Advanced Science and Technology Letters
Vol.140 (GST 2016), pp.186-191
http://dx.doi.org/10.14257/astl.2016.140.36

Hierarchical Compression of Deep Convolutional Neural Networks on Large Scale Visual Recognition for Mobile Applications

Byungjo Kim[1,1], Miyoung Lee[1], Jinkyu Kim[1], Juyeob Kim[1], and Joohyun Lee[1]

[1] Electronics and Telecommunications Research Institute (ETRI)
Gajeong-ro, Yuseong-gu, Daejeon, 305-700, South Korea
{kimbj, sharav, kimjk, juyeob, juehyun}@etri.re.kr

Abstract. This paper present hierarchical compression scheme in deep convolutional neural networks (DCNN) on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) for mobile devices. In hierarchical compression, we reduce the size of parameters and layers in VGG16 and GoogLeNet by using the iterative synapse pruning and grouping and quantization schemes hierarchically. We use the own simulation framework based on caffe to execute the hierarchical network compression and compare the test accuracy of reference model with compressed ones. The proposed scheme reduces the network parameter size to 91.61% and 67% in VGG and GoogLeNet, respectively. And with less 16bit fixed point parameters, we achieve good compression rates again, with only negligible loss of classification accuracy.

Keywords: DCNN; Convolutional Neural Network; Network Compression.

1 Introduction

Recently, DCNN can be one of promising technologies for object recognition and classification because every year DCNNs such as VGG-16/19, GoogLeNet have increased the visual recognition accuracy to the level of human's one.

As a result, DCNN will be actively used in application for object classification and detection in mobile devices, security systems, IoT devices.

But the number of weights and connection parameters in DCNN such as VGG Net and AlexNet is beyond the millions [1]-[2]. Since hardware implementation of DCNN on mobile device need connection complexity and excessive memory size, it is difficult to be implemented on mobile device with the limitation of computation resource and memory capacity.

To be implemented in actual mobile platforms, it is necessary to reduce the size of the network parameter. Recently, several researches are studied to decrease computation power and memory access, such as deep compression and quantization

ISSN: 2287-1233 ASTL

of weight and layer outputs [3]-[5].

The pruning method of CNNs with no loss of accuracy have been presented in [3]. Both weights and layer outputs in CNN are quantized while keeping the network's accuracy change below predefined target loss, compared to its 32-bit floating point counterpart [4]-[5].

In this paper, we present the efficient compression method to reduce the size of weights and layer outputs in DCNNs. We apply the iterative synapse pruning and grouping schemes, and then, quantize the compressed weight and layer outputs successively.

2 Hierarchical Compression

In this section, we describe about DCNN for large scale visual recognition and the hierarchically adaptive compression method for minimizing the number of parameters in DCNN.

2.1 Deep Convolutional Neural Networks for Large Scale Visual Recognition

GoogLeNet and Visual Geometry Group (VGG Net), having a very deep hierarchical layers, were announced and ranked first and second place in 2014 ILSVRC. In GoogLeNet, the complexity of operation have been reduced by introducing inception layers based on the concept of Network In Network (NIN).

Especially VGG Net investigate the effect of the convolutional network depth on its accuracy thorough increasing depth using an architecture with very small (3x3) convolution [2]. Despite these research, it is difficult to be implemented in hardware d into mobile applications, because of the size of the network. Table 1 shows the number of connections and parameters in various DCNNs. In case of VGG-16, it has the weight parameters of 138 million, which makes the required storage too large that all weights cannot be cached on chip. We try to condense the size of networks, so almost weights can be cached on chip memory instead of off-chip DRAM.

Table 1. Layer configuration, number of connections and parameters in various DCNN.

DCNN	AlexNet	VGG-16	VGG-19	GoogleNet
Parameters	61M	138M	144M	6.8M
Connections	837M	15.4G	19.6G	1.5G
CONV layers	5	13	16	2
FCN layers	3	3	3	1
Inception layers	-	-	-	9

2.1 Hierarchical Compression Scheme

Hierarchical compression is applied to VGG-16 and GoogLeNet in three compression steps hierarchically, where synapse pruning first condense the pre-trained synapse weight DCNN model and secondly synapse grouping is executed to the pruned network model and finally quantized the compress weight and layer outputs to fixed point models, satisfying the network's target accuracy below predefined error loss.

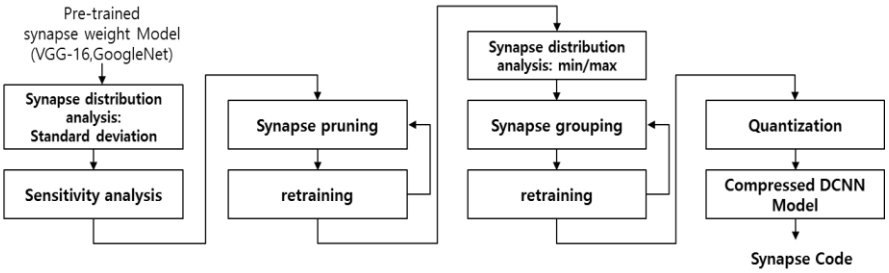

Fig. 1. Hierarchical compression procedure

Fig 1. shows the hierarchical compression procedure, where synapse pruning with retraining is followed by synapse grouping and quantization. After each compression, compressed network model, step by step, is used and applied to the next step compression schemes.

