

RESEARCH

Learning quantized neural nets by coarse gradient method for nonlinear classification



Ziang Long^{1*}, Penghang Yin² and Jack Xin¹

*Correspondence:

zlong6@uci.edu

¹Department of Mathematics,
University of California, Irvine,
Irvine, CA 92697, USA

Full list of author information is
available at the end of the article
This work was funded by NSF
Grants IIS-1632935,
DMS-1854434, DMS-1924548,
and DMS-1924935.

Abstract

Quantized or low-bit neural networks are attractive due to their inference efficiency. However, training deep neural networks with quantized activations involves minimizing a discontinuous and piecewise constant loss function. Such a loss function has zero gradient almost everywhere (a.e.), which makes the conventional gradient-based algorithms inapplicable. To this end, we study a novel class of *biased* first-order oracle, termed coarse gradient, for overcoming the vanished gradient issue. A coarse gradient is generated by replacing the a.e. zero derivative of quantized (i.e., staircase) ReLU activation composited in the chain rule with some heuristic proxy derivative called straight-through estimator (STE). Although having been widely used in training quantized networks empirically, fundamental questions like when and why the ad hoc STE trick works, still lack theoretical understanding. In this paper, we propose a class of STEs with certain monotonicity and consider their applications to the training of a two-linear-layer network with quantized activation functions for nonlinear multi-category classification. We establish performance guarantees for the proposed STEs by showing that the corresponding coarse gradient methods converge to the global minimum, which leads to a perfect classification. Lastly, we present experimental results on synthetic data as well as MNIST dataset to verify our theoretical findings and demonstrate the effectiveness of our proposed STEs.

Keywords: Quantized neural networks, Nonlinear classification, Coarse gradient descent, Discrete optimization

Mathematics Subject Classification: 90C26, 68W40

1 Introduction

Deep neural networks (DNNs) have been the main driving force for the recent wave in artificial intelligence (AI). They have achieved remarkable success in a number of domains including computer vision [14, 19], reinforcement learning [18, 23] and natural language processing [4], to name a few. However, due to the huge number of model parameters, the deployment of DNNs can be computationally and memory intensive. As such, it remains a great challenge to deploy DNNs on mobile electronics with low computational budget and limited memory storage.

Recent efforts have been made to the quantization of weights and activations of DNNs while in the hope of maintaining the accuracy. More specifically, quantization techniques constrain the weights or/and activation values to low-precision arithmetic (e.g., 4-bit) instead of using the conventional floating-point (32-bit) representation [2, 12, 17, 31–33]. In this way, the inference of quantized DNNs translates to hardware-friendly low-bit computations rather than floating-point operations. That being said, quantization brings three critical benefits for AI systems: energy efficiency, memory savings and inference acceleration.

The approximation power of weight quantized DNNs was investigated in [6, 8], while the recent paper [22] studies the approximation power of DNNs with discretized activations. On the computational side, training quantized DNNs typically calls for solving a large-scale optimization problem, yet with extra computational and mathematical challenges. Although people often quantize both the weights and activations of DNNs, they can be viewed as two relatively independent subproblems. Weight quantization basically introduces an additional set constraint that characterizes the quantized model parameters, which can be efficiently carried out by projected gradient-type methods [5, 10, 15, 16, 28, 30]. Activation quantization (i.e., quantizing ReLU), on the other hand, involves a staircase activation function with zero derivative almost everywhere (a.e.) in place of the subdifferentiable ReLU. Therefore, the resulting composite loss function is piecewise constant and cannot be minimized via the (stochastic) gradient method due to the vanished gradient.

To overcome this issue, a simple and hardware-friendly approach is to use a straight-through estimator (STE) [1, 9, 26]. More precisely, one can replace the a.e. zero derivative of quantized ReLU with an ad hoc surrogate in the backward pass, while keeping the original quantized function during the forward pass. Mathematically, STE gives rise to a *biased* first-order oracle computed by an unusual chain rule. This first-order oracle is not the gradient of the original loss function because there exists a mismatch between the forward and backward passes. Throughout this paper, this STE-induced type of “gradient” is called coarse gradient. While coarse gradient is not the true gradient, in practice it works as it miraculously points toward a descent direction (see [26] for a thorough study in the regression setting). Moreover, coarse gradient has the same computational complexity as standard gradient. Just like the standard gradient descent, the minimization procedure of training activation quantized networks simply proceeds by repeatedly moving one step at current point in the opposite of coarse gradient with some step size. The performance of the resulting coarse gradient method, e.g., convergence property, naturally relies on the choice of STE. How to choose a proper STE so that the resulting training algorithm is provably convergent is still poorly understood, especially in the nonlinear classification setting.

1.1 Related works

The idea of STE dated back to the classical perceptron algorithm [20, 21] for binary classification. Specifically, the perceptron algorithm attempts to solve the empirical risk minimization problem:

$$\min_{\mathbf{w}} \sum_{i=1}^N (\text{sign}(\mathbf{x}_i^\top \mathbf{w}) - y_i)^2, \quad (1.1)$$

where (\mathbf{x}_i, y_i) is the i^{th} training sample with $y_i \in \{\pm 1\}$ being a binary label; for a given input \mathbf{x}_i , the single-layer perceptron model with weights \mathbf{w} outputs the class prediction $\text{sign}(\mathbf{x}_i^\top \mathbf{w})$. To train perceptrons, Rosenblatt [20] proposed the following iteration for solving (1.1) with the step size $\eta > 0$:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \sum_{i=1}^N (\text{sign}(\mathbf{x}_i^\top \mathbf{w}^t) - y_i) \cdot \mathbf{x}_i \tag{1.2}$$

We note that the above perceptron algorithm is not the same as gradient descent algorithm. Assuming the differentiability, the standard chain rule computes the gradient of the i^{th} sample loss function by

$$(\text{sign}(\mathbf{x}_i^\top \mathbf{w}^t) - y_i) \cdot (\text{sign})'(\mathbf{x}_i^\top \mathbf{w}^t) \cdot \mathbf{x}_i. \tag{1.3}$$

Comparing (1.3) with (1.2), we observe that the perceptron algorithm essentially uses a coarse (and fake) gradient as if $(\text{sign})'$ composited in the chain rule was the derivative of identity function being the constant 1.

The idea of STE was extended to train deep networks with binary activations [9]. Successful experimental results have demonstrated the effectiveness of the empirical STE approach. For example, [1] proposed a STE variant which uses the derivative of sigmoid function instead of identity function. [11] used the derivative of hard tanh function, i.e., $1_{\{|x| \leq 1\}}$, as an STE in training binarized neural networks. To achieve less accuracy degradation, STE was later employed to train DNNs with quantized activations at higher bit-widths [2,3,12,29,32], where some other STEs were proposed including the derivatives of standard ReLU ($\max\{x, 0\}$) and clipped ReLU ($\min\{\max\{x, 0\}, 1\}$).

Regarding the theoretical justification, it has been established that the perceptron algorithm in (1.2) with identity STE converges and perfectly classifies linearly separable data; see, for example, [7,25] and references therein. Apart from that, to our knowledge, there had been almost no theoretical justification of STE until recently: [26] considered a two-linear-layer network with binary activation for regression problems. The training data are assumed to be instead linearly non-separable, being generated by some underlying model with true parameters. In this setting, [26] proved that the working STE is actually non-unique and that the coarse gradient algorithm is descent and converges to a valid critical point if choosing the STE to be the proxy derivative of either ReLU (i.e., $\max\{x, 0\}$) or clipped ReLU function (i.e., $\min\{\max\{x, 0\}, 1\}$). Moreover, they proved that the identity STE fails to give a convergent algorithm for learning two-layer networks, although it works for single-layer perception.

