On the Robustness of Quantization Algorithms during the Training Phase of Deep Neural Networks

Hok Fong Wong
Department of Computer Science and Engineering, The Chinese University of Hong Kong

Hoi-To Wai
Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong

Abstract

The question of enforcing quantization on the weights and activations in Deep Neural Networks (DNNs) has come to the forefront in recent years due to its relevance to restricted memory and/or computational resources. While low-precision fixed integer values could substantially reduce the memory footprint and latency, inaccurate quantization on model parameters is susceptible to significant accuracy drops. In this paper, we will revisit several key methods for quantization algorithms during neural network training. Our research also extends the branch where we have provided stronger guarantees for the Annealed Skewed Stochastic Gradient Descent algorithm (`ASkewSGD`) proposed by Leconte et al. (2023) for quantization-aware training in deep neural networks. We claim that under weaker conditions without the need for a highly differentiable loss function, the algorithm will eventually converge to the set of stationary points. We also consider the optimization problem formulation in Leconte et al's (2023) settings to be a smooth minimax optimization problem. Hence, in this paper, we will also present an application of the stochastic gradient descent ascent algorithm `SGDA` to tackle the problem. Experiments show that `ASkewSGD` and `SGDA` are able to obtain near state-of-the-art results in classical benchmarks.

Undergraduate Summer Research Internship 2024, The Chinese University of Hong Kong, Hong Kong. The LaTeX styling file is provided by AISTATS 2023.

1 INTRODUCTION

1.1 Towards the Time- and Energy-Efficient, Optimal Neural Network Design

Over the past decade, there has been an increasing awareness that deep learning models have succeeded in achieving vast improvements for tasks such as computer vision, speech recognition and generation, and natural language processing. While the accuracy of these models has significantly improved, the size of these models could be a potential problem when being deployed in resource-limited scenarios. As such, there is a growing interest in the search for efficient design, training, and deployment of neural network models, while allowing optimality trade-offs. These methods include knowledge distillation, micro-architecture design, hardware-architecture co-design, pruning of insensitive neurons, low-rank decomposition, and network quantization (Gholami et al., 2021). In this paper, we mainly concentrate on network quantization during model training, with weights mapping values from a large set of real numbers to values in a set of discrete values with lower bit widths, for efficient neural network inference.

1.2 Enforcing Quantization on Neural Networks: Challenges and Opportunities

In the recent decade, we have also seen an explosive increase in the parameters of modern neural network architectures. For example, ResNet-101 (He et al., 2016) requires 200 MB of memory (Guo, 2018). the world-renowned large language model GPT-3 proposed by OpenAI, has approximately 175 billion parameters (Brown et al., 2020). Transferring from floating point numbers to low-bit integer representations, such as 32-bit floats to 8-bit integers, could reduce the size of the neural network by a factor of 4, which is often realized in practice. Therefore, utilizing quantization techniques could potentially conquer the problem

of restricted storage capacities that prevent the wide usage of deep neural network models.

Other than shrinking the model size by using lower bit width representations, quantization has been shown to improve power efficiency and reduce network latency, as the structuredness of the quantized weight matrix enables a faster matrix-vector product (Hubara et al., 2017; Han et al., 2016). Floating point operations require complex logic for handling, as compared to integer operations. Through quantization, high-cost floating-point operations are then replaced with simple and low-cost integer-integer operations that require a smaller number of cycles on hardware. Hence, the power efficiency is improved due to the decreased computation and decreased memory bandwidth required. This generally brings a positive effect to the shorter inference time, leading to better performance.

Despite the advantages, quantization is not without drawbacks such as accuracy drop, since low-width data types cannot retain sufficient information provided by floating point numbers that are much more precise. Nevertheless, depending on the amount of precision loss, the neural network architecture, and the network training/quantization scheme, quantization can often result in minimal loss of accuracy.

What enables the success on these quantized neural networks is that the neural network models are often over-parameterized and can thus afford to lose precision without huge impact in accuracy (Gholami et al, 2021; Denil et al., 2013). Firstly, the high degree of freedom in neural network models enables vastly varying neural network parameters that all result in low-error models. Secondly, imposing quantization on weights and activations can be considered as noise injections, and hence performs as a regularizer for better network generalization (Srivastava et al., 2014). Despite the empirical success and interpretation of quantization in the real world, the theoretical understanding is still very limited. It is still a challenge to close the accuracy gap between full-precision and quantized neural networks. We aim to address the problem by introducing several state-of-the-art quantization algorithms during the training phase of deep neural networks.

1.3 Contributions

- We provide new theoretical convergence guarantees for `ASkewSGD` proposed by Leconte et al. (2023) under weaker assumptions. The approach that they have adopted relies strongly on the local approximation of the feasible set and hence does not require computationally-expensive projection or quadratic programming solvers.
- We adapt the novel mini-max optimization framework, stochastic gradient descent ascent algorithm `SGDA`, with convergence results supported by Lin et al. (2024), as a new algorithm that solves the constrained optimization problem in nonconvex-concave settings. The assumptions imposed on the algorithm is weaker or on par with `ASkewSGD`, given that the set of quantization levels is small.
- We evaluate the performance of `ASkewSGD`, `SGDA`, and other state-of-the-art algorithms by several numerical experiments.

2 RELATED WORKS

A lot of techniques have been proposed for the quantization of neural networks during the training phase. In a broad sense, quantization techniques are categorized into two major types: deterministic quantization and stochastic quantization (Guo, 2018). From this point of view, we have selected representative algorithms that are able to cover both areas, focusing on fixed codebook quantization with the quantization level set fixed in advance.

2.1 Fine-tuning Methods in Quantization-Aware Training Procedures

The process of retraining the model or training a model from scratch is often referred to as Quantization-Aware Training (QAT). The injection of noise to weights in the neural network can perturb the converged weights and thus realizing the impact at an early stage during training could be a remedy for the model to reach back to convergence again with tolerable loss. To be clear, we emphasize that the methods included in this section all use full-precision weights during the training phase and result in a quantized neural network thereafter for inference.

2.2 BinaryConnect and Straight-Through Estimators

The very first introduction to binarized neural network training was given by Courbariaux et al. (2015), which gave rise to the simplest approach to quantizing real values in neural networks - rounding. With specialized hardware, this method can even remove 2/3 of the multiplications during training time.

The rounding scheme is described below:

$$w_b = \begin{cases} +1, & w \geq 0, \\ -1, & \text{otherwise}, \end{cases} \tag{1}$$

where w is the real-valued parameter, and w_b is the binarized value of w.

In BinaryConnect (Courbariaux et al., 2015), we only binarize the weights during forward and backward propagation. On the forward path, we first obtain the binarized weights on the layers and use them to generate the outputs. However, the gradients vanish almost everywhere on the rounding function. This could be a problem to overcome during the backward pass. The general strategy for this problem is based on adopting a straight-through estimator (STE) function to gauge the gradient with respect to the full-precision weights (Hinton et al., 2012; Bengio et al., 2013), starting from the top layer and down to the first hidden layer. Assume that ℓ is the loss function, then

$$\frac{\partial \ell}{\partial w} = \frac{\partial \ell}{\partial w_b}. \tag{2}$$

Yin et al. (2019) further argued that the STE function should be chosen properly so that the expected coarse gradient correlates positively with the population gradient. Otherwise, a poor estimator can lead to instability of the training algorithm near certain local minima. A good choice for the STE would be the derivatives of clipped ReLUs (Hubara et al., 2017):

$$\frac{\partial \ell}{\partial w} = \frac{\partial \ell}{\partial w_b} \mathbb{1}(w), \text{ where } \mathbb{1}(w) = \begin{cases} 1, & |w| \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

What makes BinaryConnect even more interesting is that the deterministic rounding scheme can be replaced with random sampling from a discrete distribution.

$$w_b = \begin{cases} +1, & \text{with probability } p = \sigma(w), \\ -1, & \text{with probability } 1 - p, \end{cases} \tag{4}$$

where σ is the "hard sigmoid" function

$$\sigma(x) = \max\left(0, \min\left(\frac{x+1}{2}\right), 1\right).$$

The stochastic random rounding scheme is also simple in concept and provides us with a flexible quantization scheme.

Finally, Courbariaux et al. (2015) have decided not to binarize the gradients during the parameter update, as they claim that good precision is essential for SGD to work. The small, noisy update directions produced by stochastic gradients are averaged out, and therefore sufficient resolution for the gradients is needed. We also note that the noisy weights could act as a regularizer that assists in the generalization of neural network models. For the general quantized neural network, STE can easily be adapted by adding a quantization step that maps a real number input $w \in [0, 1]$ to a k-bit number output w_q (Zhou et al, 2016):

$$w_q = \text{round}((2^k - 1)w) - (2^{k-1} - 1). \tag{5}$$

2.3 ProxQuant

ProxQuant (Bai et al., 2018) formulates the problem of training binary neural networks as a relaxed constrained optimization problem. Instead of directly projecting the weights into the feasible region, Bai et al. propose a regularization operator to enforce the quantization of weights, thus removing the need for a non-differentiable function.

The regularization operation involves a W-shaped, non-smooth function to enforce the weights to binarized or quantized values, vanishing when stumbling on the quantization levels and reflecting the distance to the quantization levels otherwise. The regularizer function proposed is highly flexible depending on the set of quantization levels $\mathcal{Q}$ and is described as follows:

$$R(w) = \inf_{w_0 \in \mathcal{Q}} \|w_0 - w\|_1 \text{ or } R(w) = \inf_{w_0 \in \mathcal{Q}} \|w_0 - w\|_2.$$

On each parameter update, a soft "projection" is applied to the parameter by the use of a proximal operator concerning R and the adjustable strength $\lambda > 0$:

$$w_{k+1} = \text{prox}_{\eta_k \lambda_k R}\left(w_k - \eta_k \widehat{\nabla} \ell(w_k)\right), \tag{6}$$

where the proximal operator is defined as

$$\text{prox}_{\lambda R}(w_k) = \arg\min_{\bar{w} \in \mathbb{R}^n} \left(\frac{1}{2}\|w_k - \bar{w}\|_2^2 + \lambda R(\bar{w})\right).$$

To understand the notion, we consider the case where $\lambda = +\infty$, then argmin must satisfy $R(\bar{w}) = 0$, which implies that $\bar{w} \in \mathcal{Q}$ and this essentially means that the prox operator projects w onto $\mathcal{Q}$. When $\lambda < +\infty$, the prox operator corresponds to a lazy gradient descent and therefore relaxes the restriction on the enforcement of quantization.

3 PRELIMINARIES

3.1 Formalization of the Optimization Problem and Inherent Difficulty

The task of learning a quantized neural network is often described as minimizing an objective function ℓ (often referred to as training loss), and appropriate numerical or analytical optimization methods are usually applied.

