Mirror Descent View for Neural Network Quantization

Thalaiyasingam Ajanthan[*†]
Australian National University & Amazon

Kartik Gupta[*]
Australian National University & Data61, CSIRO

Philip H. S. Torr
University of Oxford

Richard Hartley
Australian National University

Puneet K. Dokania[‡]
University of Oxford & Five AI

[*]Equal contribution, [†] Work done prior to joining Amazon, [‡] Work done prior to joining Five AI

Abstract

Quantizing large Neural Networks (NN) while maintaining the performance is highly desirable for resource-limited devices due to reduced memory and time complexity. It is usually formulated as a constrained optimization problem and optimized via a modified version of gradient descent. In this work, by interpreting the continuous parameters (unconstrained) as the dual of the quantized ones, we introduce a Mirror Descent (MD) framework (Bubeck (2015)) for NN quantization. Specifically, we provide conditions on the projections (*i.e.*, mapping from continuous to quantized ones) which would enable us to derive valid mirror maps and in turn the respective MD updates. Furthermore, we present a numerically stable implementation of MD that requires storing an additional set of auxiliary variables (unconstrained), and show that it is strikingly analogous to the Straight Through Estimator (STE) based method which is typically viewed as a "trick" to avoid vanishing gradients issue. Our experiments on CIFAR-10/100, TinyImageNet, and ImageNet classification datasets with VGG-16, ResNet-18, and MobileNetV2 architectures show that our MD variants yield state-of-the-art performance.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 Introduction

Despite the success of deep neural networks in various domains, their excessive computational and memory requirements limit their practical usability for real-time applications or in resource-limited devices. Quantization is a prominent technique for network compression, where the objective is to learn a network while restricting the parameters (and activations) to take values from a small discrete set. This leads to a dramatic reduction in memory (a factor of 32 for binary quantization) and inference time – as it enables specialized implementation using bit operations.

Neural Network (NN) quantization is usually formulated as a constrained optimization problem $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$, where $f(\cdot)$ denotes the loss function by abstracting out the dependency on the dataset and $\mathcal{X} \subset \mathbb{R}^r$ denotes the set of all possible quantized solutions. Majority of the works in the literature (Ajanthan et al. (2019); Hubara et al. (2017); Yin et al. (2018)) convert this into an unconstrained problem by introducing auxiliary variables ($\tilde{\mathbf{x}}$) and optimize via (stochastic) gradient descent. Specifically, the objective and the update step take the following form:

$$\min_{\tilde{\mathbf{x}}\in\mathbb{R}^r} f(P(\tilde{\mathbf{x}}))\ , \quad \tilde{\mathbf{x}}^{k+1} = \tilde{\mathbf{x}}^k - \eta\, \nabla_{\tilde{\mathbf{x}}} f(P(\tilde{\mathbf{x}}))\big|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}^k}\ , \tag{1}$$

where $P : \mathbb{R}^r \to \mathcal{X}$ is a mapping from the unconstrained space to the quantized space (sometimes called projection) and $\eta > 0$ is the learning rate. In cases where the mapping P is not differentiable, a suitable approximation is employed (Hubara et al. (2017)).

In this work, by noting that the well-known Mirror Descent (MD) algorithm, widely used for online convex optimization (Bubeck (2015)), provides a theoretical framework to perform gradient descent in the unconstrained space (dual space, $\mathbb{R}^r$) with gradients computed in the quantized space (primal space, $\mathcal{X}$), we introduce an MD framework for NN quantization. In essence, MD extends gradient descent to non-Euclidean spaces where Euclidean projection is replaced with a more general projection defined based on the associated distance metric. Briefly, the key ingredient of MD is a concept called *mirror map* which defines both the mapping between primal and dual spaces and the exact form of the projection. Specifically, in this work, by observing P in Eq. (1) as a mapping from dual space to the primal space, we analytically derive corresponding mirror maps under certain conditions on P. This enables us to derive different variants of the MD algorithm useful for NN quantization.

Note that, MD requires the constrained set to be convex, however, the quantization set is discrete. Therefore, as discussed later in Sec. 3, to ensure quantized solutions, we employ a monotonically increasing annealing hyperparameter similar to Ajanthan et al. (2019); Bai et al. (2019). This translates into *time-varying mirror maps*, and for completeness, we theoretically analyze the convergence behaviour of MD in this case for the convex setting. Furthermore, as MD is often found to be numerically unstable (Hsieh et al. (2018)), we discuss a numerically stable implementation of MD by storing an additional set of auxiliary variables. This update is strikingly analogous to the popular Straight Through Estimator (STE) based gradient method (Bai et al. (2019); Hubara et al. (2017)) which is typically viewed as a "trick" to avoid vanishing gradients issue but here we show that it is an implementation method for MD under certain conditions on the mapping P. We believe this connection sheds some light on the practical effectiveness of STE.

In summary, we make the following contributions:

- We introduce an MD framework with time-varying mirror maps for NN quantization by deriving mirror maps from projections (P in Eq. (1)) and present two MD algorithms for quantization.
- Theoretically, we first show that MD with time-varying mirror maps converges at the same rate as the standard MD in the convex setting. Second, we discuss conditions for the convergence to a discrete solution when a monotonically increasing annealing hyperparameter is employed.
- For practical usability, we introduce a numerically stable implementation of MD and show its connection to the popular STE approximation.
- With extensive experiments on CIFAR-10/100, TinyImageNet, and ImageNet classification datasets using VGG-16, ResNet-18, and MobileNetV2 architectures we demonstrate that our MD variants yield state-of-the-art performance.

2 Preliminaries

Here we provide a brief background on the MD algorithm and NN quantization.

2.1 Mirror Descent

The Mirror Descent (MD) algorithm was first introduced in Nemirovsky and Yudin (1983) and has extensively been studied in the convex optimization literature ever since. In this section, we provide a brief overview and refer the interested reader to Chapter 4 of Bubeck (2015). In the context of MD, we consider a problem of the form:

$$\min_{\mathbf{x}\in\mathcal{X}} \; f(\mathbf{x}) \;, \tag{2}$$

where $f : \mathcal{X} \to \mathrm{I\!R}$ is a convex function and $\mathcal{X} \subset \mathrm{I\!R}^r$ is a compact convex set. The main concept of MD is to extend gradient descent to a more general non-Euclidean space (Banach space[1]), thus overcoming the dependency of gradient descent on the Euclidean geometry. The motivation for this generalization is that one might be able to exploit the geometry of the space to optimize much more efficiently. One such example is the simplex constrained optimization where MD converges at a much faster rate than the standard Projected Gradient Descent (PGD).

