Online Compressible Neural Network

Seungmo Seo[1], Sunho Lee[3], Bohyun Kim[4], Joohee Jung[5], Jongwon Choi*[1,2]

Department of Artificial Intelligence, Chung-Ang Univ[1]

Department of Advanced Imaging, Chung-Ang Univ[2]

Future Innovation Technology Center, Ministry of National Defense[3]

7th Logistics Support Command, Republic of Korea Army[4]

Education & Training Command, Republic of Korea Airforce[5]

seosm@vilab.cau.ac.kr, triwally@alumni.kaist.ac.kr, bohyuni4@naver.com, ttutoo@kakao.com, *choijw@cau.ac.kr

온라인 압축가능 인공 신경망

서승모 [1], 이선호 [3], 김보현 [4], 정주희 [5], 최종원*[1,2]

중앙대학교 AI 대학원 [1], 중앙대학교 첨단영상대학원 [2], 국방부 미래혁신연구센터 [3], 대한민국 육군 7 군수지원단 [4], 대한민국 공군 교육사령부 [5]

Abstract

With the recent advancement of deep learning, the requirements for hardware resources have also increased significantly. Many existing compression methods retrain the compressed model with the labeled datasets, so we cannot compress the model in the test environment where the label information cannot be acquired. We proposed a novel scheme to compress the pre-trained model without the label annotation in the inference phase. To implement the unsupervised compression, we obtain the pre-trained model by using a scheme of stochastic depth, which can handle the new data even with several missing blocks. Then, for searching the optimized network which has reduced size, we propose the iterative scheme to find the compressed model efficiently. We validate our proposed method on the classification benchmark dataset (Cifar-10), and our framework successfully compresses the deep learning model without the necessity of label annotation.

1. Introduction

Nowadays, deep neural networks have been applied in various industrial area such as self-driving cars, mobile phones, and many others [1]. Among these industrial fields, the attempts to apply deep neural networks to various devices have increased. However, the large model size and computational cost cause great obstacles for many applications, especially on some constrained devices with limited hardware resources [2].

To handle this issue, the existing studies such as network compression [3], and knowledge distillation [4] only focus the optimizing tiny network within the provided devices. These approaches require additional training steps, and another problem arises that trained networks cannot be applied to online strategy.

In these regards, we propose an online method that make the neural network adaptable to the target device. Based on the residual network [5], we first develop the compressible network which can skip or reuse its residual blocks. We train the neural network dynamically, so the compressed network can estimate its prediction while preserving the performance. At the end of the training, the compressible network is converted to the optimized compressed network without additional training. In Cifar-10 dataset, we validate our method, and we find that our method can achieve the compressed neural network efficiently with online execution.

2. Compressilbe Network

To construct the compressible network, we use the proposed bypassing block and recycling block. Based on the residual block in [5], these blocks are developed for network compression with efficiency. We describe these blocks in detail as follows.

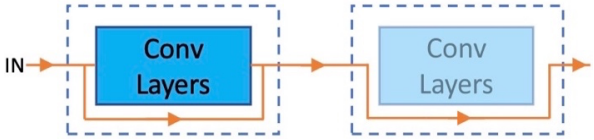

Figure 1. Bypassing Block Execution

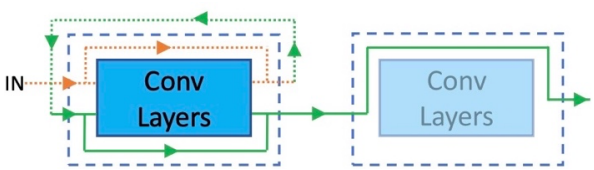

Figure 2. Recycling Block Execution

Bypassing Block. Compared to the original residual block which utilizes the identity residual connection, the bypassing block throws its input to its output directly through the skip connection [5]. By using the skip connection, the network can maintain the output channel shape. Since the bypassing block can ignore the estimation of the included layers in the original residual block, replacing the residual block with the bypassing block can reduce the model size.

Recycling Block. Instead of setting the input to the output through the residual connection, the recycling block feeds the initial input into the previous residual block to obtain the updated input. Then, the updated input becomes the output after the identity or the bottleneck residual connection of the recycling block.