1.2 Main contributions

Figure 1 shows examples of 1-bit (binary) and 2-bit (ternary) activations. We see that a quantized activation function zeros out any negative input, while being increasing on the positive half. Intuitively, a working surrogate of the quantized function used in backward pass should also enjoy this monotonicity, as conjectured by [26] which proved the effectiveness of coarse gradient for two specific STEs: derivatives of ReLU and clipped ReLU, and for binarized activation. In this work, we take a further step toward understanding the convergence of coarse gradient methods for training networks with *general quantized*

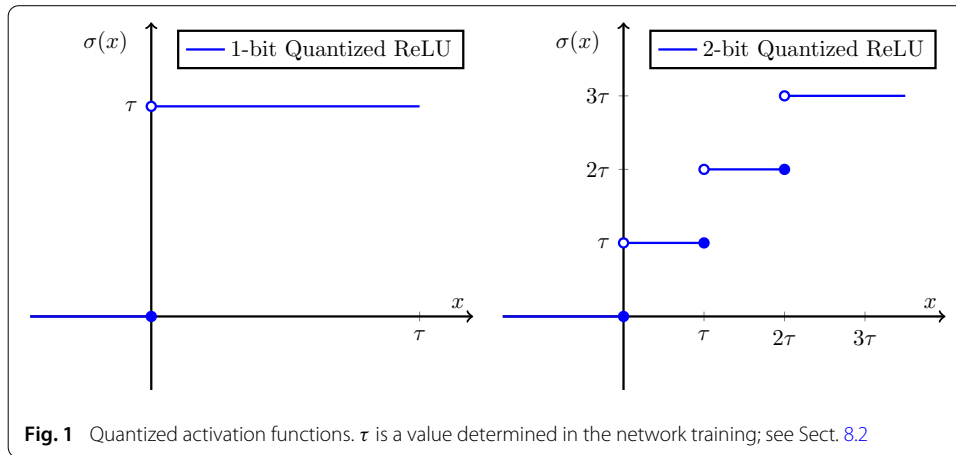


Fig. 1 Quantized activation functions. τ is a value determined in the network training; see Sect. 8.2

Table 1 Frequently used notations

Symbols	Definitions
$[n]$	$\{1, 2, \dots, n\}$
$\mathbb{1}_S(x)$	Indicator function which takes value 1 for $x \in S$ and 0 for $x \notin S$
$\ \mathbf{x}\ $	ℓ_2 -norm of vector \mathbf{x}
$\ \mathbf{W}\ $	The column-wise ℓ_2 -norm sum for a matrix \mathbf{W} . For $\mathbf{W} := [\mathbf{w}_1, \dots, \mathbf{w}_k]$, $\ \mathbf{W}\ = \sum_{j=1}^k \ \mathbf{w}_j\ $
\mathcal{H}^d	d -dimensional Hausdorff measure
$\hat{\mathbf{x}}$	The unit vector in the direction of \mathbf{x} , i.e., $\hat{\mathbf{x}} := \frac{\mathbf{x}}{\ \mathbf{x}\ }$. Additionally, $\hat{\mathbf{0}} := \mathbf{0}$.
σ	Quantized ReLU function
$\Omega_{\mathbf{W}}$	$\{\mathbf{x} \in \mathcal{X} : \langle \mathbf{W}; \mathbf{x}, y \rangle > 0\}$
Ω_v^a	$\{\mathbf{x} \in \mathcal{X} : \langle \mathbf{v}, \mathbf{x} \rangle > a\}$
$\Omega_{\mathbf{W}^j}$	$\Omega_{\mathbf{W}} \cap \Omega_{\mathbf{W}^j}^0$

activations and for classification of linearly non-separable data. A major analytical challenge we face here is that the network loss function is not in closed analytical form, in sharp contrast to [26]. We present more general results to provide meaningful guidance on how to choose STE in activation quantization. Specifically, we study multi-category classification of linearly non-separable data by a two-linear-layer network with multi-bit activations and hinge loss function. We establish the convergence of coarse gradient methods for a broad class of surrogate functions. More precisely, if a function $g : \mathbb{R} \rightarrow \mathbb{R}$ satisfies the following properties:

- $g(x) = 0$ for all $x \leq 0$,
- $g'(x) \geq \delta > 0$ for all $x > 0$ with some constant δ ,

then with proper learning rate, the corresponding coarse gradient method converges and perfectly classifies the nonlinear data when g' serves as the STE during the backward pass. This gives the affirmation of a conjecture in [26] regarding good choices of STE for a classification (rather than regression) task under weaker data assumptions, e.g., allowing non-Gaussian distributions.

1.3 Notations

We have Table 1 for notations used in this paper.

2 Problem setup

2.1 Data assumptions

In this section, we consider the n -ary classification problem in the d -dimensional space $\mathcal{X} = \mathbb{R}^d$. Let $\mathcal{Y} = [n]$ be the set of labels, and for $i \in [n]$ let \mathcal{D}_i be probabilistic distributions over $\mathcal{X} \times \mathcal{Y}$. Throughout this paper, we make the following assumptions on the data:

1. **(Separability)** There are n orthogonal subspaces $\mathcal{V}_i \subseteq \mathcal{X}, i \in [n]$ where $\dim \mathcal{V}_i = d_i$, such that

$$\mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} [\mathbf{x} \in \mathcal{V}_i \text{ and } y = i] = 1, \text{ for all } i \in [n].$$

2. **(Boundedness of data)** There exist positive constants m and M , such that

$$\mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} [m < |\mathbf{x}| < M] = 1, \text{ for all } i \in [n].$$

3. **(Boundedness of p.d.f.)** For $i \in [n]$, let p_i be the marginal probability distribution function of \mathcal{D}_i on \mathcal{V}_i . For any $\mathbf{x} \in \mathcal{V}_i$ with $m < |\mathbf{x}| < M$, it holds that

$$0 < p_i(\mathbf{x}) < p_{\max} < \infty.$$

Later on, we denote \mathcal{D} to be the evenly mixed distribution of \mathcal{D}_i for $i \in [n]$.

Remark 1 The orthogonality of subspaces \mathcal{V}_i 's in the data assumption (1) above is technically needed for our proof here. However, the convergence in Theorem 3.1 to a perfect classification with random initialization is observed in more general settings when \mathcal{V}_i 's form acute angles and contain a certain level of noise. We refer to Sect. 8.1 for supporting experimental results.

Remark 2 Assumption (3) can be relaxed to the following, while the proof remains basically the same.

\mathcal{D}_i is a mixture of n_i distributions, namely $\mathcal{D}_{i,j}$ for $j \in [n_i]$. There exists a linear decomposition of $\mathcal{V}_i = \bigoplus_{j=1}^{n_i} \mathcal{V}_{i,j}$ and $\mathcal{D}_{i,j}$; each has a marginal probability distribution function $p_{i,j}$ on $\mathcal{V}_{i,j}$. For any $\mathbf{x} \in \mathcal{V}_{i,j}$ and $m < |\mathbf{x}| < M$, it holds that

$$0 < p_{i,j}(\mathbf{x}) \leq p_{\max} < \infty.$$

2.2 Network architecture

We consider a two-layer neural architecture with k hidden neurons. Denote by $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k] \in \mathbb{R}^{d \times k}$ the weight matrix in the hidden layer. Let

$$h_j = \langle \mathbf{w}_j, \mathbf{x} \rangle$$

the input to the activation function, or the so-called pre-activation. **Throughout this paper, we make the following assumptions:**

Assumption 1 The weight matrix in the second layer $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ is fixed and known in the training process and satisfies:

1. For any $i \in [n]$, there exists some $j \in [k]$ such that $v_{i,j} > 0$.
2. If $v_{i,j} > 0$, then for any $r \in [n]$ and $r \neq i$, we have $v_{r,j} = 0$.
3. For any $i \in [n]$ and $j \in [k]$, we have $v_{i,j} < 1$.

One can easily show that as long as $k \geq n$, such a matrix $\mathbf{V} = (v_{i,j})$ is ubiquitous.

For any input data $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$, the neural net output is

$$f(\mathbf{W}; \mathbf{x}) = [o_1, \dots, o_n], \tag{2.1}$$

where

$$o_i = \langle \mathbf{v}_i, \sigma(\mathbf{h}) \rangle = \sum_{j=1}^k v_{i,j} \sigma(h_j).$$

The $\sigma(\cdot)$ is the quantized ReLU function acting element-wise; see Fig. 1, for example, of binary and ternary activation functions. More general quantized ReLU function of the bit-width b can be defined as follows:

$$\sigma(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ \text{ceil}(x) & \text{if } 0 < x < 2^b - 1, \\ 2^b - 1 & \text{if } x \geq 2^b - 1. \end{cases}$$

The prediction is given by the network output label

$$\hat{y}(\mathbf{W}, \mathbf{x}) = \underset{r \in [n]}{\operatorname{argmax}} o_r,$$

ideally $\hat{y}(\mathbf{x}) = i$ for all $\mathbf{x} \in \mathcal{V}_i$. The classification accuracy in percentage is the frequency that this event occurs (when network output label \hat{y} matches the true label) on a validation data set.