To this end, since the gradients lie in the dual space, optimization is performed by first mapping the primal point $\mathbf{x}^k \in \mathcal{B}$ (quantized space, $\mathcal{X}$) to the dual space $\mathcal{B}^*$ (unconstrained space, $\mathrm{I\!R}^r$), then performing gradient descent in the dual space, and finally mapping back the resulting point to the primal space $\mathcal{B}$. If the new point $\mathbf{x}^{k+1}$ lie outside of the constraint set $\mathcal{X} \subset \mathcal{B}$, it is projected to the set $\mathcal{X}$. Both the primal/dual mapping and the projection are determined by the *mirror map*. Specifically, the gradient of the mirror map defines the mapping from primal to dual and the projection is done via the Bregman divergence of the mirror map. We first provide the definitions for mirror map and Bregman divergence and then turn to the MD updates.

Definition 1 (Mirror map)**.** Let $\mathcal{C} \subset \mathrm{I\!R}^r$ be a convex open set such that $\mathcal{X} \subset \bar{\mathcal{C}}$ ($\bar{\mathcal{C}}$ denotes the closure of set $\mathcal{C}$) and $\mathcal{X} \cap \mathcal{C} \neq \emptyset$. Then, $\Phi : \mathcal{C} \to \mathrm{I\!R}$ is a mirror map if it satisfies:

1. Φ is strictly convex and differentiable.
2. $\nabla\Phi(\mathcal{C}) = \mathrm{I\!R}^r$, *i.e.*, $\nabla\Phi$ takes all possible values in $\mathrm{I\!R}^r$.
3. $\lim_{\mathbf{x}\to\partial\mathcal{C}} \|\nabla\Phi(\mathbf{x})\| = \infty$ ($\partial\mathcal{C}$ denotes the boundary of $\mathcal{C}$), *i.e.*, $\nabla\Phi$ diverges on the boundary of $\mathcal{C}$.

[1]A Banach space is a complete normed vector space where the norm is not necessarily derived from an inner product.

Definition 2 (Bregman divergence)**.** Let $\Phi : \mathcal{C} \to \mathbb{R}$ be a continuously differentiable, strictly convex function defined on a convex set $\mathcal{C}$. The Bregman divergence associated with Φ for points $\mathbf{p}, \mathbf{q} \in \mathcal{C}$ is the difference between the value of Φ at point $\mathbf{p}$ and the value of the first-order Taylor expansion of Φ around point $\mathbf{q}$ evaluated at point $\mathbf{p}$, *i.e.*,

$$D_\Phi(\mathbf{p}, \mathbf{q}) = \Phi(\mathbf{p}) - \Phi(\mathbf{q}) - \langle \nabla\Phi(\mathbf{q}), \mathbf{p} - \mathbf{q} \rangle \ . \tag{3}$$

Notice, $D_\Phi(\mathbf{p}, \mathbf{q}) \ge 0$ with $D_\Phi(\mathbf{p}, \mathbf{p}) = 0$, and $D_\Phi(\mathbf{p}, \mathbf{q})$ is convex on $\mathbf{p}$.

Now we are ready to provide the mirror descent strategy based on the mirror map Φ. Let $\mathbf{x}^0 \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}\cap\mathcal{C}} \Phi(\mathbf{x})$ be the initial point. Then, for iteration $k \ge 0$ and step size $\eta > 0$, the update of the MD algorithm can be written as:

$$\begin{aligned} \nabla\Phi(\mathbf{y}^{k+1}) &= \nabla\Phi(\mathbf{x}^k) - \eta\, \mathbf{g}^k \ , \\ \mathbf{x}^{k+1} &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}\cap\mathcal{C}} D_\Phi(\mathbf{x}, \mathbf{y}^{k+1}) \ , \end{aligned} \tag{4}$$

where $\mathbf{g}^k \in \partial f(\mathbf{x}^k)$ and $\mathbf{y}^{k+1} \in \mathcal{C}$. Note that, in Eq. (4), the gradient $\mathbf{g}^k$ is computed at $\mathbf{x}^k \in \mathcal{X} \cap \mathcal{C}$ (solution space) but the gradient descent is performed in $\mathbb{R}^r$ (unconstrained dual space). Moreover, by simple algebraic manipulation, it is easy to show that the above MD update (4) can be compactly written in a proximal form where the Bregman divergence of the mirror map becomes the proximal term (Beck and Teboulle (2003)):

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}\cap\mathcal{C}} \langle \eta\, \mathbf{g}^k, \mathbf{x} \rangle + D_\Phi(\mathbf{x}, \mathbf{x}^k) \ . \tag{5}$$

Note, if $\Phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, then $D_\Phi(\mathbf{x}, \mathbf{x}^k) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^k\right\|_2^2$, which when plugged back to the above problem and optimized for $\mathbf{x}$, leads to exactly the same update rule as that of PGD. However, MD allows us to choose various forms of Φ depending on the problem at hand.

2.2 Neural Network Quantization

Neural Network (NN) quantization amounts to training networks with parameters (and activations) restricted to a small discrete set representing the quantization levels. Here we discuss how one can formulate parameter quantization as a constrained optimization problem and activation quantization can be similarly formulated.

Parameter Space Formulation. Given a dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, parameter quantization can be written as:

$$\min_{\mathbf{w}\in\mathcal{Q}^m} L(\mathbf{w}; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{w}; (\mathbf{x}_i, \mathbf{y}_i)) \ . \tag{6}$$

Here, $\ell(\cdot)$ denotes the input-output mapping composed with a standard loss function (*e.g.*, cross-entropy loss), $\mathbf{w}$ is the m dimensional parameter vector, and $\mathcal{Q}$ with $|\mathcal{Q}| = d$ is a predefined discrete set representing quantization levels (*e.g.*, $\mathcal{Q} = \{-1, 1\}$ or $\mathcal{Q} = \{-1, 0, 1\}$).

