ConQ: Binary Quantization of Neural Networks via Concave Regularization

2024 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2024, LONDON, UK

CONQ: BINARY QUANTIZATION OF NEURAL NETWORKS VIA CONCAVE REGULARIZATION

Andrea Migliorati[⋆]*, Giulia Fracastoro*[⋆]*, Sophie Fosson*[†]*, Tiziano Bianchi*[⋆]*, Enrico Magli*[⋆]

[⋆] Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy

[†] Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy

ABSTRACT

The increasing demand for deep neural networks (DNNs) in resource-constrained systems propels the interest in heavily quantized architectures such as networks with binarized weights. However, despite huge progress in the field, the gap with full-precision performance is far from closed. Today's most effective methods for quantization are rooted in proximal gradient descent theory. In this work, we propose ConQ, a novel concave regularization approach to train effective DNNs with binarized weights. Motivated by theoretical investigation, we argue that the proposed concave regularizer, which allows the removal of the singularity point at 0, presents a more effective shape than previously considered models in terms of accuracy and convergence rate. We present a theoretical convergence analysis of ConQ, with specific insights on both convex and non-convex settings. An extensive experimental evaluation shows that ConQ outperforms the accuracy of competing regularization methods for networks with binarized weights.

***Index Terms*—** Concave Regularization, Quantized Neural Networks, Proximal Operators

1. INTRODUCTION

Deep neural networks (DNNs) have shown remarkable performance in diverse machine learning tasks in the computer vision field. Typically, high-performance DNNs have a huge amount of parameters, leading to significant memory usage and computational cost. However, these networks often need to be used in environments with limited memory and computational resources such as mobile devices or embedded systems. In such cases, it is desirable to compress the network into a smaller and faster version while maintaining comparable inference accuracy. In recent years, several methods employed quantization of the network parameters from floating-point 32-bit representation to half-precision (16 bits), UINT8 (8 bits), down to binary networks where parameters can only assume two values.

In binarized neural networks, parameters are quantized via a function Q to $+1/-1$ levels leading to a significant reduction of memory usage and computational complexity compared to full-precision (FP). The most straightforward solution for Q is the sign function, as introduced in BinaryConnect [1] which proposed a workaround for computing gradients in the back-propagation phase while training, despite the non-differentiability of the sign function. The so-called Straight-Through Estimator (STE) method approximates the derivative of the sign function with the identity, allowing the gradients to pass through unchanged. In such a way, the derivatives of FP parameters are used for updating the quantized parameters. STE greatly improves the performance compared to simple post-training quantization of a pre-trained network. However, the gap with FP is far from being closed. One of the possible reasons might be found in the intrinsic sub-optimality of STE, as already observed in [2]. Only a few methods proposing variations to STE have been published [3, 4].

As mentioned, a class of regularization methods to train models with binarized parameters has recently been introduced [2, 3, 4]. Specifically, one can induce effective quantization during the training by adding a suitable regularization term to the loss function. This is achieved by designing suitable proximal gradient algorithms (PGAs) for composite, regularized problems. In the pivotal paper [2], authors introduce a W-shaped regularizer and develop its corresponding PGA. Further theoretical analysis and modified proximal methods were proposed in [5, 6, 7]. Our paper introduces a novel concave regularization method for binary quantization denoted as *ConQ* which improves upon the state of the art thanks to its concave shape which allows the removal of the singularity point at 0. Our contributions are summarized as follows: *(i)* we introduce a new regularizer with a specific non-convex shape to train models with parameters binarized with $-1/+1$

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU with a partnership on "Telecommunications of the Future" (PE00000001 "RESTART" program). This study was also carried out within the Future Artificial Intelligence Research (FAIR) and received funding from the European Union Next-GenerationEU (PNRR. Piano Nazionale di Ripresa e Resilienza, Missione 4, Componente 2, Investimento 1.3 - D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the EU nor the European Commission can be considered responsible for them. The contribution of Sophie Fosson is part of the project NODES which has received funding from the MUR-M4C2 1.5 of PNRR with grant agreement n. ECS00000036.

