Deep residual learning for image recognition

Abstract

본 논문은 이전의 연구에서 다룬 네트워크보다 훨씬 더 깊어진 네트워크의 용이한 학습을 위해 *residual learning framework*를 제안한다. 이 논문에서 말하는 *residual function*이란 layer의 입력을 참고하는 형태이다. 이러한 *Residual network*는 optimize가 쉽고, 깊은 네트워크에서도 정확도를 얻을 수 있다. VGG보다 8배 깊어진 152 layer를 사용한 residual network를 ImageNet으로 평가하였다. VGG보다 좋은 성능을 내고 복잡도는 감소하였고. 3.57%의 error를 달성해서 ILSVRC 2015에서 1등을 차지했다.

1.Introduction

Network들의 깊이가 깊어질수록 좋은 성능을 보여줬다. 이에 따라 'model의 layer를 단순히 쌓으며 depth을 늘리는 것이 학습에 유리한가?' 라는 궁금증이 생기게 되었다. 하지만 Deep network의 깊이가 어느정도 깊어지면, 수렴을 방해하는 vanishing/exploding gradient문제가 있다는 것을 알 수 있다. 이는 normalized initialization*(->weight initialization)*과 intermediate normalization*(->batch noramalization)*을 통해 해결되었다. 그러나 또다른 문제가 발생한다.

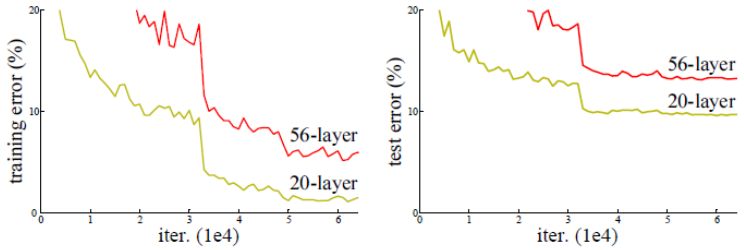

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Figure 1에서 볼 수 있듯, layer가 깊은 network의 training error와 test error가 layer가 얕은 network보다 높게 나타난다 *(Degradation Problem)*. 이것은 overfitting의 문제가 아니라 최적화가 쉽지 않기 때문에 나타나는 현상이며, 모든 시스템의 최적화가 유사하지 않음을 나타낸다.

본 논문에서는 이 문제를 *Residual Learning Framework*를 도입해서 해결한다.

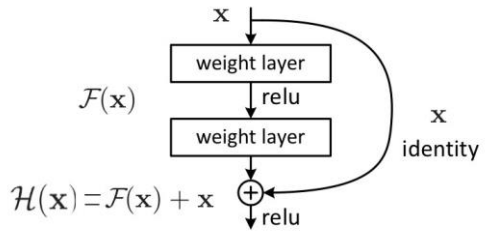

Figure 2. Residual learning: a building block. + 손그림

이는 기존의 방식으로 H(x)를 optimization하는 것이 아닌 $F(x) = 0$이 되도록 optimize를 하는 것이다. Residual Learning의 residual(잔차)은 H(x)와 x의 잔차인 F(x)를 의미하는 것이다.

$H(x) = F(x) + x$는 *shortcut connection(skip connection)*으로 구현할 수 있다. *Shortcut connection*이란 하나이상의 layer를 skip(건너뜀)하는 것이다. 본 논문에서는 Figure2와 같이 shortcut connection이 identity mapping을 수행하고, 그 출력을 stacked layer의 출력에 더해진다. Input인 x가 output에 더해지는 것이기 때문에 추가적인 파라미터가 필요하지도 않고 연산복잡도 또한 증가하지 않는다. 또한 backpropagation의 gradient 측면에서도 유리하다.

2. Related Work

Residual Representations.

Shortcut Connections.

3.Deep Residual Learning

3.1 Residual Learning

1에서 설명한 것과 같이

3.2 Identity Mapping by shortcuts

3.3 Network Architectures

Plain Network

Residual Network

3.4 Implementation

4. Experiments

4.1 ImageNet Classification

ImageNet 2012 classification dataset은 1000개의 class로 구성되어 있으며, train set은1.28million, validation set은 50k, test set은 100k개의 이미지 data가 있다. 테스트 결과는 top-1 error와

top-5 error를 모두 평가한다.

Plain Networks

Figure3의 plain network의 형태로 쌓은 18-layer network와 34-layer network를 비교해보자.

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

위의 Table2에서 보면 18-layer network에 비해, 34-layer network의 validation error가 높다.

아래 Figure4를 통해 training error와 validation error를 확인해 보자.

