Journal of Engineering and Applied Sciences 13 (Special Issue 11): 8670-8674, 2018
ISSN: 1816-949X

A Study on Channel Expansion Structure for Reducing Model Size and Speeding Up of Classifier Using Inverted Residual Block

[1]Seong-Kyun Han and [2]Soon-Chul Kwon
[1]Department of Electronic Engineering,
[2]Department of Smart Systems, Kwangwoon University, Seoul, Korea

Abstract: In this study, we propose a structure for model size reduction and speeding up of classifier using inverted residual block. Model size reduction for convolutional neural network computation in embedded systems is one of the main technique. To get a classifier structure that small and fast, we compare and analyze the experimental results of channel expansion parameter structure in inverted residual block proposed in MobileNetV2. Experiments were conducted on the Cifar-10 dataset for training and testing and compared with the method of MobileNetV2, 1.7% accuracy reduction, 60% model size reduction and 50% reduction in inference time were achieved.

Key words: Deep learning, object classification, inverted residual block, model size reduction, compared, computation

INTRODUCTION

Recently, image processing implemented by deep Neural Networks (DNNs) (LeCun *et al*., 1989; Krizhevsky, *et al*., 2012) has attracted attention. As a result, in the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), the object classifier using DNN outperformed the human's Top-5 accuracy of 94.9% (Krizhevsky *et al*., 2012) with a DNN's top-5 accuracy of 96.47% (He *et al*., 2016). However, high-accuracy models like (Krizhevsky *et al*., 2012; Szegedy *et al*., 2017) have many layers, so those models are required high computational power and memory because there are many model parameters and computation. Because of these reasons, it is difficult to implement DNN algorithms on a low-end embedded system with low computational power and memory.

To improve those problems, Xception (Chollet, 2017) and MobileNetV1 (Howard *et al*., 2017) proposed a depth-wise separable convolution that increases efficiency of convolution operation. And Sandler *et al*. (2018) proposed a linear bottleneck to reduce loss of the information in an activation space and inverted residual block to improve efficiency of the DNN. As a result, it became possible to classify objects into a smaller size model faster.

However, in an embedded system such as a smartphone, various programs are executed simultaneously. This results in less computational power and memory actually available. Therefore, in this study, we study how to perform object classification in less inference time with smaller model size. We implement a classifier using methods by Sandler *et al*. (2018), Chollet (2017) and Howard *et al*. (2017) with a basic feature extractor structure by Sandler *et al*. (2018). The channel expansion parameters in the inverted residual block proposed at Sandler *et al*. (2018) are adjusted to compare the influence of accuracy, inference time and model size and find out the most efficient channel expansion structure. After finding the most appropriate channel expansion structure, we will experiment that structure on Cifar-10, Cifar-100, Street View House Numbers (SVHN) and STL10 datasets.

MATERIALS AND METHODS

Depth-wise separable convolution: Standard convolution which is commonly used, computes the spatial direction and the channel direction at a time and outputs the result as a single value at a time. However, the depth-wise separable convolution initially introduced by Sifre and Mallat (2014) is operated by dividing the convolution operation into two orders, depth-wise for the spatial direction and point-wise for the channel.

Figure 1 and Table 1 compare the difference between the method and the computational cost according to the method by convolution operation when the number of input channel is c, size of input kernel is k×k and size of output vector is f×f×N. In the case of standard convolution, N k×k×c sized kernels are applied to input

Corresponding Author: Soon-Chul Kwon, Department of Smart Systems, Kwangwoon University, Seoul, Korea