Given the data sample $\{\mathbf{x}, y\}$, the associated hinge loss function reads

$$l(\mathbf{W}; \{\mathbf{x}, y\}) := \max \{0, 1 - f_y\} := \max \left\{ 0, 1 - \left(o_y - \max_{i \neq y} o_i \right) \right\}. \tag{2.2}$$

To train the network with quantized activation σ , we consider the following population loss minimization problem

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k}} l(\mathbf{W}) := \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}} [l(\mathbf{W}; \{\mathbf{x}, y\})], \tag{2.3}$$

where the sample loss $l(\mathbf{W}; \{\mathbf{x}, y\})$ is defined in (2.2). Let l_i be the population loss function of class i with the label $y = i, i \in [n]$. More precisely,

$$\begin{aligned} l_i(\mathbf{W}) &= \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} [\max \{0, 1 - f_i\}] \\ &= \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left[\max \left\{ 0, 1 - \left(o_i - \max_{r \neq i} o_r \right) \right\} \right]. \end{aligned}$$

Thus, we can rewrite the loss function as

$$l(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^n l_i(\mathbf{W}).$$

Note that the population loss

$$l_i(\mathbf{W}) = \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} [l(\mathbf{W}; \{\mathbf{x}, y\})]$$

fails to have simple closed-form solution even if p_i are constant functions on their supports. We do not have closed-form formula at hand to analyze the learning process, which makes our analysis challenging.

For notational convenience, we define:

$$\begin{aligned} \Omega_{\mathbf{W}} &= \{\mathbf{x} \in \mathcal{X} : l(\mathbf{W}; \{\mathbf{x}, y\}) > 0\}, \\ \Omega_{\mathbf{v}}^a &= \{\mathbf{x} \in \mathcal{X} : \langle \mathbf{v}, \mathbf{x} \rangle > a\}, \end{aligned}$$

and

$$\Omega_{\mathbf{W}}^j = \Omega_{\mathbf{W}} \cap \Omega_{\mathbf{w}_j}^0.$$

2.3 Coarse gradient methods

We see that derivative of quantized ReLU function σ is a.e. zero, which gives a trivial gradient of sample loss function with respect to (w.r.t.) \mathbf{w}_j . Indeed, differentiating the sample loss function with respect to \mathbf{w}_j , we have

$$\nabla_{\mathbf{w}_j} l(\mathbf{W}; \{\mathbf{x}, y\}) = - (v_{y,j} - v_{\xi,j}) \mathbb{1}_{\Omega_{\mathbf{W}}}(\mathbf{x}) \sigma'(h_j) \mathbf{x} = \mathbf{0}, \text{ a.e., } 1 \leq j \leq k$$

where $\xi = \operatorname{argmax}_{i \neq y} o_i$.

The partial coarse gradient w.r.t. \mathbf{w}_j associated with the sample $\{\mathbf{x}, y\}$ is given by replacing σ' with a straight-through estimator (STE) which is the derivative of function g , namely

$$\tilde{\nabla}_{\mathbf{w}_j} l(\mathbf{W}; \{\mathbf{x}, y\}) := - (v_{y,j} - v_{\xi,j}) \mathbb{1}_{\Omega_{\mathbf{W}}}(\mathbf{x}) g'(h_j) \mathbf{x}. \tag{2.4}$$

The sample coarse gradient $\tilde{\nabla} l(\mathbf{W}; \{\mathbf{x}, y\})$ is just the concatenation of $\tilde{\nabla}_{\mathbf{w}_j} l(\mathbf{W}; \{\mathbf{x}, y\})$'s. It is worth noting that coarse gradient is not an actual gradient, but some biased first-order oracle which depends on the choice of g .

Throughout this paper, we consider a class of surrogate functions during the backward pass with the following properties:

Assumption 2 $g : \mathbb{R} \rightarrow \mathbb{R}$ satisfies

1. $g(x) = 0$ for all $x \leq 0$.
2. $g'(x) \in [\delta, \bar{\delta}]$ for all $x > 0$ with some constants $0 < \delta < \bar{\delta} < \infty$.

Such a g is ubiquitous in quantized deep networks training; see Fig. 2, for example, of $g(x)$ satisfying Assumption 2. Typical examples include the classical ReLU $g(x) = \max(x, 0)$ and log-tailed ReLU [2]:

$$g(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ x & \text{if } 0 < x \leq q_b, \\ q_b + \log(x - q_b + 1) & \text{if } x > q_b, \end{cases}$$

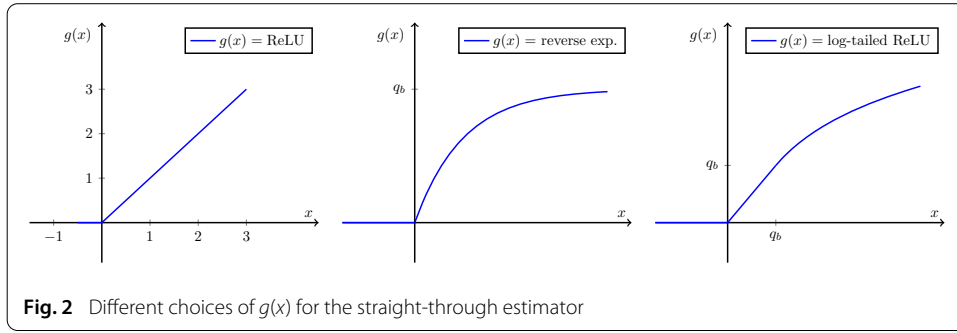
where $q_b := 2^b - 1$ is the maximum quantization level. In addition, if the input of the activation function is bounded by a constant, one also can use $g(x) = \max\{0, q_b(1 - e^{-x/q_b})\}$, which we call *reverse exponential STE*.

To train the network with quantized activation σ , we use the expectation of coarse gradient over training samples:

$$\tilde{\nabla} l(\mathbf{W}) := \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}} \tilde{\nabla} l(\mathbf{W}; \{\mathbf{x}, y\}),$$

where $\tilde{\nabla} l(\mathbf{W}; \{\mathbf{x}, y\})$ is given by (2.4). In this paper, we study the convergence of coarse gradient algorithm for solving the minimization problem (2.3), which takes the following iteration with some learning rate $\eta > 0$:

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \eta \tilde{\nabla} l(\mathbf{W}^t). \tag{2.5}$$



3 Main result and outline of proof

We show that if the iterates $\{\mathbf{W}^t\}$ are uniformly bounded in t , coarse gradient decent with the proxy function g under Assumption 2 converges to a global minimizer of the population loss, resulting in a perfect classification.

Theorem 3.1 *Suppose data assumptions (1)–(3) and STE Assumptions 1–2 hold. If the network initialization satisfies $\mathbf{w}_{j,i}^0 \neq 0$ for all $j \in [k]$ and $i \in [n]$ and \mathbf{W}^t is uniformly bounded by R in t , then for all $v_{i,j} > 0$ we have*

$$\lim_{t \rightarrow \infty} \left| \hat{\mathbf{w}}_{j,i}(\mathbf{W}^t) \right| = 0.$$

Furthermore, if \mathbf{W}^∞ is an accumulation point of $\{\mathbf{W}^t\}$ and all nonzero unit vectors $\tilde{\mathbf{w}}_{j,i}^\infty$'s are distinct for all $j \in [k]$ and $i \in [n]$, then

$$\mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}} (\hat{y}(\mathbf{W}^\infty, \mathbf{x}) \neq y) = 0.$$

We outline the major steps in the proof below.

Step 1: Decompose the population loss into n components. Recall the definition of l_i which is population loss functions for $\{\mathbf{x}, y\} \sim \mathcal{D}_i$. In Sect. 4, we show under certain decomposition of \mathbf{W} , the coarse gradient decent of each one of them only affects a corresponding component of \mathbf{W} .

Step 2: Bound the total increment of weight norm from above. Show that for all $v_{i,j} > 0$, $|\mathbf{w}_{j,i}|$'s are monotonically increasing under coarse gradient descent. Based on boundedness on \mathbf{W} , we further give an upper bound on the total increment of all $|\mathbf{w}_j|$'s, from which the convergence of coarse gradient descent follows.

