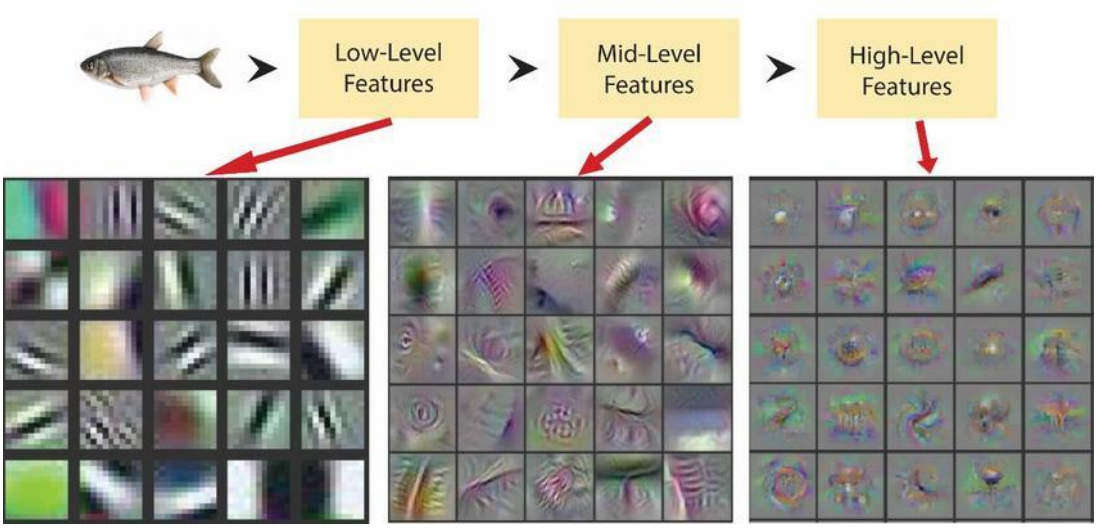
4. ResNet (Deep Residual Learning for Image Recognition)

Abstract

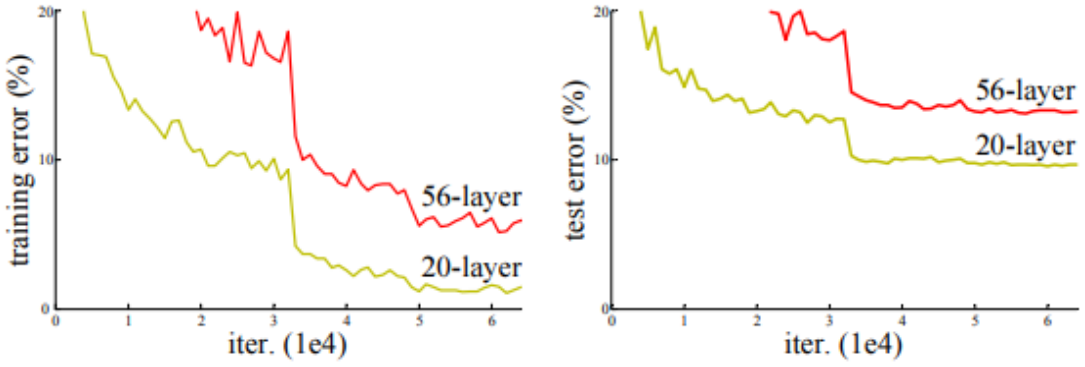
- 역대 ILSVRC 대회 결과를 보았을 때, Depth의 깊이가 모델의 성능에 큰 영향을 준다는 것을 알 수 있음
    - visual recognition task에서 depth는 매우 중요한 요소이지만, depth가 올라감에 따라 필연적으로 발생하는 문제가 있음
      → 오버 피팅, gradient의 소멸, 연산량의 증가 등
    - 따라서 심층 신경학습은 상당히 까다로운데
      이에 MicroSoft팀은 Resudual learning framework를 이용하여 이전보다 훨씬 깊은 네트워크를 더 쉽게 학습 시킬 수 있다고 함
        - Residual learning ⇒ 이전 layer의 결과를 다시 이용하는 것 이라고 볼 수 있음
        - 입력 layer를 다시 이용하는 residual function을 사용하여 더 쉬운 최적화와 깊은 네트워크에서의 정확도 향상이 가능했다고 함
        - ResNet은 VGGNet의 8배인 152 layer를 자랑하며 앙상블 기법을 적용해 오차릴 3.75%까지 줄임