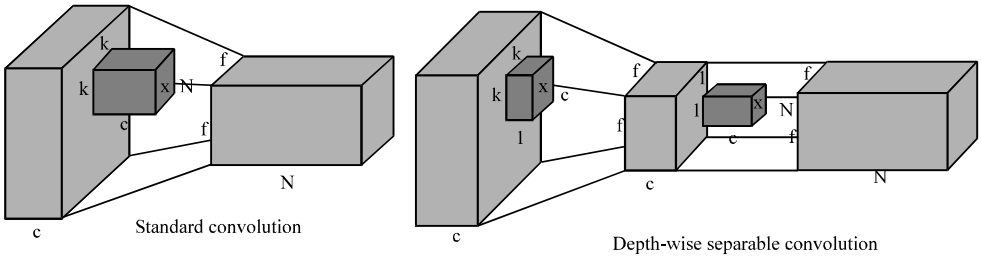

Fig. 1: Difference between two convolution methods

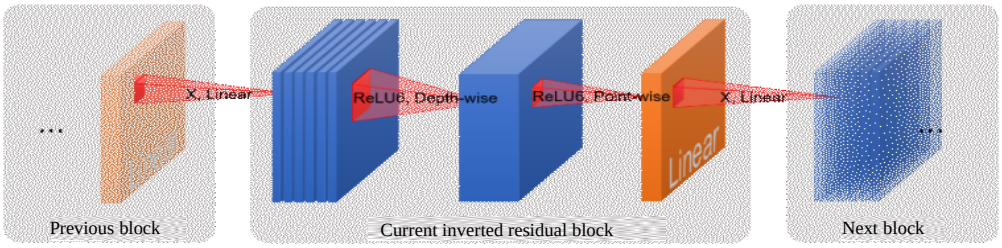

Fig. 2: Structure of linear bottleneck

Table 1: Comparison of computation cost by difference of two methods

Convolution method	Computational cost
Standard convolution	k×k×c×N×f×f
Depth-wise separable convolution	k×k×c×f×f+c×N×f×f

image and f×f×N sized vector is output by using weighted sum of spatial direction and channel direction. The computational cost of standard convolution is calculated as k×k×c×N×f×f.

However, in the case of the depth-wise separable convolution, the k×k×1 sized kernel is applied to the spatial direction without considering the channel size c of the input image, then f×f×c sized vector is medium output. The computation cost at this time is k×k×c×f×f. Next, N 1×1×c sized kernels are applied to channel direction of the medium output without considering the spatial direction. The size of the output of each convolution methods are same f×f×N sized vector but the computational cost is greatly reduced standard method's to k×k×c×f×f+c×N×f×f.

Considering the experimental result by Howard *et al.* (2017), the 3×3 depth-wise separable convolution reduces the computational cost by 8~9 times compared with the standard convolution.

Linear bottleneck: For every d-channel pixel in the deep convolution layer, the information for each layer is encoded in some manifold. The manifold existing in the high-dimensional activation space passes through the non-linear activation function like ReLU, resulting loss of information. The loss of information degrades the quality of the feature map obtained from the neural network. In this case, information loss can be prevented by embedding information existing in the high dimensional manifold of the activation space into a low dimensional subspace. Sandler *et al.* (2018) proposes a structure called linear bottleneck to prevent loss of information.

Figure 2 is an intuitive representation of a linear bottleneck structure. When this structure is used, it becomes possible to embed the information of the manifold existing in the high dimension of the activation space into the low dimensional manifold subspace. As a result, the information existing in the high dimension is simultaneously present in the low dimension, so that the loss of the high dimensional information by applying non-linear activation function can be prevented. In the other word, if the input manifold is embedded in the low dimensional subspace in the activation space, the transformation by the non-linear activation function such as ReLU acts as a good expressive function without losing information. Experimental results show that the structure with linear bottleneck improves accuracy in (Sandler *et al.*, 2018; Han *et al.*, 2017).

Inverted residual block: Residual block structure is proposed by He *et al.* (2016). Unlike a typical DNN structure, the residual block has few stacked convolutional layers together and the inputs and outputs are concatenated directly. As a result, it is possible to use