In supervised learning, the goal is to find an optimal mapping function $f(x)$ to minimize the loss function of the training samples,

$$\min_{w} \frac{1}{N} \sum_{i=1}^{N} \ell(y^{(i)}, f(x^{(i)}, w)), \tag{7}$$

where N is the number of training samples and $(x^{(i)}, y^{(i)})$ describes the i-th sample with a feature vector and a corresponding label (Bottou et al., 2018). It is a common practice to incorporate a regularization term (e.g. the lasso L_1-norm or the ridge L_2-norm of the parameter w) into the loss function to avoid overfitting.

We can also formulate an optimization problem for unsupervised learning problems such as clustering (Sun et al., 2019). For the k-means clustering problem, we need to divide a group of samples into k clusters to ensure that the difference between the samples in the same cluster is as small as possible. This induces the following optimization problem

$$\min_{S} \sum_{i=1}^{k} \sum_{x \in S_k} \|x - \mu_k\|_2^2, \tag{8}$$

where x is the feature vector of the samples, μ_k is the center of cluster k, and S is a set of clusters, each containing a set of sample features.

The task of learning a quantized neural network (QNN) adds a layer of complexity with the superposition of quantization constraints on the weights, i.e.,

$$\min_{w \in \mathcal{Q}} \ell(w), \text{where } \ell(w) = \mathbb{E}_{(x,y)\sim p_{\text{data}}}[\ell(f(x, w), y)], \tag{P1}$$

where $\mathcal{Q} \subset \mathbb{R}^n$ is the set of quantization levels, n is the number of parameters (including network weights and biases), ℓ is the training loss function as mentioned above (e.g. logistic loss and cross-entropy), $f(x, w)$ is the prediction function modeled by the deep neural network, and p_{data} is the data distribution.

Such optimization problems involving the integer-valued variables are difficult, as the optimization problem is non-convex, non-smooth, and combinatorial. Generally, this type of mixed integer non-linear programming problem (MINLP) is known to be **NP**-hard since it includes finding the solution for mixed-integer linear programming instances (MILP) (Liberti, 2019). Different geometric, analytic, and algebraic techniques have been proposed to transform the discrete problem into a continuous problem (Leconte et al., 2023). However, with the huge scale and order of the number of parameters inside a neural network, these methods can all be impractical to terminate in a reasonable time.

3.2 Local Approximation for Constraints in First-Order Optimization

In this section, we will consider a general optimization problem addressed by Muehlebach and Jordan (2022) that can be adapted to the quantized neural network optimization setting.

In particular, we shed the light on the following optimization problem:

$$\min_{w \in C} \ell(w), \text{where } C := \{w \in \mathbb{R}^n \mid g(w) \geq 0\} \tag{P2}$$

with the objective function $f : \mathbb{R}^n \mapsto \mathbb{R}$, and constraints described by the function $g : \mathbb{R}^n \mapsto \mathbb{R}^d$.

We would also need the elements from convex optimization theory and non-linear optimization theory. The definitions are stated below.

Definition 1. (Lan, 2020) A set $X \subseteq \mathbb{R}^n$ is said to be convex if it contains all of its segments, that is

$$\lambda x + (1 - \lambda)y \in X, \forall (x, y, \lambda) \in X \times X \times [0, 1]. \tag{9}$$

Let X be a convex set and $f : X \mapsto \mathbb{R}$ be a function. f is said to be convex if $\forall (x, y, \lambda) \in X \times X \times [0, 1]$, we have

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)y. \tag{10}$$

f is said to be μ-strongly convex if there exists $\mu > 0$ such that $\forall (x, y, \lambda) \in X \times X \times [0, 1]$,

$$f((1-\lambda)x+\lambda y) \leq (1-\lambda)f(x)+\lambda f(y)-\frac{\mu\lambda(1-\lambda)}{2}\|y-x\|_2^2. \tag{11}$$

Definition 2. (Bottou et al., 2018) A function $f : \mathbb{R}^n \mapsto \mathbb{R}^d$ is called L-smooth if it is continuously differentiable and its gradient is Lipschitz continuous with Lipschitz constant L:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \forall x, y \in \mathbb{R}^n. \tag{12}$$

Additionally, we will need the assumptions below to describe the algorithm in full detail.

Assumption 1. The functions ℓ is L_ℓ-smooth. $\ell(x) \to \infty$ when $|x| \to \infty$ and C is non-empty and bounded.

Assumption 2. The function g is L_g-smooth.

Assumption 3. (standing) The Mangasarian-Fromovitz constraint qualification (MFCQ) is satisfied for all $x \in \mathbb{R}^n$, i.e. $\forall x \in \mathbb{R}^d$, $\exists w \in \mathbb{R}^n$ s.t. $\nabla g_i(x)w > 0$ for all $i \in I_x$, where $I(x) = \{i \in \mathbb{Z} \mid g_i(x) \leq 0\}$.

Assumption 4. C is convex and ℓ is μ-strongly convex.

Definition 3. If the Mangasarian-Fromovitz constraint qualification holds for all $x \in \mathbb{R}^d$, the tangent and normal cones of C at $w \in C$ are given by:

$$T_C(w) = \left\{v \in \mathbb{R}^d : \nabla g_i(w)^\top v \geq 0, \forall i \in I(w)\right\}, \quad (13)$$

$$N_C(w) = \left\{ -\sum_{i \in I(w)} \lambda_i \nabla g_i(w), \lambda_i \in \mathbb{R}_+ \right\}. \quad (14)$$

Moreover, if w^* is a local minimizer of (P2), then there exists Lagrange multipliers $\lambda^* \succeq 0$ such that

$$\nabla \ell(x) - \lambda^{*\top} g(x) = 0. \quad (15)$$

Alternatively, we can say that $w^* \in \mathcal{Z}$, where

$$\mathcal{Z} := \{w \in C : 0 \in -\nabla \ell(w) - N_C(w)\}. \quad (16)$$

The detailed proofs of the tangent cone, the normal cone of the feasible set C, and the theorem of first-order necessary optimality conditions under MFCQ will be provided in Appendix A.

The main idea of Muehlebach and Jordan's (2022) work is to establish equivalences between position and velocity constraints, through a local approximation of the original nonlinear and nonconvex feasible set. While traversing along the descent direction of the loss function, we are essentially choosing a flow velocity matching the gradient. Nonetheless, with the presence of constraint limitations, we are forced to skew the velocity whenever constraints are violated. The Muehlebach-Jordan algorithm (MJ algorithm) generates iterates in $\mathbb{R}^d$ as follows:

$$\begin{cases} w_{k+1} = w_k + \gamma_k v_k \\ v_k = \arg\min_{v \in V_\alpha(w_k)} (1/2)\|v + \nabla \ell(w_k)\|_2^2, \end{cases} \quad (17)$$

where (γ_k) is a non-increasing sequence of positive step sizes, $\alpha > 0$ is a adjustable hyperparameter, and the set $V_\alpha(w)$ is defined as

$$V_\alpha(w) = \{v \in \mathbb{R}^d : \nabla g_i(w)^\top v \geq -\alpha g_i(w)\} \quad (18)$$

for a local approximation of the constraints.

We note that α controls the trade-off between two objectives: for large α, the emphasis is on the convergence to feasible set, while for small α, the focus is on reducing the objective function.

The set $V_\alpha(w)$ is guaranteed to be non-empty due to MFCQ. Additionally, when $x \in C$, the set $V_\alpha(x)$ is nothing but $T_C(x)$. Thus, the algorithm always picks the velocity that matches with the unconstrained gradient flow as closely as possible.

Theorem 1. (Muehlebach and Jordan, 2022) The iterates $(w_k)_{k \geq 0}$ are guaranteed to converge to the minimizer of ℓ at nearly a linear rate, under Assumptions 1-4.

3.3 Smoothing, Annealing, and the `ASkewSGD` Algorithm

So far, we have presented a first-order optimization algorithm by Muehlebach and Jordan (2022) for minimizing a function with inequality constraints. Still, there are a few number of gaps to be addressed in order to solve the QNN problem.

Firstly, we need to define the constraint function g. Leconte et al.'s (2023) approach to this issue is to relax the quantization levels to a set of "smoothed" interval constraints. Similar to ProxQuant, a continuous regularization function is defined for quantifying "how far away" a parameter is from its quantization levels.

Definition 4. Let $\varepsilon \in [0,1]$, $w^i \in \mathcal{Q}^i, i \in \{1,2,\ldots,n\}$, i.e. each w^i coordinate-wisely takes value from its own set of quantization levels $\mathcal{Q}^i = \{q_1^i, q_2^i, \ldots, q_{K^i}^i\}$. We define the piecewise function

$$\psi_\varepsilon^i(w^i) := \begin{cases} \varepsilon - (q_1^i - w^i)^2, & w^i < q_1^i, \\ \varepsilon - (w^i - q_{j-1}^i)^2 (w^i - q_j^i)^2, & \\ \qquad q_{j-1}^i \leq w^i < q_j^i, j = 2, \ldots, K, & \\ \varepsilon - (w^i - q_K^i)^2, & w^i \geq q_{K^i}^i, \end{cases} \quad (19)$$

for all $w^i \in \mathbb{R}$ and $i \in [n]$.

We observe that ε unanimously controls the tolerance limit of offsets from the quantization levels of each coordinate. Let g be defined as $(\psi_\varepsilon^1, \psi_\varepsilon^2, \ldots, \psi_\varepsilon^n)$, and C_ε^i be the projection of C_ε on the i-th coordinate. We notice that $w \in C_\varepsilon$ only if $\psi_\varepsilon^i(w^i) \geq 0$ for all $i \in [n]$.

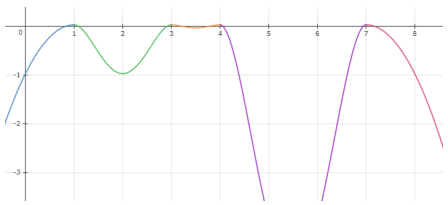

Figure 1: An illustration for a one-dimensional case where ψ is defined with $\varepsilon = 0.03$, $\mathcal{Q} = \{1,3,4,7\}$. Around each quantization level $q \in \mathcal{Q}$, there is always an admissible region for a legal choice of w controlled by ε with $\psi_\varepsilon(w) \geq 0$.

Even with constraint relaxations, the feasible region C_ε can still be complicated (see Figure 1). The feasible region, which can be disconnected and non-convex,

may violate the conditions stated for convergence of the MJ algorithm.