The approaches that directly optimize in the parameter space include BinaryConnect (BC) (Courbariaux et al. (2015)) and its variants (Hubara et al. (2017); Rastegari et al. (2016)), where the constraint set is discrete. In contrast, recent approaches (Bai et al. (2019); Yin et al. (2018)) relax this constraint set to be its convex hull:

$$\operatorname{conv}(\mathcal{Q}^m) = [q_{\min}, q_{\max}]^m \ , \tag{7}$$

where $q_{\min}$ and $q_{\max}$ represent the minimum and maximum quantization levels, respectively. In this case, a quantized solution is obtained by gradually increasing an annealing hyperparameter.

Lifted Probability Space Formulation. Another formulation is to treat NN quantization as a discrete labelling problem based on the Markov Random Field (MRF) perspective (Ajanthan et al. (2019)). Here, the equivalent relaxed optimization problem corresponding to Eq. (6) can be written as:

$$\min_{\mathbf{u}\in\Delta^m} L(\mathbf{u}\mathbf{q}; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{u}\mathbf{q}; (\mathbf{x}_i, \mathbf{y}_i)) \ , \tag{8}$$

where $\mathbf{q}$ is the vector of quantization levels with $\mathbf{w} = \mathbf{u}\mathbf{q}$ and the set Δ^m takes the following form:

$$\Delta^m = \left\{ \mathbf{u} \;\middle|\; \begin{array}{ll} \sum_\lambda u_{j:\lambda} = 1, & \forall\, j \\ u_{j:\lambda} \ge 0, & \forall\, j, \lambda \end{array} \right\} \ . \tag{9}$$

We can interpret the value $u_{j:\lambda}$ as the probability of assigning the discrete label λ to the weight w_j. Therefore Eq. (8) can be interpreted as optimizing the probability of each parameter taking a discrete label.

3 Mirror Descent Framework for Network Quantization

Before introducing the MD formulation, we first write NN quantization as a single objective unifying (6) and (8) as:

$$\min_{\mathbf{x}\in\mathcal{X}} \ f(\mathbf{x}) \ , \tag{10}$$

where $f(\cdot)$ denotes the loss function by abstracting out the dependency on the dataset $\mathcal{D}$, and $\mathcal{X}$ denotes the constraint set. As discussed in Sec. 2.2, many recent NN quantization methods optimize over the convex hull of the constraint set. Following this, we consider the solution space $\mathcal{X}$ in Eq. (10) to be convex and compact.

To employ MD, we need to choose a mirror map (refer Definition 1) suitable for the problem at hand. In fact, as discussed in Sec. 2.1, mirror map is the core component of an MD algorithm which determines the

Projection (P_{β_k})	Space	Mirror Map (Φ_{β_k})	Update Step
$\tanh(\beta_k \tilde{w})$	$\mathbf{w}$	$\Phi_{\beta_k}(w) = \frac{1}{2\beta_k}\big[(1+w)\log(1+w) + (1-w)\log(1-w)\big]$	$w^{k+1} = \frac{\frac{1+w^k}{1-w^k}\exp(-2\beta_k\eta g^k)-1}{\frac{1+w^k}{1-w^k}\exp(-2\beta_k\eta g^k)+1}$
$\mathrm{softmax}(\beta_k \tilde{\mathbf{u}})$	$\mathbf{u}$	$\Phi_{\beta_k}(\mathbf{u}) = \frac{1}{\beta_k}\left[\sum_{\lambda\in\mathcal{Q}} u_\lambda \log(u_\lambda) - u_\lambda\right]$	$u_\lambda^{k+1} = \frac{u_\lambda^k \exp(-\beta_k\eta g_\lambda^k)}{\sum_{\mu\in\mathcal{Q}} u_\mu^k \exp(-\beta_k\eta g_\mu^k)} \quad \forall\,\lambda\in\mathcal{Q}$

Table 1: *Example projections, corresponding mirror maps, and update steps obtained using Theorem 1. Here, k is the iteration index, $\eta > 0$ is the learning rate, g^k is the gradient of f computed in the primal space, $\beta_k \ge 1$ is the annealing hyperparameter, and we assume $m = 1$ without loss of generality. Notice the obtained mirror maps vary at each iteration due to β_k and the* softmax *update resembles the popular Entropic Descent Algorithm (*EDA*) (Beck and Teboulle (2003)).*

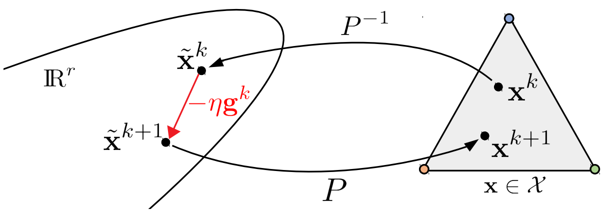

Figure 1: MD *formulation where mirror map is derived from the projection P. Note, $\mathbf{g}^k$ is computed in the primal space ($\mathcal{X}$) but it is directly used to update the auxiliary variables in the dual space.*

effectiveness of the resulting MD updates. However, there is no straightforward approach to obtain a mirror map for a given constrained optimization problem, except in certain special cases.

To this end, we observe that the usual approach to optimize the above constrained problem is via a version of projected gradient descent, where the projection is the mapping from the unconstrained auxiliary variables (full-precision) to the quantized space $\mathcal{X}$. Now, noting the analogy between the purpose of the projection operator and the mirror maps in the MD formulation, we intend to derive the mirror map analogous to a given projection. Precisely, we prove that if the *projection is strictly monotone* (and hence invertible), a valid mirror map can be derived from the projection itself. Even though this does not necessarily extend the theory of MD, this derivation is valuable as it connects existing PGD type algorithms to their corresponding MD variants. For completeness, we state it as a theorem for the case $\mathcal{X} \subset \mathbb{R}$ and the multidimensional case can be proved with an additional assumption that the vector field $P^{-1}(\mathbf{x})$ is conservative.

Theorem 1. *Let $\mathcal{C}$ be a finite open interval and $P : \mathbb{R} \to \mathcal{C}$ be a strictly monotonically increasing continuous function. Then, $\Phi(x) = \int_{x_0}^{x} P^{-1}(y)dy$ is a valid mirror map.*

Proof. This can be proved by noting that P is invertible, $\nabla\Phi(x) = P^{-1}(x)$, and $\Phi(x)$ is strictly convex. □

The MD update based on the mirror map derived from a given projection is illustrated in Fig. 1. Note that, to employ MD to the problem (10), in theory, any mirror map satisfying Definition 1 whose domain (*i.e.*, its closure) is a superset of the constraint set $\mathcal{X}$ can be chosen. The above theorem provides a method to derive a subset of all applicable mirror maps, where the closure of the domain of mirror maps is exactly equal to the constraint set $\mathcal{X}$.

