# Depthwise seperable convolutions and their computational benefit over tranditional convolutional layers

Henk Jekel (5609593)

April 28, 2023

---

These notes aim to inform the reader about the computational benefits of depthwise seperable convolutional layers over tranditional convolutional layers.

## 1 Standard convolutional layers

To understand the benefit of depthwise separable convolutions (DSC) it is necessary to review standard convolutional layers and their computational cost. As represented in figure 1, a standard convolutional layer consists of 3 steps: convolution, batch normalization (BN) and an activation function.


**Figure 1: Block diagram with three blocks: CONV, BN, and ReLu6.**

In recent convolutional neural network (CNN) architectures, BN is performed at each layer in contrast to older CNN architectures where BN was performed only once at the input layer. To compute the computational cost of the first step, the convolution, it is assumed that the convolution transforms the input space $F$ into the output space $G$ according to figure 2.


**Figure 2: Block diagram to represent the space transformation of a convolution.**

Here, $F$ gets transformed into $G$ by the convolution. The convolution is a *same* convolution such that the width and height of the space remains constant. This is achieved with padding and a stride of 1. The convolution involves $N$ kernels with each kernel $k \in \mathbb{R}^{D_k \times D_k \times M}$. As there are $N$ kernels for one convolutional layer, the number of parameters of the convolutional layer $K \in \mathbb{R}^{D_k \times D_k \times M \times N}$. The computational cost $C$ of convolution in a layer is equal to the number of multiplications in the convolution. This is true because $C_{multiplication} \gg C_{addition}$. A same convolution has therefore a computational cost $C_{conv} = D_k \times D_k \times M \times N \times D_F \times D_F$, where $D_F$ represents the spatial dimension of the input and $D_k$ represents the spatial dimension of the kernel. Rewriting results in equation 1.

$$C_{conv} = D_k^2 \times M \times N \times D_F^2 \tag{1}$$

## 2 Depthwise seperable convolutional layers

The DSC consists of two parts, the depthwise convolution and the pointwisde convolution. The depthwise convolution takes the $M$ input channels and convolves them with $M$ kernels, where each kernel convolves over its own channel, $k_{depthwise} \in \mathbb{R}^{D_k \times D_k \times 1}$. The depthwise convolution only considers spatial information within each channel and does not involve any interaction between channels to combine depth information. Its purpose is to extract spatial features independently within each channel. As the depthwise convolution is a same convolution, the input and output space are the same. The pointwise convolution takes care of combining the information dispersed through the different channels as it involves $N$ kernels of dimension $k_{pointwise} \in \mathbb{R}^{1 \times 1 \times M}$. The name pointwise

refers to the single pixel representation in the spatial dimension. Sequentially applying the depthwise and pointwise convolution is displayed in figure 3.

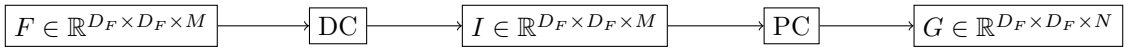


**Figure 3: Block diagram to represent the space transformation of DSC.**

Similar to computing the cost for a normal convolutional layer, the cost of the separable depthwise convolution is computed based on the number of multiplications performed. The number of multiplications for the depthwise convolution $C_{depthwise} = D_k \times D_k \times 1 \times D_f \times D_f \times M$. Rewrite results in equation 2

$$C_{depthwise} = D_k^2 \times D_f^2 \times M \tag{2}$$

The number of multiplications for the pointwise convolution is $C_{pointwise} = 1 \times 1 \times M \times D_f \times D_f \times N$. Rewriting result in equation 4.

$$C_{pointwise} = M \times D_f^2 \times N \tag{3}$$

Using equations 1 2 and 4, it was mathematically proved that the cost reduction from $C_{conv}$ to the combined cost of $C_{depthswise}$ and $C_{pointwise}$ is equal to $C_{reduction}$ as stated in equation **??**.

$$C_{reduction} = \frac{1}{N} + \frac{1}{D_k^2} \tag{4}$$

There is a small decrease in accuracy of mobilenets compared to network architectures with normal convolutions. The intuition behind this is that the DSC contains less parameters causing the potential model complexity to be lower. The authors of Mobilenet tried to reduce this effect by exploiting the seperable characteristic of DSC's. They implemented an additional ReLu function inbetween the depthwise convolution and the pointwise convolution. The resulting block is presented in figure 4. Applying the ReLu non-linearity function twice allows the trained network to be of higher degree complexity, compensating for the loss in complexity by a decrease in the number of model parameters.

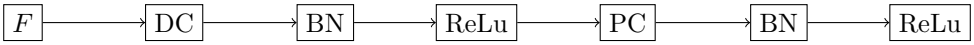


**Figure 4: Block diagram to represent the DSC block.**

DSC is a form of factorization. Factorization includes all methods that replaces standard convolutions with faster versions. Laurant Sivre came up with DSC in 2014. Later in 2017, workers at Google used DSC to introduce a new family of network architectures called MobileNets.

The above elaboration on DSC showed how factorizing the spatial dimension and the depth dimension resulted in a large reduction of computational cost. One could argue that it might be a good idea to factorize the spatial dimension too into a height and width dimension to further reduce computational cost. However, when comparing $C_{depthwise}$ and $C_{pointwise}$ the cost due the dephwise convolution is much lower than the cost due to the pointwise convolution in the earlier layers and the pointwise cost becomes almost negligible in the later layers of the network. This explains why the authors of mobilenet did not further factorized the spatial dimension. Because of the decreased amount of parameters in the DSC layers, the probability of overfitting during training decreases. It can therefore be seen as a type of regularization.