Enhancing Optimization Robustness in 1-bit Neural Networks through Stochastic Sign Descent

Nianhui Guo, Hong Guo, Christoph Meinel, and Haojin Yang

Hasso Plattner Institut, University of Potsdam, Germany
{nianhui.guo,hong.guo,christoph.meinel,haojin.yang}@hpi.de

Abstract. Binary Neural Networks (BNNs) offer a promising avenue toward achieving efficient deep-learning models but are hindered by the inherent challenge of aligning noisy floating-point gradients with binary parameters. To address this, we introduce Diode, a groundbreaking optimizer designed explicitly for BNNs that bridges this gap by utilizing the gradient's sign information in a unique, latent-weight-free approach. By focusing on the gradient sign's lower-order moment estimate for parameter updates, Diode uniformly fine-tunes binary parameters, significantly enhancing model convergence without the dependency on 32-bit latent weights or embedding buffers. This paper showcases Diode's superior performance through comprehensive evaluations on a variety of vision and Natural Language Processing (NLP) tasks. Remarkably, Diode advances the state-of-the-art by increasing BNext-18 Top-1 accuracy on ImageNet ILSVRC2012 by 0.96% with eightfold fewer training iterations. In the case of ReActNet, Diode not only matches but slightly exceeds previous benchmarks without resorting to complex multi-stage optimization strategies, effectively halving the training duration. Additionally, Diode proves its robust generalization capability on the binary BERT architecture within the GLUE benchmark, outperforming the existing BiT design by 3.3% without data augmentation and establishing a new SOTA accuracy of 78.8% with augmentation. The implementation of Diode is available at: https://github.com/GreenBitAI/bitorch-engine.

1 Introduction

Binary Neural Networks (BNNs) represent a promising approach to developing faster and more energy-efficient neural networks by using binary values, -1 or +1, for both weights and activations instead of traditional floating-point values [12]. This technique significantly reduces memory usage by 32 times and theoretically achieves up to 58 times faster computation by replacing floating-point operations with simple XNOR and bit-counting operations. Advances in BNNs, including BNext [11], BiT [19], and BMT [39], have demonstrated their effectiveness across both vision and natural language processing tasks. Given the rapid growth of large neural network model sizes and their computational costs, BNNs are pivotal in exploring scaling laws and mitigating computational expenses [17,27,39]. Notably, recent pre-trained 1-bit LLMs BitNet [36] and BitNet b1.58 [25], are paving the way for a new era of 1-bit Large Language Models (LLMs).

However, optimizing BNNs poses significant challenges due to noisy gradients and the lack of optimization algorithms tailored to their unique properties. Most advanced BNNs rely on accumulating gradients in floating-point "latent weight" buffers to adjust binary weights using traditional algorithms like SGD or Adam [21]. Yet, as identified by [14, 31], these methods are not ideal for BNNs since the concept of "latent weight" does not translate well from its floating-point counterpart. Adjusting the magnitude of these latent weights does not directly affect the binary values, leading to a mismatch in the optimization process. This inefficiency can cause BNNs to settle into suboptimal solutions and face underfitting, making their optimization more complex than that of conventional deep neural networks (DNNs). While prior works such as BOP introduce latent-free optimizers, our contribution specifically focuses on the inherent preference of BNNs for sign descent optimization.

Contributions. This work offers novel insights into the optimization landscape of Binary Neural Networks (BNNs), highlighting that leveraging the sign of the gradient, rather than the gradient magnitude, is sufficient for effective and robust optimization of BNNs. We introduce Diode, a specialized optimizer designed for BNNs, employing a sign descent update mechanism that eliminates the need for latent weights. This approach ensures uniform adjustment of binary weights based on the negative sign of the gradient sign's lower-order moment estimates, significantly reducing memory requirements by up to 50% by avoiding 32-bit latent weights or token embedding buffers. Diode not only streamlines the optimization process but also capitalizes on the reduction of gradient noise, enabling more efficient training with larger batch sizes and fewer iterations. Notably, Diode facilitates a shift away from complex multi-stage training methods, achieving superior or comparable results with single-stage training across various state-of-the-art binary CNN and BERT architectures. The core contributions of this paper are summarized as follows:

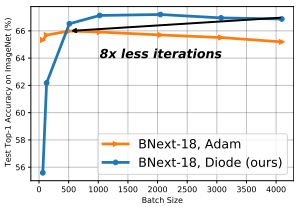


Fig. 1: Diode capitalizes more effectively on the noise reduction.

1. We unveil a groundbreaking observation that BNNs inherently prefer sign descent optimization, positioning Diode as a formidable contender against conventional training protocols.
2. Diode proves to be a game-changer in BNN optimization, significantly cutting down memory usage, simplifying the training pipeline, and making optimal use of larger batch sizes to expedite training.
3. Our comprehensive evaluation confirms Diode's adaptability and efficacy, demonstrating marked performance enhancements across diverse state-of-the-art binary architectures on both vision and NLP tasks.

2 Preliminaries

Let us first consider a neural network $y = f(z;\theta)$, where z is the input, θ is the set of parameters, and f represents the nonlinear transformation process. BNNs are a specialized class of networks where both the weights θ and activations x assume binary values (-1 or +1), resulting in reduced memory and computational requirements. However, the non-differentiability of binary operations presents a formidable challenge in BNN optimization. To address this issue, Courbariaux et al. [7] proposed a latent-weight dependent optimization pipeline for BNNs, which comprises the following steps:

Initialization: Initialize θ with ϵ, a 32-bit random value.