We now provide mirror maps and update steps for two different projections (tanh for $\mathbf{w}$-space (Eq. (6)) and softmax for $\mathbf{u}$-space (Eq. (8))) useful for NN quantization in Table 1. Given mirror maps (from Theorem 1), the MD updates are straightforwardly derived based on Eq. (5) using KKT conditions (Boyd and Vandenberghe (2009)). For the detailed derivations and pseudocode for MD-tanh, please refer to Appendix. Furthermore, the tanh projection, its inverse, and the corresponding mirror map are illustrated in Fig. 2, showing monotonicity of the inverse and strict convexity of the derived mirror map.

According to the update steps in Table 1, our MD variants corresponding to tanh and softmax projections can be performed directly in the primal space. However, for some projections (*e.g.*, multi-bit quantization), it might be non-trivial to derive the exact form of mirror maps (and the MD update), nevertheless, the MD update can be easily implemented by storing an additional set of auxiliary variables. This as discussed in Sec. 3.2 also improves the numerical stability of MD.

Note that, to ensure a discrete solution at the end of the training, the projection P is parametrized by a scalar $\beta_k \ge 1$ and it is annealed throughout the optimization. This annealing hyperparameter translates into a time varying mirror map (refer to Table 1) in our case. Intuitively, such an adaptive mirror map gradually constrains the solution space $\mathcal{X}$ to its boundary and in the limit enforces a quantized solution.

3.1 Annealing and Convergence Analysis

The classical MD literature studied the convergence behaviour of MD for the convex setting, and the adaptive

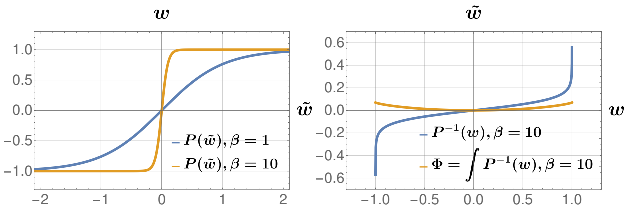

Figure 2: *Plots of* tanh*, its inverse and corresponding mirror map. Note that, the inverse is monotonically increasing and the mirror map is strictly convex. Moreover, when $\beta \to \infty$,* tanh *approaches the step function.*

mirror maps are considered in online learning (McMahan (2017)). We now prove that, in the convex setting, if the annealing hyperparameter β_k is bounded, then MD with an adaptive mirror map converges to the optimal value at the same rate of $\mathcal{O}(1/\sqrt{t})$ as the standard MD.

Theorem 2. *Let $\mathcal{X} \subset \mathbb{R}^r$ be a convex compact set and $\mathcal{C} \subset \mathbb{R}^r$ be a convex open set with $\mathcal{X} \cap \mathcal{C} \neq \emptyset$ and $\mathcal{X} \subset \bar{\mathcal{C}}$ ($\bar{\mathcal{C}}$ denotes the closure of $\mathcal{C}$). Let $\Phi : \mathcal{C} \to \mathbb{R}$ be a mirror map ρ-strongly convex[2] on $\mathcal{X} \cap \mathcal{C}$ with respect to $\|\cdot\|$, $R^2 = \sup_{\mathbf{x} \in \mathcal{X} \cap \mathcal{C}} \Phi(\mathbf{x}) - \Phi(\mathbf{x}^0)$ where $\mathbf{x}^0 = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X} \cap \mathcal{C}} \Phi(\mathbf{x})$ is the initialization and $f : \mathcal{X} \to \mathbb{R}$ be a convex function and L-Lipschitz with respect to $\|\cdot\|$. Then* MD *with mirror map $\Phi_{\beta_k}(\mathbf{x}) = \Phi(\mathbf{x})/\beta_k$ with $1 \leq \beta_k \leq B$ and $\eta = \frac{R}{L}\sqrt{\frac{2\rho}{Bt}}$ satisfies*

$$f\left(\frac{1}{t}\sum_{k=0}^{t-1} \mathbf{x}^k\right) - f(\mathbf{x}^*) \leq RL\sqrt{\frac{2B}{\rho t}} \ , \qquad (11)$$

where β_k is the annealing hyperparameter, $\eta > 0$ is the learning rate, t is the iteration index, and $\mathbf{x}^*$ is the optimal solution.

Proof. The proof is a slight modification to the proof of standard MD noting that, effectively Φ_{β_k} is ρ/B-strongly convex. Please refer to Appendix. □

Theoretical analysis of MD for nonconvex, stochastic setting is an active research area (Zhou et al. (2017a,b)) and MD has been recently shown to converge in the nonconvex stochastic setting under certain conditions (Zhang and He (2018)). We believe, similar to Theorem 2, the convergence analysis in Zhang and He (2018) can be extended to MD with adaptive mirror maps. Nevertheless, MD converges in all our experiments while outperforming the baselines in practice.

[2] A convex function $\Phi : \mathcal{C} \to \mathbb{R}$ is ρ-strongly convex with respect to $\|\cdot\|$ if $\Phi(\mathbf{x}) - \Phi(\mathbf{y}) \leq \langle \mathbf{g}, \mathbf{x} - \mathbf{y} \rangle - \frac{\rho}{2}\|x - y\|^2, \forall x, y \in \mathcal{C}$ and $g \in \partial\Phi(\mathbf{x})$.

Ensuring a discrete solution. Our original objective Eq. (10) is to obtain a discrete solution via annealing the hyperparameter $\beta_k \to \infty$. However, according to Theorem 2, β_k is capped at an arbitrarily chosen maximum value B. To this end, we now derive a constraint on the auxiliary variables $\tilde{\mathbf{x}}$ such that the primal variables converge to a discrete solution with a chosen precision $\epsilon > 0$ for a given B.

We consider the tanh projection with $m = 1$ without loss of generality and a similar derivation is possible for the softmax projection as well. Since $\beta_k \leq B$, $\tilde{x}$ has to be constrained away from zero to ensure that $\tanh(B\tilde{x})$ is close to the set $\{-1, 1\}$ with a desired precision ϵ. We now state it as a proposition below.

