Convolutional Neural Network Models and Optimization Design for Edge Computation

Buwen Liu*

Department of Electronic Science and Engineering, Southeast University, Nanjing, 211189, China

*Corresponding author: 213202725@seu.edu.cn

Abstract. Convolutional Neural Network (CNN) has been widely implemented to conduct artificial intelligence (AI) projects. Because of its various advantages, CNN plays a significant role in daily applications of AI like automatic driving, semantic segmentation, face identification etc. However, as the application scope expands, many conventional CNN modules met their obstacles, most essential of which is the high memory consumption and abundant computation source expense. For edge computation areas like embedded system and other source or power supply limited devises, many of the CNN modules are unacceptable. From another point of view, not all the time does CNN need to do full precision calculation. To explore more possibility of CNN, low-precision computation and binary neural network (BNN) which occupy less resources than traditional CNN are proposed. The central computation of CNN is multiply-accumulates (MACs), so any advanced architecture designs should base on the acceleration of MAC. Benefiting from customizable data width, reconfigurable logic resources and low power consumption, FPGA is an appealing platform to implement low-precision computation and BNN. The development history and applications of CNN and its limitations was introduced. Comparison was made of different platforms to implement CNN and specific designs of low-precision computation conducted on FPGA. The contemporary accelerating platforms for FPGA which are technically designed to decrease the energy consumption, latency and facilitate performance were reviewed.

Keywords: CNN, low-precision, Edge application, optimization.

1. Introduction

In 1998, YANN LECUN et al. [1] proposed a new concept named Convolutional Neural Network which is frequently applied in many projects and often called CNN at present. Before CNN was proposed, gradient-based learning technique was the most popular, accurate, and efficient method to conduct pattern recognition tasks such as hand-written recognition and other similar objects. The back-propagation algorithm can construct a complex decision surface after a specific neural network structure is settled that can distinguish patterns from higher dimensions without much preprocessing work. YANN LECUN proposed the first specific CNN model called LeNet-5 and successfully applied it. Due to its relatively complete structure including convolutional layer, pooling layer and fully-connected layer, LeNet-5 is regarded as the first CNN model. In the image classification assignment, graph transformer networks (GTN), as a brand-new architecture taking LeNet-5 as its core, can synthesize multiple assignments like segmentation, field extraction and many other tasks, and train them simultaneously according to a Gradient-based minimization of an overall performance indicator. During the experiments, GTN was the most outstanding model among all the tested techniques [1].

CNN outperformed in the beginning of 21th century, but there were lots of deficiencies on the first edition. Also, many modern applications were still unavailable despite the best capability by the time. Things started to change in 2012 when AlexNet became the champion of ImageNet competition. Compared with LeNet-5, AlexNet has deeper network constructions. It consists of five convolutional layers, three fully-connected layers and one softmax layer which can predict the probability of each category. Softmax can normalize and convert the original data from the precious layer into values between zero and one which can present probability distribution and used as target prediction values for multiple classifications [2].

In 2014, Jonathan Long et al. [3] defined the structure of Fully Convolutional Network, explains the application of FCN to spatially intensive prediction tasks and gives its connection with other previously proposed networks. In the previous segmentation methods [1,2], there are two main disadvantages. First, the segmentation based on image blocks is inefficient, often requiring more processing procedure before or after the main step (such as superpixel). Second, the problem of internal tension in semantics and position is essential to semantic segmentation: global information solves what while local information solves where. To deal with this problem, Jonathan Long reconfigured the classification network into FCN structure including the conversion of fully connected layer into convolution layer and upsampling by deconvolution, and applied transfer learning to fine-tune full convolutionally to increase the learning efficiency [3]. Finally, researchers defined a skip architecture synthesizing appearance information and semantic information from a shallow, fine layer and deep, coarse layer respectively to fabricate a subtle, exhaustive segmentation [3]. Besides these academic progresses, Long also occupied some new methods to get a better training result, the most important of which is the implementation of ReLu which can better correspond to the principles of imitation biology, leading to more efficient gradient descent and back-propagation avoiding gradient explosion and gradient disappearance problems [3].