We have defined a new optimization problem $\min_{w\in C_\varepsilon} \ell(w)$, with the parameter $\varepsilon \in [0,1]$ given. We also have to include ε for the expression of $V_\alpha(w)$ and $\mathcal{Z}$, which is given by the following:

$$V_{\varepsilon,\alpha}(w) = \{v \in \mathbb{R}^d : v^i \psi_\varepsilon^{'i}(w^i) \geq - \alpha\psi_\varepsilon^i(w^i) \text{ for } i \in I_\varepsilon(w)\}, \tag{20}$$

where

$$I_\varepsilon(w) = \{i \in \{1,2,\ldots,d\} : \psi_\varepsilon^i(w^i) \leq 0\},$$

and the set of KKT points $\mathcal{Z}_\varepsilon$

$$\mathcal{Z}_\varepsilon = \{w \in C_\varepsilon : 0 \in -\nabla\ell(w) - N_{C_\varepsilon}(w)\}, \tag{21}$$

where

$$N_{C_\varepsilon} = (-\lambda^1\psi_\varepsilon^{'1}(w^1), \ldots, -\lambda^n\psi_\varepsilon^{'n}(w^n)),$$

with $\lambda \succeq 0$ and $\lambda^i \neq 0$ only if $\psi_\varepsilon^i(w^i) = 0$.

We also notice that $V_{\varepsilon,\alpha}(x)$ can be empty in this case when there exists $i \in [n]$ such that $w^i = (q^i_{Q^i(w^i)} + q^i_{Q^i(w^i)+1})/2$, where MFCQ does not necessarily hold in this case. Here, $Q^i(w)$ is defined to be the unique index satisfying $q^i_{Q^i(w^i)} \leq w^i < q^i_{Q^i(w^i)+1}$. Fortunately, the set of such w is Lebesgue-measure zero and thus we can anticipate that we will never stumble upon such a point.

Since the constraints on each coordinate of w are coordinate-wise independent, we can further simplify the expression for (17), when w does not stumble on the point where $\psi^{'i}(w^i) = 0$ for some $i \in [n]$. Denote $[s_{\varepsilon,\alpha}(\widehat{\nabla}\ell(w_k), w_k)]^i$ as the solution. If $\psi_\varepsilon(w^i) > 0$ or $-\psi'_\varepsilon(w^i)\widehat{\nabla}\ell(w_k^i) \geq -\alpha\psi_\varepsilon(w^i) > 0$, $[s_{\varepsilon,\alpha}(\widehat{\nabla}\ell(w_k), w_k)]^i = -\widehat{\nabla}\ell(w_k^i)$. Otherwise, $[s_{\varepsilon,\alpha}(\widehat{\nabla}\ell(w_k), w_k)]^i = -\alpha\psi_\varepsilon(w^i)/\psi'_\varepsilon(w^i)$. For full generality, we can apply a clipping operation on $s_{\varepsilon,\alpha}$, such that $|[s_{\varepsilon,\alpha}]^i| \leq M_{\varepsilon,c}$.

Secondly, we have not yet discussed about the choice of ε. This is where the annealing process takes place. It is easily seen that $\cap_{\varepsilon>0} C_\varepsilon = \mathcal{Q}$. We can therefore solve the set of problems $\mathcal{P}_{\varepsilon_n}$ progressively with a sequence of $(\varepsilon_n)_{n\geq 0}$ and $\varepsilon_n \in [0,1]$ such that $\lim_{n\to\infty} \varepsilon_n = 0$.

4 CONVERGENCE GUARANTEES OF THE ASkewSGD ALGORITHM

In the previous section, we have presented the full picture of the ASkewSGD algorithm. The only problem that remains is whether the algorithm will converge and whether the algorithm will return a stationary point. Proving such would be crucial for the workable success of the quantization algorithm.

Algorithm 1 The ASkewSGD Algorithm

1: **for** $k = 1, 2, \ldots, T$ **do**
2: Sample a mini-batch of N_b observations $\{j_1, j_2, \ldots, j_{N_b}\}$ in $\{1, 2, \ldots, N\}$
3: Compute the stochastic gradient $\widehat{\nabla}\ell(w_k) = 1/N_b \sum_{i=1}^{N_b} \nabla\ell_{j_i}(w_k)$
4: Compute the update direction $v_k = s_{\varepsilon,\alpha}(\widehat{\nabla}\ell(w_k), w_k)$
5: $w_{k+1} \leftarrow w_k + \gamma_k v_k$
6: **end for**

In our work, we have reviewed the methods for convergence by Muehlebach and Jordan (2022), and Leconte et al. (2023). We aim to investigate the removal of some unnecessary assumptions since the relaxed problem inherits some properties that are not general in normal settings. Therefore, we will try to eliminate the need for such strong assumptions and present our proof that fills the gap between the two papers.

Muehlebach and Jordan's approach relies heavily on the properties of the Lagrangian dual of (17) and the strong convexity of ℓ; however, in our problem and the background of QNN, convexity is always left unsatisfied, the technique of shrinking the range of w to $\mathcal{Z}$ no longer works in this case (See Muehlebach and Jordan, 2022, Claim 3. Notice that using the smoothness notion only would give a nearly useless result as we can no longer show that $d(x_k)$ is monotonically increasing).

In contrast, the approach that Leconte et al. (2023) have taken is based on the convergence guarantees of the stochastic subgradient method by Davis et al. (2020). However, in ASkewSGD, we are directly given the gradient of ℓ since it is continuously differentiable. Furthermore, in order to prove that $\ell(\mathcal{Z})$ has an empty interior (due to the verification for the weak Sard's condition), the approach that Leconte et al. have taken requires the use of a highly differentiable function ℓ. This can be often difficult to realize in deep neural networks when activations are imposed.

We will stately clearly for the assumptions that Leconte et al. have adopted for the convergence of ASkewSGD.

Assumption 5. For $j \in \{1, 2, \ldots, N\}$, the function ℓ_j is d-times continuously differentiable.

Assumption 6. There exists an $M > 0$ such that for all $j \in \{1, 2, \ldots, N\}$, such that the gradient associated with the j-th sample $\nabla\ell_j(w)$, is bounded by M for all $w \in \mathbb{R}$.

Assumption 7(a). The step sizes $(\gamma_k)_{k\geq 0}$ are positive, $\sum_{j=0}^{\infty}\gamma_k=\infty$ and $\sum_{j=0}^{\infty}\gamma_k^2<\infty$.

Theorem 2. (Leconte et al., 2023) Assuming the assumptions 1, 5, 6, 7(a) holds and $0<\varepsilon\leq \inf_{1\leq i\leq d}\inf_{1\leq j<K^i}|q_j^i-q_{j+1}^i|^4/16$, where $\{q_j^i\}$ are the quantization levels, $\ell(w_k)$ converges and $\lim_{k\to\infty}d(w_k,\mathcal{Z}_\varepsilon)=0$ almost surely.

In comparison, we will state our version of convergence guarantees. However, we would need to first modify the constraint function ψ for its smoothness. In particular, we are inspired by the piecewise convex property if we only focus on an interval of two neighboring quantization levels.

Definition 5. Let $\varepsilon\in[0,1]$, $w^i\in\mathcal{Q}^i, i\in\{1,2,\ldots,n\}$, i.e. each w^i coordinate-wisely takes value from its own set of quantization levels $\mathcal{Q}^i=\{q_1^i,q_2^i,\ldots,q_{K^i}^i\}$. We define the piecewise function

$$\varphi_\varepsilon^i(w^i):=\begin{cases}\varepsilon-(q_1^i-w^i), & w^i<q_1^i,\\ \varepsilon-(q_{j-1}^i-w^i)(w^i-q_j^i), & \\ \qquad q_{j-1}^i\leq w^i<q_j^i, j=2,\ldots,K,\\ \varepsilon-(w^i-q_K^i), & w^i\geq q_{K^i}^i,\end{cases}\tag{22}$$

for all $w^i\in\mathbb{R}$ and $i\in[n]$.

Now, we can redefine g as $(\varphi_\varepsilon^1,\varphi_\varepsilon^2,\ldots,\varphi_\varepsilon^n)$.

Assumption 7(b). The step sizes $(\gamma_k)_{k\geq 0}$ is given by $\gamma_k=\gamma_0/(k+1)$, where γ_0 is a positive constant.

We note that Assumption 7(b) is a special case of Assumption 7(a).

For classical stochastic analysis, we state the following assumptions for the property of the mini-batch of data samples drawn from the dataset, ensuring an unbiased stochastic gradient with bounded variance.

Assumption 8. Suppose that $\pi(\cdot)$ is the data distribution and ξ represents samples of data drawn in a mini-batch. We also denote $\pi(\xi)$ as the probability density function of ξ defined on the probability space $\mathtt{Z}$. We have

$$\ell(x)=\int_{\mathtt{Z}}\ell(x;\xi)\pi(\xi)\mathrm{d}\xi.$$

We are now ready to state the following assumptions.

(a) The stochastic gradient is unbiased, i.e.

$$\mathbb{E}_{\xi\sim\pi}[\nabla\ell(x;\xi)]=\nabla\ell(x),\forall x\in\mathbb{R}^n.$$

(b) The stochastic gradient has a bounded variance, i.e.

$$\mathbb{E}_{\xi\sim\pi}\left[\|\nabla\ell(x;\xi)-\nabla\ell(x)\|_2^2\right]\leq\sigma^2,\forall x\in\mathbb{R}^n.$$

Theorem 3. Assuming that the Assumptions 1, 6, 7(b), 8 holds, and $0<\varepsilon\leq\inf_{1\leq i\leq d}\inf_{1\leq j<K^i}|q_j^i-q_{j+1}^i|^2/4$, where $\{q_j^i\}$ are the quantization levels, $\ell(\hat{w}_k)$ converges and $\lim_{k\to\infty}d(\hat{w}_k,\mathcal{Z}_\varepsilon)=0$ almost surely. Here, $\hat{w}_k=w_{t+N_0}$ with probability $1/(H_k(t+N_0+1))$, where $H_k=\sum_{t=0}^{k-1}1/(t+N_0+1)$ and $N_0>0$ is a sufficiently large integer.

What makes our theorem stand out is that we have removed the need for a highly differentiable loss function ℓ, where we have enjoyed the benefits of the new definition of a smooth constraint function. The proof of Theorem 3 is split into several lemmata and they are deferred to Appendix B. We have made use of Leconte et al.'s (2023) lemma stating that the cluster point of the iterates w_∞ belongs to the constraint set C_ε. The subsequent lemmata then follow the classical stochastic gradient descent analysis with several important elements: descent lemma, telescoping sum, and step size substitution. The difficulty is to quantify the size of the update direction v_k, which may even diverge without clipping. However, we observe that the pushing force for being infeasible has the same order as the iterate's distance to the feasible set when the distance is sufficiently small. This is the key to our proof.