979-8-3503-2411-2/23/$31.00 ©2024 IEEE

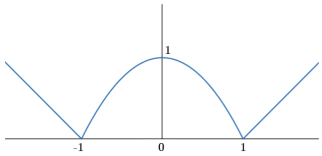

Fig. 1. The shape of the proposed regularizer r.

binarization levels; *(ii)* we design a PGA and analyze its convergence; our regularizer yields competitive results relative to state-of-the-art methods in terms of accuracy and convergence rate in the convex setting; *(iii)* we demonstrate the effectiveness of ConQ validating it on the image classification task with different deep network structures, showing that ConQ outperforms regularization competitors on the considered scenarios.

2. BACKGROUND

As mentioned, the binarization of deep models was introduced in [1]. Several works have been proposed to tackle the performance drop when compared to their FP counterpart [8, 9], and to provide theoretical insights into the guarantees under which one can efficiently train binarized models [10, 11, 12]. On the other hand, only a handful of quantization algorithms based on regularization or proximal methods were introduced as alternatives to STE. [13] introduced BinaryRelax, representing a form of lazy proximal gradient descent, while [14, 15] proposed a proximal Newton method employing an approximate diagonal Hessian. [16] formulated the training of quantized networks as a constrained optimization challenge, suggesting solutions using augmented Lagrangian methods. The ProxQuant method [2] introduced a W-shaped regularizer and analyzed its corresponding PGA, which showed improvements over the state of the art as reported in their experimental assessment. Further, ProxSGD [5] and ProxGen [6] analyzed the general problem of stochastic proximal gradient descent with convex and nonconvex regularization. In numerical experiments, ProxSGD and ProxGen were shown to improve over BinaryConnect [1], while, in some instances, ProxGen also exhibits greater accuracy than ProxQuant. Finally, in [7], BinaryConnect was reinterpreted as a dual-averaging lazy proximal gradient algorithm, giving rise to original theoretical considerations on its convergence. Here, the authors propose ProxConnect, a family of lazy proximal gradient algorithms whose performance is close to ProxQuant in several experiments on different binarized models. In [17], ProxConnect was revisited with different forward-backward quantizers.

3. PROPOSED APPROACH

In this section, we introduce our method by stating the practical and theoretical motivations that originate it. As discussed, ProxQuant [2] represents the state of the art for binarized models with regularization and proximal approach. Its key idea is the use of a W-shaped regularization defined by $\sum_{i=1}^{d} \min\{|1-x_i|, |1+x_i|\} = \|x - \text{sign}(x)\|_1$ (where $\text{sign}(x) = 1$ for $x \geq 0$ and -1 otherwise). With minima at -1 and $+1$, the function is designed to encourage the network parameters to take values around -1 and $+1$ after the typical initialization with values close to zero. This regularizer has three non-differentiable points in $\{-1, 0, 1\}$. We argue that a strictly concave shape in $[-1, 1]$ as depicted in Fig. 1 leads to a more accurate quantization. On the one hand, a smooth maximum at 0 does not force excessive penalization of a value close to zero, which instead may occur with an angular point as in [2]. In other words, by increasing the concavity, we can build a region of points with a small gradient around zero to avoid hard decisions. On the other hand, a large concavity is expected to enhance the overall convergence rate, as it moves the parameters closer to -1 and $+1$ where the slope of the regularizer is larger than 1. We remark that we keep a linear regularizer outside $[-1, 1]$, as values that are large in magnitude are less critical to quantize. Also, we point out that, by iterating the shape in $[-1, 1]$, the proposed approach could be potentially extended to quantization problems at bit depths greater than one.

The simplest way to realize a concave regularization is to consider a quadratic function $1 - x^2$ in the $[-1, +1]$ interval (Fig. 1). Although higher-order polynomials may be considered for more efficient quantization, the quadratic case has been shown to provide clear benefits compared to the linear case while at the same time allowing the theoretical analysis to be affordable. These benefits also apply against a logarithmic shape which exhibits a non-differentiable point at zero. In the following, we specialize in quadratic regularization and illustrate the convergence rate benefits compared to ProxQuant.

