How Do Adam and Training Strategies Help BNNs Optimization?

Zechun Liu*, Zhiqiang Shen*, Shichao Li, Koen Helwegen, Dong Huang, Kwang-Ting Cheng

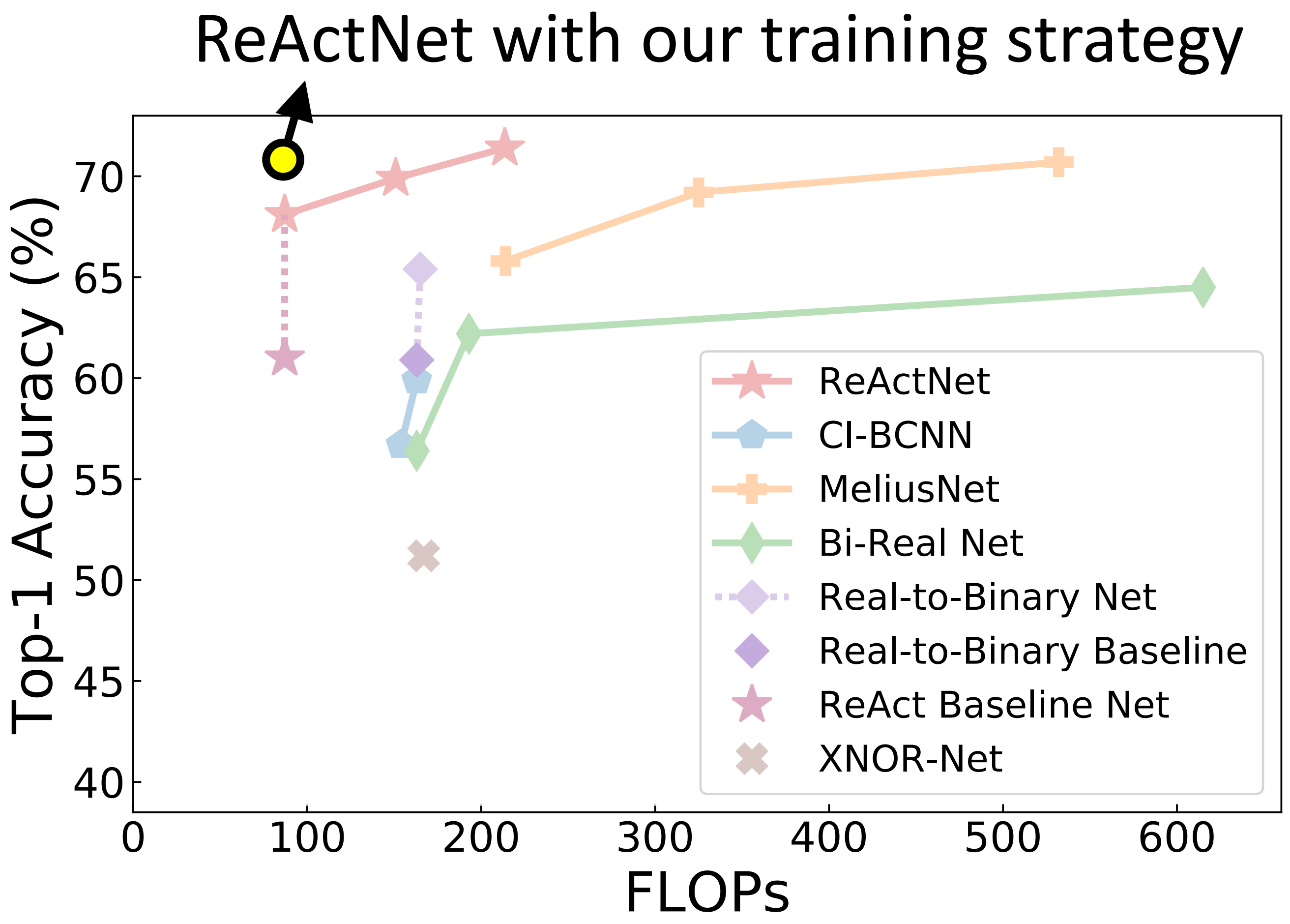
ICML 2021

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Carnegie Mellon University

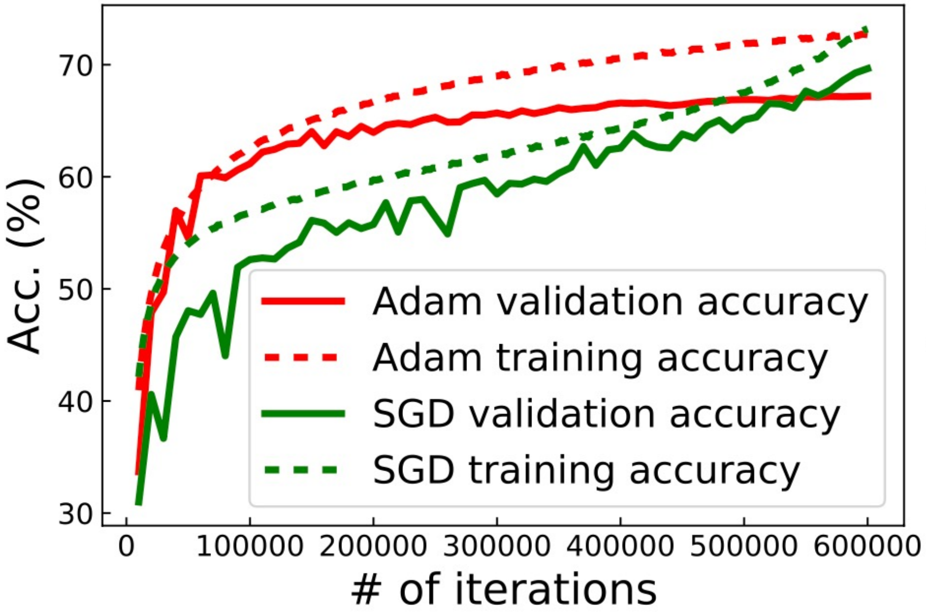