Proposition 1. For a given $B > 0$ and $0 < \epsilon < 1$, there exists a $\gamma > 0$ such that if $|\tilde{x}| \geq \gamma$ then $1 - |\tanh(B\tilde{x})| < \epsilon$. Here $|\cdot|$ denotes the absolute value and $\gamma > \tanh^{-1}(1 - \epsilon)/B$.

Proof. This is derived via a simple algebraic manipulation of tanh. Please refer to Appendix. □

3.2 Numerically Stable form of MD

We showed two examples of valid projections, their corresponding mirror maps, and the final MD updates in Table 1. Even though, in theory, these updates can be used directly, they are sometimes numerically unstable due to the operations involving multiple logarithms, exponentials, and divisions (Hsieh et al. (2018)). To this end, we provide a numerically stable way of performing MD by storing a set of auxiliary parameters during training.

A careful look at the Fig. 1 suggests that the MD update with the mirror map derived from Theorem 1 can be performed by storing auxiliary variables $\tilde{\mathbf{x}} = P^{-1}(\mathbf{x})$. In fact, once the auxiliary variable $\tilde{\mathbf{x}}^k$ is updated using gradient $\mathbf{g}^k$, it is directly mapped back to the constraint set $\mathcal{X}$ via the projection. This is mainly because of the fact that the domain of the mirror maps derived based on the Theorem 1 is exactly the same as the constraint set. Formally, with this additional set of variables, one can write the MD update (4) corresponding to the projection P as:

$$\tilde{\mathbf{x}}^{k+1} = \tilde{\mathbf{x}}^k - \eta\, \mathbf{g}^k \ , \quad \text{update in the dual space} \quad (12)$$
$$\mathbf{x}^{k+1} = P(\tilde{\mathbf{x}}^{k+1}) \in \mathcal{X} \ , \quad \text{projection to primal space}$$

where $\eta > 0$ and $\mathbf{g}^k \in \partial f(\mathbf{x}^k)$. Experimentally we observed these updates to show stable behaviour and performed remarkably well for both the tanh and softmax. We provide the pseudocode of this stable version of MD in Algorithm 2 for the tanh (MD-tanh-S) projection. Extending it to other valid projections is trivial.

Note, above updates can be seen as optimizing the function $f(P(\tilde{\mathbf{x}}))$ using gradient descent where the gradient through the projection (*i.e.*, Jacobian) $J_P = \partial P(\tilde{\mathbf{x}})/\partial\tilde{\mathbf{x}}$ is replaced with the identity matrix. This is exactly the same as the Straight Through Estimator (STE) for NN quantization (following the nomenclature of Bai et al. (2019); Yin et al. (2018)). Despite being a crude approximation, STE has shown to be highly effective for NN quantization with various network architectures and datasets (Yin et al. (2018); Zhou et al. (2016)). However, a solid understanding of the effectiveness of STE is lacking in the literature except for its convergence analysis in certain cases (Li et al. (2017); Yin et al. (2019)). In this work, by showing STE based gradient descent as an implementation method of MD under certain conditions on the projection, we provide a justification on the effectiveness of STE.

Mirror Descent *vs.* ProxQuant. The connection between the dual averaging version of MD and STE was recently hinted in ProxQuant (PQ) (Bai et al. (2019)). However, no analysis of whether an analogous mirror map exists to the given projection is provided and their final algorithm is not based on MD.

Briefly, PQ optimizes a objective of the following form:

$$\min_{\mathbf{x}\in\mathbb{R}^r} f(\mathbf{x}) + \beta R(\mathbf{x}) \ , \tag{13}$$

where f is the loss function, the regularizer R is a "W" shaped nonconvex function and β is an annealing hyperparamter similar to ours. Notice, even when the loss function f is convex, the above PQ objective would be nonconvex and has multiple local minima for a range of values of β. Therefore PQ is prone to converge to any of these local minima, whereas, our MD algorithm (even PGD) is guaranteed to converge to the global optimum regardless of the value of β.

4 Related Work

In this work, we mainly consider parameter quantization, which is usually formulated as a constrained problem and optimized using a modified projected gradient descent algorithm, where the methods (Ajanthan et al. (2019); Bai et al. (2019); Carreira-Perpinán and Idelbayev (2017); Chen et al. (2019); Courbariaux et al. (2015); Yang et al. (2019); Yin et al. (2018)) mainly differ in the constraint set, the projection used, and how backpropagation through the projection is performed. Among them, STE based gradient descent is the most popular method as it enables backpropagation through nondifferentiable projections and it has shown to be highly effective in practice (Courbariaux et al. (2015)). In fact, the success of this approach lead to various extensions by including additional layerwise scalars (Rastegari et al. (2016)), relaxing the solution space (Yin et al. (2018)), and even to quantizing activations (Hubara et al. (2017)), and/or gradients (Zhou et al. (2016)). Some recent works (Leng et al. (2018); Ye et al. (2019)) employ Alternating Direction Method of Multipliers (ADMM) framework to learn low-bit neural networks. In contrast to these, recently Helwegen et al. (2019) proposed Binary Optimizer (BOP) to avoid using "latent" real-valued weights during training. Moreover, there are methods focusing on loss aware quantization (Hou et al. (2017)), quantization for specialized hardware (Esser et al. (2015)), and quantization based on the variational approach (Achterhold et al. (2018); Louizos et al. (2017, 2019)). Some recent works (Liu et al. (2018); Martinez et al. (2019); Liu et al. (2020)) have also explored architectural modifications beneficial to increase the capacity of binarized neural networks. We have only provided a brief summary of relevant methods and for a comprehensive survey, we refer the reader to Guo (2018).

5 Experiments

Due to the popularity of binary neural networks (Courbariaux et al. (2015); Rastegari et al. (2016)), we mainly consider binary quantization and set the quantization levels as $\mathcal{Q} = \{-1, 1\}$. We perform two sets of extensive experiments for comparisons against the state-of-the-art NN binarization methods.

First, we perform *full binarization* experiments similar to Ajanthan et al. (2019) where all learnable parameters are binarized and activations are kept floating point on small scale datasets such as CIFAR-10/100 and TinyImageNet[3] with VGG-16, ResNet-18 and MobileNetV2 architectures. Note, this is more difficult than the standard setup used in the quantization literature where the first and last layers are usually kept in high precision to enhance the performance. To ensure fair comparison, we ran the comparable baselines in this setup and performed extensive cross-validation, *e.g.*, *up to* 3% *improvement for* PMF (Ajanthan et al. (2019)) due to this. In summary, our results indicate that the binary networks obtained by the MD variants outperform comparable baselines yielding state-of-the-art performance.

