FlexNet: Neural Networks with Inherent Inference-Time Bitwidth Flexibility

Yu-Shun Hsiao*, Yun-Chen Lo*, and Ren-Shuo Liu

Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan
yu_shun_hsiao@gapp.nthu.edu.tw, yunchen.lo@gapp.nthu.edu.tw, renshuo@ee.nthu.edu.tw

Convolutional neural networks (CNNs) recently emerged as a promising and successful technique to tackle artificial intelligent (AI) problems including image recognition and image generation. For example, CNNs can recognize a thousand categories of objects in the ImageNet dataset not only faster but also more accurate than people.

CNN is compute-intensive. Let us take AlexNet for example. It consists of five convolutional layers, and each layer demands 100 million to 450 million multiplications. Recognizing a small image (224 × 224) no larger than the App icons of smartphones (512 × 512) demands more than one billion multiplications in total, let alone the computing complexity of processing large images or videos.

Low-bitwidth CNNs exhibit significantly reduced computing complexity, and thus they recently attracted great attention and extensive studies. Low-bitwidth CNNs typically restrict themselves to utilizing one- to four-bit, fixed-point weight and activation values instead of floating-point values. For example, DoReFa-Net [4] proposes to set the bitwidth of weights and activations between one and four bits. Binarized neural networks (BNN) [1] and XNOR-Net [3] push the limit even further to one-bit and replace floating-point multiply operations with bitwise XNOR operations.

However, we observe a key limitation of the baseline low-bitwidth CNNs as well as their training methods that the bitwidth of these low-bitwidth CNNs has to be decided at training time, and then the bitwidth has no adjustment flexibility at inference time.

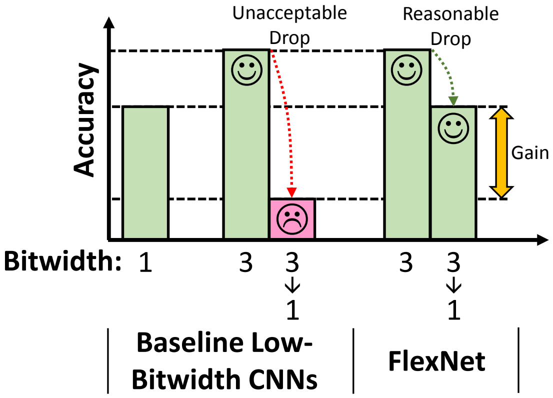

Fig. 1. Issue of the baseline, which is addressed by FlexNet

Figure 1 illustrates this limitation we observe. Let us take DoReFa-Net on ImageNet for example. One can separately train 1-bit and 3-bit CNNs, as denoted by the first and second bars in the figure. Intuitively, the 3-bit CNN achieves higher accuracy than the 1-bit counterpart, and this benefit comes at the cost of higher computing complexity (i.e., 3-bit multiplications vs. 1-bit multiplications). A less-intuitive phenomenon is that if one deliberately truncates the weights and activations of the 3-bit CNN to 1-bit at inference time, the accuracy does not hold at the level of a 1-bit CNN (the first bar) as one may expect; instead, the truncated CNN would be totally ruined, and the accuracy would be far lower than what the 1-bit CNN can achieve (denoted by the third bar) (In Figure 1 and later figures, $N \rightarrow K$ denotes truncating an N-bit CNN to a K-bit CNN.).

Lacking the above-mentioned bitwidth adjustment flexibility results in two drawbacks. First, different CNN accelerators support different computing capabilities. Some accelerators implement 1-bit multipliers, while others implement 2-, 3-, or 4- bit ones. Therefore, CNN users (e.g., smartphone manufacturers) need to rely on CNN model providers (e.g., Google or university laboratories) to release different versions of CNNs with differently trained bitwidths; otherwise, they cannot alter the bitwidth on their own. Second, some advanced CNN accelerators implement variable-bitwidth multipliers [2]. Since baseline low-bitwidth CNNs have no bitwidth adjustment flexibility, this type of accelerator is forced to store multiple versions of low-bitwidth CNNs in its storage (space overhead) and switch among these CNNs (power and performance overhead) to fully utilize the hardware flexibility.

Figure 4 illustrates the system diagram when FlexNet is utilized. This work makes the following key contributions.

- We identify the issue that the bitwidth of low-bitwidth CNNs cannot be freely adjusted at inference time. We also make cases for CNNs to have the flexibility.
- We propose FlexNet and its corresponding training method and successively achieve the flexibility.

Figure 2 and Figure 3 compare the top-5 ImageNet classification accuracy of FlexNet with baseline low-bitwidth CNNs.

* These authors contributed equally to this work.