3.1. ConQ method

Our technique is based on concave quadratic regularization associated with PGA. Given a loss function $L : \mathbb{R}^d \mapsto \mathbb{R}$, we solve

$$\min F(x) = L(x) + \lambda R(x) \tag{1}$$

$$R(x) = \sum_{i=1}^{d} r(x_i) \tag{2}$$

where $r : \mathbb{R} \mapsto \mathbb{R}$ is defined as

$$r(x) = \max\left\{1 - x^2, |x| - 1\right\} \tag{3}$$

and $\lambda > 0$ is the hyper-parameter that weights the regularization effect. The shape of the r regularizer is shown in Fig. 1. Since r is not differentiable in the quantization values, we cannot directly use gradient-based algorithms for the problem as in Eq. (1). However, we can resort to the proximal gradient algorithm which is at the core of the ConQ method.

The proximal operator of a function $G : \mathbb{R}^d \mapsto \mathbb{R}$ is defined as:

$$\text{prox}_G(z) = \underset{x\in\mathbb{R}^d}{\text{argmin}} \left\{ \frac{1}{2}\|x - z\|_2^2 + G(x) \right\}. \quad (4)$$

Let us assume that L is differentiable. The PGA for F consists of iterating a gradient step on L and a proximal operation:

$$x_{t+1} = \text{prox}_{\lambda\tau R}(x_t - \tau\nabla L(x_t)) \quad (5)$$

where $\tau > 0$ is the learning rate. The effective application of PGA depends on the possibility of computing $\text{prox}_{\lambda\tau R}$ straightforwardly. Generally, this may be challenging for non-convex, non-smooth functions such as the one we employ. Regularizers that support binary quantization over $\{-1, 1\}$ are necessarily non-convex, because they naturally have strict global minima in $\{-1, 1\}$, and non-smooth because angular points in $\{-1, 1\}$ are fundamental to induce concentration of parameter values in $\{-1, 1\}$. However, the problem defined in Eq. (4) is strongly convex for the proposed R and enjoys a simple unique solution. Since R is separable, the computation of the proximal operator is reduced to the following one-dimensional problem:

$$\text{prox}_{\lambda\tau r}(z) = \underset{x\in\mathbb{R}}{\text{argmin}} \left\{ \frac{1}{2}(x - z)^2 + \lambda\tau r(x) \right\} \quad (6)$$

If $\lambda\tau \in \left(0, \frac{1}{2}\right)$, the problem as in Eq. (6) is strongly convex and the solution is:

$$\text{prox}_{\lambda\tau r}(z) = \begin{cases} \frac{z}{1-2\lambda\tau} & \text{if } |z| < 1 - 2\lambda\tau \\ \text{sign}(z) & \text{if } |z| \in [1 - 2\lambda\tau, 1 + \lambda\tau] \\ z - \text{sign}(z)\lambda\tau & \text{if } |z| > 1 + \lambda\tau. \end{cases}$$

We summarize the training algorithm in Alg. 1, where $\tilde{\nabla}$ refers to an SGD optimizer.

Algorithm 1 ConQ: proximal gradient method with concave regularization for binary quantization

Require: Initialization x_0, learning rate τ, regularization weight λ, pre-trained FP model M
while Not converged **do**
Forward pass on M (normal SGD)
Backward pass on M (normal SGD): computation of $z_t = x_t - \tau\tilde{\nabla}L(x_t)$
Update the parameters with the proximal gradient step: $x_{t+1} = \text{prox}_{\lambda\tau R}(z_t)$
end while
Quantize the parameters of the regularized model with the sign function

4. CONVERGENCE ANALYSIS

We now analyze the convergence of ConQ in a non-stochastic setting and further discuss the differences with ProxQuant.