In this work

- Enhance the performance of state-of-the-art ReActNet from 69.4% to 70.5%.
- Understand BNN optimization

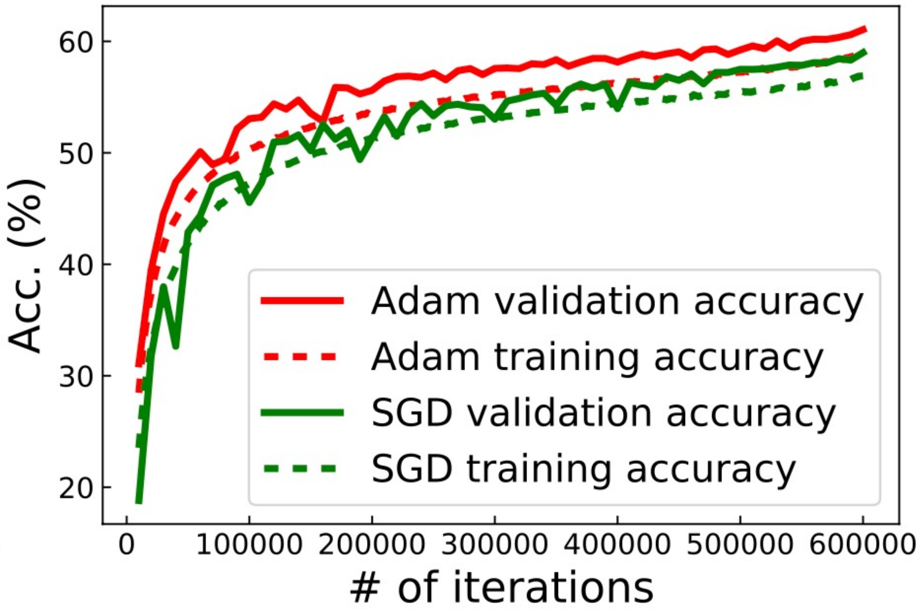


Motivation

- Real-valued network: SGD > Adam, usually use SGD
- Binary neural network: Adam > SGD, more recent works use Adam