At first step, the iterative synapse pruning remove the connections with the weights within threshold of weights distribution in pre-trained synapse weight model. To determine the pruning threshold, the standard deviation (STD) of each layer is analyzed to get the layer's weight distribution and then pruning threshold is obtained by the layer's STD multiplied with a pruning ratio. Pruning ratio of layers can be are derived through iterative experiments by checking the loss of accuracy according to the gradual increase of applied threshold. After pruning, the pruned network is retrained to get or improve the target accuracy.

After pruning step, the synapse grouping replace pruned weights with similar values by a representative value so-called centroid.

We study a unique set of centroid for each layer considering its weight distribution. Similar to the first step, we determine the centroids through retraining minimizing the network loss of the accuracy.

At final step, we quantize the floating point compressed DCNN from previous steps, we analyze the dynamic range of the input/output activations and the parameters for a given layer as a similar method in [6].

After then, we determine the dynamic fixed point bit widths of the centroid sets of convolutional and fully connected (fc) layer and layer outputs.

Finally, the compressed DCNN model applied with hierarchical compression schemes is stored as synapse code, which will be used for future hardware implementation. Three compression schemes can be applied independently and respectively or in combination, in our own framework.

3 Experimental Results

In this section, we evaluate the compression results of the hierarchical compression algorithm.

3.1 Compression Results

We use the own framework based on caffe as compression tool, train and test tools. Table 2 and 3 show the compression results summary in VGG16. First, we observe that 91.60% of weight parameter are removed from baseline floating network, keeping about 0.6% accuracy degrade. Also we represent the centroid index bit with 4bit~8bits per layers. Second, floating point network are successfully compressed to 10-bit and 6-bit dynamic fixed point weights, and 8-bit layer output with top1-accuracy drop below 2.6%.

Table 2. Hierarchical compression results in VGG16. SP: synapse pruning, SG: synapse grouping, SQ: synapse quantization

Network	SP percent(%)	SG (Average bit)	SQ		
			CONV. weights	FC weights	layer outputs
VGG16	91.610 75.40	7.125	10bits	6bits	8bits

Table 3. Classification accuracy rate (%) for hierarchical compression.

Compression Method	Top1	Top5
Floating baseline Model(No compression)	68.34	88.45
SP (percent : 75.40%)	67.71	88.35
SP (percent : 75.40%) + SG	67.104	88.062
SP (percent : 75.40%) + SG + SQ	66.272	87.636
SP (percent : **91.61**%)	65.69	87.13
SP (percent : **91.61**%) + SG	65.056	86.814
SP (percent : **91.61**%) + SG + SQ	63.014	85.796

In GoogLeNet, the number of weight parameter is reduced by 67.13% compared to baseline floating point model and we trim the network with the combinations of 16-bit convolutional weights, 4-bit fc weights and 16-bit layer outputs.

Table 4. Hierarchical compression results in GoogLeNet.

Network	SP percent(%)	SG (Average bit)	SQ		
			CONV. weights	FC weights	layer outputs
GoogleNet	67.13	7.28125	16bit	4bit	16bit

From Table 4 and 5, we discover that with less than 5% drop in accuracy, compressed network achieves high compression rates (12 ~ 20×).

Table 5. Classification accuracy rate (%) for Hierarchical compression in GoogLeNet

Compression Method	Accuracy Top1	Accuracy Top5
Floating baseline Model(No compression)	68.93	89.14
SP (percent : 67.13%)	66.53	87.75
SP (percent : 67.13%) + SG	65.14	86.89
SP (percent : 67.13%) + SQ	62.49	84.95

3.2 Classification Test

With the compressed GoogLeNet, we test the object classification with real object. The experiments are carried out on a Jetson TX1 embedded board with Quad-core ARM® Cortex® -A57 MPCore and Microsoft LifeCam. Our simulator framework is used as the main software tool in the experiments. The compressed GoogleNet takes the video frame and successfully produce the probability of object and label among the trained classes.

Fig. 2. Classification Test for the compressed GoogleNet on embedded system

4 Conclusion

In this paper, we present hierarchical compression scheme to reduce the size of parameter in DCNN for large scale visual recognition. The results shows that the number of weights parameters with quantized bits can be reduced by maximum 91.6% compared to the original parameters. Our approach achieves good compression rates, with negligible drop in the classification accuracy.

5 Future Works

We will utilize the proposed method to generate the compressed DCNN in FPGA and ASIC implementation. Ultimately, we will implement the hardware of DCNN for general object classification in mobile devices. We expect to provide an effective and compact DCNN models by our proposed scheme in implement for mobile application.

Acknowledgments. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R7117-16-0166, Brain-Inspired Neuromorphic Perception and Learning Processor)

References

1. Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet classification with deep convolutional neural networks, NIPS, pp. 1106-1114, 2012
2. Simonyan, K., Zisserman, A.: VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION, CoRR (2015)
3. Szegedy, C.: Going deeper with convolutions, arXiv preprint arXiv:1409.4842, 2014.
4. Han, S., Pool, J., Tran, J., Dally, W.: Learning both Weights and Connections for Efficient Neural Network." In Advances in Neural Information Processing Systems, pp. 1135--1143. 2015.
5. Lin, D. D., Talathi, S. S., Annapureddy, V. S.: Fixed point quantization of deep convolutional networks, In ICML, 2016.
6. Gysel, P., Motamedi, M., Ghiasi, S.: Hardware oriented approximation of convolutional neural networks, arXiv:1604.03168, 2016.