5 DERIVATION OF THE STOCHASTIC GRADIENT DESCENT ASCENT (SGDA) ALGORITHM

In this section, we aim to develop another algorithm for solving the QNN optimization problem by applying the results of Lin et al. (2024). We will first consider the minimax Lagrangian reformulation of (P2):

$$\min_{w\in\mathbb{R}^n}\max_{\lambda\geq 0}\mathcal{L}(w,\lambda),$$

where $\mathcal{L}(x,\lambda):=\ell(w)-\lambda^\top g(w)$. We emphasize that $\mathcal{L}$ is nonconvex in x but concave in λ (since $\mathcal{L}(x,\cdot)$ is linear for a fixed x), and $\{\lambda:\lambda\succeq 0\}$ is a convex set. In order to apply the algorithm, we will need to restrict the space, and the possible choice of λ within a ball of diameter D.

Even in the convex-concave settings, the gradient descent ascent (GDA) algorithm with equal step size can converge to limit cycles or even diverge in general. Thus, Lin et al. (2024) have provided the first nonasymptotic guarantees on the convergence of applying `SGDA` on minimax nonconvex-concave problems.

We will need to ensure that $\mathcal{L}$ is smooth. As such, we will decompose $\varphi^i(w^i)\geq 0$ into even smaller constraint inequalities. This can be done by some simple analytic continuations for each defined domain of the piecewise

function. By taking $\varphi^i(w^i)$ as an example, we obtain:

$$\begin{cases} f_1^i(w^i) := \varepsilon - (q_1^i - w^i) \\ f_j^i(w^i) := \varepsilon - (q_{j-1}^i - w^i)(w^i - q_j^i), \\ f_{K+1}^i(w^i) := \varepsilon - (w^i - q_K^i), \end{cases}$$

for all $w^i \in \mathbb{R}$ and $i \in [n]$.

Definition 6. A function f is ℓ-weakly convex if the function $f(x) + (\ell/2)\|x\|_2^2$ is convex.

Definition 7. A function $f_\lambda : \mathbb{R}^n \mapsto \mathbb{R}$ is the Moreau envelope of f with a positive parameter $\lambda > 0$ if $f_\lambda(x) = \min_w f(w) + (1/2)\|w - x\|_2^2$ for each $x \in \mathbb{R}^n$.

Assumption 9. The function ℓ is L-Lipschitz, i.e. $\|\ell(x) - \ell(y)\|_2 \leq L\|x - y\|_2$.

Lemma 1. Under Assumption 1, the function $\Phi := \max_{\lambda \succeq 0} \mathcal{L}(w, \lambda)$ is L-weakly convex and M-Lipschitz continuous.

Definition 8. A point x is an ε-stationary point of a differentiable function Φ if $\|\nabla\Phi(x)\|_2 \leq \varepsilon$.

Theorem 4. (Lin et al., 2024) Denote $\widehat{\Delta}_\Phi = \Phi_{1/2\ell}(w_0) - \min_w \Phi_{1/2\ell}(w)$ and $\widehat{\Delta}_0 = \Phi(w_0) - \mathcal{L}(w_0, \lambda_0)$. Under Assumption 1, 2, 8, 9, and letting the step sizes be chosen as $\eta_x = \Theta(\varepsilon_1^4/(L_\ell^3 D^2(L_\ell^2 + \sigma^2))$ and $\eta_y = \Theta(\varepsilon_1^2/L_\ell\sigma^2)$ with batch size $N_b = 1$, the iteration complexity of Algorithm 2 to return a ε_1-stationary point is bounded by

$$O\left(\left(\frac{L_\ell^3(L^2 + \sigma^2)D^2\widehat{\Delta}_\Phi}{\varepsilon_1^6} + \frac{L_\ell^3 D^2 \widehat{\Delta}_0}{\varepsilon_1^4}\right)\max\left\{1, \frac{\sigma^2}{\varepsilon_1^2}\right\}\right).$$

Lin et al. (2024) have also stressed the stronger notions of the stationarity of Φ than the stationarity of $\mathcal{L}$. To obtain an ε-stationary point of $\mathcal{L}$, we will need to pay an additional cost of $O(\varepsilon^{-4})$ stochastic gradients. Obtaining the stationary point of $\mathcal{L}$ essentially means that

$$\|\nabla\mathcal{L}(w, \lambda)\|_2 = \|\nabla\ell(w) - \lambda^\top \nabla g(w)\|_2 \leq \varepsilon,$$

thus bounding the distance of w and the KKT set in (21) within ε, ensuring the success of our algorithm.

6 NUMERICAL EXPERIMENTS

In this section, we will evaluate the performance of the algorithms presented in this paper. Specifically, we have used the full-precision SGD for comparison and tested BinaryConnect, Straight-Through Estimator (STE), `ASkewSGD`, modified `ASkewSGD` and `SGDA`

Algorithm 2 The Stochastic Gradient Descent Ascent (`SGDA`) Algorithm

1: Start with an initial dual guess $w_0 \in \mathbb{R}^n, \lambda_0 \geq 0$
2: **for** $k = 1, 2, \ldots, T$ **do**
3: Sample a mini-batch of N_b observations $\{j_1, j_2, \ldots, j_{N_b}\}$ in $\{1, 2, \ldots, N\}$
4: Compute the stochastic gradient $\widehat{\nabla}\mathcal{L}(w_{k-1}, \lambda_{k-1}) = 1/N_b \sum_{i=1}^{N_b} \nabla\mathcal{L}(w_{k-1}, \lambda_{k-1}; \xi_{j_i})$
5: $w_k \leftarrow w_{k-1} - \eta_x \widehat{\nabla}_{w_{k-1}} \mathcal{L}(w_{k-1}, \lambda_{k-1})$
6: $\lambda_k \leftarrow \mathcal{P}\left(\lambda_{k-1} + \eta_\lambda \widehat{\nabla}_{\lambda_{k-1}} \mathcal{L}(w_{k-1}, \lambda_{k-1})\right)$
7: **end for**
8: **return** $\hat{w}_k$ where $\hat{w}_k$ is drawn uniformly random from $\{w_k\}_{t=1}^T$

with weights quantized with 1, 2, and 4 bits. It is also conventional to use the notation [Wx/Ay] for describing a neural network architecture with x-bit precision weights and y-bit precision activations. We employ a k-bit quantization scheme where the quantization values $\mathcal{Q}$ are the integers from the quantization interval $[-2^{k-1}, 2^{k-1} - 1]$, e.g. $\{-8, -7, \ldots, 7\}$ for 4-bit precision.

Methods For clarity, we describe the implementation details of the aforementioned algorithms in the experiments.

For BinaryConnect (Courbariaux et al., 2015), we refer to the stochastic quantization method as defined in (4) during forward propagation, and an identity STE as described in (2) for backward propagation. No quantization operation is applied during the parameter updates. Furthermore, we clip the weights such that all values lie within the quantization interval after the update. Otherwise, the weights would grow too large without any impact on the resulting quantized neural network.

For STE (Hinton et al., 2012; Bengio et al., 2013), we refer to the deterministic quantization method as defined in (1) during forward propagation, and the STE by the derivative of the clipped ReLU as described in (3) for backward propagation. No quantization operation is applied during the parameter updates.

For `ASkewSGD` and modified `ASkewSGD`, we anneal the parameter ε. We run the algorithm until the test error does not improve, and then reduce ε with the logarithmic schedule K^t where $K = 0.88$. The iterates for the next immediate round are inherited from the iterates of the last round as a starting point. In `ASkewSGD`, we use the inequality constraints as defined in (19), whereas in modified `ASkewSGD`, we use the inequality constraints as defined in (22). Before the test set eval-

uation, we will apply the quantization to the neural network.

For `SGDA`, we employ Algorithm 2 with inequality constraints as defined in (22) equipped with the analytic continuations described in Section 5. We also test the different choices for ε.

We evaluate the performance of these quantization algorithms on three classical tasks: a convex problem finding the optimal separating hyperplane, a non-convex 2D problem on a shallow neural network with a single hidden layer, and an image classification benchmark on a deep neural network on ResNet-18 (He et al., 2015).

Task I - Logistic Regression Logistic regression is the baseline supervised machine learning algorithm to classify an observation (with features) into one of two classes (represented by the labels $\{-1, 1\}$). In this task, we will train a classifier consisting of ten 1-bit precision weights (i.e. -1 or 1) [W1/A32] taking feature vector x of dimension $d = 10$ as input. Unlike the classical logistic regression problem, we do not include the bias term for the prediction function.

We generate $n = 6000$ feature vectors $\{x_k\}_{k=1}^n$ in dimension $d = 10$ drawn independently from the uniform distribution in $[-1, 1]$. An optimal vector w^* is also chosen on the vertices of the hypercube, i.e. $w^* = \{-1, 1\}^{10}$ for the sake of illustrating how different quantization algorithms converge to the optimal point. The labels are then generated by $y_k \sim \text{Bernoulli}\left(\{1 + e^{-x_k^\top w^*}\}^{-1}\right)$.

All the methods are trained for 25 epochs with the learning rate set to 1, and the gradients are calculated on random batches of 1000 samples. The hyperparameter α for `ASkewSGD` and modified `ASkewSGD` is set to 0.1, and the clipping constant $M_{\varepsilon,c}$ is set to 1.

We plotted the training loss in Figure 2 and reported the loss of the last iteration in Table 1. The performance of `ASkewSGD`, modified `ASkewSGD`, `SGDA` is on par with or better than the full precision method. We also observe strong oscillations for BinaryConnect and STE due to the noisy gradient updates. This fact is further assisted by the plot for the batch gradient in Figure 3. For an even finer comparison, we have also reported the distance $\mathrm{d}(w_k, w^*)$ between the weight parameters (without quantization) of the classifier and the optimal separating hyperplane w^* in Table 2.

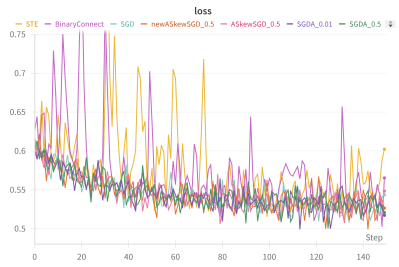

Figure 2: Training loss for Task I with a batch size of 1000, in total of 25 epochs.

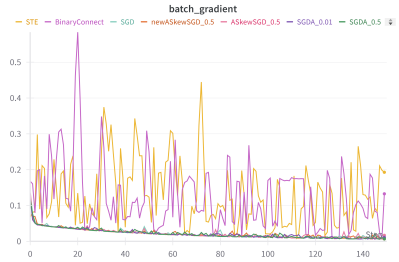

Figure 3: Batch Gradient for Task I where instability is observed for BinaryConnect and STE.

Table 1: Logistic loss after 25 training epochs.

Method	Loss
Full Precision [W32/A32]	0.5435
BinaryConnect [W1/A32]	0.5653
STE [W1/A32]	0.6022
`ASkewSGD` [W1/A32]	0.5484
Modified `ASkewSGD` [W1/A32]	0.5264
`SGDA` ($\varepsilon = 0.01$) [W1/A32]	**0.5172**
`SGDA` ($\varepsilon = 0.5$) [W1/A32]	0.5207

Table 2: The distance between the post-trained model parameters after 25 epochs and the optimal parameter $\mathrm{d}(w_k, w^*)$.

