Very Deep Convolutional Networks for Large-Scale Image Recognition

Hongzhi Liu

August 17, 2018

Abstract

Convolutional networks have recently enjoyed a great success in large-scale image and video recognition which has become possible due to the large public image repositories and high-performance computing systems. Existing work attempts to exploit temporal information on box level, but such methods are not trained end-to-end. During preparation for URPC2018, I read a thesis written by Karen Simonyan and Andrew Zisserman who are from Visual Geometry Group of Department of Engineering Science in University of Oxford. This paper investigates the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. And their main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small convolution filters. They also show that the representations generalise well to other datasets where they achieve state-of-the-art results.

1. Overview of VGG Method

With ConvNets becoming more of a commodity in the computer vision field, a number of attempts have been made to improve the original architecture of Krizhevsky *et al.* [2] in a bid to achieve better accuracy. For instance, the best-performing submissions to the ILSVRC2013 [4, 3] utilised smaller receptive window size and smaller stride of the first convolutional layer. Another line of improvements dealt with training and testing the networks densely over the whole image and over multiple scales [3, 1].

In this paper, Karen Simonyan and Andrew Zisserman address another important aspect of ConvNet architecture design-its depth. To this end, they fix other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3×3) convolution filters in all layers. As a result, they come up with significantly more accurate ConvNet architectures that is denoted as "VGG" as shown in Fig. 1, which not only achieve the state-of-the-art accuracy on ILSVRC classification and localisation tasks, but are also applicable to other image recognition datasets, where they achieve excellent performance even when used as a part of a relatively simple pipelines. They have released the two best-performing models to facilitate further research.

2. ConvNet Configurations

In this paper, in order to measure the improvement brought by the increased ConvNet depth in a fair setting, all Simonyan and Zisserman's ConvNet layer configurations are designed using the same principles inspired by [2]. The team first describe a generic layout of their ConvNetconfigurations and then detail the specific configurations used in the evaluation. The design choices are then discussed and compared to the prior art.

2.1. Architecture

During training, the input to Zisserman's ConvNets is a fixed-size 224×224 RGB image. The only preprocessing they do is subtracting the mean RGB value, computed on the training set, from each pixel. The image is passed through a stack of convolutional (conv.) layers, where the team use filters with a very small receptive 3×3 field. In one of the configurations they also utilise 1×1 convolution filters, which can be seen as a linear transformation of the input channels. The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, *i.e.* the padding is 1 pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers. Max-pooling is performed over a 2×2 pixel window, with stride 2.

A stack of convolutional layers is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels. The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU) non-linearity. They note that none of their networks contain Local Response Normalisation (LRN) normalisa-

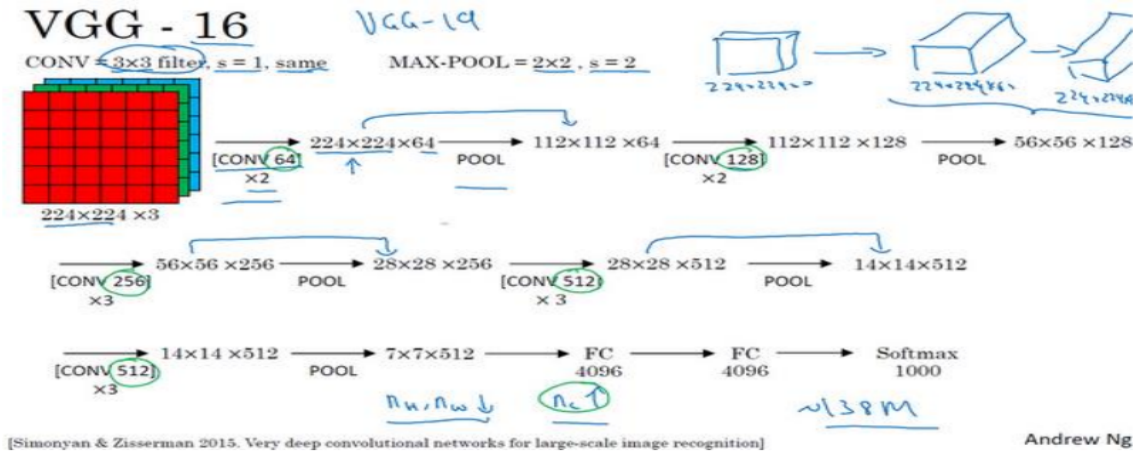

Figure 1. Architecture of VGG method which is displayed by Andrew Ng in deep learning course.

Table 1. **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

tion [2]: such normalisation does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time. Where applicable, the parameters for the LRN layer are those of [2].

2.2. Configurations

The ConvNet configurations, evaluated in this paper, are outlined in Tab. 1, one per column. In the following they refer to the nets by their names (AE). All configurations follow the generic design presented, and differ only in the depth: from 11 weight layers in the network A (8 conv. and 3 FC layers) to 19 weight layers in the network E (16 conv. and 3 FC layers). The width of conv. layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

2.3. Discussion

Simonyan and Zisserman's ConvNet configurations are quite different from the ones used in the top-performing entries of the ILSVRC-2012 [2] and ILSVRC-2013 competitions [4]. Rather than using relatively large receptive fields in the first conv. layers, they use very small 3×3 receptive fields throughout the whole net, which are convolved with the input at every pixel (with stride 1). It is easy to see that a stack of two 3×3 conv. layers has an effective receptive field of 5×5; three such layers have a 7×7 effective receptive field. First, they incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative. Second, they decrease the number of parameters: assuming that both the input and the output of a three-layer 3×3 convolution stack has C channels, the stack is parametrised by $3(3^2C^2) = 27C^2$ weights; at the same time, a single 7×7 conv. layer would require $7^2C^2 = 49C^2$ parameters, *i.e.* 81% more. This can be seen as imposing a regularisation on the 7×7 conv. filters, forcing them to have a decomposition through the 3×3 filters.

References

[1] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *Computer Science*, 2013. 1

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1

[4] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1, 3