Step 3: Show that when the coarse gradient vanishes, so does the population loss. In Sect. 6, we show that when the coarse gradient vanishes toward the end of training, the population loss is zero which implies a perfect classification.

4 Space decomposition

With $\mathcal{V} = \bigoplus_{i=1}^n \mathcal{V}_i$, we have the orthogonal complement of \mathcal{V} in $\mathcal{X} = \mathbb{R}^d$, namely \mathcal{V}_{n+1} . Now, we can decompose $\mathcal{X} = \mathbb{R}^d$ into $n + 1$ linearly independent parts:

$$\mathbb{R}^d = \mathcal{V} \bigoplus \mathcal{V}_{n+1} = \bigoplus_{i=1}^{n+1} \mathcal{V}_i$$

and for any vector $\mathbf{w}_j \in \mathbb{R}^d$, we have a unique decomposition of \mathbf{w}_j :

$$\mathbf{w}_j = \sum_{i=1}^{n+1} \mathbf{w}_{j,i}$$

where $\mathbf{w}_{j,i} \in \mathcal{V}_i$ for $i \in [n + 1]$. To simply notation, we let

$$\mathbf{W}_i = [\mathbf{w}_{1,i}, \dots, \mathbf{w}_{k,i}].$$

Lemma 4.1 For any $\mathbf{W} \in \mathbb{R}^{k \times d}$ and $i \in [n]$, we have

$$l_i(\mathbf{W}) = l_i\left(\sum_{r=1}^n \mathbf{W}_r\right) = l_i(\mathbf{W}_i).$$

Proof Note that for any $\mathbf{x} \in \mathcal{V}_i$ and $j \in [k]$, we have $\mathbf{x} \in \mathcal{V}$, so

$$\langle \mathbf{w}_{j,n+1}, \mathbf{x} \rangle = 0$$

and

$$h_j = \langle \mathbf{w}_j, \mathbf{x} \rangle = \left\langle \sum_{i=1}^k \mathbf{w}_{j,i}, \mathbf{x} \right\rangle = \langle \mathbf{w}_{j,i}, \mathbf{x} \rangle.$$

Hence,

$$f(\mathbf{W}; \mathbf{x}) = f\left(\sum_{j=1}^k \mathbf{W}_j; \mathbf{x}\right) = f(\mathbf{W}_i)$$

for all $\mathbf{W} \in \mathbb{R}^{d \times k}$, $\mathbf{x} \in \mathcal{V}_i$. The desired result follows. □

Lemma 4.2 Running the algorithm (2.5) on l_i only does not change the value of \mathbf{W}_r for all $r \neq i$. More precisely, for any $\mathbf{W} \in \mathbb{R}^{d \times k}$, let

$$\mathbf{W}' = \mathbf{W} - \eta \tilde{\nabla} l_i(\mathbf{W}),$$

then for any $r \in [n]$ and $r \neq i$

$$\mathbf{W}'_r = \mathbf{W}_r.$$

Proof of Lemma 4.2 Assume $i, r \in [n]$ and $i \neq r$. Note that

$$\mathbf{w}'_j = \mathbf{w}_j - \eta \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W})$$

and

$$\tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) = - \mathbb{E}_{\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_i} [(v_{y,j} - v_{\xi,j}) \mathbb{1}_{\Omega_{\mathbf{W}}}(\mathbf{x}) g'(h_j)\mathbf{x}] \in \mathcal{V}_i.$$

Since \mathcal{V}_i 's are linearly independent, we have

$$\mathbf{w}'_{j,i} = \mathbf{w}_{j,i} - \eta \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W})$$

and

$$\mathbf{w}'_{j,r} = \mathbf{w}_{j,r}.$$

□

By the above result, we know (2.5) is equivalent to

$$\mathbf{W}_i^{t+1} = \mathbf{W}_i^t - \frac{\eta}{n} \tilde{\nabla} l_i(\mathbf{W}^t). \tag{4.1}$$

5 Learning dynamics

In this section, we show that some components of the weight iterates have strictly increasing magnitude whenever coarse gradient does not vanish, and it quantifies the increment during each iteration.

Lemma 5.1 *Assume*

$$\hat{v}_j = \max_{i_1, i_2 \in [n]} v_{i_1, j} - v_{i_2, j},$$

we have the following estimate:

$$\mathbb{P}_{\{x, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right) \geq \frac{1}{\hat{v}_j \delta M} \left| \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) \right|.$$

Proof of Lemma 5.1

$$\begin{aligned} \left| \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) \right| &= \left| \mathbb{E}_{\{x, y\} \sim \mathcal{D}_i} \left[(v_{y, j} - v_{\xi, j}) \mathbb{1}_{\Omega_{\mathbf{W}}}(\mathbf{x}) g'(h_j) \mathbf{x} \right] \right| \\ &\leq \hat{v}_j \delta M \mathbb{E}_{\{x, y\} \sim \mathcal{D}_i} \left[\mathbb{1}_{\Omega_{\mathbf{W}}^j}(\mathbf{x}) \right] \\ &= \hat{v}_j \delta M \mathbb{P}_{\{x, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right) \end{aligned}$$

□

Lemma 5.2 *For any $j \in [k]$, if*

$$\tilde{v}_{i, j} := v_{i, j} - \max_{r \neq i} v_{r, j} > 0$$

we have

$$\left\langle \tilde{\mathbf{w}}_{j, i} - \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) \right\rangle \geq \frac{\tilde{v}_{i, j} \delta}{2 C_p} \mathbb{P}_{\{x, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right)^2,$$

where

$$C_p = \max_{v \in V, a \in \mathbb{R}} \int_{\langle v, \mathbf{x} \rangle = a} p_i(\mathbf{x}) d \mathcal{H}^{d_i-1}(\mathbf{x}).$$

Proof of Lemma 5.2 First, we prove an inequality which will be used later. Recall that $\|\mathbf{x}\| \leq M$, and that $\tilde{\nabla}_{\mathbf{w}_j} l(\mathbf{W}, \{x, y\}) \neq 0$ only when $\mathbf{x} \in \Omega_{\mathbf{W}}^j$. Hence, we have $\langle \tilde{\mathbf{w}}_{j, i}, \mathbf{x} \rangle > 0$. We have

$$\begin{aligned} \mathbb{P}_{\{x, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \cap \{ \mathbf{x} : \langle \tilde{\mathbf{w}}_{j, i}, \mathbf{x} \rangle < t \} \right) &= \int_{\Omega_{\mathbf{W}}^j} \mathbb{1}_{\{ \langle \tilde{\mathbf{w}}_{j, i}, \mathbf{x} \rangle < t \}}(\mathbf{x}) p_i(\mathbf{x}) d \mathbf{x} \\ &= \int_0^t \int_{\langle \tilde{\mathbf{w}}_{j, i}, \mathbf{x} \rangle = s} p_i(\mathbf{x}) d \mathcal{H}^{d_i-1}(\mathbf{x}) d s \\ &\leq t C_p. \end{aligned}$$

Now, we use Fubini's theorem to simplify the inner product:

$$\begin{aligned} \left\langle \tilde{\mathbf{w}}_{j, i} - \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) \right\rangle &= \mathbb{E}_{\{x, y\} \sim \mathcal{D}_i} \left[(v_{y, j} - v_{\xi, j}) \mathbb{1}_{\Omega_{\mathbf{W}}^j}(\mathbf{x}) g'(h_j) \langle \tilde{\mathbf{w}}_{j, i}, \mathbf{x} \rangle \right] \\ &\geq \tilde{v}_{i, j} \delta \int_{\Omega_{\mathbf{W}}^j \cap V_i} \langle \tilde{\mathbf{w}}_{j, i}, \mathbf{x} \rangle p_i(\mathbf{x}) d \mathbf{x} \end{aligned}$$

$$\begin{aligned}
 &= \tilde{v}_{i,j} \delta \int_{\Omega_{\mathbf{W}}^j \cap V_i} \int_0^\infty \mathbb{1}_{\{\langle \tilde{\mathbf{w}}_{j,i}, \mathbf{x} \rangle > t\}} d t p_i(\mathbf{x}) d \mathbf{x} \\
 &= \tilde{v}_{i,j} \delta \int_0^\infty \int_{\Omega_{\mathbf{W}}^j \cap V_i} \mathbb{1}_{\{\langle \tilde{\mathbf{w}}_{j,i}, \mathbf{x} \rangle > t\}} p_i(\mathbf{x}) d \mathbf{x} d t \\
 &= \tilde{v}_{i,j} \delta \int_0^\infty \mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \cap \{\mathbf{x} : \langle \tilde{\mathbf{w}}_{j,i}, \mathbf{x} \rangle > t\} \right) d t.
 \end{aligned}$$