Except the three CNN models mentioned above, there are lots of other famous and great models evolving in many specific researching areas suitable for given tasks. Deep Belief Network (DBN) was systematically discussed in [4] and [5] in 2006, just before the breaking out of deep learning. As an efficient method to initialize weights for deep autoencoder to learn codes at low dimension, which exceeds principal component analysis in data dimensionality reducing assignments [5], DBN plays as a fundamental theory of the upcoming generation of deep learning. In 2014, VGG emerged [6]. Simonyan and other researchers mainly adopted the way of adding more convolution layers to deepen the network of VGG, results implying that the deeper network is, the better it will performance in learning and classification capacity [6].

With the applying domains expanding, many shortcomings of contemporary CNN models emerge and slow down the developing speed of CNN. The fundamental difference in vision recognition principles causes CNN lots of obstacles in its way to real human-like vision, making it easily disturbed by little invisible image noise. For edge applications, existing CNN models usually consumes massive energy in training stage.

Researchers have made great efforts to reduce the power consumption. In software, a new method in training stage is generated to make CNN fit for low-precision computation and low quantization error demands. In hardware, as FPGA is widely accepted as an edge application platform, researchers restructure on-chip logic resources and orient the new architecture to low-precision computation for final implement on edge computation.

2. The Limitations of Existing CNN Models

2.1. Different Cognitive Methods

Although CNN has drawn tremendous attention on itself and proved its infinite potential in image recognition and AI area, there are still lots of significant difficulties in the training and learning stage, some of which will probably be dangerous if not properly addressed.

In computer vision (CV) region, CNN models applied could accomplish many tasks with excellent performance, but no matter how fantastic work they can handle, the fundamental operation logic is totally different from that humans follow. For computers, any image is just a bunch of data or, more accurately, matrix consisting tremendous elements which compose all pixels and finally forming a complete picture, and it's naturally difficult for computers to deal with the data differences, which is also the most essential challenge. When primary characteristics are extracted from an objective or a image, humans can observe an object in various and changing perspectives, meaning that they can always recognize a hamburger no matter it's complete or incomplete or blocked partially, no matter it's upright or upside-down or rotated by a specific angle.

However, it seems not to be a simple task for computers. A well-trained CNN could recognize a certain object despite the changing position, but that seems to be limited to a fixed posture. Once the posture is changing, even at the same position, computers will probably fail in distinguishing. Fig. 1 shows a typical adversarial example where CNN probably performs badly. Jonathan Long [3] tried to facilitate the capability of generalization recognition during training stage by data augmentation. It's a method to expand data set and generate more posture variety of the same objects. Researchers reverse or rotate some images and train CNN models on this extended back-up copy in which every component is a little different to another one. This training will help CNN more accurately extract the characteristics and gain higher robustness. But it's method cannot cover every possible status of an image especially those extreme situations like an upside-down chair or a folded T-shirt.

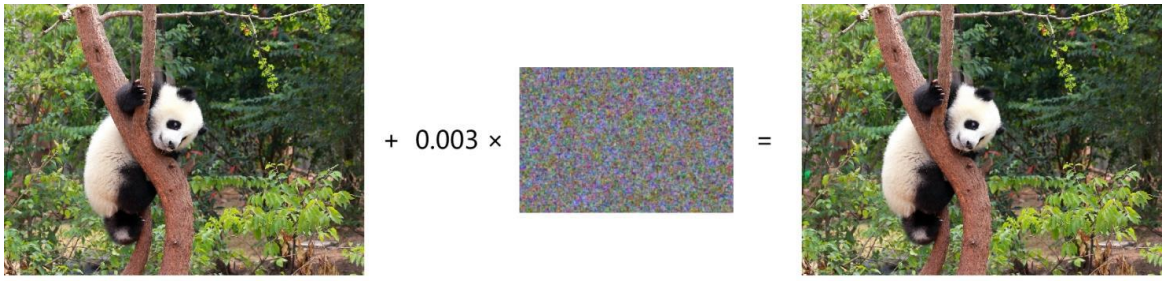


Fig. 1 Different recognition principle may lead to weird result.

