AdaBin: Improving Binary Neural Networks with Adaptive Binary Sets

Zhijun Tu[1,2], Xinghao Chen[2(✉)], Pengju Ren[1(✉)], and Yunhe Wang[2]

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
tuzhijun123@stu.xjtu.edu.cn, pengjuren@xjtu.edu.cn
[2] Huawei Noah's Ark Lab

디지털헬스학과 최우정

AdaBin: Improving Binary Neural Networks with Adaptive Binary Sets

Presentation Overview

Introduction
Overview of Binary Neural Networks and the limitations addressed by AdaBin.

Methodology
Dynamic binary set selection and optimization strategies in AdaBin.

Performance
Experimental results showing the impact of AdaBin on ResNet-18 and other models.

Conclusion
Future possibilities and the broader impact of AdaBin in AI.

AdaBin: Improving Binary Neural Networks with Adaptive Binary Sets

Introduction to Binary Neural Networks and AdaBin

Memory Efficiency
Binary Neural Networks (BNNs) offer significant reductions in memory usage by binarizing weights and activations.

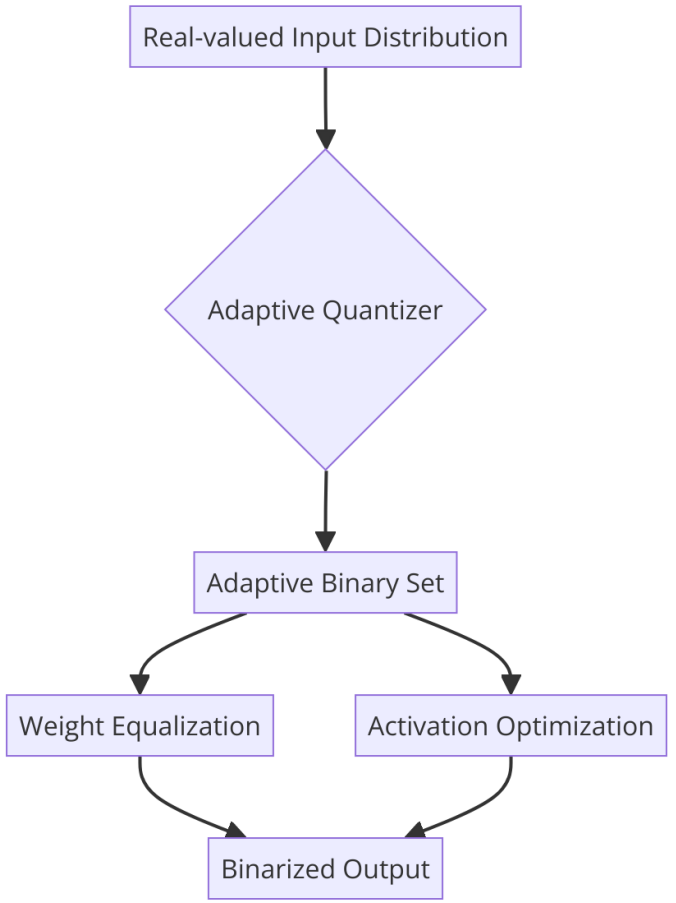
Fixed Binary Sets Limitation
Traditional BNNs use fixed binary sets that fail to capture diverse distributions of weights and activations.

AdaBin's Adaptive Approach
AdaBin dynamically adjusts binary sets for each layer to better match real-valued distributions.

AdaBin: Methodology Overview

Dynamic Binary Set Selection for Improved Performance

- **Adaptive Quantization:** AdaBin adaptively determines binary sets based on the distribution of weights and activations for each layer.
- **Enhanced Accuracy:** The approach improves the alignment of binary values with real-valued distributions, reducing quantization errors.
- **Optimization Strategies:** Optimization includes strategies for both weights and activations to minimize performance loss during binarization.