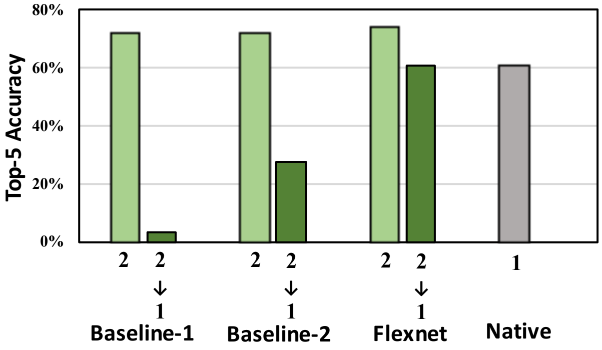

Fig. 2. Top-5 accuracy of 2-bit baselines and FlexNet

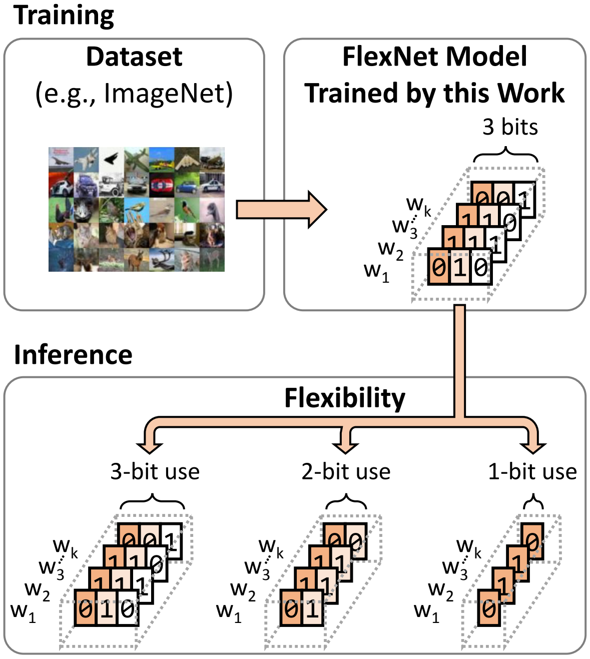

Fig. 4. System diagram when FlexNet is utilized

We train these CNNs with bitwidths set to two or three at training time and truncate their bitwidths at inference time. All of them are based on the same AlexNet architecture. Baseline-1 is trained according to DoReFa-Net [4]. Compared with Baseline-1, Baseline-2 is additionally allowed to retrain its batch normalization layers after bitwidth truncation is made.

As shown in Figure 2. After bitwidth truncation, the top-5 accuracy of Baseline-1 (3.08%), Baseline-2 (27.83%), and FlexNet (61.2%) is dramatically different. Only FlexNet can still exhibit a reasonable accuracy level. Note that FlexNet does not need any fine tuning or retraining after truncation. In comparison, the accuracy of baseline CNNs is totally ruined. Another thing worth mentioning is that we also train a native 1-bit baseline CNN, which achieves 61.2% top-5 accuracy. The truncated FlexNet can achieve the same 61.2% accuracy because of our proposed training method.

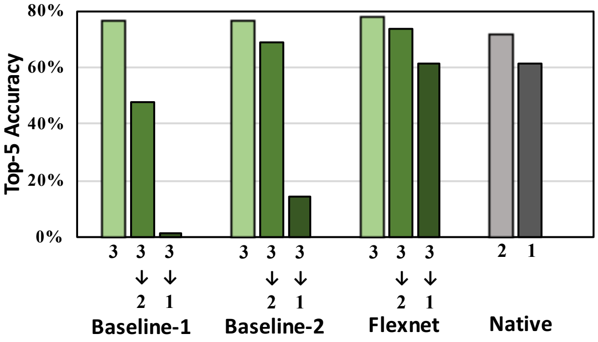

Fig. 3. Top-5 accuracy of 3-bit baselines and FlexNet

Figure 3 presents the impact of truncating the bitwidth of CNNs from 3-bit to 2-bit and 1-bit. FlexNet still significantly outperforms the two baseline counterparts. If the bitwidth is truncated from 3-bit to 2-bit, FlexNet (73.88%) outperforms Baseline-2 (68.83%) by 5.05%. If the bitwidth is further truncated to 1-bit, FlexNet (61.2%) outperforms Baseline-2 (14.2%) by 47%.

Acknowledgment

We thank the anonymous reviewers for their valuable feedback. This work is supported in part by the Ministry of Science and Technology (MOST) of Taiwan under grants 107-2218-E-007-034, 107-2218-E-007-001, and 106-2221-E-007-125. We also thank the National Center for High-performance Computing (NCHC) of Taiwan for computer time and facilities.

References

[1] M. Courbariaux and Y. Bengio. BinaryNet: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:/1602.02830*, 2016.

[2] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. J. Yoo. UNPU: A 50.6TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision. In *International Solid-State Circuits Conference (ISSCC)*, 2018.

[3] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[4] S. Zhou, Z. Ni, X. Zhou, H. Wen, Y. Wu, and Y. Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.