Introduction

- Residual learning
    - 심층 신경망은 추상화에 대한 Low / Mid / High Level의 특징을 classifier와 함께 Multi-Layer 방식으로 통합
        - 각 추상화 level은 쌓인 layer의 수에 따라 더욱 높아질 수 있으며,
          높은 추상화 특징은 high layer에서 파악할 수 있다는 것



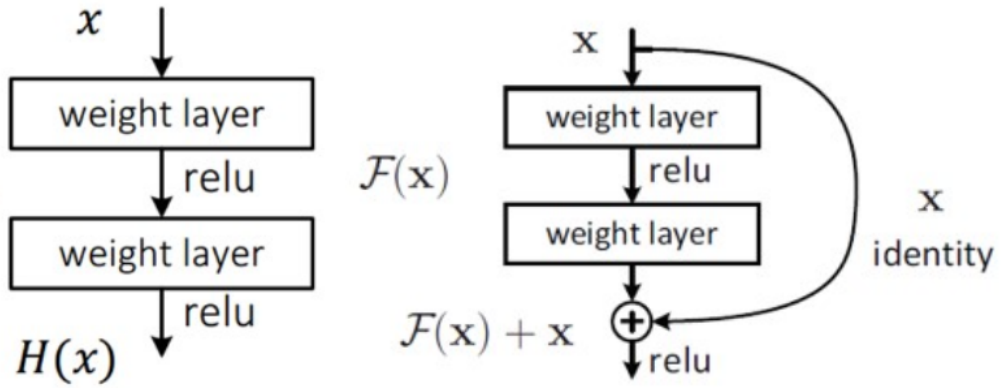
- Depth의 깊이가 중요해지면서, layer를 쌓는 만큼 더 쉽게 네트워크를 학습시킬 수 있는지에 대한 의문이 생기기 시작했고, 특히 그중에서도 Vanishing / Exploding gradient 현상이 큰 방해 요소였음
  - Vanishing Gradient - 네트워크의 깊이가 깊어질수록, 즉 Layer가 많아질수록 그라디언트가 점점 작아져서 결국 0에 가까워지는 현상
    - 활성화함수로 Sigmoid나 Tanh를 사용할 때 더욱 두드러짐, 그라디언트가 작어지면 네트워크의 초기 층에 위치한 가중치들은 거의 업데이트되지 않게 되어 학습이 효과적으로 진행되지 않음
  - Exploding Gradient - 그라디언트가 지나치게 커져서 네트워크의 가중치 업데이트가 너무 크게 일어나는 것을 말함
    - 네트워크의 가중치가 크거나 학습률이 높을 때 발생하며, 모델이 불안정해지고 종종 수치적으로 불안정해져 학습이 제대로 수행되지 않을 수 있음
- SGD를 적용한 10개의 Layer까지는 nomalization 기법과 Batch normalization과 같은 intermediate normalization layer를 사용했을 경우 문제가 없었음