4.1. Convergence in the Non-convex Setting

We start by analyzing the convergence of ConQ for non-convex $F = L + \lambda R$. Since R is non-convex, L may be either non-convex or convex. We first consider the general non-convex case, while in Sec. 4.3 we specialize to strongly convex L. To prove the convergence of ConQ, we leverage results from composite, non-convex, non-smooth optimization theory in [18, 19]. These works study the convergence of a family of descent algorithms, including PGA, by leveraging the Kurdyka-Łojasiewicz (KŁ) property defined in [18, 19]. Let $\partial F(x)$ be the limiting subdifferential of F at x. A minimizer $x^\star$ of F necessarily satisfies $0 \in \partial F(x^\star)$. Conversely, any point x that satisfies $0 \in \partial F(x)$ is said to be a critical point. The definition of KŁ property as in [19] requires that a reparametrization of the values of F exists, such that singular regions, i.e., regions where the distance between 0 and ∂F is arbitrarily small, can be turned into regions where the distance between 0 and ∂F is large (we refer to [20] for further details). We now state the convergence theorem that leverages the KŁ property, and then illustrate the family of functions enjoying it, with specific application to our binary quantization problem.

Theorem 4.1. *Let $F = L + R : \mathbb{R}^d \mapsto \mathbb{R} \cup \{+\infty\}$ be a proper, lower semi-continuous, bounded from below, Kurdyka-Łojasiewicz function. Moreover, let $L : \mathbb{R}^d \mapsto \mathbb{R}$ be differentiable and β-smooth, and R be continuous. Then, any bounded sequence $(x_t)_{t\in\mathbb{N}}$ generated by the proximal gradient algorithm as in Eq.* (5) *converges to a critical point of F. Moreover, as $\sum_k \|x_{t+1} - x_t\|_2 < +\infty$, then $(x_t)_{t\in\mathbb{N}}$ is convergent.*

Proof. This result is a direct consequence of Theorem 5.1 in [19] applied to non-convex, non-smooth, forward-backward splitting algorithms. In particular, the proximal gradient algorithm fits into the structure of Algorithm 3 defined in Section 5.1 of [19]. □

A large class of non-smooth functions enjoys the KŁ property required by Theorem 4.1. As shown by [21], semi-algebraic and subanalytic functions can be considered KŁ functions. Indeed, the considered R is semi-algebraic because it is the union of polynomials. Since the composition of analytic functions is analytic, L would be analytic if we e.g. used the analytic SoftPlus activation function. In this case, F amounts to the sum of an analytic and a semi-algebraic function, hence is sub-analytic. In conclusion, F is KŁ under reasonable assumptions. Concerning the convergence rate of the algorithm, [18, 22] state that, if the KŁ property is verified with some specific reparametrization functions, then convergence can be sublinear, linear, or composed of a finite number of steps. In our setting, since R is sharp at its minimum, the reparametrization function mainly depends on the loss function L. Theorem 4.1, which also holds for ProxQuant as its W-shaped regularizer is semi-algebraic, is a stronger result