Now, using the inequality just proved above, we have

$$\begin{aligned}
 &\mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \cap \{\mathbf{x} : \langle \tilde{\mathbf{w}}_{j,i}, \mathbf{x} \rangle > t\} \right) \\
 &= \mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right) - \mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \cap \{\mathbf{x} : \langle \tilde{\mathbf{w}}_{j,i}, \mathbf{x} \rangle < t\} \right) \\
 &\geq \max \left\{ \mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right) - t C_p, 0 \right\}.
 \end{aligned}$$

Combining the above two inequalities, we have

$$\begin{aligned}
 \langle \tilde{\mathbf{w}}_{j,i} - \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) \rangle &\geq \tilde{v}_{i,j} \delta \int_0^\infty \max \left\{ \mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right) - t C_p, 0 \right\} d t \\
 &\geq \frac{\tilde{v}_{i,j} \delta}{2C_p} \mathbb{P}_{\{\mathbf{x}, y\} \sim \mathcal{D}_i} \left(\Omega_{\mathbf{W}}^j \right)^2.
 \end{aligned}$$

□

Lemma 5.3 *If $\tilde{v}_{i,j} > 0$ in Lemma 5.2, then $\{\|\mathbf{w}_{j,i}^t\|\}$ in Eq. (2.1) is non-decreasing with coarse gradient decent (2.5). Moreover, under the same assumption, we have*

$$\left| \mathbf{w}_{j,i}^{t+1} \right| - \left| \mathbf{w}_{j,i}^t \right| \geq \frac{\eta \tilde{v}_{i,j} \delta}{2n C_p \hat{v}_j^2 \delta^2 M^2} \left| \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^t) \right|^2,$$

where C_p is defined as in Lemma 5.2 and \hat{v}_j as in Lemma 5.1.

Proof of Lemma 5.3 Since $\mathbf{w}_{j,i}^{t+1} = \mathbf{w}_{j,i}^t - \frac{\eta}{n} \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^t)$, we have

$$\left| \mathbf{w}_{j,i}^{t+1} \right| - \left| \mathbf{w}_{j,i}^t \right| \geq \left\langle \mathbf{w}_{j,i}^{t+1} - \mathbf{w}_{j,i}^t, \tilde{\mathbf{w}}_{j,i}^t \right\rangle = \left\langle -\frac{\eta}{n} \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^t), \tilde{\mathbf{w}}_{j,i}^t \right\rangle.$$

Hence, it follows from Lemmas 5.1 and 5.2 that

$$\left| \mathbf{w}_{j,i}^{t+1} \right| - \left| \mathbf{w}_{j,i}^t \right| \geq \frac{\eta \tilde{v}_{i,j} \delta}{2n C_p \hat{v}_j^2 \delta^2 M^2} \left| \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^t) \right|^2, \tag{5.1}$$

which is the desired result. □

Note that one component of \mathbf{w}_j is increasing but the weights are bounded by assumption, hence, summation of the increments over all steps should also be bounded. This gives the following proposition:

Proposition 1 *Assume $\{\|\mathbf{w}_j^t\|\}$ is bounded by R , then if $\tilde{v}_{i,j} > 0$ in Lemma 5.2, then*

$$\sum_{t=1}^\infty \left| \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^t) \right|^2 \leq \frac{2n C_p \hat{v}_j^2 \delta^2 M^2 R}{\eta \tilde{v}_{i,j} \delta} < \infty,$$

where C_p is as defined in Lemma 5.2 and \hat{v}_j defined in Lemma 5.1. This implies that

$$\lim_{t \rightarrow \infty} \left| \tilde{\nabla}_{w_j} l_i(\mathbf{W}^t) \right| = 0$$

as long as $\hat{v}_{i,j} > 0$.

Remark 3 Lemmas 5.1, 5.2, 5.3 and Proposition 1 were proved without Assumption 1. Under Assumption 1, we have $\hat{v}_j = \max_{i \in [m]} v_{i,j}$ in Lemma 5.1 and $\tilde{v}_{i,j} = \hat{v}_j$ if $v_{i,j} > 0$ and $\tilde{v}_{i,j} = -\hat{v}_j$ if $v_{i,j} = 0$ in Lemma 5.2.

6 Landscape properties

We have shown that under boundedness assumptions, the algorithm will converge to some point where the coarse gradient vanishes. However, this does not immediately indicate the convergence to a valid point because coarse gradient is a fake gradient. We will need the following lemma to prove Proposition 2, which confirms that the points with zero coarse gradient are indeed global minima.

Lemma 6.1 *Let $\Omega = \{\mathbf{x} \in \mathbb{R}^l : m < |\mathbf{x}| < M\}$, where $0 < m < M < \infty$. For $j \in [k]$, let $\Omega_j = \{\mathbf{x} : \langle \mathbf{w}_j, \mathbf{x} \rangle > a\}$, where $a \geq 0$ and $\Omega_i \neq \Omega_j$ for all $i \neq j$. If for $i \in [k]$ and $\mathbf{x} \in \Omega_i \cap \Omega$, there exists some $j \neq i$ such that $\mathbf{x} \in \Omega_j$, then*

$$\left(\bigcup_{j=1}^k \Omega_j \right) \cap \Omega = \emptyset \text{ or } \Omega.$$

Proof (Proof of Lemma 6.1) Define $\tilde{\Omega} = \bigcup_{j=1}^k \Omega_j$, by De Morgan’s law, we have

$$\tilde{\Omega}^c = \left(\bigcup_{j=1}^k \Omega_j \right)^c = \bigcap_{j=1}^k \Omega_j^c.$$

Note that k is finite and $\mathbf{0} \in \Omega_j^c$ for all $j \in [k]$, we know $\tilde{\Omega}^c$ is a generalized polyhedron and hence either

$$(\partial \tilde{\Omega}) \cap \Omega = \emptyset$$

or

$$\mathcal{H}^{l-1}((\partial \tilde{\Omega}) \cap \Omega) > 0.$$

The first case is trivial. We show that the second case contradicts our assumption. Note that

$$\partial \tilde{\Omega} = \partial \left(\bigcup_{j=1}^k \Omega_j \right) \subseteq \bigcup_{j=1}^k \partial \Omega_j,$$

we know there exists some $j^* \in [k]$ such that $\mathcal{H}^{l-1}(\partial \Omega_{j^*} \cap \Omega) > 0$. It follows from our assumption that $\tilde{\Omega} = \bigcup_{j=1}^k \Omega_j = \bigcup_{j \neq j^*} \Omega_j$, and hence,

$$\mathcal{H}^{l-1}(\partial \Omega_{j^*} \cap \partial \Omega_j) > 0.$$

Note that $\partial \Omega_j$ ’s are hyperplanes. Therefore, $\Omega_j = \Omega_{j^*}$, contradicting with our assumption that all Ω_j ’s are distinct. □

The following result shows that the coarse gradient vanishes only at a global minimizer with zero loss, except for some degenerate cases.

Proposition 2 Under Assumption 1, if $\tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) = \mathbf{0}$ for all $\tilde{v}_{i,j} > 0$ and $\tilde{\mathbf{w}}_{j,i}$'s are distinct, then $l_i(\mathbf{W}) = 0$.