(a) real networks

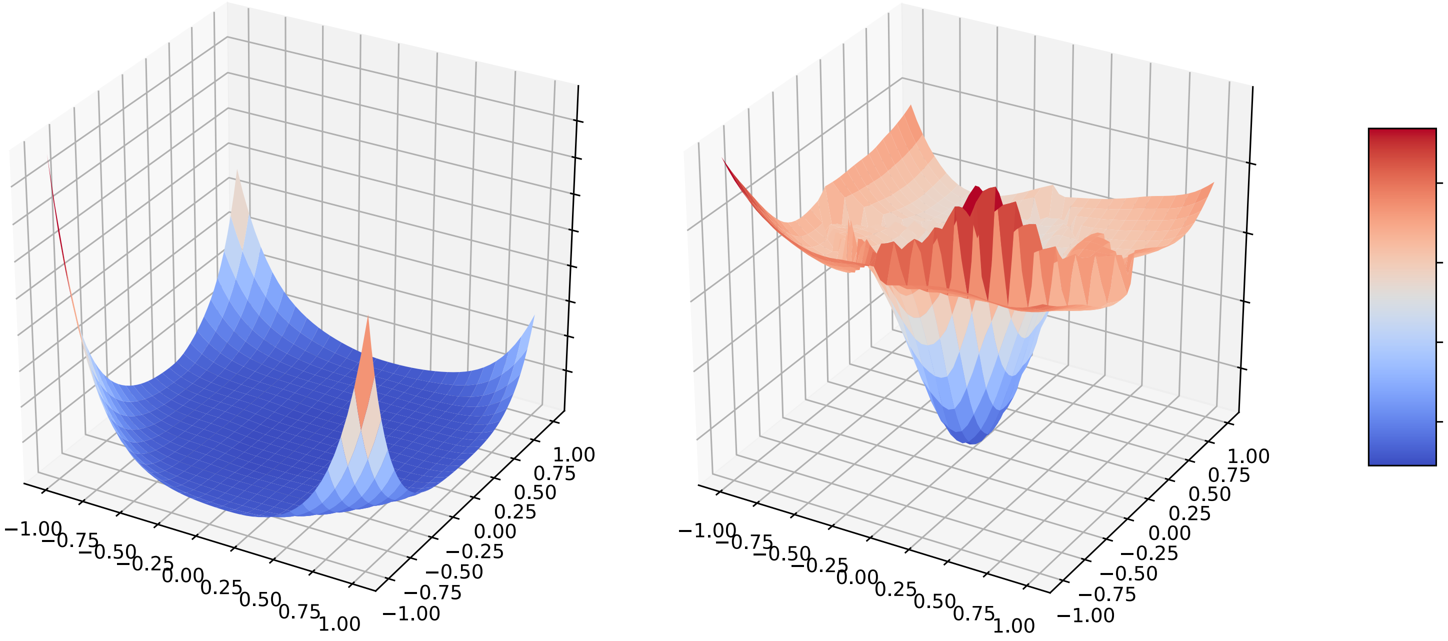


(b) binary networks

How Does Adam Help BNNs Optimization (ICML 2021)

Observation – loss landscape difference

- The actual optimization landscape from real-valued and BNNs



(a) real networks (b) binary networks

How Does Adam Help BNNs Optimization (ICML 2021)