2.2. Climbing Consumption

If CV is taken as a representative example of CNN, it's easy to tell that the cost to more accurate and more general application scenarios is higher amount of computation, higher precision of computation, much more data leading to higher memory expense and finally more consumption of energy and power supply. In the previous work, AlexNet [2] and VGG [6] were developed by adding more layers to networks and achieved practical progress in both laboratory and industry. But this not sustainable since the computation ability of any chip will always meet the extreme while the increasing precision followed by increasing amount of calculation could be theoretically infinite. Meanwhile, according to existing research achievements, only increasing the complexity of networks will lead to a series of problems like overfitting and network degradation, implying that contemporary network architectures need further optimization [7].

On one hand, CNN's computation amount could be unbearable for contemporary hardware, meaning that simply increase the calculation density is not wise. On the other hand, CNN doesn't need high precision computation at every stage of applications, such as inference stage [8]. Researchers wish that CNN could be applied in as many scenarios as possible, including many edge terminals like smart phones, vehicle systems and some Internet of things (IoT) devices. Although these applications circumstances are mostly limited by logic resources and power supply, they don't need high computing density as well. So, a practical method of breaking through obstacles is carefully selecting platform and redesign the architectures.

3. Acceleration Design for Edge Application

Since algorithms and particular architectures are based on specific acceleration platforms in order to achieve the best general performance, there should be a comprehensive review and discussion on platforms before the final decision. There are four types of chips most widely used to deploy CNN applications, CPU, GPU, ASIC and FPGA.

CPU is still the main computation unit in data processing, due to its flexibility. But may not be able to meet the efficiency demand in every application since it usually doesn't contain many

operation cores, resulting in the lack of ability of parallel computation. Contrary to CPU, GPU has massive parallel computation units. This ability directly promoted the prosperity of CNN in the last decade. However, the remarkable improvement is at the cost of tremendous soar in power consumption. ASIC could achieve both highest performance, highest efficiency, and lowest power consumption, but one architecture of ASIC could conduct only one task. Meanwhile, the developing period and risk could let many investors flinch.

The last is FPGA. It is a meaningful compromise between flexibility and operation efficiency. Just a little transformation on structure is needed when FPGA conducts different assignments. Also, compared to GPU, FPGA has higher power efficiency, making FPGA suitable for edge applications.

Reconfigurable logic resources, customizable data bit-width, low power consumption and multiple computation resources are the most attractive features of FPGA in CNN acceleration [8]. Researchers have achieved something in FPGA architecture optimization for CNN acceleration.

3.1. Reduction in Quantization Error Values in Low-precision in Training Stage

Training stage consumes most logic and memory resources when implementing CNN. Prior quantization research focused mostly on ratiocination phase, but more recently, relevant researches on CNN has paid more attention to the training stage as well. As a result of the security considerations of a CNN model which is optimized under users' circumstances, there are growing demand for CNN models whose training elements are based on the edge circumstances [9]. Sungrae Kim et al. proposed an innovative method using two cycles of bits-width allocations to train low-precision models that allocate diverse bits-width accuracy to weights and activation dynamically while ensuring that in each iteration the sum of two bit-widths is equal to 12 (e.g. 8-bit weight accuracy with 4-bit activation or 7-bit weight accuracy with 5-bit activation etc.). Sungrae Kim et al. demonstrated that double-cyclic training can reduce the quantization error to a great extent in every training cycle theoretically and proved that if weights are allocated to higher accuracy and activations possess lower accuracy, the result performed the best [9]. According to experiments, double-cyclic precision-based MAC operators use less power than the FP32 by around 92.8%, naïve W8/A8 by 62.4% and those MAC operators simply trained on single cycle by 25% [9].

