Deep Residual Networks pretrained models (ILSVRC 2015 winners)

时间 2016-02-04 21:38:57 Github (/sites/fyAnyi)

原文 https://github.com/KaimingHe/deep-residual-networks (http://www.tuicool.com/articles/hit/buay6b)

主题 R语言 (/topics/11080057)

Deep Residual Networks

By Kaiming He (http://research.microsoft.com/en-us/um/people/kahe/) , Xiangyu Zhang (https://scholar.google.com/citations?user=yuB-cfoAAAAJ&hl=en) , Shaoqing Ren (http://home.ustc.edu.cn/%7Esqren/) , Jian Sun (http://research.microsoft.com/en-us/people/jiansun/) .

Microsoft Research Asia (MSRA).

Table of Contents

Introduction

This repository contains the original models (ResNet-50, ResNet-101, and ResNet-152) described in the paper "Deep Residual Learning for Image Recognition" (http://arxiv.org/abs/1512.03385 (http://arxiv.org/abs/1512.03385)). These models are those used in ILSVRC (http://image-net.org/challenges/LSVRC/2015/) and COCO (http://mscoco.org/dataset/#detections-challenge2015) 2015 competitions, which won the 1st places in: ImageNet classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

Note- Check re-implementations with **training code** and models from Facebook AI Research (FAIR)! -- blog (http://torch.ch/blog/2016/02/04/resnets.html) , code (https://github.com/facebook/fb.resnet.torch)

If you use these models in your research, please cite:

```
@article{He2015,
    author = {Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun},
    title = {Deep Residual Learning for Image Recognition},
    journal = {arXiv preprint arXiv:1512.03385},
    year = {2015}
}
```

Disclaimer and known issues

1. These models are converted from our own implementation to a recent version of Caffe (2016/2/3, b590f1d). The numerical results using this code are as in the tables below.
2. These models are for the usage of testing or fine-tuning.
3. These models were **not** trained using this version of Caffe.
4. If you want to train these models using this version of Caffe without modifications, please notice that:
   - GPU memory might be insufficient for extremely deep models.
   - Changes of mini-batch size should impact accuracy (we use a mini-batch of 256 images on 8 GPUs, that is, 32 images per GPU).
   - Implementation of data augmentation might be different (see our paper about the data augmentation we used).
   - We randomly shuffle data at the beginning of every epoch.
   - There might be some other untested issues.
5. In our BN layers, the provided mean and variance are strictly computed using average (**not** moving average) on a sufficiently large training batch after the training procedure. The numerical results are very stable (variation of val error < 0.1%). Using moving average might lead to different results.
6. In the BN paper, the BN layer learns gamma/beta. To implement BN in this version of Caffe, we use its provided "batch_norm_layer" (which has no gamma/beta learned) followed by "scale_layer" (which learns gamma/beta).
7. We use Caffe's implementation of SGD with momentum: v := momentum*v + lr*g. If you want to port these models to other libraries (e.g., Torch, CNTK), please pay careful attention to the possibly different implementation of SGD with momentum : v := momentum*v + (1-momentum)*lr*g, which changes the effective learning rates.

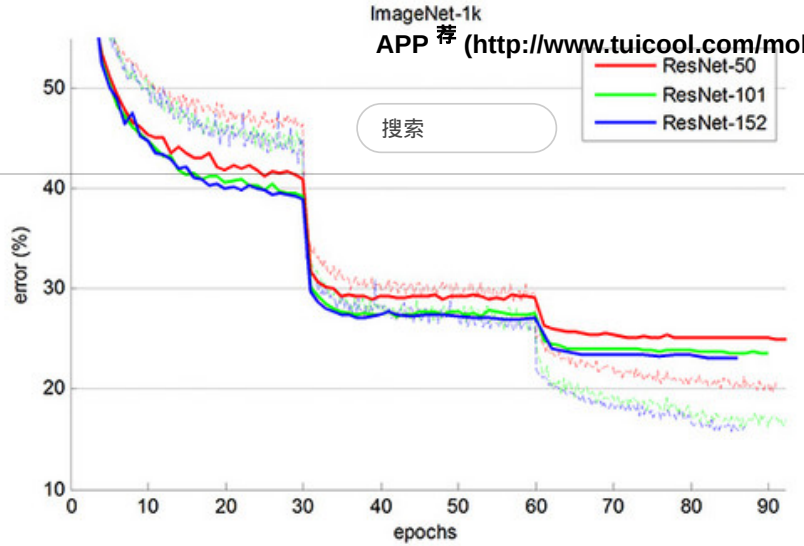
Models

1. Visualizations of network structures:
   - ResNet-50 (http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006)
   - ResNet-101 (http://ethereon.github.io/netscope/#/gist/b21e2aae116dc1ac7b50)
   - ResNet-152 (http://ethereon.github.io/netscope/#/gist/d38f3e6091952b45198b)
2. Model files:
   - MSR download: link (http://research.microsoft.com/en-us/um/people/kahe/resnet/models.zip)

- OneDrive download: link (https://onedrive.live.com/?authkey=%21AAFW2-FVoxeVRck&id=4006CBB8476FF777%211178&cid=4006CBB8476FF777)

Results

1. Curves on ImageNet (solid lines: 1-crop val error; dashed lines: training error):



2. 1-crop validation error on ImageNet (center 224x224 crop from resized image with shorter side=256):

model	top-1	top-5
VGG-16 (http://www.vlfeat.org/matconvnet/pretrained/)	28.5% (http://www.vlfeat.org/matconvnet/pretrained/)	9.9% (http://www.vlfeat.org/matconvnet/pretrained/)
ResNet-50	24.7%	7.8%
ResNet-101	23.6%	7.1%
ResNet-152	23.0%	6.7%

3. 10-crop validation error on ImageNet (averaging softmax scores of 10 224x224 crops from resized image with shorter side=256), the same as those in the paper:

model	top-1	top-5
ResNet-50	22.9%	6.7%

model	top-1	top-5
ResNet-101	21.8%	6.1%
ResNet-152	21.4%	5.7%

Third-party re-implementations

Deep residual networks are very easy to implement and train. We recommend to see also the following third-party re-implementations and extensions:

1. By Facebook AI Research (FAIR), with training code in Torch and pre-trained ResNet-18/34/50/101 models for ImageNet : blog (http://torch.ch/blog/2016/02/04/resnets.html) , code (https://github.com/facebook/fb.resnet.torch)
2. Torch, CIFAR-10, with ResNet-20 to ResNet-110, training code, and curves:code
3. Lasagne, CIFAR-10, with ResNet-32 and ResNet-56 and training code:code
4. Neon, CIFAR-10, with pre-trained ResNet-32 to ResNet-110 models, training code, and curves:code
5. Torch, MNIST, 100 layers: blog (https://deepmlblog.wordpress.com/2016/01/05/residual-networks-in-torch-mnist/) , code (https://github.com/arunpatala/residual.mnist)
6. A winning entry in Kaggle's right whale recognition challenge: blog (http://blog.kaggle.com/2016/02/04/noaa-right-whale-recognition-winners-interview-2nd-place-felix-lau/) , code (https://github.com/felixlaumon/kaggle-right-whale)
7. Neon, Place2 (mini), 40 layers: blog (http://www.nervanasys.com/using-neon-for-scene-recognition-mini-places2/) , code (https://github.com/hunterlang/mpmz/)

相关推刊

- 《匿名收藏》 P1531

我来评几句

请输入评论内容...

登录后评论

已发表评论数(0)