Method	$\mathrm{d}(w_k, w^*)$
Full Precision [W32/A32]	0.4023
BinaryConnect [W1/A32]	0.5242
STE [W1/A32]	0.7597
`ASkewSGD` [W1/A32]	0.4018
Modified `ASkewSGD` [W1/A32]	**0.4009**
`SGDA` ($\varepsilon = 0.01$) [W1/A32]	0.4156
`SGDA` ($\varepsilon = 0.5$) [W1/A32]	0.4035

Task II - Two Moons Classification We test the performance of different quantization algorithms on

a non-convex classification task. Inspired by the binary classification based on the "2 moons dataset" presented by Meng et al (2020) (see Figure 4), we train a binarized neural network with 9 weights [W1/A32], consisting of one hidden fully-connected layer with 3 neurons, and receives a dimension $d = 2$ input. We employ the ReLU function for the activation of this neural network. The binarized neural network (BNN) architecture is illustrated in Figure 5.

Our dataset consists of $n = 2000$ training samples (colored in blue and red corresponding to the two classes) and 200 test samples (colored in black), using the `make_moons` data generator from the `scikit-learn` library. Two clusters of interlacing half circles will then be generated. We will also add a random Gaussian noise with the variance of $\xi = 0.1$ to offset the point from its original position. We apply the logistic loss function in this experiment.

All methods are trained on the same neural network architecture for fair comparison. For benchmarking the result, we apply an exhaustive search to find the optimal binarized model (among $2^9 = 512$ configurations) that minimizes the test set loss. We note that the optimal model might not be unique due to the permutation invariability property (that leads to an equivalence). All the methods are trained for 25 epochs with the learning rate initially set to 1, and the gradients are calculated on random batches of 100 samples. The hyperparameter α for ASkewSGD and modified ASkewSGD is set to 4, and the clipping constant $M_{\varepsilon,c}$ is set to 1.

For this task, we plotted the training loss and additionally, the quantized loss per iteration in Figure 6. We also report the loss of the last iteration in Table 3. The performance of `ASkewSGD` falls behind our modified `ASkewSGD`, both of the `SGDA` algorithms with different ϵ and the full-precision method. We have also observed the effect of oscillation in gradient influencing the convergence from the statistics of BinaryConnect and STE. The low dimensionality of this task also enabled us to have a better view of the performance of binarized neural networks. The binarized weights have limited the development of the expressiveness ability of the neural network to form an irregular, curved decision boundary (see Figure 7). We have tested the performance by increasing the number of neurons in the hidden layer, the result is still unsatisfactory. We believe that more hidden layers should be added to the network for functional non-linearity and complication.

Table 3: Logistic loss after 25 training epochs.

Method	Loss
Full Precision [W32/A32]	0.3127
BinaryConnect [W1/A32]	0.3811
`ASkewSGD` [W1/A32]	0.3841
Modified `ASkewSGD` [W1/A32]	0.2964
`SGDA` ($\varepsilon = 0.01$) [W1/A32]	**0.2618**
`SGDA` ($\varepsilon = 0.5$) [W1/A32]	0.2632

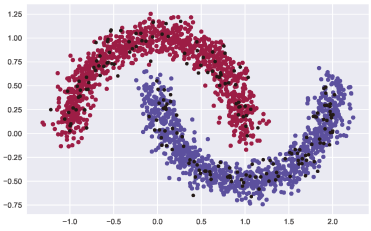

Figure 4: The 2 moons dataset, where the 2000 training samples are colored either blue or red, and the remaining 200 points belonging to the test set are colored black.

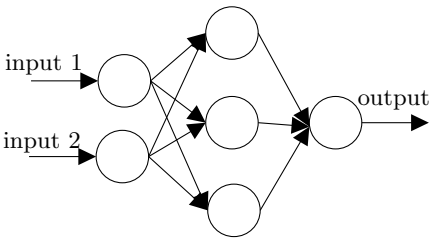

Figure 5: The BNN structure for the 2 moons classification task with a total number of 9 binary weights.

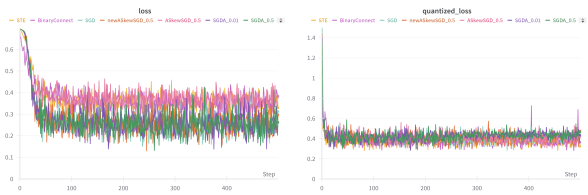

Figure 6: Training loss for Task II with a batch size of 100, in total of 25 epochs.

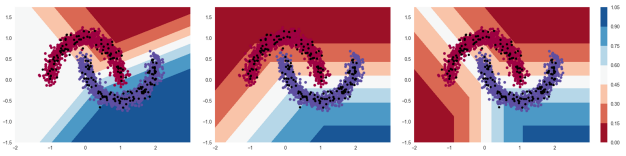

Figure 7: The representative results of the 2 moons classification problem. Left: Exhaustive search; Middle: `ASkewSGD`; Right: `SGDA`. The decision boundary is in between the red-shaded region and the blue-shaded region.

Task III - Computer Vision Task We have first tried to run `ASkewSGD` on the ResNet-18 architecture. However, with the computational resources available, it may cost more than 2 hours to train for 20 epochs. The significant accuracy drop after the start of annealing immediately becomes unbearable, which may take an unreasonable amount of time for parameter-tuning. With the limited time, we are not able to finish this task within the deadline of the project submission.

7 CONCLUSION

In this paper, we have explained the dire need for developing novel quantization techniques for deep neural networks. While there are advancements in different quantization schemes, we should keep on investigating the theoretical guarantees provided by these rules in order to keep the error at a justifiable level. We also showed the convergence guarantees of the modified `ASkewSGD` under weaker assumptions that cover more neural network architectures. For the `SGDA` algorithm, we should aim to seek the possibility of distributed optimization, as the constraints are highly structured and coordinate-wise independent (when the constraints are treated). Future work should keep on discovering the infinite possibilities of quantization models for filling in the gap in the number of bits required to maintain a good accuracy for neural networks and develop new theories about the power of the over-parameterization of deep neural network models.

Acknowledgements

The research is supported by a grant from the Engineering Faculty of The Chinese University of Hong Kong, under the Undergraduate Summer Research Internship Program (2024).

I truly appreciate Professor Hoi-To WAI's insightful advice and suggestions during the research period. His considerate pedagogical approach has greatly enhanced my understanding of the practical application of knowledge in research and has contributed to my quicker sense of mathematical optimization problems. During the research process, I was always overwhelmed by the mathematical barriers in optimization and mathematical analysis that were never brought up in discrete mathematics, and I ended up making a lot of trivial mistakes. These, are all precious learning experiences. Without Prof. WAI's insightful advice and rigorous investigations, I might not have been able to finish the project within this short summer period.

References

Y. Bai, Y.-X. Wang and E. Liberty (2019). Proxquant: Quantized Neural Networks via Proximal Operators. In *The 7th International Conference on Learning Representations 2019*.

Y. Bengio, N. Leonard and A. Courville (2013). Estimating or Propagating Gradients through Stochastic Neurons for Conditional Computation. In `arXiv:1308.3432`.

D. P. Bertsekas (1999). *Nonlinear Programming (Second Edition)*. Massachusetts: Athena Scientific.

L. Bottou, F. E. Curtis and J. Nocedal (2018). Optimization Methods for Large-Scale Machine Learning. In *SIAM Review*, **60**(2):223-311.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramech, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei (2020). Language Models are Few-Shot Learners. `arXiv:2005.14165`.

B. Chimel, R. Banner, E. Hoffer, H. B. Yaacov and D. Soudry (2023). Accurate Neural Training with 4-bit Matrix Multiplications at Standard Formats. In *The 11th International Conference on Learning Representations 2023*.

J. Choi, P. I. Chuang, Z. Wang, S. Venkataramani, V. Srinivasan and K. Gopalakrishnan (2018). Briding the Accuracy Gap for 2-bit Quantized Neural Networks (QNN). `arXiv:1807.06964v1`.

M. Courbariaux, Y. Bengio and J.-P. David (2015). BinaryConnect: Training Deep Neural Networks with Binary Weights During Propagations. In *Advances in Neural Information Processing Systems*, **28**:3123-3131.

D. Davis, D. Drusvyatskiy, S. Kakade, J. D. Lee (2020). Stochastic Subgradient Method Converges on Tame Functions. In *Foundations of Computational Mathematics*, **20**:119–154.s

M. Denil, B. Shakibi, L. Dinh, N. D. Freitas and M. Ranzato (2013). Predicting Parameters in Deep Learning. In *Advances in Neurial Information Processing Systems*, **26**:2148–2156.

A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney and K. Keutzer (2021). A Survey of Quantization Methods for Efficient Neural Network Inference. `arXiv:2103.13630v3`.

Y. Guo (2018). A Survey on Methods and Theories of Quantized Neural Networks. `arXiv:1808.04752`.

S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz and W. J. Dally (2016). EIE: Efficient Inference Engine on Compressed Deep Neural Network. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium*, 243-254, IEEE.

K. He, X. Zhang, S. Ren, J. Sun (2023). Deep Residual Learning for Image Recognition. `arXiv:1512.03385`.

R. Herzog (2023). *Lecture Notes of Non-Linear Optimization*. Interdisciplinary Center for Scientific Computing, Heidelberg University, 69120 Heidelberg, Germany.

G. Hinton, N. Srivastava and K. Swersky (2012). Neural networks for machine learning. *Coursera*, video lectures, 264.

I. Hubara, M. Courbariaux, D. Soudry, R. E.-Yaniv and Y. Bengio (2017). Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. In *Journal of Machine Learning Research*, **18**:187-191.

G. Lan (2020). *First-order and Stochastic Optimization Methods for Machine Learning*. Switzerland: Springer Nature.

L. Leconte, S. Schechtman and E. Moulines (2023). ASkewSGD: An Annealed Interval-Constrained Optimisation Method to Train Quantized Neural Networks. In *Artificial Intelligence and Statistics 2023*, **206**:3644-3663.

L. Liberti (2019). Undecidabiliity and Hardness in Mixed-Integer Nonlinear Programming. In *RAIRO Operations Research*, **53**:81-109.

T. Lin, C. Jin and M. I. Jordan (2024). On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. In *RAIRO Operations Research*, **53**:81-109.

X. Meng, R. Bachmann and M. E. Khan (2020). Training Binary Neural Networks Using the Bayesian Learning Rule. In *International Conference on Machine Learning 2020*, 6852–6861.

M. Muehlebach and M. I. Jordan (2022). On Constraints in First-Order Optimization: A View from Non-Smooth Dynamical Systems. In *Journal of Machine Learning Research*, **23**:1-47.