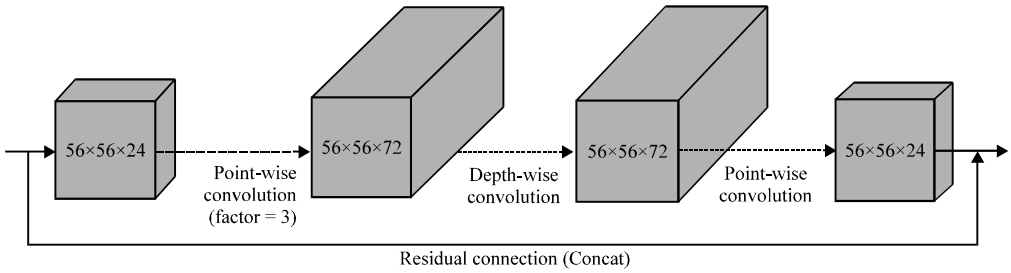

Fig. 3: One of the inverted residual block structure that proposed in this study

Table 2: The structure of proposed method

Input	Operator	t	C	n	s
$32^2 \times 3$	Conv2d	-	32	1	1
$112^2 \times 32$	Bottleneck	1	16	1	1
$112^2 \times 16$	Bottleneck	3	24	2	1
$56^2 \times 24$	Bottleneck	3	32	3	2
$28^2 \times 32$	Bottleneck	2	64	4	2
$14^2 \times 64$	Bottleneck	2	96	3	1
$14^2 \times 96$	Bottleneck	1	160	3	2
$7^2 \times 160$	Bottleneck	1	320	1	1
$7^2 \times 320$	Conv2d 4×1	-	1280	1	1
$7^2 \times 1280$	Avgpool 4×4	-	-	1	-
1×1×1280	Conv2d 1×1	-	k	-	-

the feature map that comes in as input of the residual block, so that, the feature map can be obtained more than the actual amount of calculation and parameter.

By He *et al.* (2016), the channel inside of residual block is reduced and increased. But, in the case of inverted residual block, proposed by Sandler *et al.* (2018), this method expands the channel inside of residual block opposed to He *et al.* (2016). Figure 3 shows an example of an inverted residual block. The channel expansion is multiplied by the channel expansion parameter and the point-wise convolution (Chollet, 2017) is used. The inverted residual block has the advantage that the memory efficiency is improved because the amount of information in feature map is larger than input/output of block.

Experiment

Implementation: In the experiment, the structure of the feature extractor proposed by Sandler *et al.* (2018) is modified and used. Table 2 shows the proposed structure, where t is the channel expansion parameter, c is the number of output channels, n is the number of repetitions of the block and s is the stride value. The experiment compares and analyses the experimental results according to the channel expansion parameters. For the experiment using the Cifar-10 dataset, select the model with the highest accuracy per unit model size and have additional validation using the Cifar-100, SVHN and STL10 dataset for that model.

Experimental environment: The experiment is conducted on Ubuntu 16.04LTS using Pytorch 0.4 interworking with CUDA. The Geforce GTX 1080 8GB Model is used for the CUDA. Training is conducted total of 3 steps which 1 step is 100 epochs with different learning rate. For each step, the learning rate is adjusted to 0.1, 0.01 and 0.001. the model with the highest top-1 accuracy is stored for each 1 step and the learning rate is adjusted in the next step. For each dataset, use 128 training batch size and 100 testing batch size but use 50 training and testing batch size only for STL10 dataset.

RESULTS AND DISCUSSION

Table 3 shows the experimental results according to changes in channel expansion structure when training/testing using a Cifar-10 dataset. In the Table 3, N is the number of repetitions of the corresponding block and the inner channel of the block is multiplied by the channel expansion parameter in the block of the corresponding size. Experimental results show that the structure of E is 60% smaller and 2 times faster than the method proposed by Sandler *et al.* (2018) with only 1.7% loss of accuracy.

Table 4 compares model size per unit inference time and model size per unit accuracy to compare inference time, accuracy and model size according to each channel expansion structure. As a result, the structure E showed the highest model accuracy per unit accuracy of 2.49.

Table 5 shows the experimental results of the structure E according to the changes of the dataset. Structure E shows an average of 1.7% reduction in accuracy over four datasets. However, inference times are two time faster and model sizes are 60% smaller than method by Sandler *et al.* (2018).