Forward Propagation: Compute the network output using binary operations on weights θ and activations x:

$$x_t^b, \theta_t^b = \text{sign}(x_t, \theta_t), y_t = f(x_t^b, \theta_t^b), \tag{1}$$

where sign denotes the sign function, and x_t^b and θ_t^b represent the binary activation and weight vectors, respectively.

Backward Propagation: To approximate the gradient of the sign function $\frac{\partial x^b}{\partial x}$, the Straight-Through Estimator (STE) strategy [7] is employed, which substitutes the gradient of the sign function with the gradient of the identity function:

$$\frac{\partial x_t^b}{\partial x_t} \approx \text{STE}\left(\frac{\partial x_t^b}{\partial x_t}\right) = \begin{cases} 1 & \text{if } |x_t| \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The STE enables gradient flow through the sign function during backpropagation while maintaining the binary nature of the activations x_t and weights θ_t during forward propagation:

$$\frac{\partial \mathcal{L}_t}{\partial \theta_t} = \frac{\partial \mathcal{L}_t}{\partial \theta_t^b}\frac{\partial \theta_t^b}{\partial \theta_t} \approx \frac{\partial \mathcal{L}_t}{\partial \theta_t^b}, \quad \frac{\partial \mathcal{L}_t}{\partial x_t} = \frac{\partial \mathcal{L}_t}{\partial {x^b}_t}\frac{\partial x_t^b}{\partial x_t} \approx \frac{\partial \mathcal{L}_t}{\partial {x^b}_t}, \tag{3}$$

where $\mathcal{L}_t$ denotes the loss at iteration i.

Latent Weight Update: Update the weights θ_{t+1} using a gradient-based optimization algorithm such as stochastic gradient descent (SGD), with the update rule given by:

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{\partial \mathcal{L}_t}{\partial \theta_t}, \tag{4}$$

where η represents the learning rate.

Despite the widespread adoption of latent-weight dependent optimization techniques in prior studies, methodologies such as layer-by-layer stacked Straight-Through Estimator (STE) and gradient accumulation have not fully mitigated several optimization challenges. Notably, these challenges encompass:

1. *Cumulative gradient noise explosion*;
2. *Updates vanishing*, characterized by the condition $\text{sign}(\theta_{t+1}) = \text{sign}(\theta_t)$ and a significant discrepancy between the L2 norm of θ_t and the L2 norm of its gradient, formally $\|\theta_t\|_2 \gg \|\nabla_{\theta_t}\mathcal{L}_t\|_2$.

In this context, AdamBNN [21] posits that the application of second-order momentum's regularization effect in adaptive optimizers such as Adam may offer a partial resolution to these limitations. Nonetheless, the nuanced differences between conventional optimization strategies and Binary Neural Networks (BNNs) warrant careful consideration. The scholarly discourse surrounding BNNs has yet to extensively explore the development of an optimizer specifically designed to surmount these identified barriers. This gap in research constitutes the primary motivation behind the present study.

To improve the optimization pipeline for BNNs and tackle existing challenges, we propose a novel optimization algorithm that not only considers the unique characteristics of BNNs but also incorporates a more theoretically grounded approach to handling gradient computation and weight updates. In the subsequent sections, we will introduce our proposed method and provide a detailed analysis of its properties and advantages over existing optimization strategies for BNNs.

3 Latent-Weight-Free Sign Descent Updating

To address the optimization challenges of BNNs, we propose a novel optimization algorithm called Diode, which eliminates the need for latent-weight buffers and directly updates the binary parameters. The key insight of Diode is that the gradient sign information is more critical than the gradient magnitude for BNN optimization. Leveraging this insight, Diode introduces a more efficient and robust updating rule for binary parameters.

Latent-Weight-Free Update. We resolve the mismatch between traditional optimizers and BNNs by first adapting SGD to a latent-weight-free updating rule similar to [14], where the binary parameter θ_t is updated as the sign of the moving averaged gradient:

$$u_t = \alpha u_{t-1} + (1-\alpha) g_t, \tag{5}$$

$$\theta_t = -\text{sign}(u_t), \tag{6}$$

where g_t is the gradient, u_t is the exponential moving average (EMA) of the gradient, α is the short-term EMA decay ratio, and $\text{sign}(u_t)$ represents the updating direction of θ_{t-1}. When each binary parameter θ_t is directly updated as the negative sign of the gradient's first-order moment as in Eq. 6, the binary representation θ_t is roughly determined by the averaged gradient from the past $\frac{1}{1-\alpha}$ steps. Consequently, the tuning requirements of binary representation (e.g., -1 and +1) are significantly relaxed compared to traditional latent-weight dependent settings.

Long-Short Term Aggregation. Considering the high variance of the short-term gradient distribution, simply updating θ based on the sign of first-order moments is difficult to converge. Intuitively, the unstable flipping problem can be alleviated by decreasing the weight $1-\alpha$ for g_t in Eq. 5 [14], but this involves extra hyperparameter tuning for different optimization objectives. Instead, we stabilize the updating rule by replacing u_t in Eq. 6 with a second-order

moment estimate m_t:

$$m_t = \beta m_{t-1} + (1-\beta)u_t, \tag{7}$$

where β is the long-term EMA decay ratio. During training, both short-term and long-term gradient information are considered to determine the updating decision for each binary weight θ_t.