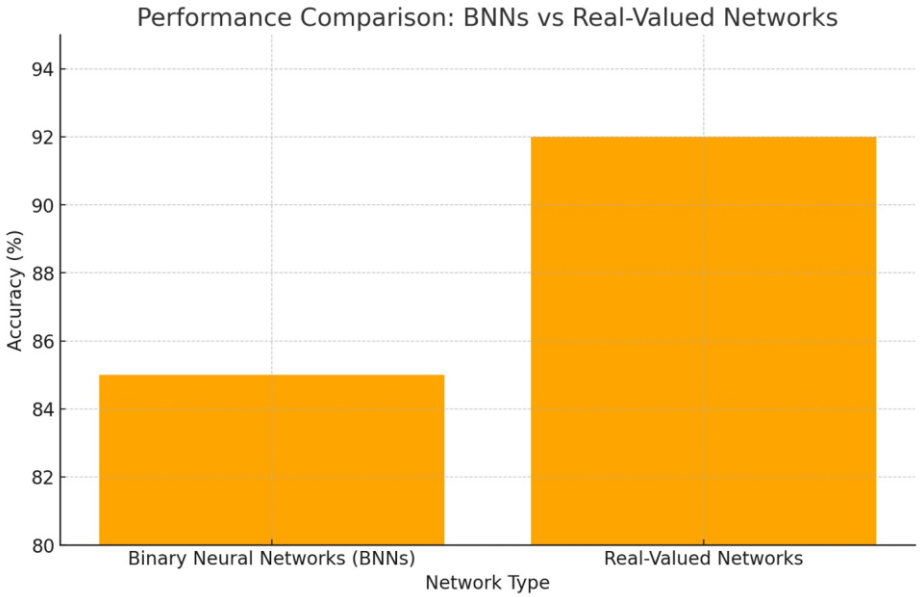
적응형 이진화 과정

Quantization Function and Optimization

Aligning Binary Sets with Real-Valued Distributions

- **Binary-Real Discrepancy Reduction:** The quantization function minimizes the gap between binary and real-valued distributions.
- **Weight Symmetry Alignment:** The quantization aligns the symmetric center of the binary distribution with real-valued weights.
- **Gradient-Based Optimization:** Activation binary sets are adjusted during training using gradient-based methods to improve accuracy.

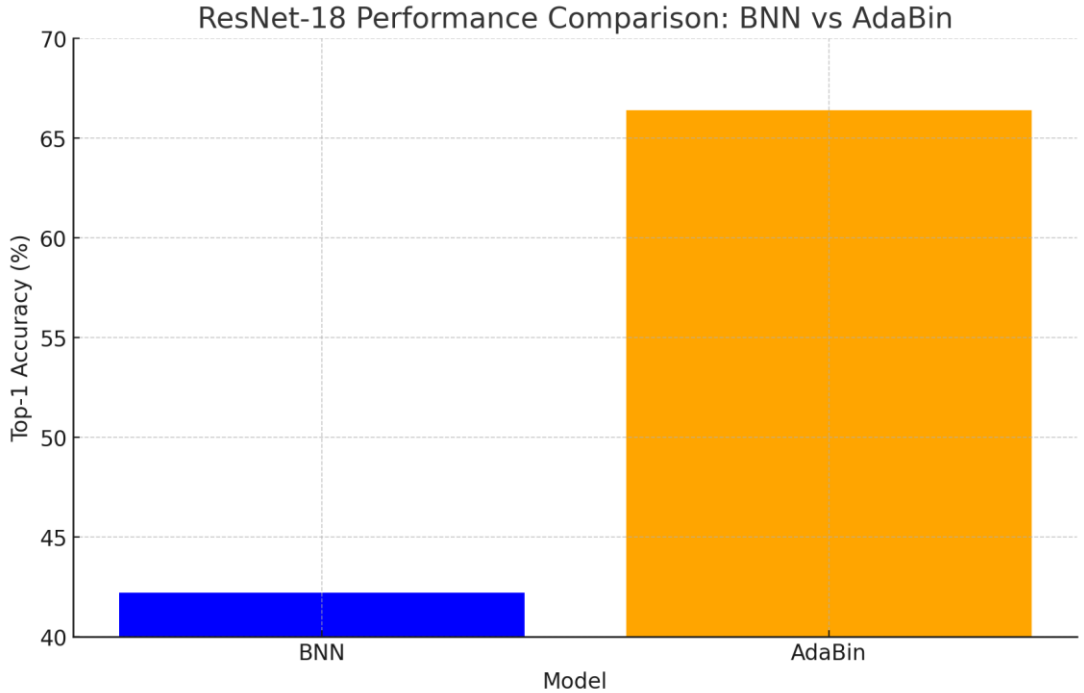
$$\hat{W} = \text{sign}(W - \text{center}) \times \text{distance}$$