Table 3: Experimental results by changing channel expansion structure with Cifar-10 dataset

			Channel expansion parameter (t)					
Input size	N	[1]	A	B	C	D	E[ours]	F	G
1122×16	1	6	6	3	3	6	3	6	3
562×24	2	6	6	3	3	6	3	6	3
282×32	3	6	3	6	3	2	2	6	3
142×64	4	6	3	6	3	2	2	2	2
142×96	3	6	3	3	6	1	1	2	2
72×160	3	6	3	3	6	1	1	2	2
Inference time (msec)		26	21	18	17	19	13	22	14
Accuracy (%)		93.47	92.24	92.28	91.64	91.91	91.77	92.63	92.15
Model size (kB)		9179	5845	6727	8295	3804	3691	5162	4672

Table 4: Model size per unit inference time and model size per unit accuracy

Structure	[1]	A	B	C	D	E[ours]	F	G
Inference time (msec)	26	21	18	17	19	13	22	14
Accuracy (%)	93.47	92.24	92.28	91.64	91.91	91.77	92.63	92.15
Model size (kB)	9179	5845	6727	8295	3804	3691	5162	4672
Inference time/model size (μsec/B)	2.83	3.59	2.68	2.05	4.99	3.52	4.26	3
Accuracy/Model size (‰/B)	1.02	1.58	1.37	1.1	2.42	2.49	1.79	1.97

Table 5: Experimental results of structure changing dataset

Dataset	Structure	Inference time (msec)	Accuracy (%)	Model size (kB)	Inference time reduction (msec)	Accuracy loss (%)	Model size reduction (kB)
Cifar-10	[1]	26	93.47	9179	13	1.70	5488 (-59.79%)
	E [ours]	13	91.77	3691			
Cifar-100	[1]	25	72.32	9863	12	1.33	5714 (-57.93%)
	E [ours]	13	70.99	4149			
SVHN	[1]	26	96.27	9402	13	0.46	5714 (-60.77%)
	E [ours]	13	95.81	3688			
STL10	[1]	13	75.15	9402	6	3.60	5711 (-60.74%)
	E [ours]	7	71.55	3691			

CONCLUSION

In this study, we have experimented to observe the changes in the accuracy, inference time and model size according to the channel expansion structure in the inverted residual block by Sandler *et al.* (2018). Experimental results show that the size of the model linearly increases in proportion to the value of the channel expansion parameter and that the smaller the size of the model, the faster the inference time is.

We have shown that applying larger channel expansion parameter values for larger input sizes of the inverted residual block input can achieve larger model size reduction with as little accuracy reduction as possible. It was confirmed from the relationship between parameters and accuracy that sufficient features can be obtained when the channel expansion parameter is smaller than the value and has different structure from Sandler *et al.* (2018).

By increasing the channel expansion parameter of a block with a large feature map size and decreasing the parameter of a small block, it is effective in preserving accuracy, reducing the amount of computation and decreasing the memory use. For the model trained with Cifar-10 dataset, the structure proposed by us is 60% smaller and 50% faster than the model proposed at (Sandler *et al.*, 2018) with only 1.7% loss of accuracy. Therefore, it is considered that the model using the structure of E is more suitable for low-end embedded systems.

REFERENCES

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. Msc Thesis, The Computer Vision Foundation, New York, USA.

Han, D., J. Kim and J. Kim, 2017. Deep pyramidal residual networks. Proceedings of the 2017 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, IEEE, Honolulu, Hawaii, ISBN:978-1-5386-0457-1, pp: 6307-6315.

He, K., X. Zhang, S. Ren and J. Sun, 2016. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 26-July 1, 2016, IEEE, Las Vegas, Nevada, USA., ISBN:9781509014385, pp: 770-778.

Howard, A.G., M. Zhu, B. Chen, D. Kalenichenko and W. Wang *et al.*, 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. Comput. Vision Pattern Recognit., 2017: 1-9.

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2012. Imagenet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems, Leen, T.K., G.D. Thomas and T. Volker (Eds.). MIT Press, Cambridge, Massachusetts, USA., ISBN:0-262-12241-3, pp: 1097-1105.

LeCun, Y., B. Boser, J.S. Denker, D. Henderson and R.E. Howard *et al*., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1: 541-551.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov and L.C. Chen, 2018. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. J. Mobile Network Communication, Vol. 1801.

Sifre, L. and S. Mallat, 2014. Rigid-motion scattering for image classification. Ph.D Thesis, Ecole Polytechnique, Palaiseau, France.

Szegedy, C., S. Ioffe, V. Vanhoucke and A.A. Alemi, 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI 31th International Conference on Artificial Intelligence (AAAI-17), February 4-9, 2017, Google, Mountain View, California, USA., pp: 4278-4284.