Activation binarization

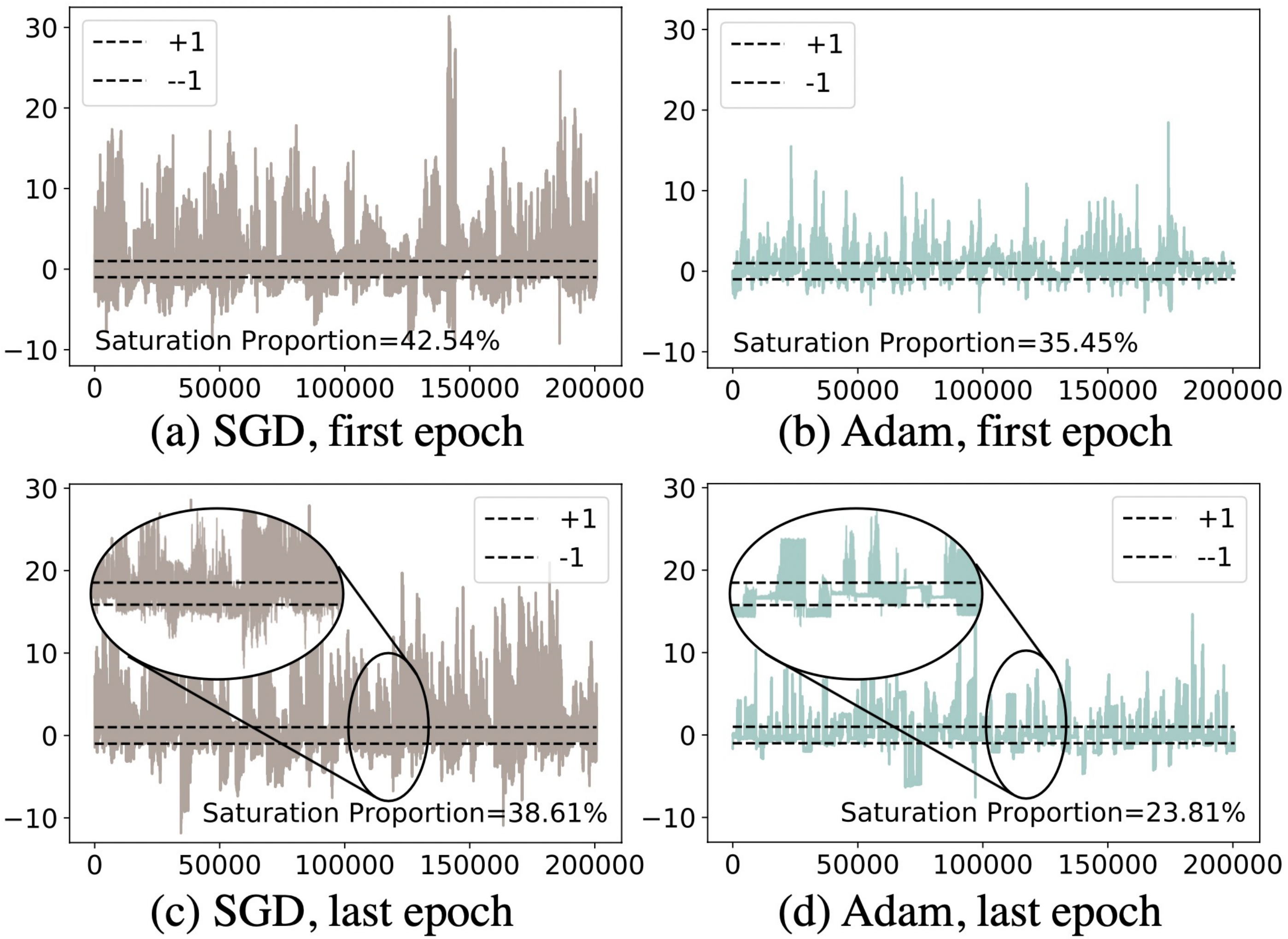
- Forward pass – cause the discretized landscape

$$a_b = \mathrm{Sign}(a_r) = \begin{cases} -1 \text{ if } & a_r < 0 \\ +1 \text{ otherwise} \end{cases}$$

- Backward pass – cause the activation saturation and zero gradient issue

$$\frac{\partial Sign(a_r)}{\partial a_r} \approx \frac{\partial Clip(-1, a_r, 1)}{\partial a_r} = \begin{cases} 1 & -1 < a_r < 1 \\ 0 & \text{otherwise} \end{cases}$$

Activation saturation



How Does Adam Help BNNs Optimization (ICML 2021)