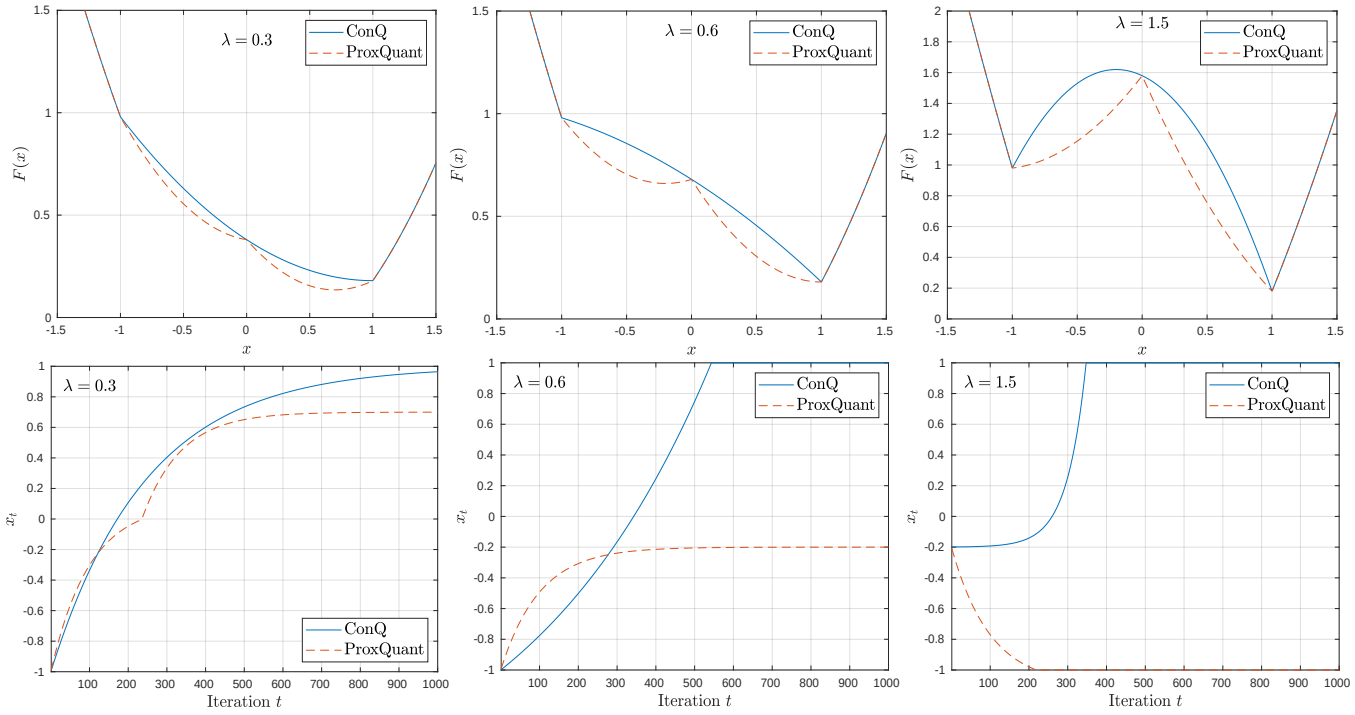

Fig. 2. Example: $L = \frac{1}{2}(x-\alpha)^2$, $\alpha = 0.4$, $\lambda \in \{0.3, 0.6, 1.5\}$. We depict $F = L + \lambda R$ (*top*) and the quantization output (*bottom*) for the proposed ConQ and ProxQuant.

than Theorem 5.1 in [2] which instead only considers convergence in an ergodic sense and assumes that a differentiable function approximates R. Specifically, smoothing R might compromise the quantization-promoting effect, marking a gap between theory and practice.

4.2. Differences between ConQ and ProxQuant

Let L be μ-strongly convex, i.e., $L(x) - \frac{1}{2}\mu\|x\|_2^2$ is convex. In this setting, the differences between ConQ and ProxQuant are evident in terms of convergence rate and accuracy, as shown in the one-dimensional toy example in Fig. 2. Here we depict $F = L + \lambda R$ (*top*), where R indicates the employed regularizer, and its quantization output (*bottom*) for the proposed ConQ and ProxQuant. Specifically, let $L = \frac{1}{2}(x-\alpha)^2$, where $x \in \mathbb{R}$ and $\alpha > 0$. The chosen L is 1-strongly convex and the minimizer $x = \alpha$ is quantized to $+1$. We apply ConQ and ProxQuant to showcase their different behavior in converging to the correct quantized solution for $\alpha = 0.4$, $\tau = 10^{-2}$ and three different values of λ as in $\lambda \in \{0.3, 0.6, 1.5\}$.