Sign Gradient Descent. As observed in Eq. 2, the gradient g_t in BNNs is not the real gradient for each θ_t, and the gradient magnitude noise accumulates layer-by-layer in backward propagation [7, 21, 23]. Consequently, the updating chances of different binary parameters θ_t are randomly distributed under traditional optimizers, and it is sensitive to initialization, which negatively impacts the subsequent BNNs optimization. Motivated by Lion [6] and recent progress in sign descent [18], we propose alleviating this issue by standardizing the update magnitude u_t in Eq. 7 for each parameter using the sign function:

$$v_t = \text{sign}(u_t), \tag{8}$$

$$m_t = \beta m_{t-1} + (1-\beta)v_t. \tag{9}$$

Compared to Adam and other BNN-specific optimizers, such as those presented in [14, 31], the updating magnitude for each binary parameter is uniformly distributed by solely incorporating the sign information of the short-term gradient u_t into the long-order moments m_t. The fundamental observation driving this method is that although the stacked STE approximation introduces random gradient magnitude noise in relation to its forward propagation layer-by-layer, the local optimization direction remains consistent. We empirically find that simply focusing on the gradient sign information alone is already reliable and robust enough for BNNs optimization.

Learning Rate and Decay. To expedite convergence, we further introduce a learning rate η in Eq. 9 as the weight of v_t, which is manually decayed to control the optimization dynamics during training. Thus, the overall updating rule can be mathematically defined as follows:

$$u_t = \alpha u_{t-1} + (1-\alpha)g_t, \tag{10}$$

$$v_t = \eta \text{sign}(u_t), \tag{11}$$

$$m_t = \beta m_{t-1} + (1-\beta)v_t, \tag{12}$$

$$\theta_t = -\text{sign}(m_t). \tag{13}$$

As demonstrated above, we propose the Diode optimizer as a variant of sign descent updating in the latent-weight-free setting. Unlike traditional optimizers used in BNN optimization, Diode eliminates the need for an additional 32-bit buffer for each binary parameter θ. By considering only the sign information of the gradient during training, we can further reduce memory requirements by quantizing g_t to lower bits. This can result in a nearly 50% reduction in memory needs compared to using popular optimizers like Adam. Our work is the first to demonstrate the potential acceleration effect of BNNs not only for inference but also for training.

4 Experiments

We conducted comprehensive experiments to validate the performance of the Diode optimizer in vision and NLP domains, specifically focusing on the ImageNet task for vision and the GLUE benchmark for NLP. We utilized leading binary CNN models and a top-tier binary transformer model for evaluation. In Section 4.1, the experimental setup for each domain is detailed, followed by a presentation of results for both vision and NLP tasks in Sections 4.2 and 4.3. A thorough ablation study in Section 4.4 investigates the impact of various factors including batch size, EMA decay ratios, learning rate, and its decay schedule.

4.1 Setups

Vision Classification and Metrics. Our first evaluation of Diode's performance focuses on the ImageNet ILSVRC2012 dataset [9], which comprises 1,000 classes, 1.2 million training images, 50,000 validation images, and 100,000 test images. We gauge the models' performance using top-1 and top-5 classification accuracy on the validation set. Standard data augmentation and preprocessing techniques are employed. We utilize ReActNet [23] and BNext-18 [11] as representative binary CNN models. To align the model capacity with traditional binary ResNet-18 [7,24], we remove the 4-bit linear layer from the original BNext-18 design.

Natural Language Understanding and Metrics. Furthermore, we assess the Diode optimizer in the NLP domain using the BERT model [10] and the General Language Understanding Evaluation (GLUE) benchmark [35], a widely recognized suite of natural language understanding (NLU) tasks designed to measure the model's capacity to perform various linguistic reasoning tasks. The GLUE benchmark encompasses eight diverse tasks, including sentiment analysis (SST-2), textual entailment (MNLI, RTE), question answering (QNLI), and paraphrase detection (QQP, MRPC), among others (CoLA, STS-B). We measure the performance of the binary BERT using task-specific evaluation metrics, such as accuracy (MNLI, QNLI, SST-2, RTE), F1-score (QQP, MRPC), pearson correlation (STS-B), and matthews correlation coefficient (CoLA), as defined by the GLUE benchmark. We employ the most recent SOTA design BIT [19] as the binary BERT backbone. For a fair and rigorous evaluation, we adhere predominantly to the BIT experimental setup as outlined in their official Github repository [20]. Pre-trained BERT-Base [10] models on GLUE tasks are chosen as full-precision baselines and are subsequently finetuned on each task using respective task-specific training data.

Training Settings. For all evaluations, we empirically set the short-term EMA decay coefficient $\alpha = 0.99$, and long-term EMA decay coefficient $\beta = 0.9999$ based on grid search. Since there is no latent weight for binary parameter optimization, the θ in BNNs can be randomly initialized from a Bernoulli distribution (-1, +1) with the probability $\mathcal{P} = 0.5$. When BNNs are trained from scratch without pre-trained knowledge, the hyperparameter η in Eq.11 does not

share the same meaning as the learning rate in full precision model optimization. Instead, it works as a relative decay coefficient for the short-term updates voting (Eq.12). Consequently, the absolute initialization value does not affect the optimization, and we initialize η as 1.0 if there is no special statement. Core techniques such as Knowledge distillation, cosine learning rate scheduler, and long training are utilized to align with previous work [11, 19, 23].

For ImageNet, we adhere to the protocols of BNext-18 and ReActNet, choosing a single-stage training strategy for ReActNet to avoid the complexities and costs associated with two-stage training, following the rationale in [31]. The optimization of full-precision components remain consistent with previous studies [11, 23].

In GLUE benchmarks, we train the binary BERT-Base model without employing BIT's multi-stage distillation technique due to reproducibility challenges, as indicated by unresolved issues in the official GitHub repository [20]. Despite this, we selected BIT for its state-of-the-art (SOTA) performance in our evaluations.