- 심층 신경망의 경우 성능이 최고 수준에 도달할 때 degradation 문제가 발생했고, 네트워크의 깊이가 깊어 짐에 따라 정확도가 포화하고 급속하게 감소하는것을 의미
  - degradation 문제의 원인은 오버 피팅이 아닌 그저 layer의 수가 더 추가 되었기 때문인데 test error만이 아닌, training error도 함께 높아졌기 때문임, 이는 서로 다른 system들이 최적화되는 방식이 다르다는 점을 의미
- deeper 모델에서도 제한된 상황에서는 최적화될 수 있는 방법이 있음
  - 추가된 레이어가 identity mapping이고, 추가되지 않은 다른 레이어들은 더 얕은 모델에서 학습된 layer를 사용하는것
    - 이 같이 제한된 상황에서의 deeper 모델은 shallower 모델보다 더 낮은 training error를 가져야 하고, 마이크로소프트 팀은 이 개념을 모델에 적용
  - 기존 네트워크는 입력 x를 받고, layer를 거쳐 H(x)를 출력하는데 이는 입력값 x를 타겟값 y로 mapping하는 함수 H(x)를 얻는 것이 목적
    - ResNet의 Residual Learning은 H(x)가 아닌 출력과 입력의 차인 H(x) - x를 얻도록 목표를 수정
    - Residual Function인 F(x) = H(x) - x를 최소화 시켜야하고 이는 즉, 츨력과 입력의 차를 줄인다는 의미
    - x의 값은 도중에 바꾸지 못하는 입력값이므로 F(x)가 0이 되는 것이 최적의 해이고,
      결국 0 = H(x) - x로 H(x) = x
    - 즉 H(x)를 x로 mapping 하는 것이 학습의 목표
    - H(x) = x라는 최적의 목표값이 사전에 pre-conditioning으로 제공되기에 Identity mapping이 Identity mapping인 F(x)가 학습이 더 쉬워지는 것
    - H(x) = F(x) + x 이므로 네트워크 구조 또한 크게 변경할 필요 없이 단순히 입력에서 출력으로 바로 연결되는 shorcut만 추가하면 되기 때문, 입력과 같은 x가 그대로 출력에 연결되기에 파라미터 수에 영향이 없으며, 덧셈이 늘어나는 것을 제외하면 shortcut연결을 통한 연산량 증가는 없음
    - 곱셈 연산에서 덧셈 연산으로 변형되어 몇 개의 layer를 건너뛰는 효과가 있어 forward와 backward path가 단순해지는 효과가 있었으며, gradient의 소멸문제를 해결할 수 있었음
      - 기존 신경망은 입력x를 받아 여러 Layer를 통과하면서 최종적으로 출력H(x)를 만들어내는 것이 목표임 이때, H(x)는 입력 x로부터 어떤 타

겟 y를 예측하기 위한 매핑 함수

- ResNet에서는 매핑을 조금 다르게 접근 → 목표함수 H(x)자체를 직접 학습시키는것이 아닌, 입력 x와 출력 H(x)의 차이, 즉 잔차(Residual) F(x) = H(x) - x를 학습 시키는 것
- ResNet의 기본 가정은 신경망이 F(x)를 학습하는 것이 H(x)를 직접 학습하는 것보다 쉬울 수 있다는 것, 만약 최적의 함수가 입력을 그대로 출력으로 전달하는 것이라면 이상적으로는 F(x)는 0이 될 것 이므로 네트워크는 해당 잔차함수인 F(x)를 최소화 하려고 시도하게 됨
- x가 Residual Block을 통과한 후, 결과 F(x)에 다시 입력 x를 더해줌, 이렇게 함으로써 만약 네트워크가 학습해야 할 함수가 단순히 입력 x를 그대로 전달하는 것이라면, F(x)는 0에 가까운 값이 되어야 하고 네트워크는 이러한 아이덴티티 매핑(Identity mapping)을 학습하는 데 집중하게 됨