In the first case ($\lambda = 0.3$) ConQ exhibits a strongly convex $F(x)$ and converges to the right solution for any x_0, with a global linear convergence rate. In the same setting, $F(x)$ for ProxQuant has a unique minimum, but its convergence is not overall linear and substantially slows down before overpassing zero, which might be critical. Indeed, if we stop the algorithm at iteration $t = 200$ and binarize the current x_t, only ConQ provides the correct quantized value. In the second case ($\lambda = 0.6$), $F(x)$ is non-convex for ConQ, but with a unique minimizer. For any x_0, ConQ converges to the right solution in a finite number of steps as expected from [22]. In contrast, $F(x)$ for ProxQuant has a local minimum at $\alpha - \lambda < 0$, which yields an incorrect quantization. One can easily prove that ProxQuant converges to it whenever $x_0 < -\frac{\tau\alpha}{1-\tau} = -0.004$. Finally, in the third case ($\lambda = 1.5$), $F(x)$ for ConQ or ProxQuant has two minima at the quantization values. ConQ converges to the correct solution whenever x_0 is at the right of the negative minimizer, namely $x_0 > \frac{\alpha}{1-2\lambda}$, while ProxQuant converges whenever $x_0 > -\frac{\tau\alpha}{1-\tau}$. Hence, ConQ has a larger region of attraction to the correct solution. To sum up, ProxQuant exhibits drawbacks compared to our ConQ. Firstly, even if F has a unique minimum, ProxQuant has no global linear convergence rate guaranteed. Secondly, when more minima occur, ProxQuant has a smaller region of attraction towards the optimal quantization, affecting overall accuracy. Beyond W-shaped regularization, non-convexity issues occur whenever a regularizer has a non-differentiable point at 0. For example, this is the case of ℓ_q regularization with $q \in (0,1)$, defined by $r(x) = (|x - \text{sign}(x)|)^q$ and used in [6].

4.3. Convergence for Strongly Convex Loss

In this section, we analyze the convergence of ConQ for strongly convex L. In particular, we provide conditions on λ such that ConQ is Q-linearly convergent. As illustrated in Sec. 4.2, the convex analysis cannot be performed for ProxQuant and other similar regularizers as they are non-differentiable at 0 [2, 6].

Theorem 4.2. *Let us consider the minimization of $F = L + \lambda R$, where L is μ-strongly convex and β-smooth, and R is defined by Eqs.* (2)-(3). *Let $\mu \geq 2\lambda + \epsilon$ for some $\epsilon > 0$. Then, it is possible to design a PGA that enjoys a Q-linear convergence, i.e., it generates a sequence $(x_t)_{t\in\mathbb{N}}$, such that:*

$$\|x_{t+1} - x^\star\|_2 \leq c\|x_t - x^\star\|_2, \tag{7}$$

where $x^\star$ is the global minimizer and $c \in (0, 1)$. L is said to be β-smooth if its gradient is β-Lipschitz.

5. EXPERIMENTAL EVALUATION

We evaluate the performance of our method on the image classification task by testing different network structures on the CIFAR-10 dataset.

5.1. Datasets, Models, and Setup

We report our classification performance on the CIFAR-10 dataset on which we employ standard normalization and augmentation techniques. We test on progressively larger models such as ResNets [23] with different depths and VGG-Small [24]. The training pipeline is composed as follows. First, we train a standard FP model for 200 epochs with no proximal regularization. Secondly, we introduce the proposed proximal update and fine-tune for other 200 epochs. Finally, the network is quantized and trained for further 100 epochs with frozen gradients to stabilize the batch normalization layers, following the procedure in [2]. We train with the Adam optimizer [25] with an initial learning rate set to 0.01, momentum 0.9, and employ learning rate decay as in [1]. We set $\lambda = 10^{-4}$ for all experiments, where λ is the hyperparameter that weights the regularization effect. Specifically, we empirically found that, for the considered models and datasets, $\lambda = 10^{-4}$ leads to the best performance. We hypothesize the chosen λ is the best for ensuring a smooth regularization that does not introduce instability in the training, while steadily pushing the parameters toward the $-1/+1$ quantization levels. All models are implemented in PyTorch [26] and run on NVIDIA GeForce GTX Titan X GPUs.