4.2 Main Results on ImageNet ILSVRC2012

The experimental results for ImageNet ILSVRC2012 are presented in Table 1. In our experiments with the ResNet-18 backbone, we employ the Adam optimizer as the baseline. As evident from the table, our proposed Diode optimizer achieves a state-of-the-art Top-1 accuracy of 68.2% with BNext-18†, surpassing the BNext-18† baseline by 0.7%. Notably, even when compared to the BNext-18 baseline, which includes an additional 4-bit capacity in each basic block, our BNext-18† with the Diode optimizer outperforms BNext-18 by 0.3%, emphasizing a fascinating observation in BNN optimization. These results imply that traditional optimization techniques for BNNs may not fully harness the true potential of 1-bit representation, potentially due to a mismatch in update rules between noisy 32-bit gradients and 1-bit weight representation. When compared to the previous work ReCU [37], which also addresses the inefficient update problem in BNNs, our Diode optimizer resolves the issue without introducing extra optimization constraints, keeping the optimization process as simple as full-precision model optimization.

Furthermore, we apply the Diode optimizer to the widely-used ReActNet-A architecture, which is based on the MobileNet backbone. Impressively, the proposed Diode optimizer enhances ReActNet-A by 0.2%, without resorting to the costly two-stage optimization needed for weight pre-training. The final accuracy of 69.6% is on par with that of full-precision ResNet-18, demonstrating the robustness of Diode across various architectures while maintaining a modest optimization resource cost.

4.3 Main Results on GLUE

The evaluation results for the natural language processing task GLUE are displayed in Table 2. In the case without data augmentation and multi-stage dis-

Table 1: Performance comparison with the most recent state-of-the-art BNNs on ImageNet, under the same backbones. The "W-A" indicates the convolution bit-width except for the input layer. "QOPs" means the INT-4/8 operations in total. "*" indicates using the ReActNet-BiReal design. "†" indicates removing additional 4-bit linear layers from BNext-18 backbone. We define the binary ratio following [11], where ratio=BOPS/(OPS*64).

Methods	Bits(W-A)	BOPs (10^9)	QOPs (10^8)	FLOPs (10^8)	OPs (10^8)	#Param (MB)	Binary Ratio (%)	Top-1 (%)
ResNet-18 [13]	32-32	-	-	18.0	-	44.6	0	69.6
BNN [15]	1-1	1.70	-	1.20	1.47	4.2	18.1	42.2
XNOR-Net [33]	1-1	1.70	-	1.20	1.47	4.2	18.1	51.2
XNOR-Net ++ [5]	1-1	1.70	-	1.20	1.47	4.2	18.1	57.1
Bi-RealNet-18 [24]	1-1	1.68	-	1.39	1.65	4.2	15.9	56.4
BOP [14]	1-1	1.68	-	1.63	1.89	4.2	13.8	56.4
Real2Binary-Net [26]	1-1	1.67	-	1.56	1.82	5.1	14.3	65.4
ReActNet-BiR18 [22]	1-1	1.68	-	1.63	1.89	4.2	13.8	65.9
ReCU* [37]	1-1	1.68	-	1.63	1.89	4.2	13.8	66.4
BNext-18 [11]	1-1	1.68	1.35	-	0.43	2.2	61.0	67.9
BNext-18 † (Adam, **ours**)	1-1	1.68	**0.24**	-	**0.27**	**2.0**	**97.2**	67.5
BNext-18 † (Diode, **ours**)	1-1	1.68	**0.24**	-	**0.27**	**2.0**	**97.2**	**68.2**
ReActNet-A [24] (Adam)	1-1	4.82	-	0.12	0.87	7.4	86.5	69.4
ReActNet-A [31] (GFilter)	1-1	4.82	-	0.12	0.87	7.4	86.5	69.7
ReActNet-A (Diode, **ours**)	1-1	4.82	-	0.12	0.87	7.4	86.5	**69.6**

tillation, the Diode optimizer enhances BIT‡ (Adam) by an average of 3.7%. For sub-tasks with limited training sets, Diode exhibits substantially higher improvements than larger sub-tasks, such as CoLA (+9.3%) and STS-B (+4.4%), corroborating our hypothesis that a tailor-designed optimizer like Diode is a more robust and efficient optimization solution for binary transformer. Remarkably, even when compared to BIT† (Adam) [20], which employs extra multi-stage distillation for weight pretraining, the single-stage optimized BIT‡ (Diode) delivers superior performance by 2.3%.

To assess the impact of our sign gradient descent design, we combine BIT‡ with existing latent-weight-free optimizers BOP [14] and Gradient Filter [31], both of which aggregate gradient magnitude information during training. Notably, the original BNN-specific optimizers have only been evaluated on vision tasks using CNN backbones, and our evaluation provides the first assessment of transformer architecture and NLP tasks. As illustrated in Table 2, both BOP and Gradient Filter fail to achieve the accuracy level of the original BIT‡ (Adam), and the models experience significant accuracy loss on sample-limited tasks such as CoLA, STS-B, MRPC, and RTE. The considerable performance gap between traditional BNN-specific optimizers and Diode on BERT optimization suggests that the optimization of binary transformers introduces additional difficulty than CNNs. The sign descent design furnishes Diode with good robustness across various architectures.

We then combine the Diode optimizer and BIT† based on pre-trained BIT weight initialization to examine the impact of initialization. Surprisingly, the performance further improves to 75.7%, which is 3.3% higher than the baseline BIT† (Adam), setting a new state-of-the-art performance for binary BERT without data augmentation. This result indicates that Diode can benefit from improved initialization.

Table 2: Comparison of BERT quantization methods on the GLUE with & without data augmentation. The E-W-A represents the BERT backbone's quantization level of embeddings, weights, and activations. "‡" represents results without using the multi-stage distillation technique."†" indicates results based on released pre-trained models on the official Github repository [20]. "*" means that data augmentation is not needed for MNLI, QQP. "↑" & "↓" indicates improved & decreased performance compared with the baseline optimizer Adam.

