MPEG-NNR 의 영상 압축을 위한 CNN 의 압축 표현 기법

문현철, 김재곤
한국항공대학교
hcmoon@kau.kr, jgkim@kau.ac.kr

Compressed Representation of CNN for Image Compression in MPEG-NNR

HyeonCheol Moon and Jae-Gon Kim
Korea Aerospace University

요 약

MPEG-NNR (Compression of Neural Network for Multimedia Content Description and Analysis) aims to define a compressed and interoperable representation of trained neural networks. In this paper, we present a low-rank approximation to compress a CNN used for image compression, which is one of MPEG-NNR use cases. In the presented method, the low-rank approximation decomposes one 2D kernel matrix of weights into two 1D kernel matrix values in each convolution layer to reduce the data amount of weights. The evaluation results show that the model size of the original CNN is reduced to half as well as the inference runtime is reduced up to about 30% with negligible loss in PSNR.

1. Introduction

Artificial neural networks are being widely adopted for a broad range of tasks in multimedia analysis and processing, media coding, data analysis, and many other fields. Many applications require the deployment of a trained network instance on a larger number of devices with limited computation power and memory. For such applications, interoperable and compact representations of neural networks are required. To address this issue, exchange formats have been developed that can interoperate in various deep learning framework and optionally apply compression. [1]. In addition, recently, MPEG is developing an interoperable compressed representation of neural networks called NNR (Compression of Neural Networks for Multimedia Content Description and Analysis).

In this paper, we present a low-rank approximation to compress a CNN used for video compression, which is one of MPEG-NNR use cases.

2. MPEG-NNR

MPEG activities on NNR aims to define a compressed, interpretable and interoperable representation for trained neural networks. Therefore, NNR summarized the requirements that depend on the use cases to which neural network is applied, and configured the framework to reflect this [2].

Figure 1 shows an MPEG-NNR framework in the use case of video compression. In the video compression case, neural network model is applied in a tool-by tool basis. In addition, an NN model applied to each coding tool is transmitted to the encoder or decoder in a form of coded representation. In addition, the acceleration library optionally optimized according to the given requirement such as compression rate, and the optimized networks are transmitted to the encoder or decoder.

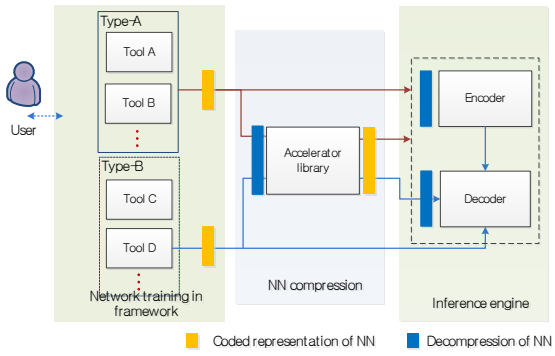

Figure 1. MPEG-NNR framework for video compression [2]

Figure 2 shows the framework for evaluation which meet requirement in the NNR. The framework considers the memory consumption in SW (O and R_size) and evaluation performance (O and R_per) as well as the size of model (Os and Cs size) to be transmitted. O_Per and R_Per should be as close as possible to

each other, and the memory to be transferred and the memory consumption on the device must meet the requirements of that each use case.

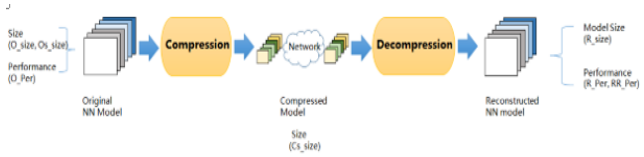

Figure 2. Evaluation framework in NNR [3]