5.2. Results

Table 1 reports a detailed comparison against the proximal methods ProxQuant [2] and ProxGen [6] obtained by exactly replicating their experimental setup. Specifically, ProxGen is a family of methods including a revised ProxQuant that considers preconditioners in the proximal operation. As reported in [6], ProxGen improves ProxQuant for large networks, with slight differences with different regularizer variants, obtained by modifying the ℓ_1-based W-shape to ℓ_q, $q \in (0, 1)$-based shapes. Table 1 shows that proposed ConQ outperforms both ProxQuant and ProxGen on ResNet20, ResNet32, ResNet44, and ResNet56, demonstrating the superiority of the shape of the proposed concave regularizer compared to the ℓ_1, W-shaped and ℓ_q shaped regularizers used in [2, 6]. Furthermore, we compare on CIFAR10 with other state-of-the-art methods such as BinaryConnect [1], DoReFa [27], LQ-Nets [24], IR-Net [28], DSQ [29], BayesBiNN [31], MD [32], and

Model	Baseline	ProxQuant	ProxGen	ConQ
ResNet20	91.94	90.65	90.50	**91.41**
ResNet32	92.75	91.47	91.78	**92.19**
ResNet44	93.04	92.05	92.32	**92.53**
ResNet56	93.46	92.30	92.48	**92.65**

Table 1. Performance comparison (% test accuracy) with ProxQuant [2] and ProxGen [6] on the CIFAR-10 dataset. Results are averaged over 4 runs for ProxQuant and 10 runs for ConQ.

Model	Method	W/A	Acc. (%)
ResNet20	FP	32/32	91.94
	DoReFa [27]	1/32	90.0
	LQ-Nets [24]		90.1
	IR-Net [28]		90.2
	DSQ [29]		90.2
	ProxGen [6]		90.5
	ProxQuant [2]		90.65
	ConQ (10 runs avg.)		**91.4**
VGG-Small	FP	32/32	94.1
	BinaryConnect [1]	1/32	89.75
	LAB [14]		89.5
	BWN [30]		90.1
	ProxQuant [2]		90.11
	BayesBiNN [31]		90.68
	MD-softmax-s [32]		91.3
	MD-tanh-s [32]		91.4
	PMF [33]		91.4
	ConQ (10 runs avg.)		**92.2**
ResNet18	FP	32/32	94.8
	BinaryConnect [1]	1/32	91.92
	BayesBiNN [31]		92.28
	ProxQuant [2]		92.32
	MD-softmax [32]		91.28
	MD-softmax-s [32]		93.1
	ConQ (10 runs avg.)		**93.1**

Table 2. Performance comparison (% test accuracy) on the CIFAR-10 dataset with competing techniques.

PMF [33]. ConQ improves over competing methods with the ResNet20 architecture and with bigger models such as VGG-Small and ResNet18, which exhibit respectively $4.66M$ and $11M$ parameters. Specifically, Table 2 shows that ConQ is competitive with other techniques outperforming the majority of them, only tying with [32] on the ResNet18 architecture. Note that the accuracy for our method has been obtained by averaging the maximum test accuracy values over 10 different runs to remove training variability.

6. CONCLUSIONS

In this work, we proposed a proximal gradient method to train binary quantized neural networks called ConQ. Unlike other state-of-the-art regularization methods for binary neural networks, our regularizer is smooth in 0. Removing the singularity at 0 is the key to limiting the non-convexity of the problem thus improving the accuracy and the convergence speed. In the case of a strongly convex loss function, we proved that ConQ enjoys a linear convergence rate, differently from W-shaped regularizers. Extensive numerical experiments show that ConQ outperforms state-of-the-art regularization methods for binarized neural networks.

7. REFERENCES

[1] M. Courbariaux et al., "BinaryConnect: Training deep neural networks with binary weights during propagations," *Advances in neural information processing systems*, vol. 28, 2015.

[2] Yu Bai et al., "Proxquant: Quantized neural networks via proximal operators," in *International Conf. on Learning Representations*, 2019.

[3] J. Lee et al., "Network quantization with element-wise gradient scaling," in *Proceedings of the IEEE/CVF conf. on computer vision and pattern recognition*, 2021, pp. 6448–6457.

[4] Louis Leconte et al., "Askewsgd: an annealed interval-constrained optimisation method to train quantized neural networks," in *Int. Conf. on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 3644–3663.