Proof of Proposition 2 For quantized ReLU function, let $q_b := \max_{x \in \mathbb{R}} \sigma(x)$ be the maximum quantization level, so that

$$\sigma(x) = \sum_{a=0}^{q_b-1} \mathbb{1}_{\{x>a\}}(x).$$

Note that

$$f_i(\mathbf{W}; \mathbf{x}) = o_i - o_\xi = \sum_{j=1}^k (v_{i,j} - v_{\xi,j}) \sigma(h_j) = \sum_{j=1}^k (v_{i,j} - v_{\xi,j}) \sum_{a=0}^{q_b} \mathbb{1}_{\Omega_{\mathbf{w}_j}^a}(\mathbf{x}).$$

By assumption, $\tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) = \mathbf{0}$ for all $\tilde{v}_{i,j} > 0$ which implies $\mathbb{1}_{\Omega_{\mathbf{w}_j}(\mathbf{x})} \mathbb{1}_{\Omega_{\mathbf{w}_j}^a(\mathbf{x})} = 0$ for all $\tilde{v}_{i,j} > 0$ and $a \in [n]$ almost surely. Now, for any $\mathbf{x} \in \Omega_{\mathbf{w}_j}^a$, we have $\mathbf{x} \notin \Omega_{\mathbf{w}_j}$. Note that $\mathbf{x} \in \Omega_{\mathbf{w}_j}$ if and only if $o_i - o_\xi \geq 1$, then for any $\mathbf{x} \in \Omega_{\mathbf{w}_j}^a$, since $v_{i,j} - v_{\xi,j} < 1$, there exist $j' \neq j$ and $a' \in [n]$ such that $v_{i,j'} > 0$ and $\mathbf{x} \in \Omega_{\mathbf{w}_{j'}}^{a'}$. By Lemma 6.1, $\mathbb{P}_{\{\mathbf{x},y\} \sim \mathcal{D}_i} [\Omega_{\mathbf{w}_j}] = 0$ is empty, and thus, $l_i(\mathbf{W}) = 0$. \square

The following lemma shows that the expected coarse gradient is continuous except at $\mathbf{w}_{j,i} = \mathbf{0}$ for some $j \in [k]$.

Lemma 6.2 Consider the network in (2.1). $\tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W})$ is continuous on

$$\left\{ \mathbf{W} \in \mathbb{R}^{k \times d} : |\mathbf{w}_{j,i}| > 0 \text{ for all } j \in [k], i \in [n] \right\}.$$

Proof of Lemma 6.2 It suffices to prove the result for $j \in [k]$. Note that

$$\tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}) = \mathbb{E}_{\{\mathbf{x},y\} \sim \mathcal{D}_i} \left[- (v_{y,j} - v_{\xi,j}) \mathbb{1}_{\Omega_{\mathbf{w}_j}(\mathbf{x})} g'(h_j) \mathbf{x} \right]$$

For any \mathbf{W}^0 satisfying our assumption, we know

$$\lim_{\mathbf{W} \rightarrow \mathbf{W}^0} \mathbb{1}_{\Omega_{\mathbf{w}_j}(\mathbf{x})} g'(h_j) = \mathbb{1}_{\Omega_{\mathbf{w}_j^0}(\mathbf{x})} g'(h_j^0), \text{ a.e.}$$

The desired result follows from the dominant convergence theorem. \square

7 Proof of main results

Equipped with the technical lemmas, we present:

Proof of Theorem 3.1 It is easily noticed from Assumption 1 that $v_{i,j} > 0$ if and only if $\tilde{v}_{i,j} > 0$. By Lemma 5.3, if $v_{i,j} > 0$ and $|\mathbf{w}_{j,i}^0| > 0$, then $|\mathbf{w}_{j,i}^t| > 0$ for all t . Since \mathbf{W} is randomly initialized, we can ignore the possibility that $\mathbf{w}_{j,i}^0 = \mathbf{0}$ for some $j \in [k]$ and $i \in [n]$. Moreover, Proposition 1 and Eq. (2.5) imply for all $v_{i,j} > 0$

$$\lim_{t \rightarrow \infty} \left| \tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^t) \right| = 0.$$

Suppose \mathbf{W}^∞ is an accumulation point and $\mathbf{w}_{j,r}^\infty \neq \mathbf{0}$ for all $j \in [k]$ and $r \in [n]$, we know for all $v_{i,j} > 0$

$$\tilde{\nabla}_{\mathbf{w}_j} l_i(\mathbf{W}^\infty) = \mathbf{0}.$$

Next, we consider the case when $w_{j,r} = \mathbf{0}$ for some $j \in [k]$ and $r \in [n]$. Lemma 5.2 implies $v_{r,j} = 0$. We construct a new sequence

$$\hat{w}_{j,r}^t = \begin{cases} w_{j,r}^t & \text{if } w_{j,r}^\infty \neq 0, \\ \mathbf{0} & \text{if } w_{j,r}^\infty = 0, \end{cases}$$

and

$$\hat{W}_r^t = [\hat{w}_{1,r}^t, \dots, \hat{w}_{k,r}^t].$$

With

$$\hat{o}_r = \sum_{j=1}^k v_{r,j} \sigma(\hat{h}_j) = \sum_{j=1}^k v_{r,j} \sigma(\langle \hat{w}_{j,r}, \mathbf{x} \rangle),$$

we know $\hat{o}_r = o_r$ for all $r \in [n]$. Hence, we have

$$l(\hat{W}^t, \{\mathbf{x}, i\}) = \text{ReLU}(1 - \hat{o}_i + \hat{o}_\xi) = l(W^t, \{\mathbf{x}, i\}).$$

This implies that $\Omega_{\hat{W}^t} = \Omega_{W^t}$, so we have for all $j \in [k]$,

$$\left| \langle \tilde{\nabla}_{w_j} l_i(\hat{W}_1^t), \hat{w}_{j,i}^t \rangle \right| \leq \left| \langle \tilde{\nabla}_{w_j} l_i(W_i^t), \hat{w}_{j,i}^t \rangle \right| \leq \left| \tilde{\nabla}_{w_j} l_i(W_i^t) \right|.$$

Letting t go to infinity on both side, we get

$$\left| \langle \tilde{\nabla}_{w_j} l_i(\hat{W}^\infty), \hat{w}_{j,i}^\infty \rangle \right| = 0.$$

By Lemmas 5.1 and 5.2, we know

$$\tilde{\nabla}_{w_j} l_i(W^\infty) = \tilde{\nabla}_{w_j} l_i(W_i^\infty) = 0,$$

so $\tilde{\nabla}_W l_i(W^\infty) = 0$. By Proposition 2, $l_i(W^t) = 0$, which completes the proof. □

8 Experiments

In this section, we conduct experiments on both synthetic and MNIST data to verify and complement our theoretical findings. Experiments on larger networks and data sets will left for a future work.

8.1 Synthetic data

Let $\{e_1, e_2, e_3, e_4\}$ be orthonormal basis of \mathbb{R}^4 , θ be an acute angle and $v_1 = e_1, v_2 = \sin \theta e_2 + \cos \theta e_3, v_3 = e_3, v_4 = e_4$. Now, we have two linearly independent subspaces of \mathbb{R}^4 , namely $\mathcal{V}_1 = \text{Span}(\{v_1, v_2\})$ and $\mathcal{V}_2 = \text{Span}(\{v_3, v_4\})$. We can easily calculate that the angle between \mathcal{V}_1 and \mathcal{V}_2 is θ . Next, with

$$S_r = \left\{ \frac{j}{10} : j \in [20] - [9] \right\}, S_\varphi = \left\{ \frac{j\pi}{40} : j \in [80] \right\},$$

we define

$$\hat{\mathcal{X}}_1 = \{r(\cos \varphi v_1 + \sin \varphi v_2) : r \in S_r, \varphi \in S_\varphi\}$$

and

$$\hat{\mathcal{X}}_2 = \{r(\cos \varphi v_3 + \sin \varphi v_4) : r \in S_r, \varphi \in S_\varphi\}.$$

Let $\hat{\mathcal{D}}_i$ be uniform distributed on $\hat{\mathcal{X}}_i \times \{i\}$ and $\hat{\mathcal{D}}$ be a mixture of $\hat{\mathcal{D}}_1$ and $\hat{\mathcal{D}}_2$. Let $\hat{\mathcal{X}} = \hat{\mathcal{X}}_1 \cup \hat{\mathcal{X}}_2$.

