolution of signs of weight filters in the last training epoch (or 600 iterations) of VGG-11. Each of the three 27×200 blocks corresponds to evolution of the $3 \times 3 \times 3$ convolutional filter over 200 iterations. Ternary weights over the last 600 iterations of training, red/green/blue for sign values $1/0/-1$.

References

[1] Yoshua Bengio, Nicholas Léonard, Aaron Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv preprint, arXiv:1308.3432, 2013.

[2] Yaniv Blumenfeld, Dar Gilboa, Daniel Soudry, A mean field theory of quantized deep networks: the quantization-depth trade-off, in: Advances in Neural Information Processing Systems, 2019, pp. 7036–7046.

[3] Zhaowei Cai, Xiaodong He, Jian Sun, Nuno Vasconcelos, Deep learning with low precision by half-wave gaussian quantization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[4] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, Kailash Gopalakrishnan Pact, Parameterized clipping activation for quantized neural networks, arXiv preprint, arXiv:1805.06085, 2018.

[5] Matthieu Courbariaux, Yoshua Bengio, Jean-Pierre David Binaryconnect, Training deep neural networks with binary weights during propagations, in: Advances in Neural Information Processing Systems, 2015, pp. 3123–3131.

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[7] Song Han, Huizi Mao, William J. Dally, Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint, arXiv:1510.00149, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[9] Geoffrey Hinton, Neural networks for machine learning, coursera. Coursera, video lectures, 2012.

[10] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio, Binarized neural networks: training neural networks with weights and activations constrained to +1 or -1, arXiv preprint, arXiv:1602.02830, 2016.

[11] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio, Quantized neural networks: training neural networks with low precision weights and activations, J. Mach. Learn. Res. 18 (2018) 1–30.

[12] Eric Jang, Shixiang Gu, Ben Poole, Categorical reparameterization with gumbel-softmax, in: International Conference on Learning Representations (ICLR), 2017.

[13] Alex Krizhevsky, Learning multiple layers of features from tiny images, Tech Report, 2009.

[14] Fengfu Li, Bo Zhang, Bin Liu, Ternary weight networks, arXiv preprint, arXiv:1605.04711, 2016.

[15] Hao Li, Soham De, Zheng Xu, Christoph Studer, Hanan Samet, Tom Goldstein, Training quantized nets: a deeper understanding, in: Advances in Neural Information Processing Systems, 2017, pp. 5811–5821.

[16] Tao Lin, Sebastian U. Stich, Luis Barba, Daniil Dmitriev, Martin Jaggi, Dynamic model pruning with feedback, in: International Conference on Learning Representations, 2020.

[17] Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, Max Welling, Relaxed quantization for discretized neural networks, in: International Conference on Learning Representations, 2019.

[18] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi Xnor-net, Imagenet classification using binary convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 525–542.

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

[20] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Nissanka Bodhi Priyantha, Jie Liu, Diana Marculescu, Single-path mobile automl: efficient convnet design and nas hyperparameter optimization, IEEE J. Sel. Top. Signal Process. (2020).

[21] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso García, Stephen Tiedemann, Thomas Kemp, Akira Nakamura, Mixed precision dnns: all you need is a good parametrization, in: International Conference on Learning Representations (ICLR), 2020.

[22] Xia Xiao, Zigeng Wang, Sanguthevar Rajasekaran Autoprune, Automatic network pruning by regularizing auxiliary parameters, in: Advances in Neural Information Processing Systems, 2019, pp. 13681–13691.

[23] Canran Xu, Ruijiang Li, Relation embedding with dihedral group in knowledge graph, in: Annual Conference of the Association for Computational Linguistics, 2019.

[24] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley J. Osher, Yingyong Qi, Jack Xin, Understanding straight-through estimator in training activation quantized neural nets, in: International Conference on Learning Representations, 2019.

[25] Penghang Yin, Shuai Zhang, Jack Xin, Yingyong Qi, Training ternary neural networks with exact proximal operator, arXiv:1612.06052 [abs], 2016.

[26] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, Yurong Chen, Incremental network quantization: towards lossless CNNs with low-precision weights, arXiv preprint, arXiv:1702.03044, 2017.

[27] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, Yuheng Zou Dorefa-net, Training low bitwidth convolutional neural networks with low bitwidth gradients, arXiv preprint, arXiv:1606.06160, 2016.

[28] Chenzhuo Zhu, Song Han, Huizi Mao, William J. Dally, Trained ternary quantization, arXiv preprint, arXiv:1612.01064, 2016.