Why Adam can alleviate activation saturation

- SGD update: $v_t = \gamma v_{t-1} + g_t$

- Adam update: $u_t = \frac{\hat{v}_t}{\sqrt{\hat{m}_t + \epsilon}}$ $\quad m_t = \beta m_{t-1} + g_t^2$

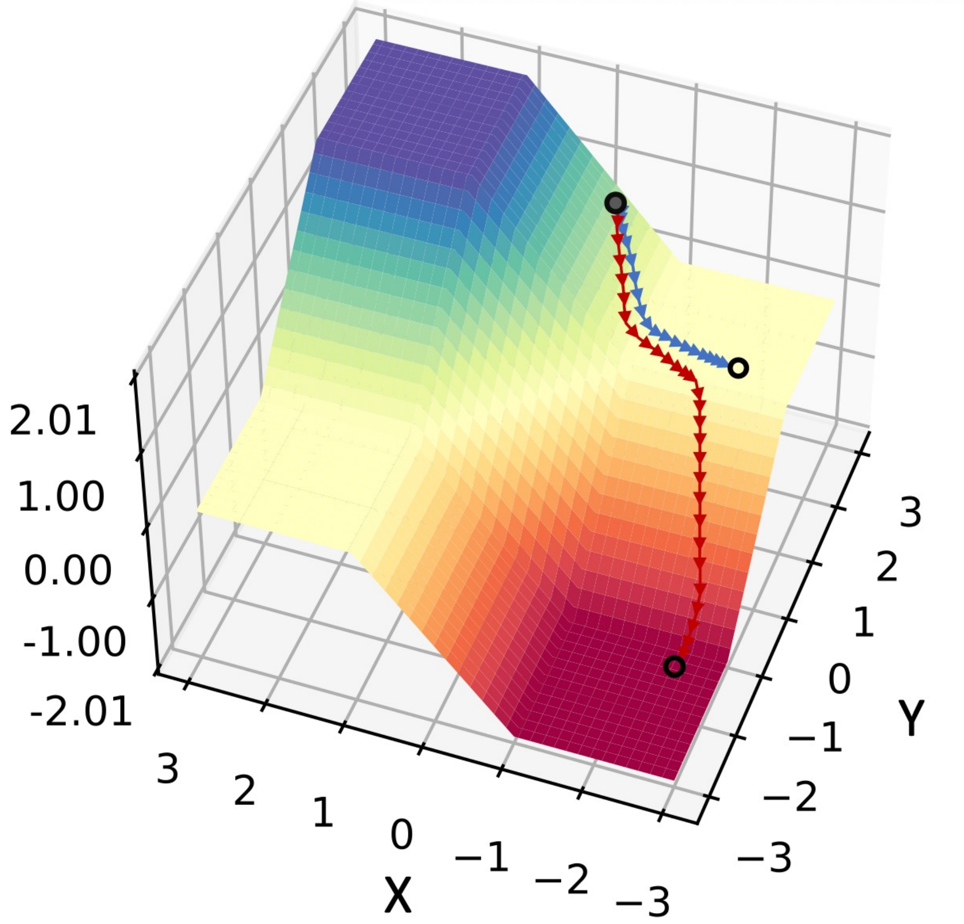
- Adam naturally leverages the accumulation in the second momentum to amplify the learning rate regarding the gradients with small historical values.

How Does Adam Help BNNs Optimization (ICML 2021)

SGD vs Adam

- SGD update: $v_t = \gamma v_{t-1} + g_t$

- Adam update: $u_t = \dfrac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$