A. M.-C. So (2021). *Handouts of ENGG5501 - Foundations of Optimization*. Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting.. In *The Journal of Machine Learning Research*, **15**(1):1929-1958.

S. Sun, Z. Cao, H. Zhu and J. Zhao (2019). A Survey of Optimization Methods from a Machine Learning Perspective. `arXiv:1906.06821`.

H.-T. Wai (2024). *Lecture Notes of ESTR2520 - Optimization Methods*. Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong.

S. Zhou, Y. Wu, Z. Ni (2016). DoReFa-net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients. `arXiv:1606.06160`.

A CONCEPTS IN OPTIMIZATION THEORY - COROLLARIES DUE TO THE MANGASARIAN-FROMOVITZ CONSTRAINT QUALIFICATION, AND STRONG DUALITY OF THE CONSTRAINT ON VELOCITY

Definition 9. The Clarke's tangent cone of C contains all $\delta x \in T_C(x)$ if there exists two sequences $x_j \to x, x_j \in C, t_j \downarrow 0$ such that $(x_j - x)/t_j \to \delta x$. The normal cone is defined as follows: $N_C(x) = \{\lambda \in \mathbb{R}^n \mid \lambda^\top \delta x \leq 0, \forall \delta x \in T_C(x)\}$.

Lemma 2. Suppose that $x \in C$ and the Mangasarian-Fromovitz constraint qualification holds for every $x \in C$, then every $\delta x \in T_C(x)$ satisfies $\nabla g_i(x)\delta x \geq 0, \forall i \in I_x$. The converse also holds.

Proof. $(\Rightarrow)$: $\delta x \in T_C(x)$ implies that there exists two sequences $\{x_j\} \to x, \{x_j\} \subset C, t_j \downarrow 0$ for all $j \in \mathbb{N}$ and

$$\frac{x_j - x}{t_j} \to \delta x,$$

which implies that

$$\frac{g(x_j) - g(x)}{x_j - x} \cdot \frac{x_j - x}{t_j} \geq 0.$$

This is because $x_j \in C$ implies that $g_i(x_j) \geq 0$ and $g_i(x) \leq 0$ for all $i \in I(x)$.

$(\Leftarrow)$: Adapted from R. Herzog, 2023, a simplified version. Let δx satisfy $\nabla g_i(x)\delta x \geq 0, \forall i \in I_x$, also let δy be given by MFCQ such that $\nabla g_i(w)\delta y > 0, \forall i \in I(x)$. Put $\ell(t) := \delta x + t \cdot \delta y$. Then for all $t > 0$, we have $\nabla g_i(x)\ell(t) > 0, \forall i \in I(x)$, implying that $\ell(t)$ are all feasible MFCQ vectors.

Now, we claim that $\ell(t) \in T_C(x)$ for all $t \in \mathbb{R}_{++}$. Let $\gamma(t) := x + t\ell(t)$, $t \in (-\varepsilon, \varepsilon)$, for an infinitestimally small ϵ, given by the continuity of g. Then, $y(t) \in C$ for every $t \in [0, \epsilon)$ and $\gamma(0) = x, \gamma'(0) = \ell(t)$. For an arbitrary sequence $\{t_j\} \downarrow 0$ and $x_k = \gamma(t_j) \to x$ we have

$$\ell(t) = \gamma'(0) = \lim_{j\to\infty} \frac{\gamma(t_j) - \gamma(0)}{t_j - 0} = \lim_{j\to\infty} \frac{x_j - x}{t_j} \in T_C(x).$$

Since $T_C(x)$ is closed, $\delta x = \lim_{t\to 0} \ell(t) \in T(x)$. □

Now, we can simplify the tangent cone and the normal cone for any $x \in C$ as follows, due to the Mangasarian-Fromovitz constraint qualification:

$$T_C(x) = \{x \mid \nabla g_i(x)^\top x \geq 0, \forall i \in I_x\}, N_C(x) = \left\{ \lambda \in \mathbb{R}^d_+ \middle| - \sum_{i\in I(x)} \lambda_i \nabla g_i(x) \right\}.$$

Theorem 5. Suppose that x^* is a local minimizer of $\mathcal{P}$ which satisfies the MFCQ. Then there exist Lagrange multipliers λ^* (not necessarily unique) such that the KKT conditions are satisfied. The set of Lagrange multipliers $\Lambda(x^*)$ is compact.

See Chapter 8 of R. Herzog's (2023) lecture notes for the proof.

Definition 10. The indicator function for a convex set X is defined as follows:

$$I_X(x) = \begin{cases} 0, x \in X, \\ \infty, \text{ otherwise.} \end{cases}$$

Lemma 3. Strong duality holds for the optimization problem of the update

$$v_k = \underset{v\in V_{\varepsilon,\alpha}(w_k)}{\arg\min}\ (1/2) \left\| v + \widehat{\nabla}\ell(w_k) \right\|^2$$

in (17) when $V_{\varepsilon,\alpha}$ is non-empty for every $w_k \in \mathbb{R}^n$. For $\alpha \geq 0$, the dual can be rewritten as

$$\max_{\lambda\succeq 0} -\frac{1}{2} \left\| \lambda^\top \nabla g(w_k) - \ell(w_k) \right\|^2 - \alpha\lambda^\top g(w_k) \tag{23}$$

Proof. Let $D_{w_k} : \{\lambda \in \mathbb{R}^n : \lambda_i = 0 \text{ if } i \notin I(w_k)\}$. For clarity, we reformulate the primal problem in Lagrangian:

$$\min_{v \in \mathbb{R}^n} \max_{\lambda \in D_{w_k}} \frac{1}{2} \|v + \nabla \ell(w_k)\|^2 - \lambda^\top (\nabla g(w_k)^\top v + \alpha g(w_k)). \tag{24}$$

Then, the corresponding dual is as follows:

$$\max_{\lambda \in D_{w_k}} \min_{v \in \mathbb{R}^n} \frac{1}{2} \|v + \nabla \ell(w_k)\|^2 - \lambda^\top (\nabla g(w_k)^\top v + \alpha g(w_k)), \tag{25}$$

which then resolves into the required by picking $v^* = \lambda^\top \nabla g(w_k) - \nabla \ell(w_k)$ and the addition of the constant term $-(1/2) \|\nabla \ell(w_k)\|^2$.

We show that Slater's condition holds for $V_{\varepsilon,\alpha}$, i.e. there exists a $v \in \mathbb{R}^n$ such that $\nabla g_i(w_k)^\top v + \alpha g_i(w_k) > 0$ for all $i \in I_\varepsilon(w_k)$.

Let $\bar{v} \in \mathbb{R}^n$ satisfy $\nabla g_i(w_k)\bar{v} + \alpha g_i(w_k) = 0, \forall i \in I_\varepsilon$. By MFCQ, there exists a $w \in \mathbb{R}^n$ such that $\nabla g_i(w_k)w > 0, \forall i \in I_\varepsilon(w_k)$. Picking $v = \bar{v} + \xi w$ for some $\xi > 0$ satisfies the required condition.

Strong duality follows, from the fact that the optimization problem for the update is convex and Slater's condition holds. As a result, $\lambda(w_k)$ satisfies the following stationarity condition:

$$\nabla g(w_k)^\top (\lambda(w_k) \nabla g(w_k) - \nabla \ell(w_k)) + \alpha g(w_k) \in \partial \mu_{D_x}(\lambda(w_k)), \tag{26}$$

where $\mu_{D_{w_k}}$ is the indicator function of the set D_{w_k}. □

B PROOF OF THEOREM 3

We will first restate our version of the theorem about the convergence of `ASkewSGD`.

Theorem 3 Assuming that the Assumptions 1, 6, 7(b), 8 holds, and $0 < \varepsilon \leq \inf_{1\leq i\leq d} \inf_{1\leq j<K^i} |q_j^i - q_{j+1}^i|^2/4$, where $\{q_j^i\}$ are the quantization levels, $\ell(\hat{w}_k)$ converges and $\lim_{k\to\infty} d(\hat{w}_k, \mathcal{Z}_\varepsilon) = 0$ almost surely. Here, $\hat{w}_k = w_{t+N_0}$ with probability $1/(H_k(t+N_0+1))$, where $H_k = \sum_{t=0}^{k-1} 1/(t+N_0+1)$ and $N_0 > 0$ is a sufficiently large integer.

The following lemmata, 4-8, will be devoted to proving this theorem.

Lemma 4. Under Assumptions 1, 6, 7(b), it holds that $\limsup_{k\to\infty} \mathrm{d}(w_k, C_\varepsilon) = 0$ almost surely.

Proof. We would reproduce the proof from Leconte et al. (2023) to provide clarity to the reader. Before proceeding, we denote $M_1 = \max\{M_{\varepsilon,c}, M\}$. Now, for all $i \in [n]$, we have $\|\widehat{\nabla}\ell(w_k)\|_2 \leq M_1$ and $|v_k^i| \leq M_1$, and thus $|w_{k+1}^i - w_k^i| \leq \gamma_k M_1$.

Three crucial steps for the correctness of this Lemma will then be introduced. The three steps involved will correspond to Claims 1-3, Claims 4-5, and Claims 7-8.

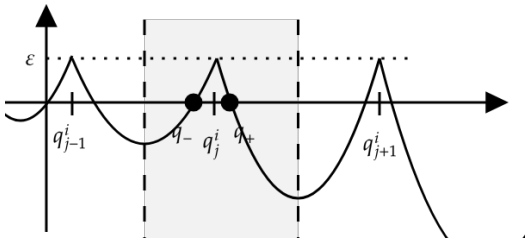

Figure 8: After $k > k_0$, w_k^i is guaranteed to stay within the interval $[(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$.

Claim 1. For $i \in \{1, 2, \ldots, d\}$, and for $2 \leq j \leq K_i - 1$ if the set $[(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$ is visited by w_k^i infinitely often, then there is k_0 such that for all $k > k_0$, $w_k^i \in [(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$.

Fix a j and denote $[q_-, q_+]$ the set $C_\varepsilon^i \cap [(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$ (see Figure 8, and the region corresponding to $[(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$ is shaded), where C_ε^i is the projection of C_ε on the i-th coordinate. Define $k_0 = \sup\{k : \gamma_k M_1 \geq \max(q_- - (q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2 - q_+)\}$. Consider $k \geq k_0$, suppose $(q_j^i + q_{j-1}^i)/2 \leq w_k^i < q_-$ (the left side of the interval), then the iterate is pushed to the right and $w_k^i \leq w_{k+1}^i$. Furthermore, by definition of k_0, it holds that $w_{k+1}^i \leq q_- + \gamma_k M_1 \leq (q_j^i + q_{j+1}^i)/2$. This implies, that in this case w_{k+1}^i stays in $[(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$. Otherwise, suppose $q_+ \leq w_k^i < (q_j^i + q_{j+1}^i)/2)$ (the right side of the interval), then the iterate is pushed to the left and $w_k^i \geq w_{k+1}^i \geq q_+ - \gamma_k M_1 \geq (q_j^i + q_{j-1}^i)/2$. Finally, if $w_k^i \in [q_-, q_+]$, then by the definition of k_0, we obtain $w_{k+1}^i \in [(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$. Thus, we have shown that for $k \geq k_0$ if w_k^i stays within the interval $[(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$, then for all $k' \geq k$, $w_{k'}^i \in [(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$.