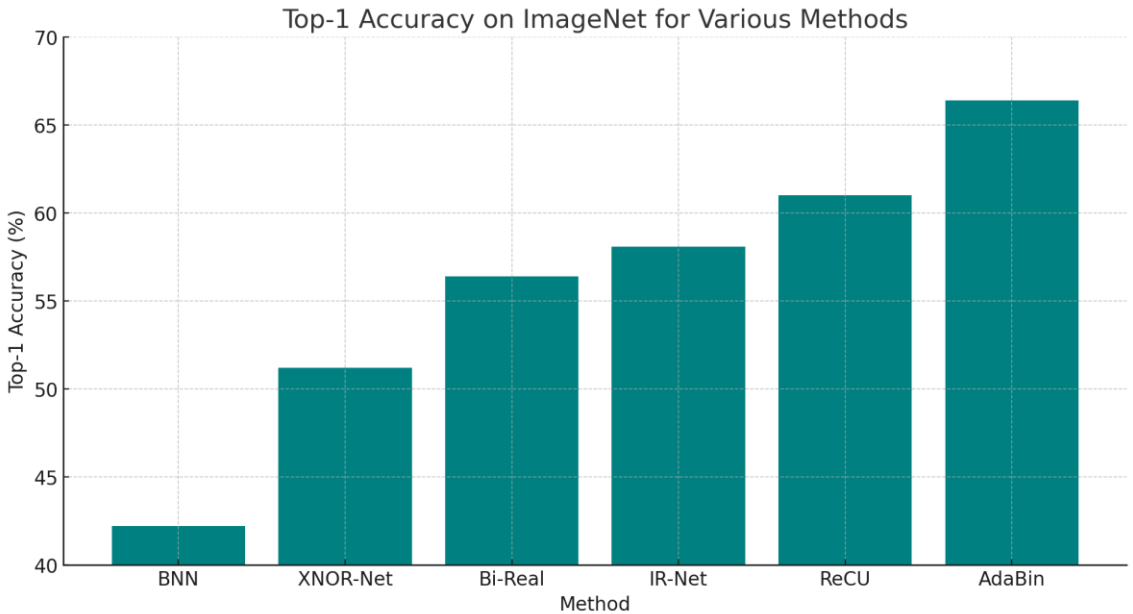


Experimental Results - ResNet-18

AdaBin's Impact on Accuracy and Performance

- **Top-1 Accuracy Improvement:** AdaBin achieved a 66.4% Top-1 accuracy on ImageNet, surpassing traditional BNN methods.
- **Performance Attribution:** The performance gain is due to better adaptation of binary sets to layer-specific distributions.
- **Comparison with Standard BNNs:** AdaBin significantly outperforms traditional BNNs, reducing the performance gap with real-valued networks.