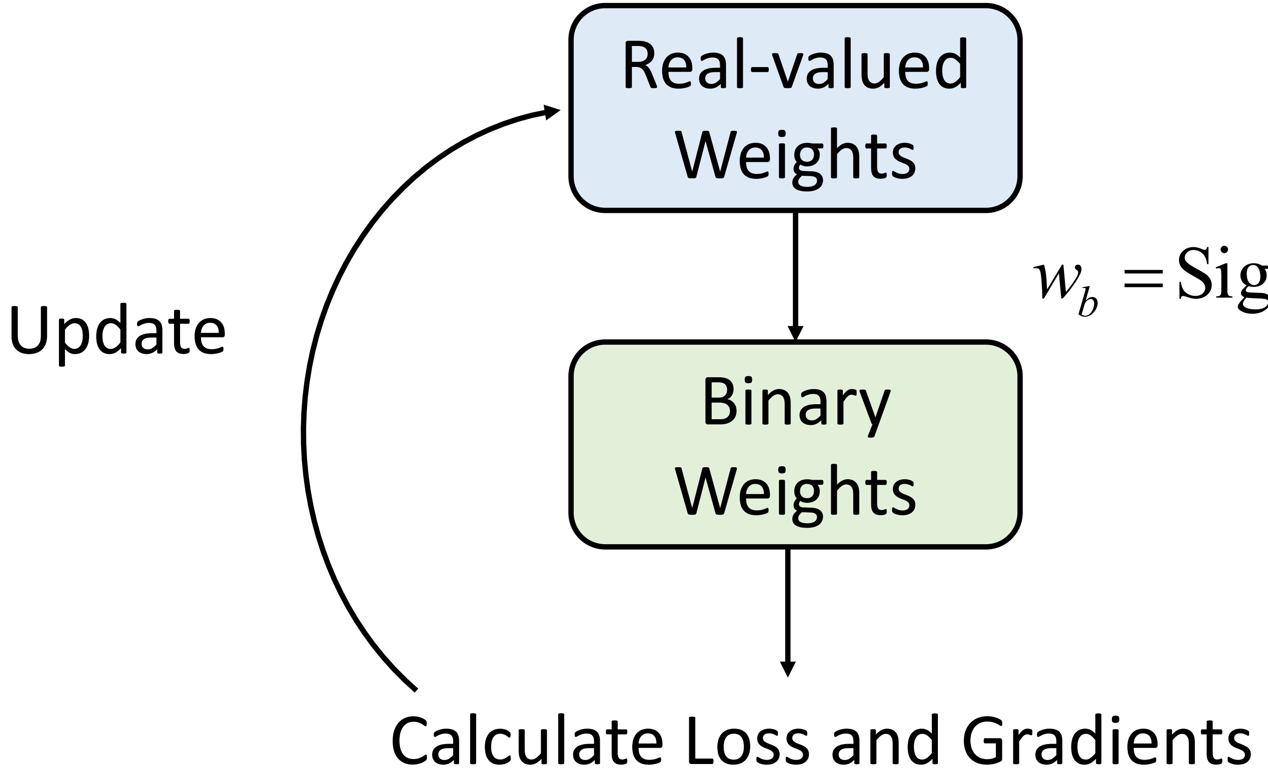


(c)

How Does Adam Help BNNs Optimization (ICML 2021)