3.2. DSP Redesign for Low-precision Computation

Just as it was mentioned above, low-precision computation is suitable for resources limited conditions. The key point of any design is accelerating MAC in low-precision computation [8] which is fitting for the DSP module to perform. First, Boutros et al. [10] created a base-line DSP that resembles the Arria-10 [11] FPGA model from Intel and can multiply two 18-bit numbers by one 27-bit number. This original model has 4 multipliers: M1 for 18×18, M2 for 9×9, M3 and M4 for 9×18. If a 27-bit multiplication is needed, this model will divide the formula into four parts, and the structure will be transformed:

$$A[26:0]\times B[26:0]=(A[26:9]+A[8:0])\times(B[26:9]+B[8:0]) \tag{1}$$

In this way, the 27-bit multiplication is taken apart by multiplication distribution law and each subordinate multiplication can be delivered to corresponding multiplier according to the bit-width and finally add all results together. For two 18×18 multiplications, one could be sent to M1, the other will be divided into two 9×18 multiplications first, then sent to M3 and M4, leaving M2 unused.

For further low-precision computation, Boutros et al. [10] proposed three methods to reconstruct these four blocks to make them support 4-bit computation: the first design is to split up all four modules into individual 4-bit multiplication units. The second one involves adding four extra 4-bit multipliers and fracturing M2 and M3 which are not on the critical path. The last one entails splitting up M2 and M3 and adding two more 4-bit multipliers to the rest modules respectively [10]. Fig. 2 shows a simple sketch of the baseline and enhanced DSP blocks [8]. In experiments, researchers implemented different models at 8-bit and 4-bit, the results of which show imply that the enhanced DSP could perform 1.62× and 2.97× better than the original model respectively [10].

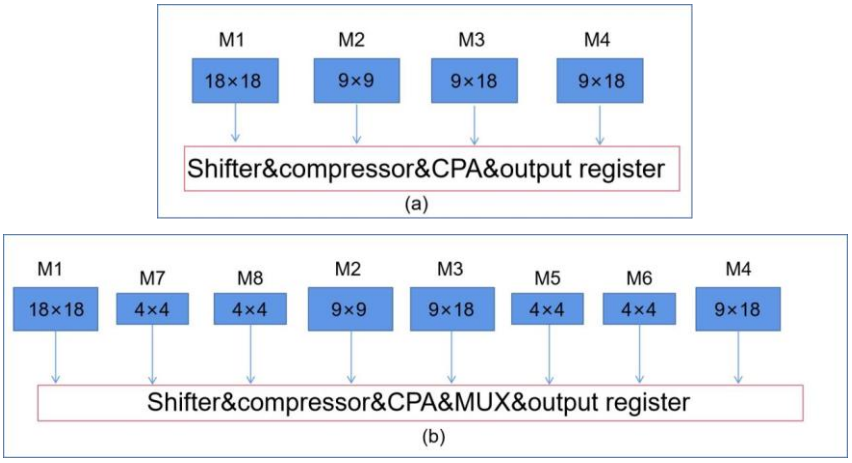


Fig. 2 Baseline DSP block (a) and enhanced DSP block (b) [8]

3.3. ALM Design for Low-precision Computation

To break the obstacle that high accuracy always leads to high power consumption and get FPGA chip mostly utilized in CNN acceleration tasks, it is of great significance to reconstruct the ALM/CLB blocks which is occupies the most area on FPGA. Boutros et al. [12] restructured the LUT units of FPGA in three different ways: The first is the ALM with an additional carrier chain design that have been suggested. Second one is the architecture consisting adders of 4-bit for ALM, fracturing each 4-input LUT into two 3-input LUTs and contains two more complete adders and multiplexing as the second architecture improvement. This allows ALM to conduct 8-bit precision computations and use additional adders. The final architecture improvement is logical array block with a shadow multiplier aiding. These enhancing designs can improve the density of MAC operation on-chip. According to the experiments, the best performance could reach up to 6.1× MAC density elevation by shadow multiplier combined with 4-bit adder architectures at the cost of larger area expense and higher critical path latency delay [12].