Similarly, we can prove the two statements below corresponding to the edge cases.

Claim 2. For $i \in \{1, 2, \ldots, d\}$, if the set $(-\infty, (q_1^i + q_2^i)/2)$ is visited by w_k^i infinitely often, then there is k_0 such that for all $k > k_0$, $w_k^i \in (-\infty, (q_1^i + q_2^i)/2)$.

Claim 3. For $i \in \{1, 2, \ldots, d\}$, if the set $[(q_{K^i-1}^i + q_{K^i}^i)/2, +\infty)$ is visited by w_k^i infinitely often, then there is k_0 such that for all $k > k_0$, $w_k^i \in [(q_{K^i-1}^i + q_{K^i}^i)/2, +\infty)$.

We can then leave our attention to one single interval $[(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$, $(-\infty, (q_1^i + q_2^i)/2)$, or $[(q_{K^i-1}^i + q_{K^i}^i)/2, +\infty)$.

In the following claims, we will let w_k^i reside in an interval as defined in the context of Claims 1-3, i.e. $k \geq k_0$.

Claim 4. There is $k_1 \geq k_0$, such that if there are two indices $m_+ \geq m_- > k_0$ such that $w_{m_-} < q_- < q_+ < w_{m_+}$, then there is m, satisfying $m_- \leq m \leq m_+$, such that $w_m^i \in [q_-, q_+]$.

Define $k_1 = \sup\{k : \gamma_k M_1 \geq q_+ - q_-\}$. Let m_-, m_+ as the same in the claim and consider $m = \inf\{k \geq m_- : w_k^i \geq q_-\}$. It holds that $w_{m-1}^i < q_- \leq w_m^i \leq w_{m-1}^i + \gamma_k M_1$. Since $m \geq k_1$, this implies that $w_m^i \leq q_- + \gamma_k M_1 \leq q_+$, which proves the claim.

Claim 5. There is $k_1 \geq k_0$, such that if there are two indices $m_- \geq m_+ > k_0$ such that $w_{m_-} < q_- < q_+ < w_{m_+}$, then there is m, satisfying $m_+ \leq m \leq m_-$, such that $w_m^i \in [q_-, q_+]$.

The proof of Claim 5 is almost the same as in Claim 4.

Claims 4, and 5 show that there are only three behaviors of w_k^i. These three conditions will be treated by Claims 6, 7, and 8, respectively.

Claim 6. Let $k \geq k_0$. If w_k^i visits $[q_-, q_+]$ infinitely often, then $\limsup_{k\to\infty} w_k^i \leq c_+$ and $\limsup_{k\to\infty} w_k^i \geq c_-$.

We first show the case of $\limsup_{k\to\infty} w_k^i \leq c_+$. Suppose that $w_k^i > q_+$, then $w_{k+1}^i \leq w_k^i$, and otherwise if $w_k^i \leq q_+$ then $w_{k+1}^i \leq q_+ + \gamma_k M_1$. We note that γ_k is non-increasing and thus no iterates $w_{k'}$ where $k' \geq k$ can go further than $q_+ + \gamma_k M$ since it immediately receives the pushing force after the constraint violation. Tending k to infinity yields $\limsup_{k\to\infty} w_k^i \leq c_+$. It is easy to see that the other case also holds.

Claim 7. If for all $k \geq k_0$ large enough, $w_k^i > q_+$, then $w_k^i \to q_+$.

For all k large enough, the sequence w_k^i receives a pushing force from the right and thus is decreasing. From the given condition that $w_k^i > q_+$, w_k^i is decreasing and bounded below, thus the sequence has a limit. Assume the contrary that $\lim_{k\to\infty} w_k^i \neq q_+$. It holds that $w_{k+m+1}^i \leq w_k^i - M_+ \sum_{i=0}^m \gamma_{k+i}$, where $M_+ = \inf\{\min(M_{\varepsilon,c}, \alpha|\psi_\varepsilon^i(w)|/\psi_\varepsilon^{'i}(w)) : w \in [w_+, (q_j^i + q_{j+1}^i)/2)\} > 0$. Since $\sum_{j=0}^\infty \gamma_j = +\infty$, we have reached a contradiction. $w_k^i \to q_+$.

Claim 8. If for all $k \geq k_0$ large enough, $w_k^i < q_-$, then $w_k^i \to q_-$.

It holds that $w_{k+m+1}^i \geq w_k^i + M_- \sum_{i=0}^m \gamma_{k+i}$, where $M_- = \inf\{\min(M_{\varepsilon,c}, \alpha|\psi_\varepsilon^i(w)|/\psi_\varepsilon^{'i}(w)) : w \in ((q_j^i + q_{j-1}^i)/2, w_-]\} > 0$. Similar to Claim 7, it is impossible that $\lim_{k\to\infty} w_k^i \neq q_-$.

Lemma 5. Denote $[q_-, q_+]$ the set $C_\varepsilon^i \cap [(q_j^i + q_{j-1}^i)/2, (q_j^i + q_{j+1}^i)/2)$, where C_ε^i is the projection of C_ε on the i-th coordinate. Let $0 < \varepsilon \leq \inf_{1\leq i\leq d} \inf_{1\leq j<K^i} |q_j^i - q_{j+1}^i|^2/4$ and $(q_j^i + c_{j-1}^i)/2 - q_- < \varepsilon_1 < 0, 0 < \varepsilon_2 < (q_j^i + q_{j+1}^i)/2 - q_+$. The following statements on the small perturbations $\varepsilon_1, \varepsilon_2$ are true:

(a) $|-\alpha\psi_\varepsilon^i(q_- + \varepsilon_1)/(\psi_\varepsilon^{'i}(q_- + \varepsilon_1))| = O(\varepsilon_1)$;

(b) $|-\alpha\psi_\varepsilon^i(q_+ + \varepsilon_2)/(\psi_\varepsilon^{'i}(q_+ + \varepsilon_2))| = O(\varepsilon_2)$;

Proof. We will prove the case of q_- and it is easy to see (by symmetry) that the statement holds for the case of q_+.

Notice that $\psi_\varepsilon^i(q_-) = \varepsilon + (q_- - q_j^i)(q_- - q_{j-1}^i) = 0$. $\psi_\varepsilon^i(q_- + \varepsilon_1) = \varepsilon + (q_- + \varepsilon_1 - q_j^i)(q_- + \varepsilon_1 - q_{j-1}^i) = \varepsilon_1(2q_- - q_{j-1}^i - q_j^i) + \varepsilon_1^2 < 0$. Also, $\psi_\varepsilon^{'i}(q_- + \varepsilon_1) = 2(q_- + \varepsilon_1) - q_{j-1}^i - q_j^i > 0$.

Thus,

$$\begin{aligned}
\frac{-\alpha\psi_\varepsilon^i(q_- + \varepsilon_1)}{\psi_\varepsilon^{'i}(q_- + \varepsilon_1)} &= \frac{-\alpha(\varepsilon_1(2q_- - q_{j-1}^i - q_j^i) + \varepsilon_1^2)}{2(q_- + \varepsilon_1) - q_{j-1}^i - q_j^i} \\
&= \frac{-\alpha(1 + \varepsilon_1/(2q_- - q_{j-1}^i - q_j^i))}{1/\varepsilon_1 + 2/(2q_- - q_{j-1}^i - q_j^i)} \\
&\leq \frac{-\alpha\varepsilon_1/(2q_- - q_{j-1}^i - q_j^i)}{2/(2q_- - q_{j-1}^i - q_j^i)} \\
&\leq -\alpha\varepsilon_1/2.
\end{aligned}$$

□

Lemma 6. (Descent Lemma for `ASkewSGD`) Let $\varepsilon_1 > 0$. Then, there exists $K \geq 0$, such that $\forall k \geq K$, $1 - \gamma_k L/2 > 1/2$ and

$$\mathbb{E}\left[\ell(w_{k+1})|w_k\right] \leq \ell(w_k) - \frac{\gamma_k}{2} \sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2 + \frac{\gamma_k^2 \sigma^2 L}{2} + \gamma_k \sum_{i \in I_{k,\varepsilon_1}} MO(\varepsilon_1) + \gamma_k^2 \sum_{i \in I_{k,\varepsilon_1}} \frac{L}{2} O(\varepsilon_1^2),$$

where $i \notin I_{k,\varepsilon}$ if $w_k^i \in (q_-^i + \varepsilon_1, q_+^i - \varepsilon_1)$ or $v_k^i = -[\widehat{\nabla} \ell(w_k)]^i$.

Proof. By Lemma 1, for any $\varepsilon_1 > 0$, there exists K_1 such that $\mathrm{d}(w_k, C_\varepsilon) < \varepsilon_1$ for all $k \geq K_1$.

By Assumption 3, γ_k can reach an arbitrarily small positive value so that $1 - \gamma_k L/2 > 1/2$. Denote the smallest possible k as K_2.

Set $K = \min\{K_1, K_2\}$.

Consider the update rule for the model parameter $w_{k+1} = w_k + \gamma_k v_k$.