[5] Yang Yang et al., "ProxSGD: Training structured neural networks under regularization and constraints," in *International Conference on Learning Representations (ICLR)*, 2020.

[6] Jihun Yun, Aurelie C Lozano, and Eunho Yang, "Adaptive proximal gradient methods for structured neural networks," in *Advances in Neural Information Processing Systems*, M. Ranzato et al., Eds., 2021, vol. 34, pp. 24365–24378.

[7] Tim Dockhorn et al., "Demystifying and generalizing BinaryConnect," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13202–13216, 2021.

[8] H. Qin et al., "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, pp. 107281, 2020.

[9] Chunyu Yuan and Sos S Agaian, "A comprehensive review of binary neural network," *Artificial Intelligence Review*, pp. 1–65, 2023.

[10] Hao Li et al., "Training quantized nets: A deeper understanding," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] Alexander G. Anderson and Cory P. Berg, "The high-dimensional geometry of binary neural networks," in *International Conference on Learning Representations*, 2018.

[12] Haotong Qin et al., "BiBench: Benchmarking and analyzing network binarization," in *Proceedings of Int. Conf. on Machine Learning*, 2023.

[13] Penghang Yin et al., "Binaryrelax: A relaxation approach for training deep neural networks with quantized weights," *SIAM Journal on Imaging Sciences*, vol. 11, no. 4, pp. 2205–2223, 2018.

[14] Lu Hou et al., "Loss-aware binarization of deep networks," in *International Conf. on Learning Representations*, 2017.

[15] Lu Hou and James T. Kwok, "Loss-aware weight quantization of deep networks," in *International Conf. on Learning Representations*, 2018.

[16] Miguel A Carreira-Perpinán, "Model compression as constrained optimization, with application to neural nets. part i: General framework," *arXiv preprint arXiv:1707.01209*, 2017.

[17] Yiwei Lu et al., "Understanding neural network binarization with forward and backward proximal quantizers," in *Advances in Neural Information Processing Systems*, 2023.

[18] H. Attouch et al., "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Operations Res.*, vol. 35, no. 2, pp. 438–457, 2010.

[19] H. Attouch et al., "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Math. Programm.*, vol. 137, pp. 91–129, 2013.

[20] J. Bolte et al., "Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity," *Trans. Amer. Math. Soc.*, vol. 362, no. 6, pp. 3319–3363, 2010.

[21] K. Kurdyka, "On gradients of functions definable in o-minimal structures," in *Annales de l'institut Fourier*, 1998, vol. 48, pp. 769–783.

[22] P. Frankel et al., "Splitting methods with variable metric for Kurdyka-Łojasiewicz functions and general convergence rates," *Journal of Optimization Theory and Applications*, vol. 165, pp. 874–900, 2015.

[23] Kaiming He et al., "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[24] D. Zhang et al., "Lq-nets: Learned quantization for highly accurate and compact deep neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 365–382.

[25] Diederik P Kingma and Jimmy Ba, "A method for stochastic optimization," in *Int. conf. on Learning Representations*, 2015, vol. 5, p. 6.

[26] Adam Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, pp. 8024–8035. Curran Associates, Inc., 2019.

[27] S. Zhou et al., "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

[28] H. Qin et al., "Forward and backward information retention for accurate binary neural networks," in *Proceedings of the IEEE/CVF conf. on computer vision and pattern recognition*, 2020.

[29] Ruihao Gong et al., "Differentiable soft quantization: Bridging full-precision and low-bit neural networks," in *Proceedings of the IEEE/CVF international conf. on computer vision*, 2019.

[30] M. Rastegari et al., "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.

[31] X. et al Meng, "Training binary neural networks using the bayesian learning rule," in *International conference on machine learning*. PMLR, 2020, pp. 6852–6861.

[32] T. Ajanthan et al., "Mirror descent view for neural network quantization," in *International conference on artificial intelligence and statistics*. PMLR, 2021, pp. 2809–2817.

[33] T. Ajanthan et al., "Proximal mean-field for neural network quantization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4871–4880.