The activation function σ is 4-bit quantized ReLU:

$$\sigma(x) = \begin{cases} 0 & \text{if } x < 0, \\ \text{ceil}(x) & \text{if } 0 \leq x < 15, \\ 15 & \text{if } x \geq 15. \end{cases}$$

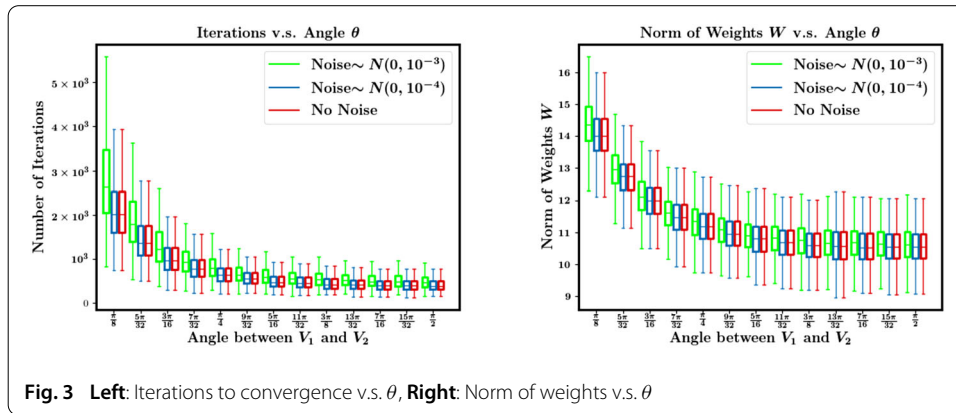


Fig. 3 Left: Iterations to convergence v.s. θ , Right: Norm of weights v.s. θ

For simplicity, we take $k = 24$ and $v_{ij} = \frac{1}{2}$ if $j - 12(i - 1) \in [12]$ for $i \in [2]$ and $j \in [24]$ and 0 otherwise. Now, our neural network becomes

$$f_i = \frac{(-1)^{i-1}}{2} \left[\sum_{j=1}^{12} \sigma(h_j) - \sum_{j=1}^{12} \sigma(h_{j+12}) \right]$$

where $h_j = \langle \mathbf{w}_j, \mathbf{x} \rangle$ and $\mathbf{x} \in \mathbb{R}^4$. The population loss is given by

$$l(\mathbf{W}) = \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}} [l(\mathbf{W}; \{\mathbf{x}, y\})] = \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}} [\max\{1 - f_i\}].$$

We choose the ReLU STE (i.e., $g(x) = \max\{0, x\}$) and use the coarse gradient

$$\begin{aligned} \tilde{\nabla}_{\mathbf{W}} l(\mathbf{W}) &= \mathbb{E}_{\{\mathbf{x}, y\} \sim \mathcal{D}} [\tilde{\nabla}_{\mathbf{W}} l(\mathbf{W}, \{\mathbf{x}, y\})] \\ &= \frac{1}{|\mathcal{X}|} \left[\sum_{\mathbf{x} \in \mathcal{X}_1} \tilde{\nabla}_{\mathbf{W}} l(\mathbf{W}; \{\mathbf{x}, 1\}) + \sum_{\mathbf{x} \in \mathcal{X}_2} \tilde{\nabla}_{\mathbf{W}} l(\mathbf{W}; \{\mathbf{x}, 2\}) \right]. \end{aligned}$$

Taking learning rate $\eta = 1$, Eq. 2.5 becomes

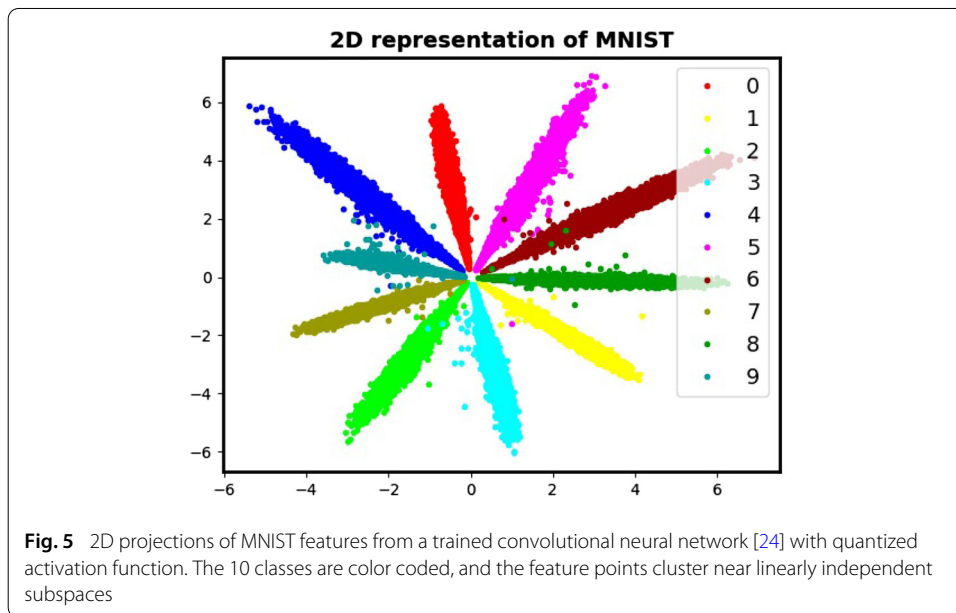
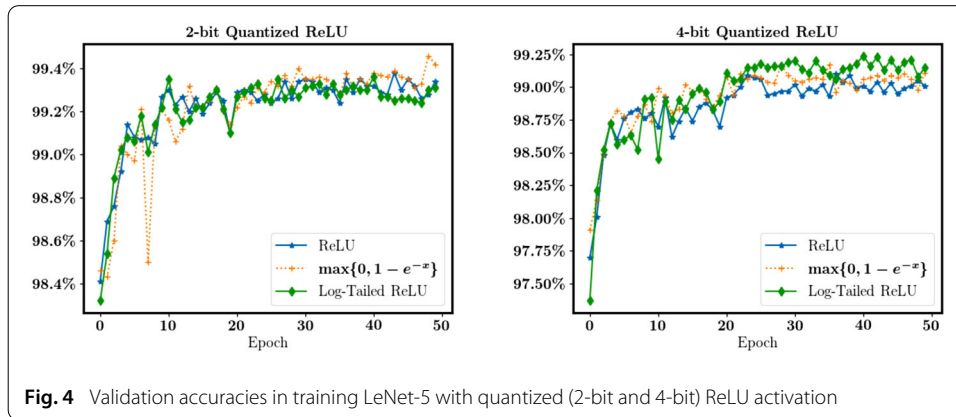
$$\mathbf{W}^{t+1} = \mathbf{W}^t - \tilde{\nabla}_{\mathbf{W}} l(\mathbf{W}^t).$$

We find that the coarse gradient method converges to a global minimum with zero loss. As shown in box plots of Fig. 3, the convergence still holds when the subspaces \mathcal{V}_1 and \mathcal{V}_2 form an acute angle, and even when the data come from two levels of Gaussian noise perturbations of \mathcal{V}_1 and \mathcal{V}_2 . The convergence is faster and with a smaller weight norm when θ increases toward $\frac{\pi}{2}$ or \mathcal{V}_2 are orthogonal to each other. This observation clearly supports the robustness of Theorem 1 beyond the regime of orthogonal classes.

8.2 MNIST experiments

Our theory works for a board range of STEs, while their empirical performances on deeper networks may differ. In this subsection, we compare the performances of the three types of STEs in Fig. 2.