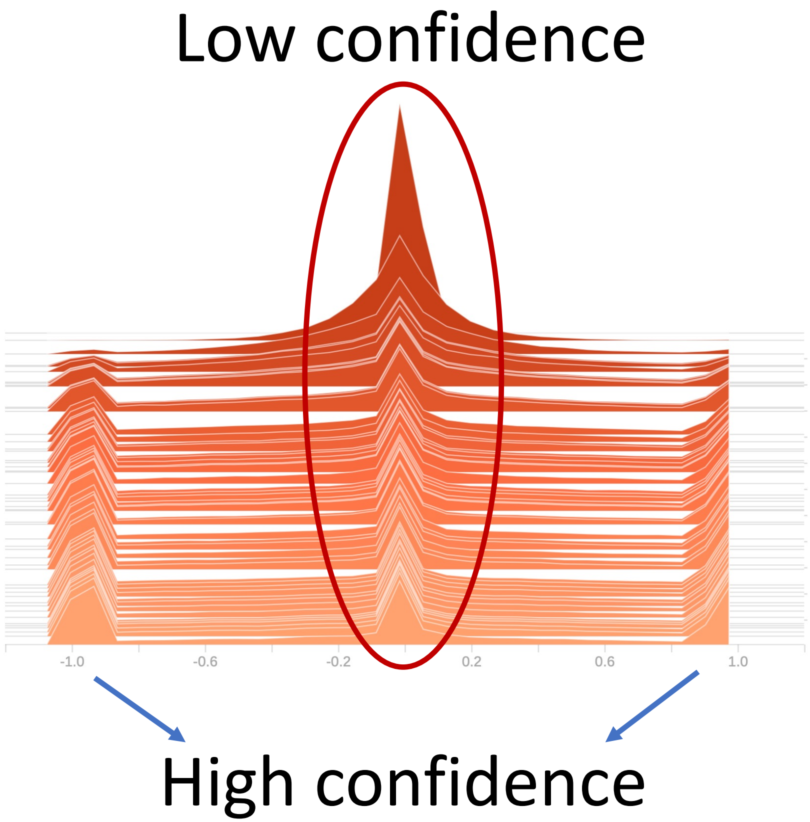
Role of real-valued weights

- Weight binarization and update process in the BNNs:



$$w_b = \mathrm{Sign}(w_r) = \begin{cases} -1 & \text{if } w_r < 0 \\ +1 & \text{otherwise} \end{cases}$$

