Deep Residual Learning for Image Recognition

Nama Kelompok :

1. Ari Cahya Saputra (1103190093)
2. Alvin Anandra Brilliyandy (1103190111)

Kelas : TK-42-PIL

INTRODUCTION

- Deep convolutional neural networks, have led to a series of breakthroughs for image classification. Deep networks naturally integrate low/mid/highlevel features and classifiers in an end-to-end multilayer fashion, and the “levels” of features can be enrichedby the number of stacked layers (depth). The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution by construction to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart.

RELATED WORK

- Residual Representations : In image recognition, VLAD [18] is a representation that encodes by the residual vectors with respect to a dictionary, and Fisher Vector [30] can be formulated as a probabilistic version [18] of VLAD. Both of them are powerful shallow representations for image retrieval and classification [4, 48]. For vector quantization, encoding residual vectors [17] is shown to be more effective than encoding original vectors.
- Shortcut Connections An early practice of training multi-layer perceptrons (MLPs) is to add a linear layer connected from the network input to the output [34, 49]. In [44, 24], a few intermediate layers are directly connected to auxiliary classifiers for addressing vanishing/exploding gradients. The papers of [39, 38, 31, 47] propose methods for centering layer responses, gradients, and propagated errors, implemented by shortcut connections. In [44], an “inception” layer is composed of a shortcut branch and a few deeper branches.

DEEP RESIDUAL LEARNING

- 3.1. Residual Learning The degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers. With the residual learning reformulation, if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers toward zero to approach identity mappings.
- 3.2. Identity Mapping by Shortcuts We adopt residual learning to every few stacked layers. Formally, in this paper we consider a building block defined as:
  - y = F(x, {Wi}) + x.
- Here 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 function F(x, {Wi}) can represent multiple convolutional layers. The element wise addition is performed on two feature map

DEEP RESIDUAL LEARNING

- 3.3. Network Architectures
  - Plain Network : 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. We 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. The total number of weighted layers is 34
  - Residual Network : Based on the above plain network, we 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
- 3.4. Implementation We use SGD with a mini-batch size of 256. The learning rate starts from 0.1 and is divided by 10 when the error plateaus, and the models are trained for up to 60 × 104 iterations. We use a weight decay of 0.0001 and a momentum of 0.9.

DEEP RESIDUAL LEARNING

- 4.1. ImageNet Classification
  - Plain Networks. To reveal the reasons, we compare their training/validation errors during the training procedure. We have observed the degradation problem – the 34-layer plain net has higher training error throughout the whole training procedure, even though the solution space of the 18-layer plain network is a subspace of that of the 34-layer one.
  - Residual Networks. Next we evaluate 18-layer and 34- layer residual nets (ResNets). The baseline architectures are the same as the above plain nets, expect that a shortcut connection is added to each pair of 3×3 filters. the situation is reversed with residual learning – the 34-layer ResNet is better than the 18-layer ResNet (by 2.8%).
  - Identity vs. Projection Shortcuts.

DEEP RESIDUAL LEARNING

- Deeper Bottleneck Architectures. We modify the building block as a bottleneck design4 . For each residual function F, we use a stack of 3 layers instead of 2. 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.
- 50-layer ResNet We replace each 2-layer block in the 34-layer net with this 3-layer bottleneck block, resulting in a 50- layer ResNet .
- 101-layer and 152-layer ResNets We construct 101- layer and 152-layer ResNets by using more 3-layer blocks. The 50/101/152-layer ResNets are more accurate than the 34-layer ones by considerable margins Comparisons with State-of-the-art Methods. Our baseline 34-layer ResNets have achieved very competitive accuracy. Our 152-layer ResNet has a single-model top-5 validation error of 4.49%.

CIFAR-10 AND ANALYSIS

- Analysis of Layer Responses : For ResNets, this analysis reveals the response strength of the residual functions. ResNets have generally smaller responses than their plain counterparts. We also notice that the deeper ResNet has smaller magnitudes of responses, as evidenced by the comparisons among ResNet-20, 56, and 110. When there are more layers, an individual layer of ResNets tends to modify the signal less.
- Exploring Over 1000 layers : We explore an aggressively deep model of over 1000 layers. But there are still open problems on such aggressively deep models. The testing result of this 1202-layer network is worse than that of our 110-layer network, although both have similar training error.

OBJECT DETECTION PASCAL AND MS COCO

- We adopt Faster R-CNN as the detection method. Here we are interested in the improvements of replacing VGG-16 with ResNet-101. Most remarkably, on the challenging COCO dataset we obtain a 6.0% increase in COCO's standard metric, which is a 28% relative improvement.

Thank You !!!