Student ID : 20194293

Name : Go, Kyeong Ryeol

[AI 502] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

1. Paper Summary

In computer vision area, convolutional neural networks have made substantial advances. However, the networks are being more non-efficient in perspective of size and speed. Many real-world applications such as robotics, self-driving car and augmented reality, the task must be implemented in a timely manner. This computational restriction motivates many new efficient networks that can be generally categorized into either compressing pretrained networks or training small networks directly. This paper belongs to the second category and present a class of network architectures primarily focused on optimizing for latency. It is named as 'MobileNets' for mobile and embedded vision applications whose architectures are outlined in Table 1 in paper.

MobileNets are built primarily from the depth-wise separable convolutions (except for the first layer with full convolution) which factorize a standard convolution into two components; a depth-wise convolution and a pointwise convolution. The former one applies a filter to each input channel and latter one applies a 1 x 1 convolution to combine the outputs of the depth-wise convolution. (As a side remark, 94.86% of computation time is spent in the pointwise convolutions. In addition, 74.59% and 24.33% of parameters are assigned to the pointwise convolutions and fully connected layers, respectively.) In this way, the receptive field stays the same. Since the depth-wise convolution break the interaction between the number of output channels and the size of the kernel, computation and model size can be considerably reduced.

For analysis, let's assume that a standard convolutional layer takes a $\mathrm{D_f} \times D_f \times M$ feature map as an input and produces a $\mathrm{D}_g \times D_g \times N$ feature map as an output. Then, the kernel size would be $\mathrm{D_k} \times D_k \times M \times N$, which implies that the computation cost would be $\mathrm{D_k} \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f$. When this is replaced by the depth-wise separable convolutions, then the kernel size would be $\mathrm{D_k} \times D_k \times M$ in the depth-wise convolution which is N times smaller than usual. Moreover, in the pointwise convolution, the kernel size would be $1 \times 1 \times M \times \mathrm{N}$. Therefore, the total computation cost would be $\mathrm{D_k} \cdot D_k \cdot M \cdot D_f \cdot D_f + \mathrm{M} \cdot \mathrm{N} \cdot \mathrm{D_f} \cdot \mathrm{D_f}$ that is $\left(\frac{1}{\mathrm{N}} + \frac{1}{\mathrm{D_k^2}}\right)$ times smaller than the standard convolutional layer.

The author also introduces some extra hyperparameters, width multiplier(α) and resolution multiplier(ρ), in case of applications that require the model to be smaller and faster in the expense of a little bit of accuracy. The width multiplier, which is multiplied to input and output channels, thins a network uniformly at each layer and the resolution multiplier, which is multiplied to width and height of inputs, changes the resolution. Then, the resulting model would have $\frac{1}{\alpha^2}$ and about $\frac{1}{\rho^2}$ times smaller cost.

In the experiment, less regularization and data augmentation techniques are used and little or no weight decay was followed on the depth-wise filters since the small network is less vulnerable to the overfitting problem. It is shown that on ImageNet dataset, MobileNets saved tremendous multi-adds and parameters while the accuracy is decreased only 1% comparing to using full convolutions. Also, the thinner MobileNets with width/resolution multiplier performed better than the corresponding shallow model with similar computation and number of parameters, which justifies the use of multipliers. Varying the width/resolution multiplier, the trade-off between accuracy and multi-adds(and the number of parameters) was also observed and several applications supports the effectiveness of MobileNets.

2. Discussion

Here, I want to offer two discussion points. To begin with, is the residual learning compatible with the depth-wise separable convolutional layers? As ResNet paper states, residual learning boosts the learning, which may require less parameters to achieve the same accuracy. Next, what if Batch Normalization in MobileNets is replaced with Layer Normalization? Since depth-wise convolutions may miss the dependence among the channels, Layer Normalization(or Group Normalization) may be helpful.