Real-valued weights distribution

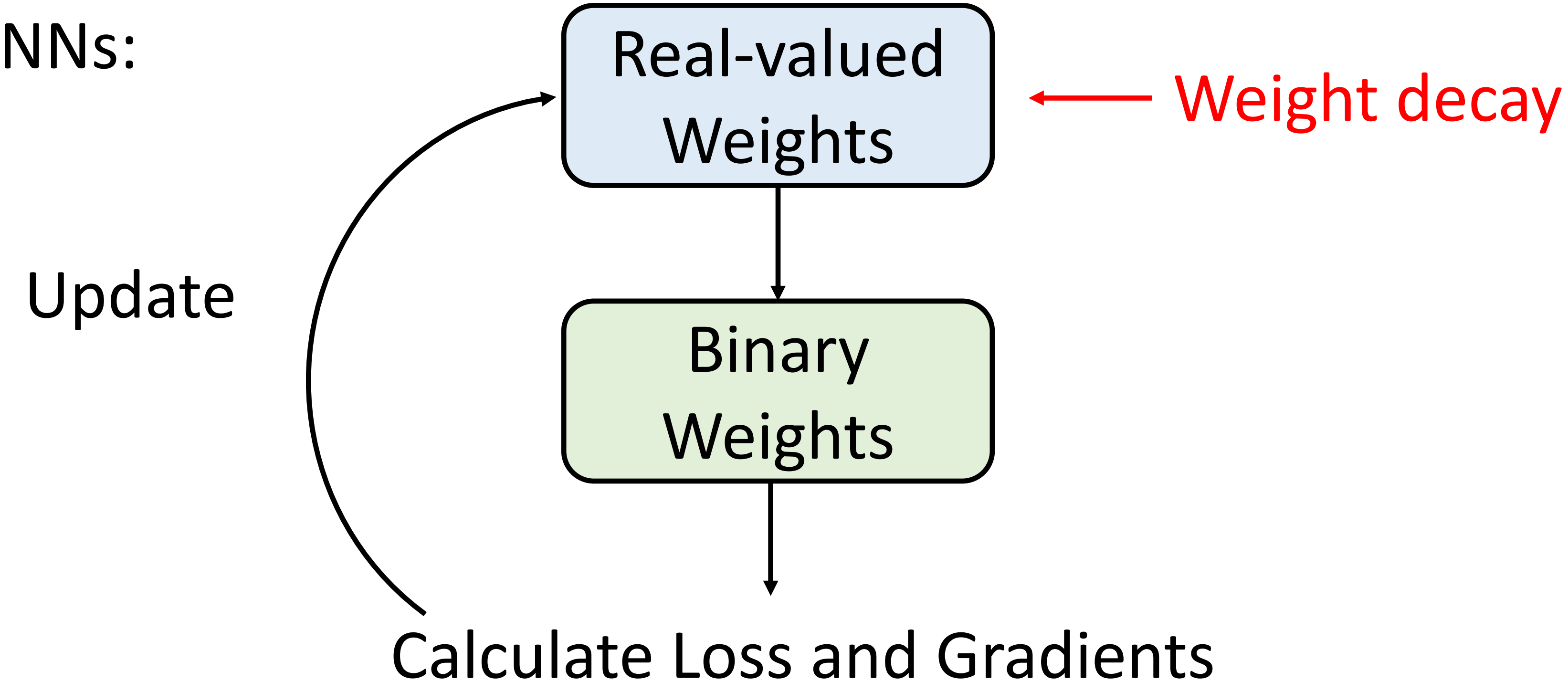


Visualization of final real-valued weights distribution in BNNs

How Does Adam Help BNNs Optimization (ICML 2021)

Another aspect in optimization: weight decay

- The role of weight decay in BNNs is tricky.
- In Real-valued network, weight decay prevents over-fitting
- In BNNs:



Weight decay: a dilemma in stability and initialization dependency

- High weight decay:

  Decrease the magnitude of real-valued weights

  Thus, binary weights are easy to change the sign and unstable.

- Low weight decay:

  Binary weights will be more stable to stay in the current status

  Tend to be largely depend on the initial value.

Two metrics to depict the effect

- FF ratio (optimization stability)

whether a weight changes its sign after the updating at t^{th} iter.

$$\mathbf{I_{FF}} = \frac{|\mathrm{Sign}(w_{t+1}) - \mathrm{Sign}(w_t)|_{abs}}{2},$$

$$\mathbf{FF_{ratio}} = \frac{\sum_{l=1}^{L} \sum_{w \in W_l} \mathbf{I_{FF}}}{N_{\text{total}}},$$

- C2I ratio (correlation-to-initialization)

whether a weight has different sign to its initial sign.

$$\mathbf{I_{C2I}} = \frac{|\mathrm{Sign}(w_{\text{final}}) - \mathrm{Sign}(w_{\text{init}})|_{abs}}{2},$$

$$\mathbf{C2I_{ratio}} = 1 - \frac{1}{2} \frac{\sum_{l=1}^{L} \sum_{w \in W_l} \mathbf{I_{C2I}}}{N_{\text{total}}},$$

How Does Adam Help BNNs Optimization (ICML 2021)

Disentangle the FF ratio and C2I ration

- Two step training:

  (1) Step 1: *Binarize activation, add weight decay*

  real-valued networks have no worry about the FF ratio

  (2) Step 2: *Binarize activation + weight, zero weight decay*

  Improve stability and utilize the good initialization from *Step1*

Experiments

- Dataset: imageNet
- Comparison with the state-of-the-art BNNs.

Table 2. Comparison with state-of-the-art methods that binarize both weights and activations.

Networks	Top1 Acc %	Top5 Acc %
BNNs (Courbariaux et al., 2016)	42.2	67.1
ABC-Net (Lin et al., 2017)	42.7	67.6
DoReFa-Net (Zhou et al., 2016)	43.6	-
XNOR-ResNet-18 (Rastegari et al., 2016)	51.2	69.3
Bi-RealNet-18 (Liu et al., 2018b)	56.4	79.5
CI-BCNN-18 (Wang et al., 2019)	59.9	84.2
MoBiNet (Phan et al., 2020a)	54.4	77.5
BinarizeMobileNet (Phan et al., 2020b)	51.1	74.2
PCNN (Gu et al., 2019)	57.3	80.0
StrongBaseline (Brais Martinez, 2020)	60.9	83.0
Real-to-Binary Net (Brais Martinez, 2020)	65.4	86.2
MeliusNet29 (Bethge et al., 2020)	65.8	–
ReActNet ResNet-based (Liu et al., 2020)	65.5	86.1
ReActNet-A (Liu et al., 2020)	69.4	88.6
StrongBaseline + Our training strategy	63.2	84.0
ReActNet-A + Our training strategy	**70.5**	**89.1**

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Carnegie Mellon University

Thank you






Zechun Liu*, Zhiqiang Shen*, Shichao Li, Koen Helwegen, Dong Huang, Kwang-Ting Cheng