MobileNets- Efficient Convolutional Neural Networks for Mobile Vision Applications

Mobile net architecture
1

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

1

paper, Factorized Networks[34] introduces a similar factorized convolution as well as the use of topological connections.

2

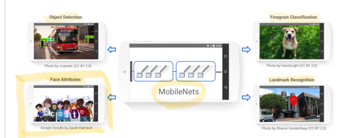


Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

Factorization
2

various factorizations have been proposed to speed up pretrained networks [14, 20].

Distillation
2

Another method for training small networks is distillation [9] which uses a larger network to teach a smaller network

Standard conv layer
2

The standard convolutional layer is parameterized by convolution kernel $\mathbf{K}$ of size $D_K \times D_K \times M \times N$ where D_K is the spatial dimension of the kernel assumed to be square and M is number of input channels and N is the number of output channels as defined previously.

The output feature map for standard convolution assuming stride one and padding is computed as:

$$\mathbf{G}_{k,l,n} = \sum_{i,j,m} \mathbf{K}_{i,j,m,n} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \quad (1)$$

Standard convolutions have the computational cost of:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2)$$

D_K : kernel size. M : input size

N : output size

Feature map $F \in \mathbb{R}^{D_F \times D_F \times N}$

Feature map $G \in \mathbb{R}^{D_G \times D_G \times M}$

2

3.1. 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 1× 1 convolution called a pointwise convolution.

3

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 1×1 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.

Depthwise conv
3

Depthwise convolution with one filter per input channel (input depth) can be written as:

$$\hat{\mathbf{G}}_{k,l,m} = \sum_{i,j} \hat{\mathbf{K}}_{i,j,m} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \quad (3)$$

Computational cost

$= D_K \cdot D_K \cdot M \cdot D_F \cdot D_F$.

3

Depthwise separable convolutions cost:

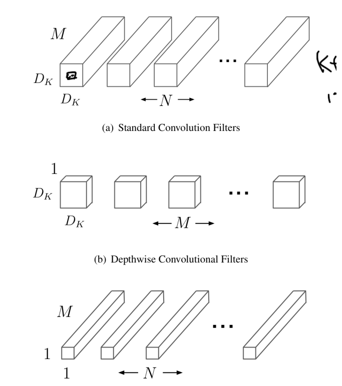
$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (5)$$

which is the sum of the depthwise and 1×1 pointwise convolutions.

By expressing convolution as a two step process of filtering and combining we get a reduction in computation of:

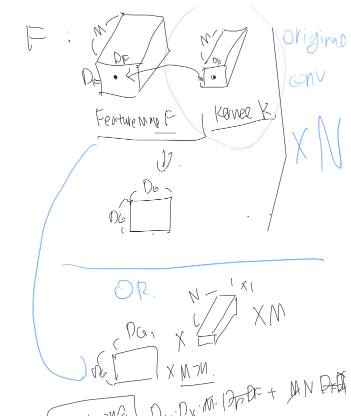
$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

3



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 2. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.



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

만큼 감소.

$\frac{N}{D_K^2} + M : N$

4

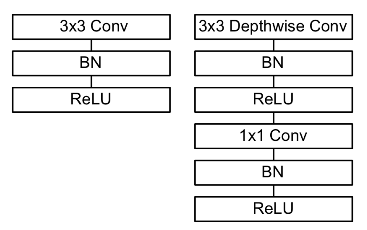


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

GEMM
4

unstructured sparse matrix operations are not typically faster than dense matrix operations until a very high level of sparsity. Our model structure puts nearly all of the computation into dense 1 × 1 convolutions. This can be implemented with highly optimized general matrix multiply (GEMM) functions

typically sparse matrix operation

dense operation → dense matrix.

1x1 conv.

4

1×1 convolutions do not require this reordering in memory and can be implemented directly with GEMM which is one of the most optimized numerical linear algebra algorithms. MobileNet spends 95% of it's computation time in 1×1 convolutions which also has 75% of the parameters as can be seen in Table 2. Nearly all of the additional parameters are in the fully connected layer.

Alpha
4

In order to construct these smaller and less computationally expensive models we introduce a very simple parameter acalled width multiplier. The role of the width multiplier α is to thin a network uniformly at each layer

$\alpha \in (0, 1]$ 보통 0.25, 0.5, 0.75

4

and width multiplier α, the number of input channels M becomes αM and the number of output channels N becomes αN.

The computational cost of a depthwise separable convolution with width multiplier α is:

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F \quad (6)$$

Resolution
5

ply this to the input image and the internal representation of every layer is subsequently reduced by the same multiplier. In practice we implicitly set ρ by setting the input resolution.

We can now express the computational cost for the core layers of our network as depthwise separable convolutions with width multiplier α and resolution multiplier ρ:

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F \quad (7)$$

Result of mobilenet
5

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Table 5. Narrow vs Shallow MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

7

4.7. Face Embeddings

The FaceNet model is a state of the art face recognition model [25]. It builds face embeddings based on the triplet loss. To build a mobile FaceNet model we use distillation to train by minimizing the squared differences of the output

8

Table 14. MobileNet Distilled from FaceNet

Model	1e-4 Accuracy	Million Mult-Adds	Million Parameters
FaceNet [25]	83%	1600	7.5
1.0 MobileNet-160	79.4%	286	4.9
1.0 MobileNet-128	78.3%	185	5.5
0.75 MobileNet-128	75.2%	166	3.4
0.75 MobileNet-128	72.5%	108	3.8

of FaceNet and MobileNet on the training data. Results for very small MobileNet models can be found in table 14.