While the bypassing block can reduce both the computation cost and model size, the recycling block does not result in the reduction of the computational flops. However, by using the recycling blocks, we can reduce the memory size of our model and make the block operations more diverse, which is helpful to find the optimal compression network in a wide range. We present the proposed block computations in Figure 1 and Figure 2.

3. Dynamic Network Training

To construct the compressible network, we use the proposed bypassing block and recycling block. Based on the residual block in [5], these blocks are developed for network compression with efficiency. We describe these blocks in detail as follows.

Network Training. In our method, we use stochastic depth to train the residual network. By using the stochastic depth, performance can be preserved even if the network is executed with fewer computation paths, which we called dynamic inference. The loss function with the stochastic depth applied is as follows;

$$L_{total} = E_{x \in X}[\mathrm{CE}(f^{p}(x), y)] \quad (1)$$

CE denotes the cross-entropy loss function [7], x means the input data, and y is the target label. We present the stochastic model path as f^{p}, and the path is determined by the probability p. We use the ResNet-18 [5] as the base model. For stochastic depth training, p is set as [0.0, 0.1, 0.2, 0.1] when ResNet-18 has [2, 2, 2, 2] bottleneck block combination. According to the probability set here, the residual blocks are converted into the two proposed block types (bypassing and recycling blocks). The conversion probabilities of the two block types are equal to the set p.

4. Online Inference with Compression

At the end of the training, we use the trained compressible network on the testdataset. For each batch of test data, the compressible network is executed multiple times.

Unlabeled Prediction. Before estimating the optimized compressed network, we apply the bootstrapping aggregation to obtain the optimal predicted output. Various paths determined by probability set p yield different predictions from the same output. However, because the accuracy of the output cannot be judged during the inference step, we average the output values from the various paths to estimate the accurate derived prediction.

Optimized Compressed Network. By using the accurate predictions we described, we can achieve the optimal compression path. We take 10 randomized paths according to the probability set p and calculate how similar the paths are to the unlabeled predictions we described. Cross-Entropy is used as a criterion for judging distributions from various compressed paths, and the path which has the most similar distributions to the unlabeled prediction is selected as the optimal path. We utilize the derived optimal path as a compressed network for online inference. Through this process, we find the optimal converted network which makes the best accurate predictions. By using the optimized compressed network, resources for obtaining a compressed network are reduced.

5. Experimental Result

To validate our method, experiments are conducted in open benchmark dataset. We recorded the accuracy of full and compressed models and the detailed inference times.

Implementation Details. We use the Cifar-10 dataset for the classification which consists of 10 class image data. We use the ResNet-18 [5] architecture and an SGD optimizer that the learning rate is set at 0.01.

Method	Accuracy
Full Model	88.65%
Compressed Model	72.34%

Table 1. Cifar-10 Result

Method	Inference Time
Full Model	0.00127
Compressed Model	0.00090

Table 2. Inference Time

We recorded the performance and inference time at the above tables. *Full Model* presents the performance of the non-compressed networks and the *Compressed Model* is the experimental results from the optimized compressed model followed by our method. From these result, it can be seen that our method estimate the predictions faster with the slightly low accuracy. The performance of the compressed network is notable even if the network has not been fine-tuned or additionally trained. Furthermore, the reduced inference time allows the network to be used in various devices.

6. Conclusion

In this paper, we propose an online adaptive neural network to fit the target device without additional training steps. By using our method, neural networks can be adaptable to the target devices. We develop a compressible network which can convert its residual blocks for reducing the model size and computation cost. From the validation experiment, we can show that our method achieves notable performance with a reduced computation cost. We argue that our proposed method provides a meaningful perspective as the online compression scheme.

ACKNOWLEMENT

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program(Chung-Ang University)) and (2022-0-00601, Military AI Development and Management Program).

REFERENCE

[1] Yu, Jiahui, and Thomas S. Huang. "Universally slimmable networks and improved training techniques." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[2] Guerra, Luis, Zhuang, Reid and Drummond. "Switchable precision neural networks." arXiv preprint arXiv:2002.02815. 2020.

[3] Kim, Hyeji, Muhammad Umar Karim Khan, and Chong-Min Kyung. "Efficient neural network compression." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[4] Gou, Jianping, et al. "Knowledge distillation: A survey." International Journal of Computer Vision. 2021.

[5] He, Zhang, Ren, and Sun. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.