Convolutional Neural Networks at Constrained Time Cost

Kaiming He, Jian Sun
Microsoft Research

In industrial and commercial scenarios, engineers and developers are often faced with the requirement of constrained time budget. This paper investigates the accuracy of CNN architectures at constrained time cost during both training and testing stages. Our investigations involve the depth, width, filter sizes, and strides of the architectures. Because the time cost is constrained, the differences among the architectures must be exhibited as trade-offs between those factors. For example, if the depth is increased, the width and/or filter sizes need to be properly reduced. In the core of our designs is "layer replacement" - a few layers are replaced with some other layers that preserve time cost. Based on this strategy, we progressively modify a model and investigate the accuracy through a series of controlled experiments. This not only results in a more accurate model with the same time cost as a baseline model, but also facilitates the understandings of the impacts of different factors to the accuracy.

From the controlled experiments, we draw the following empirical observations about the depth. (1) The network depth is clearly of high priority for improving accuracy, *even if the width and/or filter sizes are reduced to compensate the time cost*. This is not a straightforward observation even though the benefits of depth have been recently demonstrated, because in previous comparisons the extra layers are added without trading off other factors, and thus increase the complexity. (2) While the depth is important, the accuracy gets stagnant or even degraded if the depth is overly increased. This is observed even if width and/filter sizes are not traded off (so the time cost increases with depth).

We obtain a model that achieves 11.8% top-5 error (10-view test) on ImageNet and only takes 3 to 4 days training on a single GPU. Our model is more accurate and also faster than several competitive models in recent papers. Our model has 40% less complexity than "*AlexNet*" and has 4.2% lower top-5 error.

Trade-offs between Depth and Filter Sizes

We first investigate the trade-offs between depth d and filter sizes s. We replace a larger filter with a cascade of smaller filters. We denote a layer configuration as: $n_{l-1} \cdot s_l^2 \cdot n_l$, which is also its theoretical complexity with n denoting the filter number and s denoting the filter size.

An example replacement can be written as the complexity involved:

$$\begin{aligned} & 256 \cdot 3^2 \cdot 256 \\ \Rightarrow \quad & 256 \cdot 2^2 \cdot 256 + 256 \cdot 2^2 \cdot 256. \end{aligned} \tag{1}$$

This replacement means that a 3×3 layer with 256 input/output channels is replaced by two 2×2 layers with 256 input/output channels.

Fig. 1 summarizes the relations among the models A, B, C, D, and E. Here deep models have smaller filter sizes. When the time complexity is roughly the same, the deeper networks with smaller filters show better results than the shallower networks with larger filters.

Trade-offs between Depth and Width

Next we investigate the trade-offs between depth d and width n. We increase depth while properly reducing the number of filters per layer, without changing the filter sizes. We replace the three 3×3 layers with six 3×3 layers. The complexity involved is:

$$\begin{aligned} & 128 \cdot 3^2 \cdot 256 + (256 \cdot 3^2 \cdot 256) \times 2 \\ = \quad & 128 \cdot 3^2 \cdot 160 + (160 \cdot 3^2 \cdot 160) \times 4 + 160 \cdot 3^2 \cdot 256. \end{aligned} \tag{2}$$

Here we fix the number of the input channels (128) of the first layer, and the number of output filters (256) of the last layer, so avoid impacting the previous/next stages. With this replacement, the width reduces from 256 to 160 (except the last layer).

Fig. 2 summarizes the relations among the models of A, F, G, H, and I. Here the deeper models are narrower. We find that increasing the depth leads to considerable gains, even the width needs to be properly reduced.

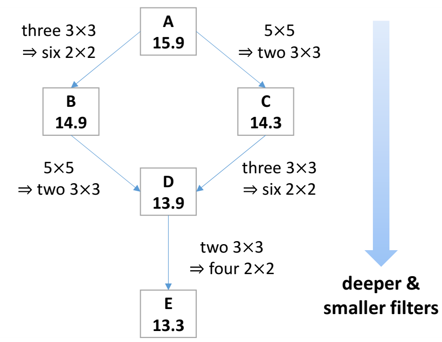

Figure 1: The relations of the models about **depth and filter sizes**.

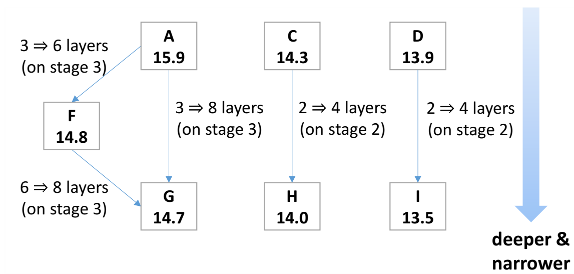

Figure 2: The relations of the models about **depth and width**.

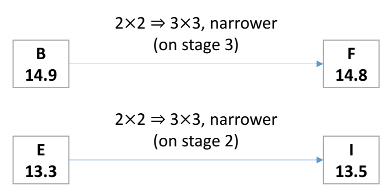

Figure 3: The relations of the models about **width and filter sizes**.

Trade-offs between Width and Filter Sizes

We can also fix the depth and investigate the trade-offs between width and filter sizes. The models B and F exhibit this trade-off on the last six convolutional layers:

$$\begin{aligned} & 128 \cdot 2^2 \cdot 256 + (256 \cdot 2^2 \cdot 256) \times 5 \\ \Rightarrow \quad & 128 \cdot 3^2 \cdot 160 + (160 \cdot 3^2 \cdot 160) \times 4 + 160 \cdot 3^2 \cdot 256. \end{aligned} \tag{3}$$

This means that the first five 2×2 layers with 256 filters are replaced with five 3×3 layers with 160 filters.

Fig. 3 shows the relations of these models. Unlike the depth that has a high priority, the width and filter sizes (3×3 or 2×2) do not show apparent priorities to each other.

More trade-offs involving depth, width, filter sizes, strides, and pooling are investigated in the main paper.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.