Method	#Bits(E-W-A)	Size (MB)	FLOPs (G)	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Avg.
BERT [10]	32-32-32	418	22.5	84.9/85.5	91.4	92.1	93.2	59.7	90.1	86.3	72.2	83.9
Without data augmentation												
Q-BERT [34]	2-8-8	43.0	6.5	76.6/77.0	–	–	84.6	–	–	68.3	52.7	–
Q-BERT [34]	2-8-8	43.0	6.5	47.2/47.3	67.0	61.3	80.6	0	4.4	68.4	52.7	47.7
TernaryBERT [38]	2-2-8	28.0	6.4	83.3/83.3	90.1	–	–	50.7	–	87.5	68.2	–
BinaryBERT [1]	1-1-8	16.5	3.1	84.2/84.7	91.2	91.5	92.6	53.4	88.6	85.5	72.2	82.7
BinaryBERT [1]	1-1-4	16.5	1.5	83.9/84.2	91.2	90.9	92.3	44.4	87.2	83.3	65.3	79.9
BinaryBERT [1]	1-1-2	16.5	0.8	62.7/63.9	79.9	52.6	82.5	14.6	6.5	68.3	52.7	53.7
BinaryBERT [1]	1-1-1	16.5	0.4	35.6/35.3	66.2	51.5	53.2	0	6.1	68.3	52.7	41.0
BiBERT [28]	1-1-1	13.4	0.4	66.1/67.5	84.8	72.6	88.7	25.4	33.6	72.5	57.4	63.2
BiT ‡ (Adam) [20]	1-1-1	13.4	0.4	77.1/77.5	82.9	85.7	87.7	25.1	71.1	79.7	58.8	71.0
BiT ‡ (BOP [14])	1-1-1	13.4	0.4	76.6/76.8 ↓	86.6	84.9 ↓	87.7	18.9 ↓	15.9 ↓	70.3 ↓	54.5 ↓	61.9 ↓
BiT ‡ (Gdient Filter [31])	1-1-1	13.4	0.4	76.6/76.8 ↓	86.6	84.8 ↓	87.0 ↓	22.6 ↓	17.2 ↓	70.8 ↓	54.5 ↓	62.5 ↓
BIT ‡ (Diode, **ours**)	1-1-1	13.4	0.4	**79.0/78.3** ↑	**88.4** ↑	**87.6** ↑	**91.0** ↑	**34.3** ↑	**75.5** ↑	**81.3** ↑	**60.6** ↑	**74.7** ↑
BiT † (Adam) [20]	1-1-1	13.4	0.4	79.5/79.4	88.9	85.9	88.5	31.5	67.4	79.2	59.5	72.4
BiT † (Diode, **ours**)	1-1-1	13.4	0.4	79.5/79.4	88.9	**87.6** ↑	**91.0** ↑	**34.3** ↑	**78.3** ↑	**82.6** ↑	**63.9** ↑	**75.7** ↑
With data augmentation [19]												
TernaryBERT [38]	2-2-8	28.0	6.4	83.3/83.3*	90.1*	90.0	92.9	47.8	84.3	82.6	68.4	80.3
BinaryBERT [1]	1-1-8	16.5	3.1	84.2/84.7*	91.2*	91.6	93.2	55.5	89.2	86.0	74.0	83.3
BinaryBERT [1]	1-1-4	16.5	1.5	83.9/84.2*	91.2*	91.4	93.7	53.3	88.6	86.0	71.5	82.6
BinaryBERT [1]	1-1-2	16.5	0.8	62.7/63.9*	79.9*	51.0	89.6	33.0	11.4	71.0	55.9	57.6
BinaryBERT [1]	1-1-1	16.5	0.4	35.6/35.3*	66.2*	66.1	78.3	7.3	22.1	69.3	57.7	48.7
BiBERT [28]	1-1-1	13.4	0.4	66.1/67.5*	84.8*	76.0	90.9	37.8	56.7	78.8	61.0	68.8
BiT † (Adam) [20]	1-1-1	13.4	0.4	79.5/79.4	88.9	85.5	90.6	36.2	82.3	85.5	66.0	76.7
BiT † (Diode, **ours**)	1-1-1	13.4	0.4	79.5/79.4*	88.9*	**87.6** ↑	**92.3** ↑	**42.3** ↑	**83.6** ↑	**87.5** ↑	**69.0** ↑	**78.8** ↑

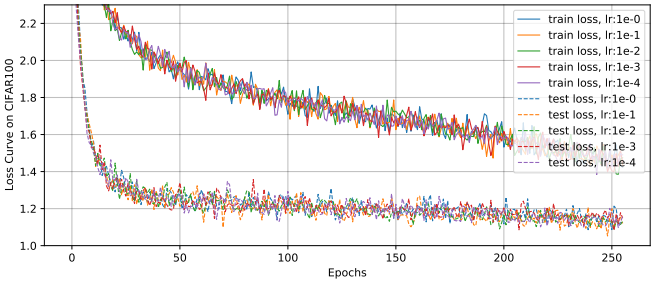
Lastly, we evaluate BIT† optimization combined with data augmentation, as suggested by [16,19,29]. Compared to the officially released BIT† (Adam), BIT† (Diode) consistently outperforms the baseline model across most tasks, with an average improvement of 2.1%. Notably, for the CoLA task, binary BERT-Base improves by 6%, reaching the 40%+ Mcc level for the first time. For the MRPC task, binary BERT-Base attains an F1-score of 87.5%, surpassing the 32-bit BERT-Base by 1.2%.