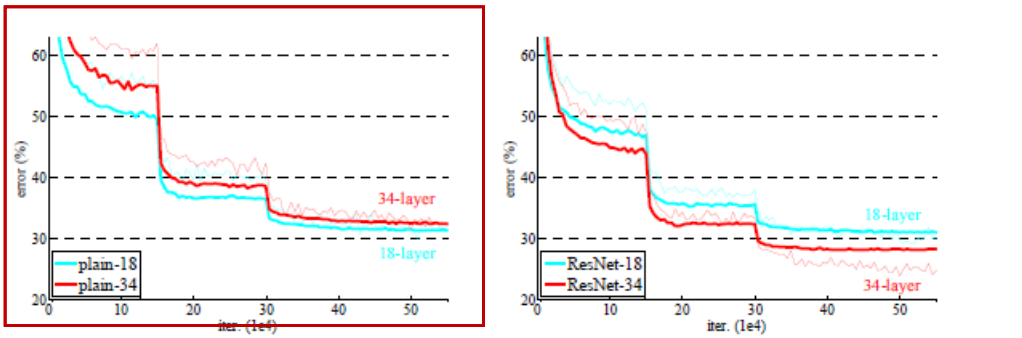

Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

(왼쪽-plain net 오른쪽-residual net / 굵은 선-validation error 얇은 선-training error)

Plain network의 경우 깊은 network가 (34-layer net) 오히려 error가 높게 나타난다.

본 논문에서는 이 문제가 vanishing gradients에 의한 것은 아니라고 말한다. plain network가 batch normalization을 사용하였기 때문에 forward propagated signal의 분산이 0이 되고, backward propagated gradients가 healthy norm을 보이기 때문이다.

Residual Networks

18-layer 및 34-layer residual network(이하 ResNet)를 평가한다. Figure3의 34-layer residual network와 동일하게 각각의 3x3 filter pair에 shortcut connection을 추가했다. 모든 shortcut connection은 identity mapping을 사용하며, 차원을 맞추기 위하여 zero-padding을 사용한다.

(추가되는 parameter 없다고 3번째 말하는중.....ㅡ,.ㅡ)

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

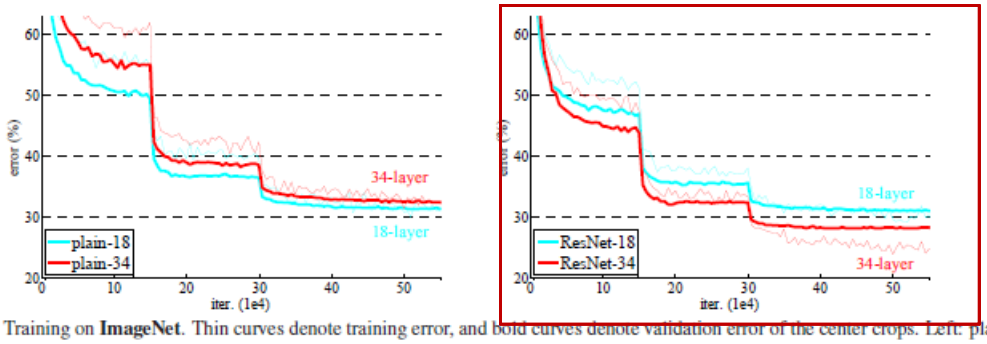

Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

(왼쪽-plain net 오른쪽-residual net / 굵은 선-validation error 얇은 선-training error)

위의 Table2와 Figure4로 알 수 있는 세가지 observation이 있다.

첫째, 34-layer ResNet이 18-layer ResNet보다 우수한 성능을 보인다. 이는 위의 Plain network에서 직면한 성능저하 문제를 잘 해결하고 깊은 network가 높은 정확성을 보장하는 것을 의미한다.

둘째, 34-layer ResNet과 34-layer plain network와 비교할 때, validation data에 대한 top-1 error를 3.5% 줄였다. 이는 extremely deep systems에서도 residual learning이 효과적임을 나타낸다.

셋째, Table2에서 18-layer ResNet과 18-layer plain network를 비교하면 accuracy는 비슷하지만, Figure4에서 보면 ResNet의 경우가 더 빨리 수렴하는 것을 알 수 있다. 이는 SGD가 ResNet에서 더 빠르게 수렴한기 때문에 optimization이 더 쉽다는 것을 의미한다.

Identity vs. Projection Shortcuts

model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, **10-crop** testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

위에서 parameter-free한 identity shortcut이 학습에 용이하다는 것을 알아봤다.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$ projection shortcuts에 대해 조사해보겠다.