3.4. Binary Neural Network

In 2016, Courbariaux et al. [13] proposed the concept of binary neural network (BNN). In this model, weights and activations are constrained and represented to either +1 or -1 by binarization function which is a simple sign function. If the real value of a variable is not negative, the binarized result is +1, and the rest variables will all be constrained to -1. In this way, conventional CNN multiplication is transformed into XNOR operations, and accumulation is transformed into popcount operation which is just counting the number of ones [8]. With regard to sign function's feature that its derivative is zero almost everywhere, Courbariaux et al. designed straight through estimator (STE) which could skip the sign function or set the default value of the gradient between output and input as 1. After the use of the STE, the real-valued weights can be updated with a common optimizer such as SGD. While considering that the boundary is not set to the real value of weights, possibly making the accumulation infinite, Courbariaux et al. applied a function, Clip(), to clip the weights and constrain them between -1 to +1. In the experiments, the specially designed GPU kernel running MNIST MLP performed 7 times better than general GPU kernels with negligible loss in accuracy [13].

In terms of XNOR and popcount operations, BNN is highly suitable to FPGA and other resources limited environments. R X Zhou et al. [14] implemented CNV model on PYNQ-z2 to realize a BNN hardware circuit algorithm, trained it with CIFAR-10 vehicle data set and finally constructed an efficient vehicle classification system. CNV is a convolutional network topology inspired by CNN and VGG-16. In the first place, researchers divided 50000 pictures into 1000 batches and adopted mini-batch gradient descent, updating parameters by batch. CNV was primitively deployed on GPU and achieved the accuracy of 89%. Then, after the values of weights were binarized, researchers applied CNV on PYNQ-z2 and finally constructed a vehicle classifier of expected accuracy [14].

4. Conclusion

In this article, an overview on the development, exiting deficiencies of CNN and optimization designs on edge applications featuring low-precision computation on FPGA is delivered. Historically, it was not until CNN was proposed that neural network started to widely draw attention from researchers and after DBN came out, deep learning commenced with rapid development and evolution. As the most recent researches and products with regard to CNN show, market is getting more concerned with the application of CNN on edge devices since these are the closest applications to the most consumers who possess the strongest consuming potential. So, it's still a long-term developing direction of CNN to perform better on resources-efficient environments, implying that low-power designs like low-precision computation and BNN are especially promising. In natural language processing and other conventional areas, CNN has formed substantial foundation both theoretically and in practice, and with the existing basis and technology maturing further, CNN could be expected to be deployed in more domains like AI in computer games, artistic creation information processing in biology and even financial field, revealing the infinite possibility of CNN to help humans in an increasing number of regions.

References

[1] Lecun Y, Bottou L. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[2] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks. Advances in neural information processing systems, 2012, 25(2).

[3] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(4):640-651.

[4] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 2006, 18(7):1527-1554.

[5] HINTON GE, SALAKHUTDINOV RR. Reducing the dimensionality of data with neural networks. Science,2006,313(5786):504-507.

[6] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science, 2014.

[7] Li Yandong, Hao Zongbo, Lei Hang. Survey of convolutional neural network. computer application, 2016, 36(009):2508-2515.

[8] Li Zhengjie, Zhang Yufan, Wang Jian, Lai Jinmei, A survey of FPGA design for AI era. J. Semicond., 2020, 41(2): 021402.

[9] S. Kim and H. Kim, "Power-Efficient Double-Cyclic Low-Precision Training for Convolutional Neural Networks," 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Korea, Republic of, 2022, 344-347.

[10] Boutros A, Yazdanshenas S, Betz V. Embracing Diversity: Enhanced DSP Blocks for Low-Precision Deep Learning on FPGAs, 2018 28th International Conference on Field Programmable Logic and Applications (FPL). 2018.

[11] Intel Arria 10 core fabric and general purpose I/Os handbook. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/arria-10/a10_handbook.pdf, 2018.

[12] Boutros A, Eldafrawy M, Yazdanshenas S, et al. Math Doesn't Have to be Hard: Logic Block Architectures to Enhance Low-Precision Multiply-Accumulate on FPGAs, 2019 ACM/SIGDA International Symposium. ACM, 2019.

[13] Courbariaux M, Bengio Y. BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. 2016.

[14] R X Zhou, Z Y Wang, Z Y Yuan. Design and implementation of deep learning circuits based on FPGA. Science and Education Guide-Electronic edition (late half), 2022.