In summary, our Diode optimizer consistently enhances BNNs' performance across various settings and simplifies optimization requirements for both vision and NLP tasks. Our experimental results provide empirical evidence for the superiority of customized optimizer designs, such as Diode, for BNN optimization.

4.4 Ablation Study and Analysis

In this subsection, we present a comprehensive ablation study aimed at gaining deeper insights into the optimization process of the proposed Diode optimizer and its associated hyperparameters.

Learning Rate, Scheduler, and Optimization Dynamics. Although we refer to the hyperparameter η as the "learning rate," its meaning and behavior during training differ from the "learning rate" in traditional optimizers. To investigate the effects of η and identify the optimal configuration, we perform an extensive ablation study using a BNext-18† model. The baseline model is op-



(a) The loss curve on CIFAR100 with different lr η initialization.

η	**Top-1 Acc (%)**
1.0	74.50±0.03
1e-1	74.59±0.06
1e-2	74.32±0.14
1e-3	74.46±0.04
1e-4	74.45±0.12

(b) The validation Top-1 accuracy of BNext-18† on CIFAR100 dataset with different initial lr η in the proposed Diode optimizer.

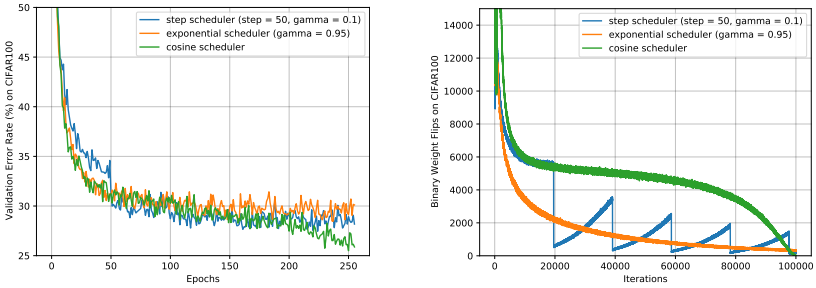
Fig. 2: The optimization process of BNext-18† on CIFAR100 with different initial learning rate η. As η serves as a relative decay coefficient for the binarized short-term EMA value $Sign(u_t)$, its absolute value has minimal impact on the optimization process.

timized under nearly identical settings, except for varying initial learning rates $1.0, 1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}$. We set the EMA decay coefficients to (0.99, 0.9999).

The optimization process and evaluation results are depicted in Fig. 2. Unlike conventional optimizers, the initial η in Diode exhibits minimal influence on the optimization process in terms of convergence speed and final accuracy. As a result, we can initialize η to 1.0 in Diode without requiring hyperparameter tuning. Additionally, we examine the effects of the η decay strategy using the three most popular schedulers provided by Pytorch: StepLR, ExponentialLR, and CosineLR. Due to the signed momentum update approach, the update magnitude of all binary parameters is regulated by η. As illustrated in Fig. 3b, a rapid η decay strategy causes BNext-18† to experience inefficient updating and underfitting issues. Conversely, a zero-decay strategy for η encourages the model to slide away from the existing situation, leading to convergence difficulties. Our empirical evaluation indicates that the CosineLR scheduler offers the best trade-off and therefore best accuracy Fig. 3a. The core principle behind this is that the BNNs optimization benefits from a smooth but coherent convergence trend.

Table 3: Top-1 accuracy comparison among BNext-18† optimizations with different settings of short-term EMA ratio α and long-term EMA ratio β in the Diode optimizer.

Short-Term EMA Ratio α	0.9		0.95		0.99	
Long-Term EMA Ratio β	0.999	0.9999	0.999	0.9999	0.999	0.9999
Top-1 Accuracy (%)	71.57	73.92	72.07	74.06	72.01	74.51



(a) The validation error curve. **(b)** The weight flips curve.

Fig. 3: Ablation study about the influences of different learning rate decay strategies for Diode optimizer (StepLR, CosineLR, ExponentialLR). We can easily see that the CosineLR scheduler guides the model to a lower flip ratio loss landscape area and therefore better accuracy.

Examining the Impacts of Lower Order Moments. Unlike existing latent-weight-dependent optimizers, Diode directly updates the binary parameter θ_t based on the negative sign of lower-order moments Eq. 13. As a result, BNNs with a Diode optimizer are more sensitive to the estimation coefficients of lower-order moments (α, β). We investigate the effects of the two momentum decay coefficients by sweeping α over the range $\{0.9, 0.95, 0.99\}$ and β over the range $\{0.999, 0.9999\}$. The validation loss curve, weight updating curve, first-order moment u_t, and second-order moment m_t under different hyperparameter settings are recorded in Fig. 4. As observed, the Diode optimizer benefits from the momentum mechanism. The frequency of binary weight flips is directly related to (α, β). With a suitable flipping frequency over time, the binary model converges more rapidly and attains a higher performance level, as shown in Table 3.

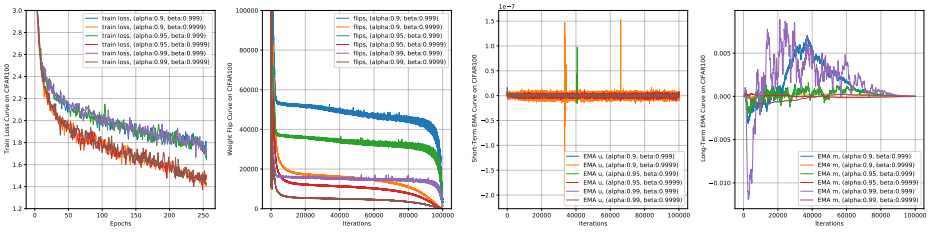


Fig. 4: Alation study on different short-term EMA ratio α and long-term EMA ratio β, based on BNext-18† and CIFAR100. The backward gradients are quite noisy and the short-term and long-term EMA smooth strategy help the model convergence by stabilizing the binary weight updating.

