SUMMARY

Residual Network

Introduction

Deep networks are hard to train because of the notorious vanishing gradient problem — as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly causing a degradation problem The problem is not due to overfitting as both training and test increases simultaneously with increasing depth.Increasing network depth does not work by simply stacking layers together.

The core idea of ResNet is introducing a so-called “identity shortcut connection” that skips one or more layers.The authors of paper argue that stacking layers shouldn’t degrade the network performance, because we could simply stack identity mappings upon the current network, and the resulting architecture would perform the same. This indicates that the deeper model should not produce a training error higher than its shallower counterparts. They hypothesize that letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlying mapping. And the residual block explicitly allows it to do precisely that.

Formally, denoting the desired underlying mapping as H(x), we let the stacked nonlinear layers fit another mapping of $F(x) := H(x)−x$. The original mapping is recast into $F(x)+x$ Authors hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers

Identity Mapping by Shortcuts

A typical building block in paper looks like - **$y = F(x, \{Wi\}) + x$ - (1)** ,where x and y are the input and output vectors of the layers considered. The function F(x, {Wi}) represents the residual mapping to be learned. The shortcut connections in Eqn.(1) introduce neither extra parameters nor computation complexity. This is not only attractive in practice but also important in comparing plain and residual networks .The dimensions of x and F must be equal in Eqn.(1)

If this is not the case (e.g., when changing the input/output channels), we can perform a linear projection Ws by the shortcut connections to match the dimensions   **$y = F(x, \{Wi\}) + Wsx$.**

Network Architectures

Plain Network - Baselines are mainly inspired by the philosophy of VGG nets. The convolutional layers .

mostly have 3×3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer.Perform downsampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax

Residual Network - Based on the above plain network, insert shortcut connections which turn the network into its counterpart residual version. The identity shortcuts can be directly used when the input and output are of the same dimensions . When the dimensions increase,consider two options: (A) The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no extra parameter; (B) The projection shortcut in Eqn.(2) is used to match dimensions (done by 1×1 convolutions). For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

Experiments

1 ImageNet Classification - Firstly evaluating over 18 and 34 layer deep network, deeper 34-layer plain net has higher validation error than the shallower 18-layer plain net. Now evaluating over 18 and 34 layer deep Resnet,– the 34-layer ResNet is better than the 18-layer ResNet (by 2.8%). More importantly, the 34-layer ResNet exhibits considerably lower training error and is generalizable to the validation data. This indicates that the degradation problem is well addressed

Deeper Bottleneck Architectures - Because of concerns on the training time,the building block is modified as a bottleneck design. For each residual function F, we use a stack of 3 layers instead of 2 (Fig. 5). The three layers are 1×1, 3×3, and 1×1 convolutions, where the 1×1 layers are responsible for reducing and then increasing (restoring) dimensions, leaving the 3×3 layer a bottleneck with smaller input/output dimensions.The parameter-free identity shortcuts are particularly important for the bottleneck architectures.

2 Cifar10 - Evaluating 20, 32, 44, and 56-layer plain networks.The deep plain nets suffer from increased depth, and exhibit higher training error when going deeper. This phenomenon is similar to that on ImageNet and on MNIST suggesting that such an optimization difficulty is a fundamental problem. ResNets manage to overcome the optimization difficulty and demonstrate accuracy gains when the depth

There are still problems on such aggressively deep models. The testing result of this 1202-layer network is worse than that of the 110-layer network, although both have similar training errors, reason being the overfitting.

3 Object Detection on PASCAL and MS COCO - Resnet has good generalization performance on other recognition tasks adopting Faster R-CNN as the detection method. On the COCO dataset we obtain a 6.0%

increase in COCO’s standard metric, which is a 28% relative improvement. This gain is solely due to the learned representations.