3. Compression Methods

The low-rank approximation which applied to convolution layer is processed as follows [4]. The trained convolution weights in 2D-CNN are of 4D-types (filter size, filter size, the number of input channel, the number of output channel). Equation (1) shows the approximated equation in each channel. The dimension of W_{st} is expressed as d x d, where d is the filter size of convolution layer. In addition, the dimension of U_s, V_t which matrices to be approximated as a low-rank are expressed as d x R, where R is the target number of size for low-rank approximations.

$$W_{st} \triangleq U_s {V_t}^T, \quad (1)$$

So, the purpose of this method is to find (U, V) which give the minimum difference between W and $(\mathrm{U} \times \mathrm{V})$. For the filter reconstruction optimization, the cost function is as follows.

$$\min(U,V) = \sum_{S,T} \sum \|W_{st} - U_s V_t^T\|^2 \quad (2)$$

4. Experimental Results

The tool-by-tool case is the post-processing filter of VTM 3.0 with CNN. The test sequences were used in the JVET CTC [5] in all of class. In addition, testing environment is All Intra, the PSNR were compared with the original frame to compare the performances.

Figure 3 shows the CNN structure for post-processing filter in VTM 3.0

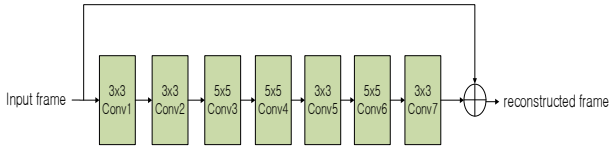

Figure 3. CNN structure for post-processing filtering

Table 1 shows the performance and model size results when applying the low-rank approximation method only to the 2nd and 3rd layers among the 7 layers. In the 2nd and 3rd layers, the number of input and output channel is all 64, and the number of target channels is set to 32, 24 to apply low-rank approximation. In addition, the NN model is defined as the .mat file.

Table 2 shows the comparison with the evaluation results and runtime against the original model into the low-rank approximation. In terms of PSNR, it has been found that there is generally loss of 0.02~0.03dB compared to the original CNN, depending on the number of layers to which the compression method is applied. In addition, the modified model with applied low rank approximation reduces not only the model size but also the runtime lowers (about 17% gain)).

Table 1: Model size of each compression method

Compression method	Model size (KB)
Original	692
LR-CNN: 2nd layer	495 (72%)
LR-CNN: 2nd, 3rd layers	341 (49%)

Table 2. Evaluation results (PSNR & Runtime)

Compression method	PSNR (dB)	Runtime (%)
VTM 3.0	38.90	-
Original CNN (Baseline)	39.95	100%
LR-CNN: 2nd layer applied	38.93	94%
LR-CNN: 2nd, 3rd layers applied	38.92	83%

5. Conclusions

In this paper, we presented the evaluation results on the compression of CNN in use case of video compression in MPEG-NNR. The results of the evaluation were presented in accordance with the procedure of the evaluation framework defined in MPEG-NNR.

The evaluation results showed that the low-rank approximation led a slightly loss in terms of PSNR, but the inference runtime reduced about 17%. Therefore, it is noted that the low-rank approximation would be a useful tool for network networks compression in the image compress use case.

ACKNOWLEDGMENT

This work was supported by National Standards Technology Promotion Program of KATS grant funded by Korea Government (MOTIE) (10084981).

References

[1] Available at [Online]: https://onnx.ai/

[2] W. Bailer, et al, “Use cases and requirements for compressed representation of neural networks,” ISO/IEC JTC1/SC29/WG11 N17740, July. 2018.

[3] W. Bailer, et al, “Evaluation Framework of Compression of Neural Networks for Multimedia Content Description and Analysis,” ISO/IEC JTC1/SC29/WG11 N18462, Apr. 2019.

[4] C. Tai, et al, “Convolution Neural Networks with Low-Rank Regularization,” In Proc. Computer Vision and Pattern Recognition (CVPR), Feb. 2016.

[5] A. Segall, et al, “JVET common test conditions and evaluation procedures for HDR/WCG Video Coding,” JVET document, JVET-D1020, Oct. 2016