Batch Size's Role in Diminishing Noise and Hastening Convergence. To examine whether Diode takes better advantage of the reduction in gradient noise than Adam and check the impact of batch size ($\mathcal{B}$) for training, we compare Diode to Adam using BNext-18† on the ImageNet dataset, as depicted in Fig. 1. The batch size is varied across the range {64, 128, 512, 1024, 2048, 3072, 4096}, and all models are trained for 128 epochs, which is a quarter of the duration used in the BNext-18 paper [11]. As demonstrated in the figure, the Diode optimizer is less effective than Adam when the batch size is smaller than 128, which is in line with the observation in the previous sign descent researches [3,4,6,18,32]. However, the accuracy of Diode increases fast along with the larger batch sizes and is more efficient than Adam's in leveraging the benefits of larger batch sizes. For instance, the Diode optimizer with $\mathcal{B}$=2048 surpasses the best Adam setting with $\mathcal{B}$=512 by 1.3%, while using a quarter of iterations. When the batch size is increased to 4096, Diode still outperforms the best Adam setting ($\mathcal{B}$=512) by 0.96% using 8x less training iterations. Basically, the accuracy gap between Diode and Adam keeps on increasing when the batch size is larger than 128. This reveals the potential of the sign descent method combined with increased batch size on training BNNs.

5 Related Work

Latent-weight Optimization in BNNs. The optimization of Binarized Neural Networks (BNNs) often relies on 32-bit latent-weight techniques, as introduced by Courbariaux et al. [7]. This method uses 32-bit buffers for accumulating floating-point gradients, which are then optimized using standard techniques like SGD and Adam. Although initial attempts, such as XNOR-Net [33], ReActNet [23], and IRNet [30], aimed to refine optimization by reducing approximation errors, later research, including BOP [14], AdamBNN [21], and Gradient Filter [31], focused specifically on BNN optimization. These studies highlight the inefficiencies of the traditional latent-weight approach for BNNs, pointing towards the benefits of latent-weight-free optimization. However, despite these advancements, existing custom optimizers, which still depend on noisy floating gradients, have not consistently outperformed Adam in complex task scenarios as noted in our evaluation on GLUE tasks (Table. 2). We argue that both the conventional latent-weight optimization and the reliance on floating gradients may limit BNNs' performance.

Sign Descent Optimization. The concept of sign gradient descent, introduced to reduce communication in distributed systems, has evolved significantly. Bernstein et al. [3, 4] showed that signSGD could converge under both large and mini-batch conditions with a majority vote mechanism. Further analysis by Balles and Hennig [2] positioned signSGD as a variant of Adam with specific parameter settings. More recent work, such as that by Crawshaw et al. [8] and Chen et al. [6], has advanced our understanding of signSGD, with the latter introducing Lion, a sign-based optimizer surpassing Adam in multiple tasks. This body of research, indicating the efficacy and reliability of gradient signs over

magnitudes, particularly in transformer training, informs our development of Diode, a sign descent-based optimizer designed for the unique requirements of BNNs.

6 Limitations

Reduced Efficacy with Small Batch Sizes. While the proposed Diode exhibits promising potential for training binary neural networks in the domains of vision and natural language processing, its effectiveness is less pronounced when the batch size is small. This may diminish its applicability in scenarios with limited computational resources, such as embedded systems. A possible approach to addressing this limitation is to integrate gradient accumulation techniques into the optimization process, whereby gradients are accumulated or taken into account over multiple batches.

Underexplored Theoretical Perspectives behind Sign Descent. Our method aims to mitigate the inefficient training of BNNs and develops an innovative sign descent variant within latent-weight-free settings. Although we have empirically demonstrated Diode's superiority across a variety of tasks and architectures, particularly in the context of large batch sizes, the theoretical underpinnings of the benefits associated with sign descent optimization for BNNs warrant further investigation. A deeper understanding of the dynamics of sign descent optimization, which has been a long-standing question in the literature [18], may offer valuable insights for enhancing the optimization of BNNs.

7 Conclusion

In this paper, we introduce Diode, an optimization algorithm designed for Binary Neural Networks (BNNs) that prioritizes gradient sign information over magnitude, employing a latent-weight-free update rule. This approach addresses BNN-specific challenges, leading to quicker convergence, improved accuracy, and greater robustness. Through extensive testing on vision and NLP tasks, Diode demonstrates superior performance compared to conventional optimizers like Adam, proving to be an effective tool for a broad spectrum of BNN applications. Future work will explore Diode's integration with binary large language models (LLMs) to further enhance their optimization.

Acknowledgments

The authors acknowledge the financial support by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV) within the project EKAPEx 67KI32002A