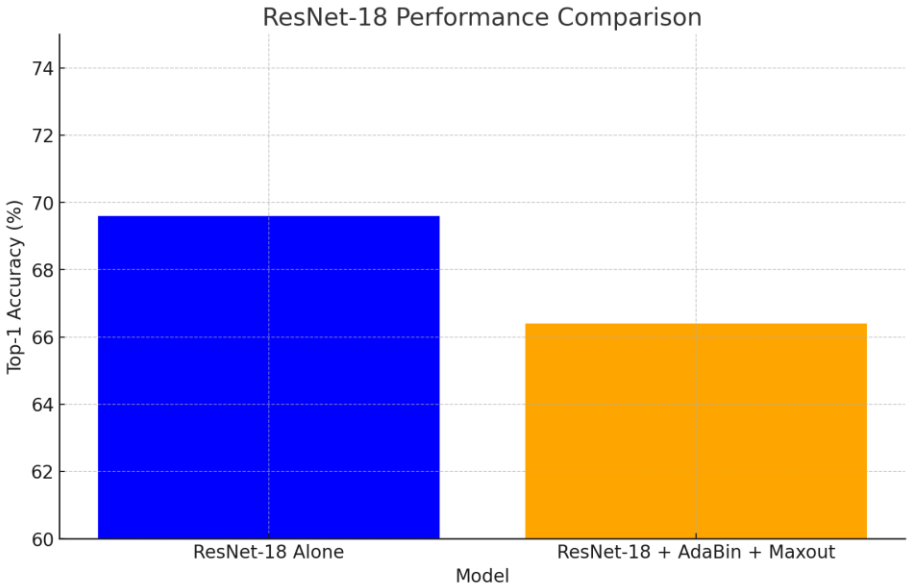




AdaBin + Maxout: Achieving State-of-the-Art Performance

Performance Gains Compared to ResNet-18 Alone

ResNet-18 vs AdaBin + Maxout
AdaBin combined with Maxout significantly outperforms ResNet-18 alone, achieving state-of-the-art (SOTA) results.



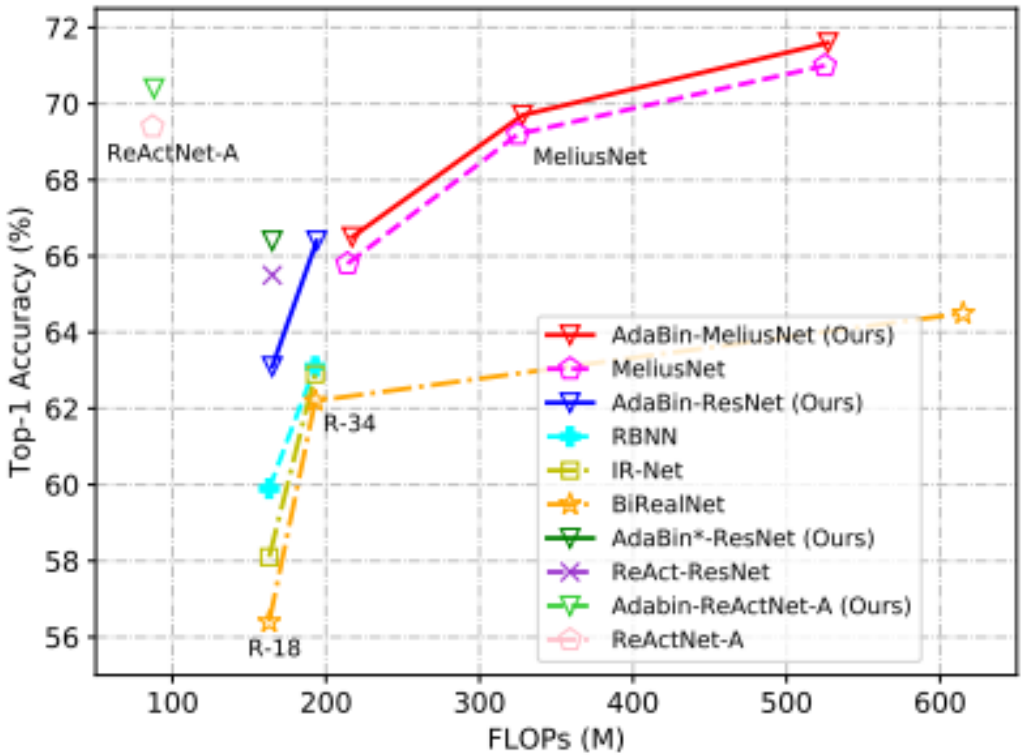
Cross-Model Performance Comparison
The combination of AdaBin + Maxout shows a substantial performance improvement over other models, with gains up to 9.7%.

Broader Impact
The approach demonstrates its effectiveness across various tasks, setting a new benchmark for binary networks.

Further Comparisons with Binary-Specific Architectures

Performance of AdaBin with BDenseNet and MeliusNet

- **BDenseNet and MeliusNet:** AdaBin was applied to binary-specific architectures such as BDenseNet and MeliusNet.
- **Consistent Outperformance:** Across multiple architectures, AdaBin consistently outperformed the original networks by 0.5% to 1.1%.
- **Enhanced Capacity:** This demonstrates AdaBin's ability to enhance the capacity and quality of binary networks with minimal computational overhead.