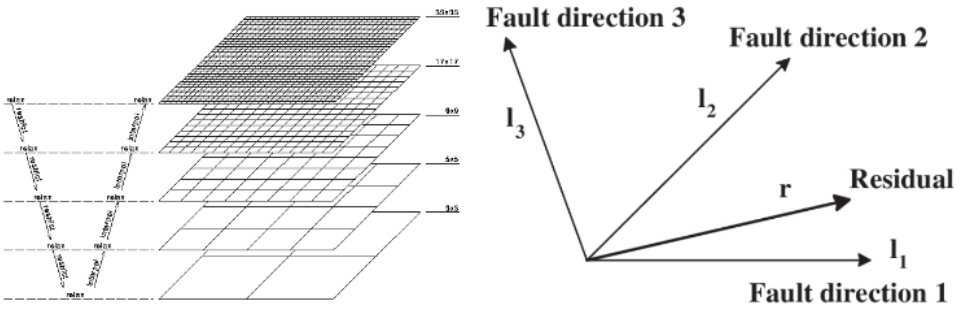


기존 네트워크와 ResNet의 구조

- ResNet은 Depth가 매우 깊어도 최적화가 쉬웠지만, 일반적인 CNN Model인 PlainNet은 Depth가 증가하면 Training Error도 함께 증가함
- 반면에 ResNet은 매우 깊은 구조 덕분에, 높은 정확도를 쉽게 얻을 수 있었고 그 결과 이전의 다른 네트워크보다 월등함

Related Work

- Residual Representations
  - 벡터 양자화에 있어 residual vector를 인코딩하는 것이 original vector보다 훨씬 효과적
    - 벡터 양자화란 특징 벡터 X를 클래스 벡터 Y로 mapping하는 것을 의미



multi grid 방식과 residual vector 방식

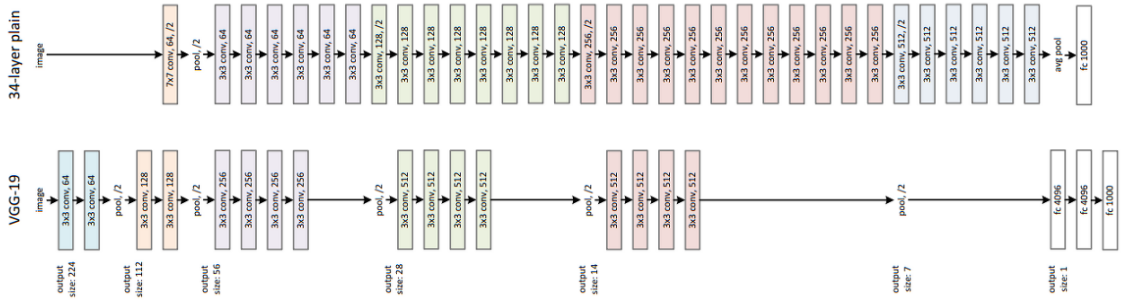
- Low-Level 비전 및 컴퓨터 그래픽 문제에서 편미분 방정식을 풀기 위해 멀티 그리드 방식을 많이 사용해왔는데, 이 방식은 시스템을 여러 scale의 하위 문제로 재구성하는 것
  - 하위 문제는 더 큰 scale과 더 작은 scale간의 residual을 담당
- 멀티 그리드 방식 대신에 두 scale 간의 residual 벡터를 가리키는 변수에 의존하는 방식이 있는데 이를 계층 기반 pre-conditioning이라고 한다. 이 방식은 해의 residual 특성에 대해 다루지 않는 기존 방식보다 훨씬 빨리 수렴하는 특징이 있음
- 합리적인 문제 재구성과 전제 조건(pre-conditioning)은 최적화를 더 간단하게 수행해준다는 것을 의미

- Shortcut Connections
  - ResNet의 Shortcut Connection은 parameter가 전혀 추가되지 않으며, 0으로 수렴하지 않기에 절대 닫힐 일이 없어 항상 모든 정보가 통과되어 지속적으로 residual function을 학습하는 것이 가능

