Computer Vision and Image Understanding 000 (2017) 1–9

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Systematic evaluation of convolution neural network advances on the Imagenet

Dmytro Mishkin[a,*], Nikolay Sergievskiy[b], Jiri Matas[a]

[a] Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague. Karlovo namesti, 13. Prague 2, 12135, Czech Republic
[b] ELVEES NeoTek, Proyezd 4922, 4 build. 2, Zelenograd, Moscow, 124498, Russian Federation

ARTICLE INFO

Article history:
Received 7 June 2016
Revised 11 March 2017
Accepted 11 May 2017
Available online xxx

Keywords:
CNN
Benchmark
Non-linearity
Pooling
ImageNet

ABSTRACT

The paper systematically studies the impact of a range of recent advances in convolution neural network (CNN) architectures and learning methods on the object categorization (ILSVRC) problem. The evaluation tests the influence of the following choices of the architecture: non-linearity (ReLU, ELU, maxout, compatability with batch normalization), pooling variants (stochastic, max, average, mixed), network width, classifier design (convolutional, fully-connected, SPP), image pre-processing, and of learning parameters: learning rate, batch size, cleanliness of the data, etc.

The performance gains of the proposed modifications are first tested individually and then in combination. The sum of individual gains is greater than the observed improvement when all modifications are introduced, but the "deficit" is small suggesting independence of their benefits.

We show that the use of 128 × 128 pixel images is sufficient to make qualitative conclusions about optimal network structure that hold for the full size Caffe and VGG nets. The results are obtained an order of magnitude faster than with the standard 224 pixel images.

1. Introduction

Deep convolution networks have become the mainstream method for solving various computer vision tasks, such as image classification (Russakovsky et al., 2015), object detection (Everingham et al., 2010; Russakovsky et al., 2015), semantic segmentation (Dai et al., 2016), image retrieval (Tolias et al., 2016), tracking (Nam and Han, 2015), text detection (Jaderberg et al., 2014), stereo matching (Žbontar and LeCun, 2014), and many other.

Besides two classic works on training neural networks – (LeCun et al., 1998b) and Bengio (2012), which are still highly relevant, there is very little guidance or theory on the plethora of design choices and hyper-parameter settings of CNNs with the consequence that researchers proceed by trial-and-error experimentation and architecture copying, sticking to established net types. With good results in ImageNet competition, the AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2015) and GoogLeNet(Inception) (Szegedy et al., 2015) have become the de-facto standard.

Theory-grounded recommendations for the selection the number of neurons (Ithapu et al., 2017; Schmidhuber, 1997), network depth (Gao and Jojic, 2016), effective receptive field size (Luo et al., 2016), etc. have been published. The topic of local minima in deep network optimization is well covered by Choromanska et al. (2014) and by Soudry and Carmon (2016). However, the latest state of art results have been achieved by hand-crafted architectures (Zagoruyko and Komodakis, 2016) or by large-scale "trial-end-error" reinforcement learning search (Zoph and Le, 0000).

Improvements of many components of the CNN architecture like the non-linearity type, pooling, structure and learning have been recently proposed. First applied in the ILSVRC (Russakovsky et al., 2015) competition, they have been adopted in different research areas.

The contributions of the recent CNN improvements and their interaction have not been systematically evaluated. We survey the recent developments and perform a large scale experimental study that considers the choice of non-linearity, pooling, learning rate policy, classifier design, network width, batch normalization (Ioffe and Szegedy, 2015). We did not include ResNets (He et al., 2016a) – a recent development achieving excellent results – since they have been well covered in papers (He et al., 2016b; Larsson et al., 2016; Szegedy et al., 2016; Zagoruyko and Komodakis, 2016).

There are three main contributions of the paper. First, we survey and present baseline results for a wide variety of architectures and design choices both individually and in combination. Based

* Corresponding author.
E-mail address: ducha.aiki@gmail.com (D. Mishkin).

http://dx.doi.org/10.1016/j.cviu.2017.05.007
1077-3142/

Table 1
List of hyper-parameters tested.

Hyper-parameter	Variants
Non-linearity	linear, tanh, sigmoid, ReLU, VLReLU, RReLU, PReLU, ELU, maxout, APL, combination
Batch Normalization (BN)	before non-linearity. after non-linearity
BN + non-linearity	linear, tanh, sigmoid, ReLU, VLReLU, RReLU, PReLU, ELU, maxout
Pooling	max, average, stochastic, max+average, strided convolution
Pooling window size	3×3, 2×2, 3×3 with zero-padding
Learning rate decay policy	step, square, square root, linear
Colorspace & Pre-processing	RGB, HSV, YCrCb, grayscale, learned, CLAHE, histogram equalized
Classifier design	pooling-FC-FC-clf, SPP-FC-FC-clf, pooling-conv-conv-clf-avepool, pooling-conv-conv-avepool-clf
Network width	$1/4, 1/2\sqrt{2}, 1/2, 1/\sqrt{2}, 1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}$
Input image size	64, 96, 128, 180, 224
Dataset size	200k, 400k, 600k, 800k, 1200k(full)
Batch size	1, 32, 64, 128, 256, 512, 1024
Percentage of noisy data	0, 5%, 10%, 15%, 32%
Using bias	yes/no

on a large-scale evaluation, we provide novel recommendations and insights into deep convolutional network structure. Second, we show that for popular architectures – AlexNet, GoogLeNet, VGGNet – the recommendations based on results obtained on small images hold for common image size 224×224 or even 300×300 pixels which allows very fast testing. Last, but not least, the benchmark is fully reproducible and all scripts and data are available online.[1]

The paper is structured as follows. In Section 2.1, we explain and validate experiment design. In Section 3, the influence of a range of hyper-parameters is evaluated in isolation. The related literature is review the corresponding in experiment sections. Section 4 is devoted to the combination of best hyper-parameter setting and to "squeezing-the-last-percentage-points" for a given architecture recommendation. The paper is concluded in Section 5.