By the smoothness of ℓ, we obtain

$$\ell(w_{k+1}) \leq \ell(w_k) + \gamma_k v_k^\top \nabla \ell(w_k) + \frac{\gamma_k^2 L}{2} \|v_k\|_2^2.$$

Taking the conditional expectation $\mathbb{E}\left[\cdot|w_k\right]$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\ell(w_{k+1})|w_k\right] \leq & \quad \ell(w_k) + \gamma_k \mathbb{E}\left[v_k^\top \nabla \ell(w_k)|w_k\right] + \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\|v_k\|_2^2 |w_k\right] \\
\leq & \quad \ell(w_k) - \sum_{i \notin I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[v_k^i [\nabla \ell(w_k)]^i | w_k\right] + \sum_{i \in I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[v_k^i [\nabla \ell(w_k)]^i | w_k\right] \\
& + \sum_{i \notin I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|v_k^i\right|^2 \middle| w_k\right] + \sum_{i \in I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|v_k^i\right|^2 \middle| w_k\right] \\
\leq & \quad \ell(w_k) - \sum_{i \notin I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[[\widehat{\nabla} \ell(w_k)]^i [\nabla \ell(w_k)]^i \middle| w_k\right] + \sum_{i \in I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[v_k^i [\nabla \ell(w_k)]^i \middle| w_k\right] \\
& + \sum_{i \notin I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|[\widehat{\nabla} \ell(w_k)]^i\right|^2 \middle| w_k\right] + \sum_{i \in I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|v_k^i\right|^2 \middle| w_k\right] \\
\leq & \quad \ell(w_k) - \sum_{i \notin I_{k,\varepsilon_1}} \gamma_k \left|[\nabla \ell(w_k)]^i\right|^2 + \sum_{i \in I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[v_k^i [\nabla \ell(w_k)]^i \middle| w_k\right] \\
& + \sum_{i \notin I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|[\widehat{\nabla} \ell(w_k)]^i\right|^2 \middle| w_k\right] + \sum_{i \in I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|v_k^i\right|^2 \middle| w_k\right].
\end{aligned}$$

Note that

$$\begin{aligned}
\mathbb{E}\left[\left|[\widehat{\nabla} \ell(w_k)]^i\right|^2 \middle| w_k\right] = & \quad \mathbb{E}\left[\left|[\widehat{\nabla} \ell(w_k)]^i - [\nabla \ell(w_k)]^i + [\nabla \ell(w_k)]^i\right|^2 \middle| w_k\right] \\
= & \quad \mathbb{E}\left[\left|[\widehat{\nabla} \ell(w_k)]^i - [\nabla \ell(w_k)]^i\right|^2 \middle| w_k\right] \\
& + 2[\nabla \ell(w_k)]^i \mathbb{E}\left[[\widehat{\nabla} \ell(w_k)]^i - [\nabla \ell(w_k)]^i \middle| w_k\right] \\
& + \left|[\nabla \ell(w_k)]^i\right|^2 \\
\leq & \quad \sigma^2 + \left|[\nabla \ell(w_k)]^i\right|^2.
\end{aligned}$$

We notice that $i \notin I_{k,\varepsilon_1}$ implies that v_k^i takes $-\alpha \psi_\varepsilon^i(w_k^i)/(\psi_\varepsilon^{'i}(w_k^i))$, and following this hereby, we simply use the

notation v_k^i to denote the pushing force. Thus,

$$\begin{aligned}
\mathbb{E}\left[\ell(w_{k+1})|w_k\right] \leq \quad & \ell(w_k) - (\gamma_k - \frac{\gamma_k^2 L}{2}) \sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2 + \sum_{i \in I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[v_k^i [\nabla \ell(w_k)]^i \middle| w_k\right] \\
& + \sum_{i \in I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|v_k^i\right|^2 \middle| w_k\right] + \frac{\gamma_k^2 \sigma^2 L}{2} \\
\leq \quad & \ell(w_k) - \frac{\gamma_k}{2} \sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2 + \sum_{i \in I_{k,\varepsilon_1}} \gamma_k \mathbb{E}\left[|v_k^i||[\nabla \ell(w_k)]^i| \middle| w_k\right] \\
& + \sum_{i \in I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} \mathbb{E}\left[\left|v_k^i\right|^2 \middle| w_k\right] + \frac{\gamma_k^2 \sigma^2 L}{2} \\
\leq \quad & \ell(w_k) - \frac{\gamma_k}{2} \sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2 + \sum_{i \in I_{k,\varepsilon_1}} \gamma_k M_\ell O(\varepsilon_1) \\
& + \sum_{i \in I_{k,\varepsilon_1}} \frac{\gamma_k^2 L}{2} O(\varepsilon_1^2) + \frac{\gamma_k^2 \sigma^2 L}{2}.
\end{aligned}$$

□

Lemma 7. (Telescoping Sum Argument) Let $T > 0$, $\varepsilon_1 > 0$. Then, there exists $K \geq 0$, such that $\forall k \geq K$,

$$\begin{aligned}
& \sum_{k=0}^{T-1} \frac{\gamma_0}{2(k+K-1)} \left(\sum_{i \notin I_{k+K,\varepsilon}} \mathbb{E}\left[\left|[\nabla \ell(w_{k+K})]^i\right|^2\right] + \sum_{i \in I_{k+K,\varepsilon}} \mathbb{E}\left[\left|v_{k+K}^i\right|^2\right] \right) \\
\leq \quad & \mathbb{E}[\ell(w_K) - \ell(w_{K+T})] - \sum_{k=0}^{T-1} \frac{\gamma_0^2 \sigma^2 L}{2(k+K+1)^2} + \sum_{k=0}^{T-1} \frac{\gamma_0}{k+K+1} O(\varepsilon_1).
\end{aligned}$$

Proof Continuing from Lemma 3, we now have

$$\mathbb{E}\left[\ell(w_{k+1})|w_k\right] \leq \quad \ell(w_k) - \frac{\gamma_k}{2} \sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2 + \gamma_k O(\varepsilon_1) + \frac{\gamma_k^2 \sigma^2 L}{2}.$$

Taking the full expectation of the last inequality, we have

$$\begin{aligned}
\mathbb{E}\left[\ell(w_{k+1})\right] \leq \quad & \mathbb{E}\left[\ell(w_k)\right] - \frac{\gamma_k}{2} \mathbb{E}\left[\sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2\right] + \gamma_k O(\varepsilon_1) + \frac{\gamma_k^2 \sigma^2 L}{2}, \\
\frac{\gamma_k}{2} \mathbb{E}\left[\sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2\right] \leq \quad & \mathbb{E}\left[\ell(w_k)\right] - \mathbb{E}\left[\ell(w_{k+1})\right] + \gamma_k O(\varepsilon_1) + \frac{\gamma_k^2 \sigma^2 L}{2} \\
\frac{\gamma_k}{2} \mathbb{E}\left[\sum_{i \notin I_{k,\varepsilon_1}} \left|[\nabla \ell(w_k)]^i\right|^2\right] + \frac{\gamma_k}{2} \mathbb{E}\left[\sum_{i \in I_{k,\varepsilon_1}} \left|v_k^i\right|^2\right] \leq \quad & \mathbb{E}\left[\ell(w_k)\right] - \mathbb{E}\left[\ell(w_{k+1})\right] + \gamma_k O(\varepsilon_1) + \frac{\gamma_k^2 \sigma^2 L}{2}.
\end{aligned}$$

Summing over T epochs and rearranging the terms, we now have

$$\begin{aligned}
& \sum_{k=0}^{T-1} \frac{\gamma_{k+K}}{2} \left(\sum_{i \notin I_{k+K,\varepsilon}} \mathbb{E}\left[\left|[\nabla \ell(w_{k+K})]^i\right|^2\right] + \sum_{i \in I_{k+K,\varepsilon}} \mathbb{E}\left[\left|v_{k+K}^i\right|^2\right] \right) \\
\leq \quad & \mathbb{E}[\ell(w_K) - \ell(w_{K+T})] - \sum_{k=0}^{T-1} \frac{\gamma_{k+K}^2 \sigma^2 L}{2} + \sum_{k=0}^{T-1} \gamma_k O(\varepsilon_1).
\end{aligned}$$

Substituting $\gamma_{k+K} = \gamma_0/(k+K-1)$ concludes the proof. □

Lemma 8. (Heavy Tail Substitution) Let $\varepsilon_1 > 0$. Then, there exists $K \geq 0$, such that when $T \to \infty$, $\sum_{i \notin \bar{I}_\varepsilon} \mathbb{E}\left[\left|[\nabla \ell(\hat{w}_T)]^i\right|^2\right] + \sum_{i \in \bar{I}_\varepsilon} \mathbb{E}\left[\left|\hat{v}_T^i\right|^2\right] \to O(\varepsilon_1)$, where $\hat{w}_T = w_{t+K}, \hat{v}_T = v_{t+K}$ with probability $1/(H_T(t+K+1))$, $\bar{I}_\varepsilon$ is dependent on $\hat{w}_T, \hat{v}_T$, and $H_T = \sum_{i=0}^{T-1} 1/(i+K+1)$.

Proof

$$\mathbb{E}\left[\left|[\nabla \ell(\hat{w}_T)]^i\right|^2\right] = \sum_{t=0}^{T-1} \mathbb{P}(\hat{w}_T = w_{t+K}) \mathbb{E}\left[\left|[\nabla \ell(w_{t+K})]^i\right|^2\right] = \sum_{t=0}^{T-1} \frac{\mathbb{E}\left[\left|[\nabla \ell(w_{t+K})]^i\right|^2\right]}{H_T(t+K+1)}$$

Similarly,

$$\mathbb{E}\left[\left|\hat{v}_T^i\right|^2\right] = \sum_{t=0}^{T-1} \mathbb{P}(\hat{v}_T = v_{t+K}) \mathbb{E}\left[\left|v_T^i\right|^2\right] = \sum_{t=0}^{T-1} \frac{\mathbb{E}\left[\left|v_T^i\right|^2\right]}{H_T(t+K+1)}$$

Therefore,

$$\begin{aligned} & \sum_{i \notin \bar{I}_\varepsilon} \mathbb{E}\left[\left|[\nabla \ell(\hat{w}_T)]^i\right|^2\right] + \sum_{i \in \bar{I}_\varepsilon} \mathbb{E}\left[\left|\hat{v}_T\right|^2\right] \\ \leq \; & \frac{2}{\gamma_0 H_T} \left(\mathbb{E}[\ell(w_K) - \ell(w_{K+T})] - \sum_{k=0}^{T-1} \frac{\gamma_0^2 \sigma^2 L}{2(k+K+1)^2} + \sum_{k=0}^{T-1} \frac{\gamma_0}{k+K+1} O(\varepsilon_1) \right). \end{aligned}$$

The above inequality implies that for some constant $C > 0$,

$$\sum_{i \notin \bar{I}_\varepsilon} \mathbb{E}\left[\left|[\nabla \ell(\hat{w}_T)]^i\right|^2\right] + \sum_{i \in \bar{I}_\varepsilon} \mathbb{E}\left[\left|\hat{v}_T\right|^2\right] \leq \frac{C}{\log T} + O(\varepsilon_1).$$

Thus, when $T \to \infty$,

$$\sum_{i \notin \bar{I}_\varepsilon} \mathbb{E}\left[\left|[\nabla \ell(\hat{w}_T)]^i\right|^2\right] + \sum_{i \in \bar{I}_\varepsilon} \mathbb{E}\left[\left|\hat{v}_T\right|^2\right] \to O(\varepsilon_1)$$

and this concludes the theorem by recalling that ε_1 can be an arbitrary small constant due to Lemma 1. Also, we can show that the upper bound of ε_1 can be set to be irrespective of T and only depends on the number of parameters n, the pre-selected hyperparameter α, the smoothness constant L, and the upper bound of the (deterministic or stochastic) gradient.

Note that the update direction v_k also goes to $O(\varepsilon_1)$ and this implies the convergence of the cost function $\ell(\hat{w}_k)$. □