Secondly, we evaluated our MD-tanh-S approach on large scale ImageNet dataset with ResNet-18 in two different setups: 1) only parameters are binarized; and 2) both activations and parameters are binarized. We follow a similar experimental setup for our approach as has been used in baselines. For both the setups, our MD-tanh-S variant outperforms all the recent baselines *even without requiring layerwise scalars* and sets new

[3] `https://tiny-imagenet.herokuapp.com/`

	Algorithm	Space	CIFAR-10 VGG-16	CIFAR-10 ResNet-18	CIFAR-100 VGG-16	CIFAR-100 ResNet-18	TinyImageNet ResNet-18
	REF (float)	**w**	93.33	94.84	71.50	76.31	58.35
	BC	**w**	89.04	91.64	59.13	72.14	49.65
	PQ	**w**	85.41	90.76	39.61	65.13	44.32
	PQ*	**w**	90.11	92.32	55.10	68.35	49.97
	PMF	**u**	90.51	92.73	61.52	71.85	51.00
	PMF*	**u**	91.40	93.24	**64.71**	71.56	51.52
	GD-tanh	**w**	91.47	93.27	60.67	71.46	51.43
Ours	MD-softmax	**u**	90.47	91.28	56.25	68.49	46.52
Ours	MD-tanh	**w**	**91.64**	92.97	61.31	72.13	**54.62**
Ours	MD-softmax-S	**u**	91.30	**93.28**	63.97	**72.18**	51.81
Ours	MD-tanh-S	**w**	91.53	93.18	61.69	**72.18**	52.32

Table 2: *Classification accuracies on the test set where all the parameters are binarized.* PQ* *denotes performance with biases, fully-connected layers, and shortcut layers in float (original* PQ *setup) whereas* PQ *represents full quantization.* PMF* *denotes the performance of* PMF *after crossvalidation and the original results from the paper are denoted as* PMF. *Note our* MD *variants obtained accuracies virtually the same as the best performing method and it outperformed the best method by a large margin in much harder TinyImageNet dataset.*

	Algorithm	CIFAR-10	CIFAR-100
	REF (float)	93.67	73.97
	BC	86.84	65.04
Ours	MD-softmax-S	89.71	66.40
Ours	MD-tanh-S	**89.99**	**66.63**

Table 3: *Classification accuracies on the test set for MobileNetV2 where all the parameters are binarized. Our stable* MD *variants significantly outperformed* BC *while* MD-tanh-S *is slightly better than* MD-softmax-S.

state-of-the-art for binarized networks on ImageNet.

For all the experiments, standard multi-class cross-entropy loss is used unless otherwise mentioned. We crossvalidate the hyperparameters such as learning rate, learning rate scale, rate of increase of annealing hyperparameter β, and their respective schedules. We provide the hyperparameter tuning search space and the final hyperparameters in Appendix B. Our algorithm is implemented in PyTorch (Paszke et al. (2017)) and the experiments are performed on NVIDIA Tesla-P100 GPUs. Our PyTorch code is available online[4].

5.1 Full Binarization of Parameters

We evaluate both of our MD variants corresponding to tanh and softmax projections and their numerically stable counterparts as noted in Eq. (12). The results are compared against parameter quantization methods, namely BinaryConnect (BC) (Courbariaux et al. (2015)), ProxQuant (PQ) (Bai et al. (2019)) and Proximal Mean-Field (PMF) (Ajanthan et al. (2019)). In addition, for completeness, we also compare against a standard PGD variant corresponding to the tanh projection (denoted as GD-tanh), *i.e.*, minimizing $f(\tanh(\tilde{\mathbf{x}}))$ using gradient descent. Note that, numerous techniques have emerged with BC as the workhorse algorithm by relaxing constraints such as the layer-wise scalars (Rastegari et al. (2016)), and similar extensions are straightforward even in our case though our variants perform well even without using layer-wise scalars.

The classification accuracies of binary networks obtained by both variants of our algorithm, namely, MD-tanh and MD-softmax, their numerically stable versions (denoted with suffix "-S") and the baselines BC, PQ, PMF, GD-tanh and the floating point Reference Network (REF) are reported in Table 2. Both the numerically stable MD variants consistently produce better or on par results compared to other binarization methods while narrowing the performance gap between binary networks and floating point counterparts to a large extent, on multiple datasets.

Our stable MD-variant perform slightly better than MD-softmax, whereas for tanh, MD updates either perform on par or sometimes even better than numerically stable version of MD-tanh. We believe, the main reason for this empirical variation in results for our MD variants is due to numerical instability caused by the floating-point arithmetic of logarithm and exponential functions in update steps for MD (refer to Table 1). Furthermore, even though our two MD-variants, namely MD-softmax and MD-tanh optimize in different spaces, their performance is similar in most cases.

Note, PQ (Bai et al. (2019)) does not quantize the fully-connected layers, biases, and shortcut layers. For fair comparison as previously mentioned, we crossvalidate PQ with all layers binarized and original PQ settings,

[4] `https://github.com/kartikgupta-at-anu/md-bnn`

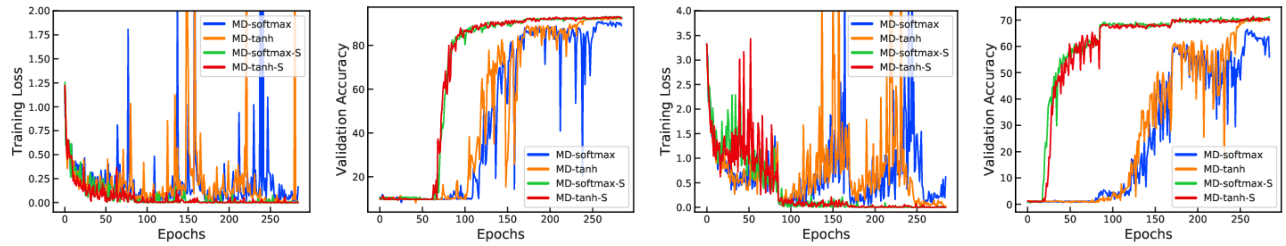

Figure 3: *Training curves for binarization for CIFAR-10 (first two) and CIFAR-100 (last two) with ResNet-18. Compared to original* MD *variants, stable* MD *variants are less noisy and after the initial exploration phase (up to* 60 *in CIFAR-10 and* 25 *epochs CIFAR-100), the validation accuracies rise sharply and show gradual improvement afterwards.*

and report the results denoted as PQ and PQ* respectively in Table 2. Our MD-variants outperform PQ consistently on multiple datasets in equivalent experimental settings. This clearly shows that entropic or tanh-based regularization with our annealing scheme is superior to a simple "W" shaped regularizer and emphasizes that MD is a suitable framework for quantization. Furthermore, the superior performance of MD-tanh against GD-tanh and on par or better performance of MD-softmax against PMF for binary quantization empirically validates that MD is useful even in a nonconvex stochastic setting. This hypothesis along with our numerically stable form of MD can be particularly useful to explore other projections that are useful for quantization and/or network compression in general.