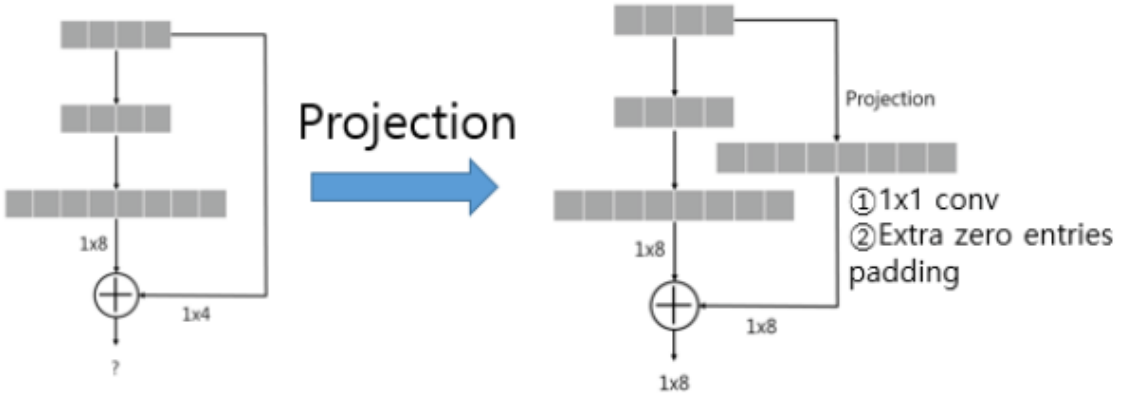
Deep Residual Learning

- Residual Learning
  - Identity mapping이 최적일 가능성이 낮다고 하지만 ResNet에서 제안하는 재구성 방식은 문제에 pre-conditioning을 추가하는데 도움을 줌
  - pre-conditioning으로 인해 Optimal function이 zero mapping보다 identity mapping에 더 가깝다면, solver가 identity mapping을 참조하여 작은 변화를 학습하는 것이 새로운 function을 ㅎ학습하는 것보다 더 쉬울 것이라고 주장함
- Identity Mapping by Shortcuts
  - 파라미터나 연산 복잡성을 추가하지 않는다. 이 때, F + x 연산을 위해 x와 F의 차원이 같아야 하는데, 이들이 서로 다를 경우 linear projection인 Ws를 곱하여 차원을 같게 만들 수 있는데
    Ws는 차원을 매칭 시켜줄 때에만 사용한다

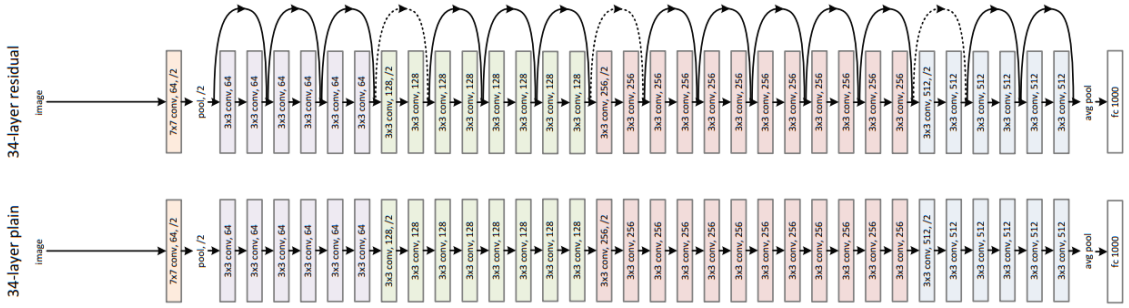
- Plain Network
  - Baseline Model로 사용한 plainNet은 VGGNet에서 영감을 받아 conv filter의 사이즈가 3 x 3이고, 다음 2가지 규칙에 기반하여 설계 됨
    - Output feature map의 size가 같은 layer들은 모두 같은 수의 conv filter를 사용함
    - Output feature map의 seize가 반으로 ㅈ루어들면 time complexity를 동일하게 유지하기 위해 필터 수를 2배로 늘림
    - 추가적으로 만약 downsampling을 수행한다면 pooling을 사용하는 것이 아니라 stride가 2인 conv filter를 사용하며 마지막으로 모델 끝단에 GAP을 사용하고 사이즈가 1,000인 FC layer와 Softmax를 사용함
    - 결과적으로 전체 layer수는 34인데 이는 VGGNet보다 적은 필터와 복잡성을 가짐