(in **3.2 ,** projection shortcut : W_i는 x의 dimension을 맞추기 위해서 사용)

Table3의 3가지의 옵션->(A,B,C).

A. zero-padding shortcut은 차원을 높이기 위해 사용, 모든 shortcut은 parameter-free.

B. projection shortcut은 차원을 높이기 위해 사용, 다른 shortcut은 모두 identity

C.모든 shortcut은 projection.

성능은 C>B>A이지만, 차이가 미미하다. Projection shortcut은 성능 저하를 해결하는데 크게 효과를 내지 못한다는 것을 알 수 있다.

(본 논문에서는 옵션 C를 사용하지 않는다. Memory/time complex와 model size를 줄이기 위해,)

Deeper Bottleneck Architectures

ImageNet data set사용을 위한 deep network구조를 알아보자. 이경우 Training 시간을 고려하여 building block architecture구조에 Bottleneck 구조를 사용한다.

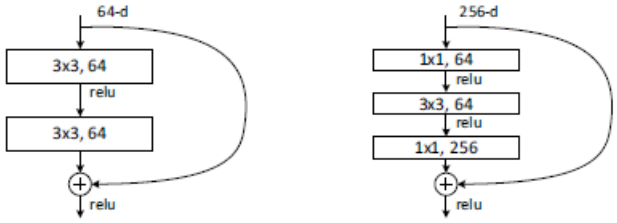

Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

(왼쪽 -> ResNet34, 오른쪽 -> bottleneck 구조 for ResNet50/101/152)

Figure5 오른쪽 bottleneck구조에서 1x1 convolution layer는 차원을 조절하기 위해 사용되며, 3x3 convolution layer는 input/output의 차원을 줄이는 용도로 사용된다.

50-layer ResNet

34-layer ResNet의 2-layer block들을 3-layer bottleneck block으로 대체하여(Figure5의 왼쪽 구조를 오른쪽 구조로 대체) 50-layer ResNet을 구성했다. 옵션*B. (projection shortcut은 차원을 높이기 위해 사용, 다른 shortcut은 모두 identity)를 사용*했다.

101-layer and 152-layer ResNets

3-layer bottleneck block을 더 추가하여 101-layer/152-layer ResNet을 구성했다. 34-layer보다 50/101/152의 정확도가 더 높다. 특히 152-layer ResNet은 VGG와 비교하였을 때, 복잡도와 연산량이 적다.

Comparisons with State-of-the-art Methods

method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

method	top-5 err. (test)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

Table.4에서는 previous best single-model의 성능과 비교한다. 본논문의 baseline인 34-layer ResNet은 previous best에 비준하는 정확도를 달성했으며, 152-layer ResNet의 single-model top-5 error는 4.49%를 달성했다. 또한, 서로 다른 depth의 ResNet을 6개 ensemble하여 top-5 test error를 3.57%까지 달성했고. 이는 ILSVRC 2015 classification task에서 1위를 차지했다.

4.2 CIFAR-10 and Analysis

6n+2개의 stacked weighted layer로 구성된 간단한 architecture를 사용했다.

input은 per-pixel mean subtracted 32x32 이미지이다.

첫 번째 layer는 3x3 conv layer이다.

다음에는 크기가 각각 {32, 16, 8}인 feature map에 3x3 conv가 적용된 6n개의 layer stack을 사용한다. 각 size마다 2n개의 layer로 구성된다.

filter의 수는 각각 {16, 32, 64}개 이다.

subsampling은 strides가 2인 conv layer로 수행한다.

네트워크의 종단에는 global average pooling과 softmax를 포함한 FC layer로 구성된다.

shortcut connection은 모두 identity shortcut로, 구조상 차이를 제외하고는 parameter 등의 모든 조건이 plain network와 동일하다.

학습 진행법은 다음과 같다.

4개의 pixel이 각 side에 padding되며, padded image와 horizontal flip 중에서 무작위로 32x32 crop을 샘플링 한다.

He initialization으로 weight 초기화를 수행한다.

decay는 0.0001, momentum은 0.9이다.

learning rate는 0.1부터 시작하여 32000/48000번 째 iteration에서 rate를 10으로 나누어 적용한다.

2개의 GPU에서 mini-batch size를 128로 했으며, 총 64000번의 iteration 동안 학습한다.

성능 테스트 시에는 32x32의 원본 이미지에 대한 single view만 평가한다.

참고 : https://datascienceschool.net/view-notebook/958022040c544257aa7ba88643d6c032/

논문을 리뷰한건지 번역한건지 모를정도로 양이 많아졌네욤 8ㅁ8