The training curves for our MD variants for CIFAR-10 and CIFAR-100 datasets with ResNet-18 are shown in Fig. 3. The original MD variants show unstable behaviour during training. Regardless, by storing auxiliary variables, the MD updates are demonstrated to be quite stable. This clear distinction between MD variants emphasizes the significance of practical considerations while implementing MD especially in NN optimization.

To further demonstrate the superiority of MD, we tested on a more resource efficient MobileNetV2 (Sandler et al. (2018)) and the results are summarized in Table 3. In short, our MD variants are able to fully-binarize MobileNetV2 with minimal loss in accuracy on CIFAR datasets. For more experiments such as training curves comparison to other methods and ternary quantization results please refer to the Appendix B.

5.2 Binarization on ImageNet

We evaluated our MD-tanh-S against state-of-the-art methods on ImageNet with ResNet-18 for parameter binarization and the results are reported in Table 4. Following the standard practice, we do not quantize the first convolution layer, last fully-connected layer, biases, and batchnorm parameters for all the compared methods and in this case, we set a *new state-of-the-art for binarization* with achieving merely $< \mathbf{4}\%$ reduction compared to the floating-point network. Note that, the standard practice to quantize on ImageNet is to use floating-point scalars in each layer (Rastegari et al. (2016); Yang et al. (2019)), however, our method outperforms all the methods without requiring layerwise scalars. In addition, MD-tanh-S yields the best results even when trained from scratch with $< \mathbf{1}\%$ reduction compared to finetuning from a pretrained network. Furthermore, MD enables the training of fully-binarized networks with no additional scalars (except the batchnorm parameters) from scratch, which is considered to be difficult for ImageNet (Rastegari et al. (2016)). More ablation study experiments can be found in Appendix B.

Similar to the above, the results for both parameters

	Algorithm	Top-1	Top-5
	REF (float)	70.61	89.40
	BWN* (Rastegari et al. (2016))	60.80	83.00
	BR (Yin et al. (2018))	63.20	85.10
	ELQ (Zhou et al. (2018))	64.72	86.04
	ADMM (Leng et al. (2018))	64.80	86.20
	QN (Yang et al. (2019))	66.50	**87.03**
Ours	MD-tanh-S*$^{+}$	56.67	79.66
	MD-tanh-S*	65.92	86.29
	MD-tanh-S	**66.78**	87.01

Table 4: *ImageNet classification accuracies for binary quantization (only parameters) on ResNet-18. Here, * indicates training from scratch and* $^{+}$ *indicates full-binarization except the batchnorm parameters. Note* MD-tanh-S *outperforms all other methods setting new state-of-the-art on ImageNet binarization. It might seem that the improvement over* QN *is marginal, however,* QN *requires layerwise scalars and pretraining (Yang et al. (2019)), whereas* MD-tanh-S *does not require layerwise scaling and obtains near state-of-the-art results even without pretraining.*

	Algorithm	Top-1	Top-5
	REF (float)	70.6	89.4
	BinaryNet (Hubara et al. (2016))	42.2	67.1
	Dorefa-Net (Zhou et al. (2016))	52.5	76.7
	XNOR-Net (Rastegari et al. (2016))	51.2	73.2
	Bireal-Net (Liu et al. (2018))	56.4	79.5
	Bireal-Net (Liu et al. (2018)) (PReLU)	59.0	81.3
	PCNN (J=1) (Gu et al. (2019))	57.3	80.0
	QN (Yang et al. (2019))	53.6	75.3
	BOP (Helwegen et al. (2019))	54.2	77.2
	GBCN (Liu et al. (2019))	57.8	80.9
	IR-Net (Qin et al. (2020))	58.1	80.0
	Noisy Supervision (Han et al. (2020))	59.4	81.7
Ours	MD-tanh-S	60.3	82.3
Ours	MD-tanh-S (KL div. loss)	**62.8**	**84.3**

Table 5: *ImageNet classification accuracies for binary quantization (both parameters and activations) on ResNet-18. Here, our* MD-tanh-S *and Han et al. (2020) use Bireal-Net-18 (Liu et al. (2018)) with PReLU activations as baseline architecture. We also show results of our method with* KL *divergence loss between softmax output of our binary network and* REF *(trained ResNet-34 on ImageNet). Note,* MD-tanh-S *clearly outperforms all the methods.*

and activations binarized networks are reported in Table 5. For this experiment, we use Bireal-Net-18 (Liu et al. (2018)) with PReLU activations as network architecture for our proposed algorithm. Even in this experiment, the first convolution layer, last fully-connected layer, biases, and batchnorm parameters are not binarized for all the methods. Our MD-tanh-S achieves state-of-the-art results even when both parameters and activations are binarized, beating all the comparable baselines by almost **1**%. In addition, similar to Zhuang et al. (2018); Martinez et al. (2019); Liu et al. (2020), we ran MD-tanh-S with KL Divergence loss (replacing the cross-entropy loss) between the softmax output of our binary network and REF (trained ResNet-34 on ImageNet). Our MD-tanh-S with KL divergence loss outperforms previous methods by a significant margin of $> \mathbf{3}\%$, which clearly reflects the efficacy of MD.

