MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Abstract

MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks.

1. Introduction

This paper describes an efficient network architecture and a set of two hyper-parameters in order to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications.

1. Prior Work

A different approach for obtaining small networks is shrinking, factorizing or compressing pretrained networks.

Another method for training small networks is distillation which uses a larger network to teach a smaller network.

1. MobileNet Architecture、
   1. Depthwise Separable Convolution

The MobileNet model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1X1 convolution called a pointwise convolution. For MobileNets the depthwise convolution applies

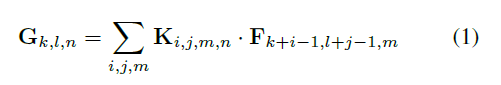
a single filter to each input channel. The pointwise convolution then applies a 1X1 convolution to combine the outputs the depthwise convolution. The depthwise separable convolution splits this

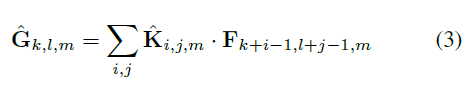
into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.

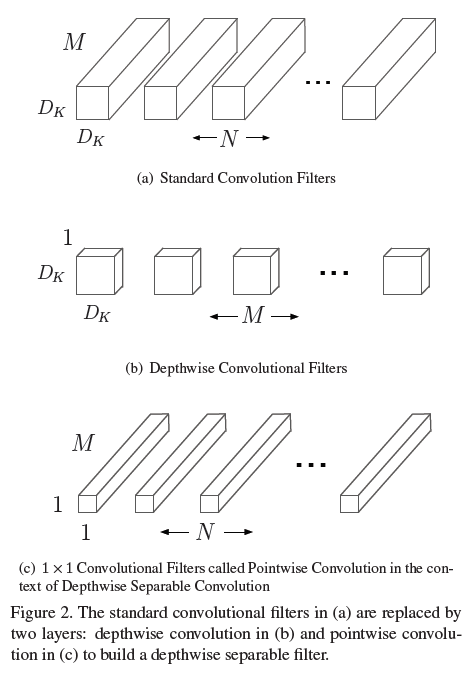
The filtering and combination steps can be split into two steps via the use of factorized convolutions called depthwise separable convolutions for substantial reduction in computational cost.

Depthwise separable convolution are made up of two layers: depthwise convolutions and pointwise convolutions. We use depthwise convolutions to apply a single filter per each input channel (input depth). Pointwise convolution, a simple 1X1 convolution, is then used to create a linear combination

of the output of the depthwise layer. MobileNets use both batchnorm and ReLU nonlinearities for both layers

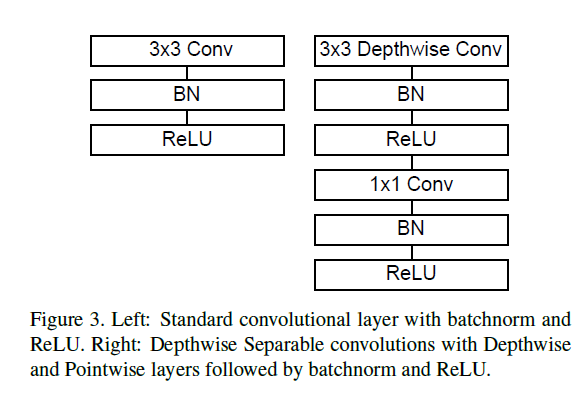






* 1. Network Structure and Training

Contrary to training large models we use less regularization and data augmentation techniques because small models have less trouble with overfitting.



* 1. Width Multiplier: Thinner Models

Although the base MobileNet architecture is already small and low latency, many times a specific use case or application may require the model to be smaller and faster. In order to construct these smaller and less computationally expensive models we introduce a very simple parameter alpha

called width multiplier.

Width multiplier can be applied to any model structure to define a new smaller model with a reasonable accuracy, latency and size trade off. It is used to define a new reduced structure that needs to be trained from scratch.



* 1. Resolution Multiplier: Reduced Representation

The second hyper-parameter to reduce the computational cost of a neural network is a resolution multiplier rou.



1. Experiments
   1. Model Choices
   2. Model Shrinking Hyperparameters
   3. Fine Grained Recognition
   4. Large Scale Geolocalizaton
   5. Face Attributes
   6. Object Detection
   7. Face Embeddings
2. Conclusion

We proposed a new model architecture called MobileNets based on depthwise separable convolutions.