2. Evaluation

Standard CaffeNet parameters and architecture are shown in Table 2. The full list of tested attributes is given in Table 1.

2.1. Evaluation framework

All tested networks were trained on the 1000 object category classification problem on the ImageNet dataset (Russakovsky et al., 2015). The set consists of a 1.2M image training set, a 50k image validation set and a 100k image test set. The test set is not used in the experiments. The commonly used pre-processing includes image rescaling to $256 \times N$, where $N \geq 256$, and then cropping a random 224×224 square (Howard, 2013; Krizhevsky et al., 2012). The setup achieves good results in classification, but training a network of this size takes several days even on modern GPUs. We thus propose to limit the image size to $144 \times N$ where $N \geq 128$ (denoted as ImageNet-128px). For example, the CaffeNet (Jia et al., 2014) is trained within 24 h using NVIDIA GTX980 on ImageNet-128px.

2.1.1. Architectures

The input size reduction is validated by training CaffeNet, GoogLeNet and VGGNet on both the reduced and standard image sizes. The results are shown in Fig. 1. The reduction of the input image size leads to a consistent drop of around 6% in top-1 accuracy for all three popular architectures and does not change their relative order (VGGNet > GoogLeNet > CaffeNet) or accuracy difference.

[1] https://www.github.com/ducha-aiki/caffenet-benchmark.

Table 2
The basic CaffeNet architecture used in most experiments. Pad 1 – zero-padding on the image boundary with1 pixel. Group 2 convolution – filters are split into 2 separate groups. The architecture is denoted in "shorthand" as 96C11/4 → MP3/2 → 192G2C5/2 → MP3/2 → 384G2C3 → 384C3 → 256G2C3 → MP3/2 → 2048C3 → 2048C1 → 1000C1.

input	image 128×128 px, random crop from $144 \times N$, random mirror
pre-process	out = 0.04 (BGR - (104; 117; 124))
conv1	conv $11 \times 11 \times 96$, stride 4
	ReLU
pool1	max pool 3×3, stride 2
conv2	conv $5 \times 5 \times 192$, stride 2, pad 1, group 2
	ReLU
pool2	max pool 3×3, stride 2
conv3	conv $3 \times 3 \times 384$, pad 1
	ReLU
conv4	conv $3 \times 3 \times 384$, pad 1, group 2
	ReLU
conv5	conv $3 \times 3 \times 256$, pad 1, group 2
	ReLU
pool5	max pool 3×3, stride 2
fc6	fully-connected 4096
	ReLU
drop6	dropout ratio 0.5
fc7	fully-connected 4096
	ReLU
drop7	dropout ratio 0.5
fc8-clf	softmax-1000

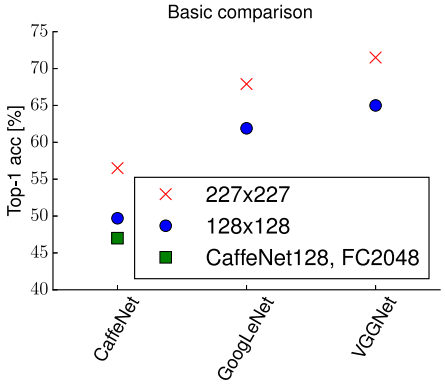

Fig. 1. Impact of image and network size on top-1 accuracy.

In order to decrease the probability of overfitting and to make experiments less demanding in memory, another change of CaffeNet is made. A number of filters in fully-connected layers 6 and 7 were reduced by a factor of two, from 4096 to 2048. The results validating the resolution reduction are presented in Fig. 1.

The parameters and architecture of the standard CaffeNet are shown in Table 2. For experiments we used CaffeNet with $2 \times$ thinner fully-connected layers, named as CaffeNet128-FC2048. The architecture can be denoted as 96C11/4 → MP3/2 → 192G2C5/2 → MP3/2 → 384G2C3 → 384C3 → 256G2C3 → MP3/2 → 2048C3 → 2048C1 → 1000C1. Here we used fully-convolutional notation for fully-connected layers, which are equivalent when image input size is fixed to 128×128 px. The default activation function is ReLU and it is put after every convolution layer, except the last 1000-way softmax classifier.

2.1.2. Learning

SGD with momentum 0.9 is used for learning, the initial learning rate is set to 0.01, decreased by a factor of ten after every 100k iterations until learning stops after 320k iterations. The L2 weight decay for convolutional weights is set to 0.0005 and it is not applied to bias. The dropout (Srivastava et al., 2014) with probability 0.5 is used before the two last layers. All the networks were initial-

Please cite this article as: D. Mishkin et al., Systematic evaluation of convolution neural network advances on the Imagenet, Computer Vision and Image Understanding (2017), http://dx.doi.org/10.1016/j.cviu.2017.05.007