References

1. Bai, H., Zhang, W., Hou, L., Shang, L., Jin, J., Jiang, X., Liu, Q., Lyu, M., King, I.: Binarybert: Pushing the limit of bert quantization. arXiv preprint arXiv:2012.15701 (2020) 9
2. Balles, L., Hennig, P.: Dissecting adam: The sign, magnitude and variance of stochastic gradients. In: International Conference on Machine Learning. pp. 404–413. PMLR (2018) 12
3. Bernstein, J., Wang, Y.X., Azizzadenesheli, K., Anandkumar, A.: signsgd: Compressed optimisation for non-convex problems. In: International Conference on Machine Learning. pp. 560–569. PMLR (2018) 12
4. Bernstein, J., Zhao, J., Azizzadenesheli, K., Anandkumar, A.: signsgd with majority vote is communication efficient and fault tolerant. arXiv preprint arXiv:1810.05291 (2018) 12
5. Bulat, A., Tzimiropoulos, G.: Xnor-net++: Improved binary neural networks. arXiv preprint arXiv:1909.13863 (2019) 8
6. Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.J., et al.: Symbolic discovery of optimization algorithms. arXiv preprint arXiv:2302.06675 (2023) 5, 12
7. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830 (2016) 3, 5, 6, 12
8. Crawshaw, M., Liu, M., Orabona, F., Zhang, W., Zhuang, Z.: Robustness to unbounded smoothness of generalized signsgd. arXiv preprint arXiv:2208.11195 (2022) 12
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 6
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 6, 9
11. Guo, N., Bethge, J., Meinel, C., Yang, H.: Join the high accuracy club on imagenet with a binary neural network ticket. arXiv preprint arXiv:2211.12933 (2022) 1, 6, 7, 8, 12
12. Guo, N., Bethge, J., Yang, H., Zhong, K., Ning, X., Meinel, C., Wang, Y.: Boolnet: minimizing the energy consumption of binary neural networks. arXiv preprint arXiv:2106.06991 (2021) 1
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 8
14. Helwegen, K., Widdicombe, J., Geiger, L., Liu, Z., Cheng, K.T., Nusselder, R.: Latent weights do not exist: Rethinking binarized neural network optimization. Advances in neural information processing systems **32** (2019) 2, 4, 5, 8, 9, 12
15. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 4114–4122 (2016) 8
16. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351 (2019) 9

17. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020) 1
18. Kunstner, F., Chen, J., Lavington, J.W., Schmidt, M.: Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. arXiv preprint arXiv:2304.13960 (2023) 5, 12, 13
19. Liu, Z., Oguz, B., Pappu, A., Xiao, L., Yih, S., Li, M., Krishnamoorthi, R., Mehdad, Y.: Bit: Robustly binarized multi-distilled transformer. arXiv preprint arXiv:2205.13016 (2022) 1, 6, 7, 9
20. Liu, Z., Oguz, B., Pappu, A., Xiao, L., Yih, S., Li, M., Krishnamoorthi, R., Mehdad, Y.: Official source code of bit on github (2022), https://github.com/facebookresearch/bit 6, 7, 8, 9
21. Liu, Z., Shen, Z., Li, S., Helwegen, K., Huang, D., Cheng, K.T.: How Do Adam and Training Strategies Help BNNs Optimization? In: International Conference on Machine Learning. pp. 6936–6946. PMLR (2021), http://arxiv.org/abs/2106.11309 2, 4, 5, 12
22. Liu, Z., Shen, Z., Savvides, M., Cheng, K.: Official source code of reactnet on github (2020), https://github.com/liuzechun/ReActNet, accessed: 2021-10-01 8
23. Liu, Z., Shen, Z., Savvides, M., Cheng, K.: Reactnet: Towards precise binary neural network with generalized activation functions. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV. Lecture Notes in Computer Science, vol. 12359, pp. 143–159. Springer (2020). https://doi.org/10.1007/978-3-030-58568-6_9, https://doi.org/10.1007/978-3-030-58568-6_9 5, 6, 7, 12
24. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In: Proceedings of the European conference on computer vision (ECCV). pp. 722–737 (2018) 6, 8
25. Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., Dong, L., Wang, R., Xue, J., Wei, F.: The era of 1-bit llms: All large language models are in 1.58 bits. arXiv preprint arXiv:2402.17764 (2024) 1
26. Martinez, B., Yang, J., Bulat, A., Tzimiropoulos, G.: Training binary neural networks with real-to-binary convolutions. In: International Conference on Learning Representations (2020) 8
27. Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., et al.: Efficient large-scale language model training on gpu clusters using megatron-lm. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1–15 (2021) 1
28. Qin, H., Ding, Y., Zhang, M., Yan, Q., Liu, A., Dang, Q., Liu, Z., Liu, X.: Bibert: Accurate fully binarized bert. arXiv preprint arXiv:2203.06390 (2022) 9
29. Qin, H., Ding, Y., Zhang, M., Yan, Q., Liu, A., Dang, Q., Liu, Z., Liu, X.: Bibert: Accurate fully binarized bert. In: International Conference on Learning Representations (ICLR) (2022) 9
30. Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., Song, J.: Forward and backward information retention for accurate binary neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2250–2259 (2020) 12
31. Quist, J., Li, Y., van Gemert, J.: Understanding weight-magnitude hyperparameters in training binary networks. arXiv preprint arXiv:2303.02452 (2023) 2, 5, 7, 8, 9, 12

32. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10428–10436 (2020) 12
33. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. pp. 525–542. Springer (2016) 8, 12
34. Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Q-bert: Hessian based ultra low precision quantization of bert. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8815–8821 (2020) 9
35. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018) 6
36. Wang, H., Ma, S., Dong, L., Huang, S., Wang, H., Ma, L., Yang, F., Wang, R., Wu, Y., Wei, F.: Bitnet: Scaling 1-bit transformers for large language models. arXiv preprint arXiv:2310.11453 (2023) 1
37. Xu, Z., Lin, M., Liu, J., Chen, J., Shao, L., Gao, Y., Tian, Y., Ji, R.: Recu: Reviving the dead weights in binary neural networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5198–5208 (2021) 7, 8
38. Zhang, W., Hou, L., Yin, Y., Shang, L., Chen, X., Jiang, X., Liu, Q.: Ternarybert: Distillation-aware ultra-low bit bert. arXiv preprint arXiv:2009.12812 (2020) 9
39. Zhang, Y., Garg, A., Cao, Y., Lew, Ł., Ghorbani, B., Zhang, Z., Firat, O.: Binarized neural machine translation. arXiv preprint arXiv:2302.04907 (2023) 1