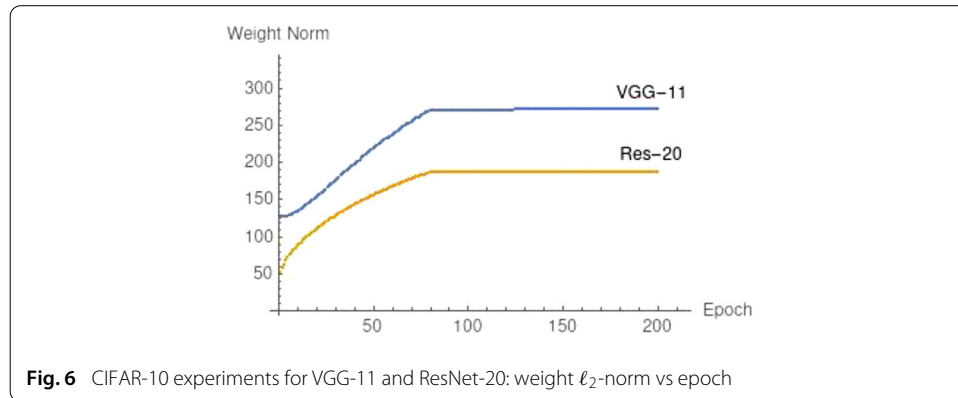
As in [2], we resort to a modified batch normalization layer [13] and add it before each activation layer. As such, the inputs to quantized activation layers always follow unit Gaussian distribution. Then, the scaling factor τ applied to the output of quantized activation layers can be pre-computed via k -means approach and get fixed during the whole training process. The optimizer we use to train quantized LeNet-5 is the (stochastic) coarse gradient method with momentum = 0.9. The batch size is 64, and learning rate is initialized



to be 0.1 and then decays by a factor of 10 after every 20 epochs. The three backward pass substitutions g for the straight-through estimator are (1) ReLU $g(x) = \max\{x, 0\}$, (2) reverse exponential $g(x) = \max\{0, q_b(1 - e^{-x/q_b})\}$ and (3) log-tailed ReLU. The validation accuracy for each epoch is shown in Fig. 4. The validation accuracies at bit-widths 2 and 4 are listed in Table 2. Our results show that these STEs all perform very well and give satisfactory accuracy. Specifically, reverse exponential and log-tailed STEs are comparable, both of which are slightly better than ReLU STE. In Fig. 5, we show 2D projections of MNIST features at the end of 100 epoch training of a 7 layer convolutional neural network [24] with quantized activation. The features are extracted from input to the last fully connected layer. The data points cluster near linearly independent subspaces. Together with Sect. 8.1, we have numerical evidence that the linearly independent subspace data structure (working as an extension of subspace orthogonality) occurs for high-level features in a deep network for a nearly perfect classification, rendering support to the realism of our theoretical study. Enlarging angles between linear subspaces can improve classification accuracy, see [27] for such an effort on MNIST and CIFAR-10 data sets via linear feature transform.

Table 2 Validation accuracy (%) on MNIST with LeNet-5

$g(x)$	Bit-width (b)	Valid. accuracy
	32	99.45
ReLU	2	99.10
	4	99.38
Reverse exp.	2	99.17
	4	99.46
Log-tailed ReLU	2	99.24
	4	99.36



8.3 CIFAR-10 experiments

In this experiment, we train VGG-11/ResNet-20 with 4-bit activation function on CIFAR-10 data set to numerically validate the boundedness assumption upon the ℓ_2 -norm of weight. The optimizer is momentum SGD with no weight decay. We used initial learning rate = 0.1, with a decay factor of 0.1 at the 80-th and 140-th epoch.

We see from Fig. 6 that the ℓ_2 norm of weights is bounded during the training process. This figure also shows that the norm of weights is generally increasing in epochs which coincides with our theoretical finding shown in Lemma 5.3.

9 Summary

We studied a novel and important biased first-order oracle, called coarse gradient, in training quantized neural networks. The effectiveness of coarse gradient relies on the choice of STE used in backward pass only. We proved the convergence of coarse gradient methods for a class of STEs bearing certain monotonicity in nonlinear classification using one-hidden-layer networks. In experiments on LeNet and MNIST data set, we considered three different proxy functions satisfying the monotonicity condition for backward pass: ReLU, reverse exponential function and log-tailed ReLU for training LeNet-5 with quantized activations. All of them exhibited good performance which verified our theoretical findings. In future work, we plan to expand theoretical understanding of coarse gradient descent for deep activation quantized networks.

Acknowledgements

This work was partially supported by NSF Grants IIS-1632935, DMS-1854434, DMS-1924548, DMS-1924935 and DMS-1952644. On behalf of all authors, the corresponding author states that there is no conflict of interest.

Author details

¹Department of Mathematics, University of California, Irvine, Irvine, CA 92697, USA, ²Department of Mathematics and Statistics, University at Albany, State University of New York, Albany, NY 12222, USA.

Received: 9 February 2021 Accepted: 6 June 2021

Published online: 31 July 2021

References

- Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint [arXiv:1308.3432](https://arxiv.org/abs/1308.3432) (2013)
- Cai, Z., He, X., Sun, J., Vasconcelos, N.: Deep learning with low precision by half-wave gaussian quantization. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
- Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.-J., Srinivasan, V., Gopalakrishnan, K.: Pact: parameterized clipping activation for quantized neural networks. arXiv preprint [arXiv:1805.06085](https://arxiv.org/abs/1805.06085) (2018)
- Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: International Conference on Machine Learning. ACM, pp. 160–167 (2008)
- Courbariaux, M., Bengio, Y., David, J.-P.: Binaryconnect: training deep neural networks with binary weights during propagations. In: Advances in Neural Information Processing Systems, pp. 3123–3131 (2015)
- Ding, Y., Liu, J., Xiong, J., Shi, Y.: On the universal approximability and complexity bounds of quantized relu neural networks. arXiv preprint [arXiv:1802.03646](https://arxiv.org/abs/1802.03646) (2018)
- Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. *Mach. Learn.* **37**, 277–296 (1999)
- He, J., Li, L., Xu, J., Zheng, C.: ReLU deep neural networks and linear finite elements. *J. Comput. Math.* **38**, 502–527 (2020)
- Hinton, G.: Neural Networks for Machine Learning, Coursera. Coursera, Video Lectures (2012)
- Hou, L., Kwok, J.T.: Loss-aware weight quantization of deep networks. In: International Conference on Learning Representations (2018)
- Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: Advances in Neural Information Processing Systems (2016)
- Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: training neural networks with low precision weights and activations. *J. Mach. Learn. Res.* **18**, 1–30 (2018)
- Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (2015)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
- Li, F., Zhang, B., Liu, B.: Ternary weight networks. arXiv preprint [arXiv:1605.04711](https://arxiv.org/abs/1605.04711) (2016)
- Li, H., De, S., Xu, Z., Studer, C., Samet, H., Goldstein, T.: Training quantized nets: a deeper understanding. In: Advances in Neural Information Processing Systems, pp. 5811–5821 (2017)
- Louizos, C., Reisser, M., Blankevoort, T., Gavves, E., Welling, M.: Relaxed quantization for discretized neural networks. In: International Conference on Learning Representations (2019)
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. *Nature* **518**, 529 (2015)
- Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
- Rosenblatt, F.: The Perceptron, a Perceiving and Recognizing Automaton Project Para. Cornell Aeronautical Laboratory, Buffalo (1957)
- Rosenblatt, F.: Principles of neurodynamics. In: Spartan Book (1962)
- Shen, Z., Yang, H., Zhang, S.: Deep network approximation with discrepancy being reciprocal of width to power of depth. arXiv preprint [arXiv:2006.12231](https://arxiv.org/abs/2006.12231) (2020)
- Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneer-shelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484 (2016)
- Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., Liu, W.: Cosface: large margin cosine loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
- Widrow, B., Lehr, M.A.: 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proc. IEEE* **78**, 1415–1442 (1990)
- Yin, P., Lyu, J., Zhang, S., Osher, S.J., Qi, Y., Xin, J.: Understanding straight-through estimator in training activation quantized neural nets. In: International Conference on Learning Representations (2019)
- Yin, P., Xin, J., Qi, Y.: Linear feature transform and enhancement of classification on deep neural network. *J. Sci. Comput.* **76**, 1396–1406 (2018)
- Yin, P., Zhang, S., Lyu, J., Osher, S., Qi, Y., Xin, J.: Binaryrelax: a relaxation approach for training deep neural networks with quantized weights. *SIAM J. Imag. Sci.* **11**, 2205–2223 (2018)
- Yin, P., Zhang, S., Lyu, J., Osher, S., Qi, Y., Xin, J.: Blended coarse gradient descent for full quantization of deep neural networks. *Res. Math. Sci.* **6**, 14 (2019)

30. Yin, P., Zhang, S., Qi, Y., Xin, J.: Quantization and training of low bit-width convolutional neural networks for object detection. *J. Comput. Math.* **37**, 349–359 (2019)
31. Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y.: Incremental network quantization: towards lossless CNNs with low-precision weights. In: International Conference on Learning Representations (2017)
32. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint [arXiv:1606.06160](https://arxiv.org/abs/1606.06160) (2016)
33. Zhu, C., Han, S., Mao, H., Dally, W.J.: Trained ternary quantization. In: International Conference on Learning Representations (2017)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.