Efficient and Accurate Approximations of Nonlinear Convolutional Networks

Xiangyu Zhang[1], Jianhua Zou[1], Xiang Ming[1], Kaiming He[2], Jian Sun[2]
[1]Xi'an Jiaotong University. [2]Microsoft Research.

This paper addresses efficient test-time computation of deep convolutional neural networks (CNNs). Since the success of CNNs for large-scale image classification, the accuracy of the newly networks has been continuously improving. However, the computational cost of these networks (especially the more accurate but larger models) also increases significantly. The expensive test-time evaluation of the models can make them impractical in real-world systems. It is of practical importance to accelerate the test-time computation of CNNs.

There have been a few studies on approximating deep CNNs for accelerating test-time evaluation. A commonly used assumption is that the convolutional filters are approximately low-rank along certain dimensions. So the original filters can be approximately decomposed into a series of smaller filters, and the complexity is reduced. These methods have shown promising speedup ratios on a single [1] or a few layers [2] with some degradation of accuracy.

The algorithms and approximations in the previous work are developed for reconstructing linear filters and linear responses. However, the nonlinearity like the Rectified Linear Units (ReLU) is not involved in their optimization. Ignoring the nonlinearity will impact the quality of the approximated layers. Let us consider a case that the filters are approximated by reconstructing the linear responses. Because the ReLU will follow, the model accuracy is more sensitive to the reconstruction error of the positive responses than to that of the negative responses.

Moreover, it is a challenging task of accelerating the whole network (instead of just one or a very few layers). The errors will be accumulated if several layers are approximated, especially when the model is deep. Actually, in the recent work [1, 2] the approximations are applied on a single layer of large CNN models, such as those trained on ImageNet. It is insufficient for practical usage to speedup one or a few layers, especially for the deeper models which have been shown very accurate.

The main contributions of this paper are as follows:

(i) Nonlinear Approximation

In this paper, a method for accelerating *nonlinear* convolutional networks is proposed (see Fig. 1). It is based on minimizing the reconstruction error of *nonlinear* responses, subject to a low-rank constraint that can be used to reduce computation. We formulate the approximation as:

$$\min_{\mathbf{M},\mathbf{b}} \sum_i \|r(\mathbf{y}_i) - r(\mathbf{M}\mathbf{y}_i + \mathbf{b})\|_2^2, \quad s.t. \quad rank(\mathbf{M}) \le d'. \tag{1}$$

Here $r(\mathbf{y}_i) = r(\mathbf{W}\mathbf{x}_i)$ is the nonlinear response computed by the original filters, and $r(\mathbf{M}\mathbf{y}_i + \mathbf{b}) = r(\mathbf{M}\mathbf{W}\mathbf{x}_i + \mathbf{b})$ is the nonlinear response computed by the approximated filters, where $\mathbf{x}_i$ is the input of the layer. The above problem is challenging due to the nonlinearity and the low-rank constraint. To find a feasible solution, we relax it as:

$$\min_{\mathbf{M},\mathbf{b},\{\mathbf{z}_i\}} \sum_i \|r(\mathbf{y}_i) - r(\mathbf{z}_i)\|_2^2 + \lambda\|\mathbf{z}_i - (\mathbf{M}\mathbf{y}_i + \mathbf{b})\|_2^2 \quad s.t. \quad rank(\mathbf{M}) \le d'. \tag{2}$$

Here $\{\mathbf{z}_i\}$ is a set of auxiliary variables of the same size as $\{\mathbf{y}_i\}$. λ is a penalty parameter. If $\lambda \to \infty$, the solution to (2) will converge to the solution to (1). We adopt an alternating solver, fixing $\{\mathbf{z}_i\}$ and solving for M, $\mathbf{b}$ and vice versa.

(ii) Asymmetric Reconstruction for Multi-Layer

We further propose to minimize an asymmetric reconstruction error, which effectively reduces the accumulated error of multiple approximated layers:

$$\min_{\mathbf{M},\mathbf{b}} \sum_i \|r(\mathbf{W}\mathbf{x}_i) - r(\mathbf{M}\mathbf{W}\hat{\mathbf{x}}_i + \mathbf{b})\|_2^2, \quad s.t. \quad rank(\mathbf{M}) \le d'. \tag{3}$$

Here in the first term $\mathbf{x}_i$ is the non-approximate input, while in the second term $\hat{\mathbf{x}}_i$ is the approximate input due to the previous layer. We need not use $\hat{\mathbf{x}}_i$ in the first term, because $r(\mathbf{W}\mathbf{x}_i)$ is the real outcome of the original network and thus is more precise. On the other hand, we do not use $\mathbf{x}_i$ in the second term, because $r(\mathbf{M}\mathbf{W}\hat{\mathbf{x}}_i + \mathbf{b})$ is the actual operation of the approximated layer. This asymmetric version can reduce the accumulative errors when multiple layers are approximated. The optimization problem in (3) can be solved using the same algorithm as for (1).

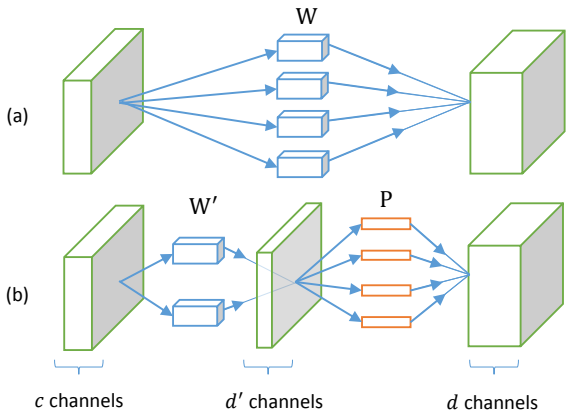

Figure 1: Illustration of the approximation. (a) An original layer with complexity $O(dk^2c)$. (b) An approximated layer with complexity reduced to $O(d'k^2c) + O(dd')$. Here k is the spatial size of filters

(iii) Rank Selection for Whole-Model Acceleration

In the above, the optimization is based on a target d' of each layer. d' is the only parameter that determines the complexity of an accelerated layer. But given a desired speedup ratio of the *whole model*, we need to determine the proper rank d' used for each layer. Our strategy is based on an empirical observation that the PCA energy is related to the classification accuracy after approximations. We assume that the whole-model classification accuracy is roughly related to the product of the PCA energy of all layers. Then we optimize d' for each layer to maximize the product of the PCA energy.

We evaluate our method on a 7-convolutional-layer model trained on ImageNet. We investigate the cases of accelerating each single layer and the whole model. Experiments show that our method is more accurate than the recent method of Jaderberg *et al.*'s [2] under the same speedup ratios. A *whole-model* speedup ratio of 4× is demonstrated, and its degradation is merely 0.9%. When our model is accelerated to have a comparably fast speed as the "AlexNet", our accuracy is 4.7% higher.

[1] Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014.

[2] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *BMVC*, 2014.

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