6 Discussion

In this work, we have introduced an MD framework for NN quantization by deriving mirror maps corresponding to two projections useful for quantization and provided two algorithms for quantization. Theoretically, we provided a convergence analysis in the convex setting for time-varying mirror maps and discussed conditions to ensure convergence to a discrete solution when an annealing hyperparameter is employed. In addition, we have discussed a numerically stable implementation of MD by storing an additional set of auxiliary variables and showed that this update is strikingly analogous to the popular STE based gradient method. The superior performance of our MD formulation even with simple projections such as tanh and softmax is encouraging and we believe, MD would be a suitable framework for not just NN quantization but for network compression in general. In the future, we intend to focus more on the theoretical aspects of MD in conjunction with stochastic momentum based optimizers such as Adam.

7 Acknowledgements

This work was supported by the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1, EPSRC/MURI grant EP/N019474/1, and the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016). We would like to thank Pawan Kumar for useful discussions on mirror descent and we acknowledge the Royal Academy of Engineering, FiveAI, Data61, CSIRO, and National Computing Infrastructure, Australia.

References

Achterhold, J., Kohler, J. M., Schmeink, A., and Genewein, T. (2018). Variational network quantization. *ICLR*.

Ajanthan, T., Dokania, P. K., Hartley, R., and Torr, P. H. (2019). Proximal mean-field for neural network quantization. *ICCV*.

Bai, Y., Wang, Y.-X., and Liberty, E. (2019). Proxquant: Quantized neural networks via proximal operators. *ICLR*.

Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*.

Boyd, S. and Vandenberghe, L. (2009). *Convex optimization*. Cambridge university press.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*.

Carreira-Perpinán, M. A. and Idelbayev, Y. (2017). Model compression as constrained optimization, with application to neural nets. part ii: Quantization. *NeurIPS Workshop on Optimization for Machine Learning*.

Chen, S., Wang, W., and Pan, S. J. (2019). MetaQuant: learning to quantize by learning to penetrate non-differentiable quantization. *NeurIPS*.

Courbariaux, M., Bengio, Y., and David, J.-P. (2015). Binaryconnect: Training deep neural networks with binary weights during propagations. *NeurIPS*.

Esser, S. K., Appuswamy, R., Merolla, P. A., Arthur, J. V., and Modha, D. S. (2015). Backpropagation for energy-efficient neuromorphic computing. *NeurIPS*.

Gu, J., Li, C., Zhang, B., Han, J., Cao, X., Liu, J., and Doermann, D. (2019). Projection convolutional neural networks for 1-bit cnns via discrete back propagation. *AAAI*.

Guo, Y. (2018). A survey on methods and theories of quantized neural networks. *arXiv preprint arXiv:1808.04752*.

Han, K., Wang, Y., Xu, Y., Xu, C., Wu, E., and Xu, C. (2020). Training binary neural networks through learning with noisy supervision. *ICML*.

Helwegen, K., Widdicombe, J., Geiger, L., Liu, Z., Cheng, K.-T., and Nusselder, R. (2019). Latent weights do not exist: Rethinking binarized neural network optimization. *NeurIPS*.

Hou, L., Yao, Q., and Kwok, J. T. (2017). Loss-aware binarization of deep networks. *ICLR*.

Hsieh, Y.-P., Kavis, A., Rolland, P., and Cevher, V. (2018). Mirrored langevin dynamics. *NeurIPS*.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks. *NeurIPS*.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. *JMLR*.

Leng, C., Dou, Z., Li, H., Zhu, S., and Jin, R. (2018). Extremely low bit neural network: Squeeze the last bit out with ADMM. *AAAI*.

Li, H., De, S., Xu, Z., Studer, C., Samet, H., and Goldstein, T. (2017). Training quantized nets: A deeper understanding. *NeurIPS*.

Liu, C., Ding, W., Hu, Y., Zhang, B., Liu, J., and Guo, G. (2019). Gbcns: Genetic binary convolutional networks for enhancing the performance of 1-bit dcnns. *CoRR*.

Liu, Z., Shen, Z., Savvides, M., and Cheng, K.-T. (2020). Reactnet: Towards precise binary neural network with generalized activation functions. *arXiv preprint arXiv:2003.03488*.

Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., and Cheng, K.-T. (2018). Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. *ECCV*.

Louizos, C., Reisser, M., Blankevoort, T., Gavves, E., and Welling, M. (2019). Relaxed quantization for discretized neural networks. *ICLR*.

Louizos, C., Ullrich, K., and Welling, M. (2017). Bayesian compression for deep learning. *NeurIPS*.

Martinez, B., Yang, J., Bulat, A., and Tzimiropoulos, G. (2019). Training binary neural networks with real-to-binary convolutions. In *International Conference on Learning Representations*.

McMahan, H. B. (2017). A survey of algorithms and analysis for adaptive online learning. *JMLR*.

Nemirovsky, A. S. and Yudin, D. B. (1983). Problem complexity and method efficiency in optimization.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch.

Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., and Song, J. (2020). Forward and backward information retention for accurate binary neural networks. *CVPR*.

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. *ECCV*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: inverted residuals and linear bottlenecks. *CVPR*.

Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., and Hua, X.-s. (2019). Quantization networks. *CVPR*.

Ye, S., Feng, X., Zhang, T., Ma, X., Lin, S., Li, Z., Xu, K., Wen, W., Liu, S., Tang, J., et al. (2019). Progressive dnn compression: A key to achieve ultra-high weight pruning and quantization rates using admm. *arXiv preprint arXiv:1903.09769*.

Yin, P., Lyu, J., Zhang, S., Osher, S., Qi, Y., and Xin, J. (2019). Understanding straight-through estimator in training activation quantized neural nets. *ICLR*.

Yin, P., Zhang, S., Lyu, J., Osher, S., Qi, Y., and Xin, J. (2018). Binaryrelax: A relaxation approach for training deep neural networks with quantized weights. *SIIMS*.

Zhang, S. and He, N. (2018). On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization. *CoRR*.

Zhou, A., Yao, A., Wang, K., and Chen, Y. (2018). Explicit loss-error-aware quantization for low-bit deep neural networks. *CVPR*.

Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., and Zou, Y. (2016). Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*.

Zhou, Z., Mertikopoulos, P., Bambos, N., Boyd, S., and Glynn, P. (2017a). Mirror descent in non-convex stochastic programming. *CoRR*.

Zhou, Z., Mertikopoulos, P., Bambos, N., Boyd, S., and Glynn, P. W. (2017b). Stochastic mirror descent in variationally coherent optimization problems. *NeurIPS*.

Zhuang, B., Shen, C., Tan, M., Liu, L., and Reid, I. (2018). Towards effective low-bitwidth convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7920–7928.