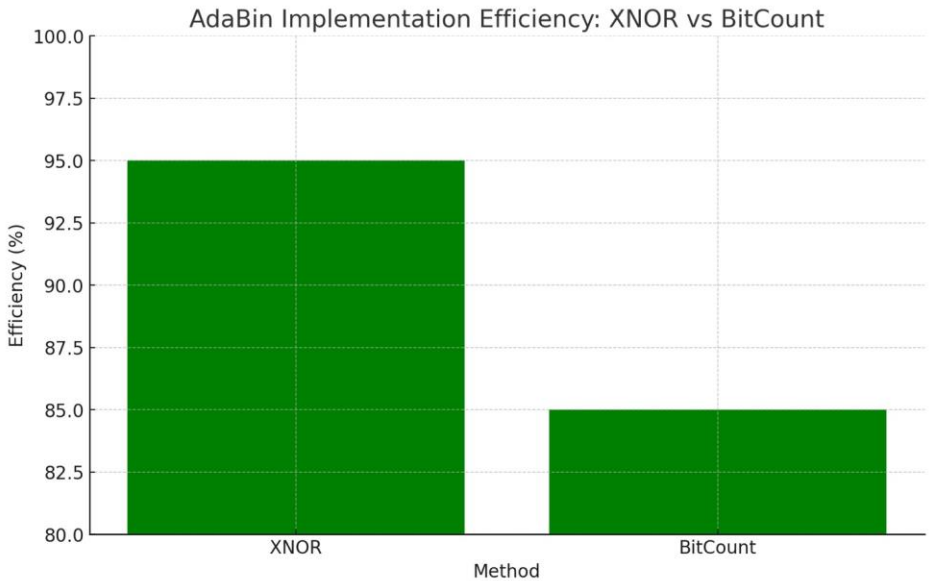


(a) FLOPs vs. ImageNet accuracy

Technical Discussion: Efficiency and Implementation

XNOR and BitCount in AdaBin Implementation

- **Implementation with XNOR and BitCount:** AdaBin can be efficiently implemented using XNOR and BitCount operations, resulting in 60-85% acceleration.
- **Memory Efficiency:** Memory usage can be reduced by 31%, making it a viable solution for resource-constrained environments.
- **Proven Effectiveness:** AdaBin's effectiveness is demonstrated across various datasets and tasks, including object detection on PASCAL VOC.

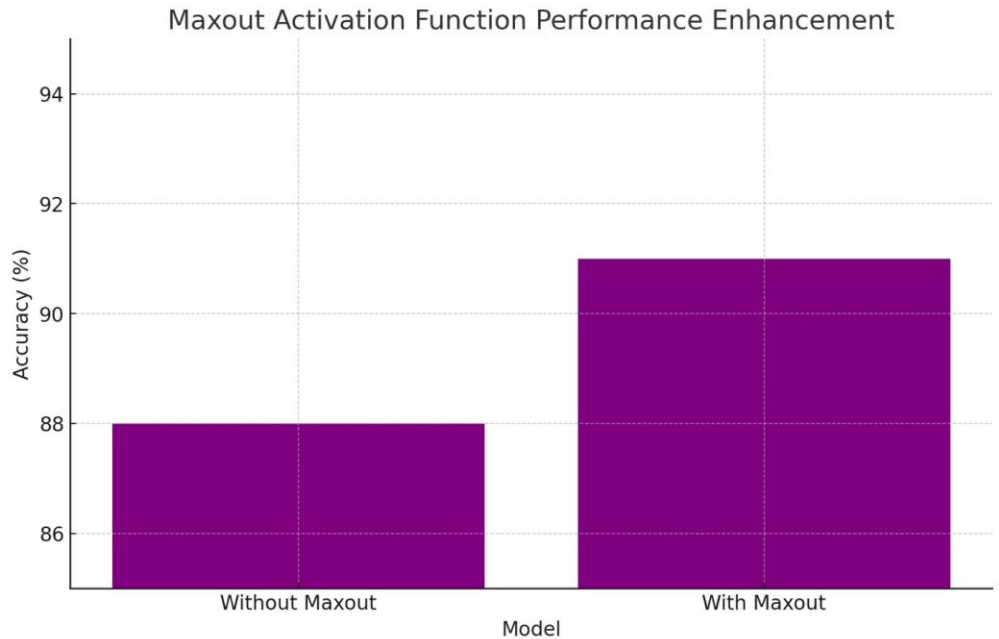


Maxout's Role in Enhancing Performance

Improving Non-Linearity and Performance in Binary Networks

Enhancing Non-Linearity
Maxout activation function further improves the performance of binary networks by enhancing non-linearity.

