MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Link

arxiv

Summary

- MobileNet model is based on depthwise separable convolution which is a form of factorized convolution. Standard convolution is split into two layers. First a depthwise convolution applies a single filter to each input channel. Then pointwise convolution applies 1x1 convolution to combine the outputs of the depthwise convolution. Standard convolution has a computational cost of $D_K \times D_K \times M \times N \times D_F \times D_F$ where D_K is the convolution kernel size, M is the number of input channels, N is the number of output channels, D_F is the spatial width and height. The depthwise convolution has computational cost of $D_K \times D_K \times M \times D_F \times D_F$ and the pointwise convolution has computational cost of $M \times N \times D_F \times D_F$. So the reduction in depthwise separable convolution is

  $$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

  Since N is number of output channels it's usually much lower than the second term when we use small value of kernel like3. So the saving in computation cost is almost $1/D_K^2$. For kernel size 3 we can thus expect to gain 8 to 9 times computational speed up.

  Standard convolution has memory requirement of $D_K \times D_K \times M \times N$. Depthwise separable convolution has memory requirement of $D_K \times D_K \times M + M \times N$. So the saving is memory is

  $$\frac{D_K \times D_K \times M + M \times N}{D_K \times D_K \times M \times N} = \frac{1}{N} + \frac{1}{D_K^2}$$

  i.e., same as the saving in computational cost.

- MobileNet spends 95% of its computation time in 1×1 convolution. Unlike convolution with larger kernels efficient implementation of 1×1 convolution does not require restructuring in memory which further improves the time and memory requirement.

- Contrary to large model MobileNet has less trouble with ovrefitting and thus require little or no regularization and data augmentation.

- MobileNet uses two hyperparameters to control model capacity

  1. Width multiplier(α) thins the network uniformly at each layer. For a given layer and width multiplier *alpha* the number of channels become αM and the number of output channels become αN. The computational cost of a depthwise separable convolution layer with width multiplier α is

     $$D_K \times D_K \times \alpha M \times D_F \times D_F + \alpha M \times \alpha N \times D_F \times D_F$$

     Width multiplier has the effect of reducing computational cost and number of parameters quadratically roughly by α^2.

  2. Resolution multiplier ρ is applied to input image and the internal representation of every layer is subsequently reduced by the same multiplier. This reduces the computational cost by ρ^2 but doesn't change number of parameters.

  Computational cost in the presence of both width multiplier and resolution multiplier is

  $$D_K \times D_K \times \alpha M \times \rho D_F \times \rho D_F + \alpha M \times \alpha N \times \rho D_F \times \rho D_F$$

- MobileNet is nearly as accurate as VGG16 while being 32 times smaller and 27 times less computationally expensive on ImageNet. Reduced version of MobileNet ($\alpha = 0.5$ and resolution 160×160 is 3% more accurate than AlexNet (absolute accuracy) while being 45 times smaller and having 9.5% less computation. It is also 3% better than SqueezeNet at about the same size and 22 times less computation.