Performance Gains
In experiments, Maxout contributed up to 9.7% performance improvement in specific configurations.

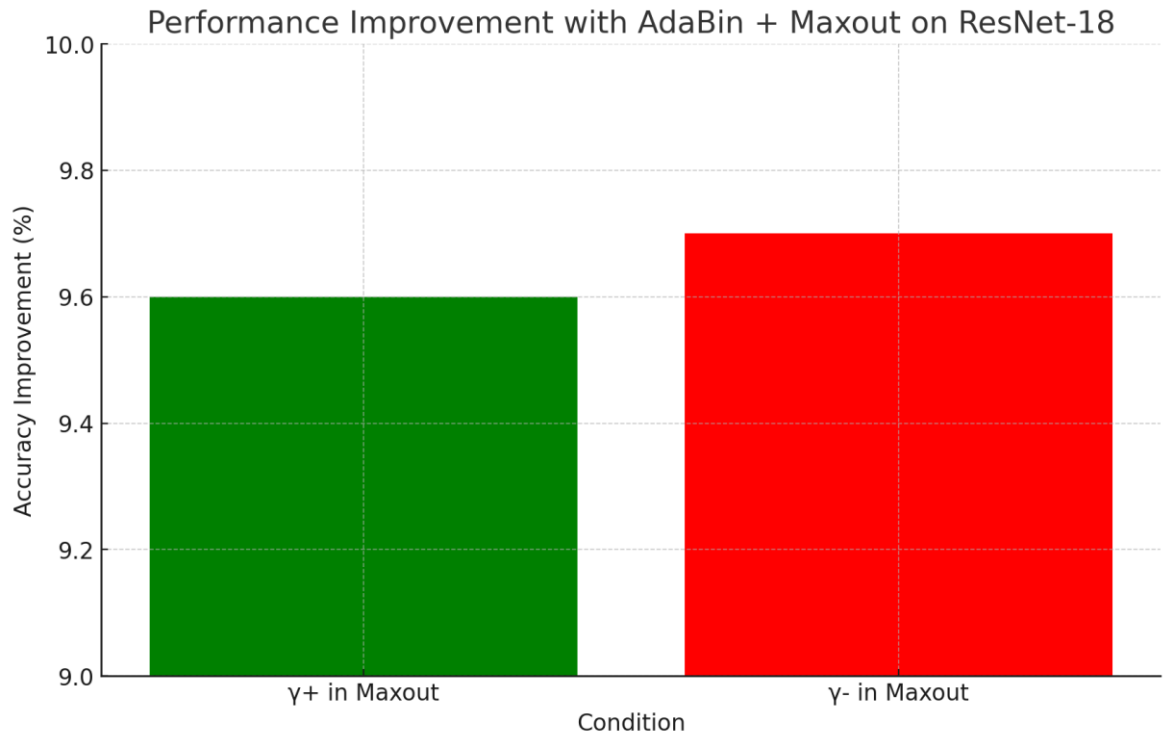


SOTA Performance
Combining AdaBin with Maxout results in state-of-the-art performance on challenging datasets.

Achieving SOTA Performance with AdaBin + Maxout

Comparative Analysis Across Models

- **SOTA Achievements:** Combining AdaBin and Maxout with ResNet leads to state-of-the-art performance across various models.
- **Cross-Model Analysis:** The approach shows consistent improvements over standard models, with up to 9.7% performance gains.
- **Broader Impact:** This method highlights the potential for binary networks to rival real-valued counterparts in complex tasks.



Conclusion and Q&A

Final Thoughts and Discussion

AdaBin's Contribution
Significant improvements in binary neural network performance with adaptive binary sets.

Future Directions
Potential for expanding AdaBin's applications in various AI models and tasks.

Q&A
Open floor for questions and further discussion.