- Residual Network
  - Residual Network는 Plain 모델에 기반하여 Shortcut connection을 추가하여 구성
  - input과 output의 차원이 같다면 identity shortcut을 바로 사용하면 되지만, dimension이 증가했을 경우 두 가지 선택권이 있음
    1. zero padding을 적용하여 차원을 키워줌
    2. 앞서 다뤘던 projection shortcut을 사용함 (1 x 1 convolution)



- 이때, shortcut이 feature map을 2 size씩 건너뛰므로 stride를 2로 설정



- Implementation
  - 모델 구현은 다음과 같음
    1. 짧은 쪽이 [256, 480]사이가 되도록 random하게 resize 수행
    2. horizontal flip 부분적으로 적용 및 per-pixel mean을 빼줌
    3. 224 × 224 사이즈로 random 하게 crop 수행
    4. standard color augmentation 적용
    5. z에 Batch Normalization 적용
    6. He 초기화 방법으로 가중치 초기화
    7. Optimizer : SGD (mini-batch size : 256)
    8. Learning rate : 0.1에서 시작 (학습이 정체될 때 10씩 나눠 줌)
    9. Weight decay : 0.0001
    10. Momentum : 0.9
    11. 60 X 10^4 반복 수행
    12. dropout 미사용

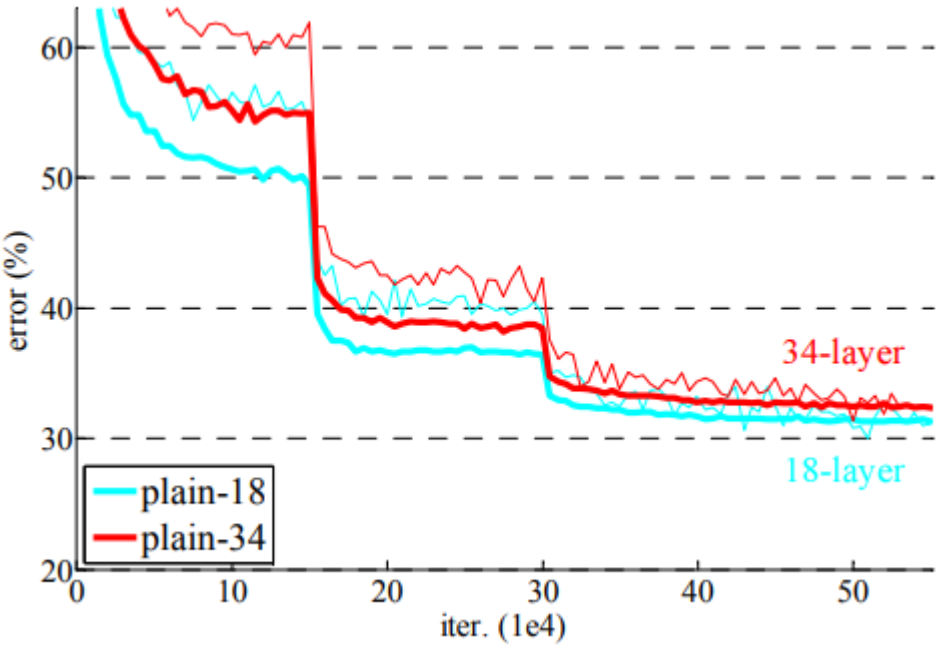
- 테스트 단계에서는 10-cross validation 방식을 적용하고, multiple scale을 적용해 짧은 쪽이 {224, 256, 384, 480, 640}중 하나가 되도록 resize 한 후, 평균 score를 산출

Experiments

- ImageNet Classification
  - plainNet과 ResNet을 대상으로 ImageNet을 이용해 수행한 실험의 결과와 그 특징에 대해 알아보고 모델 구조의 세부적인 내용은 아래 표를 보면 알 수 있다.

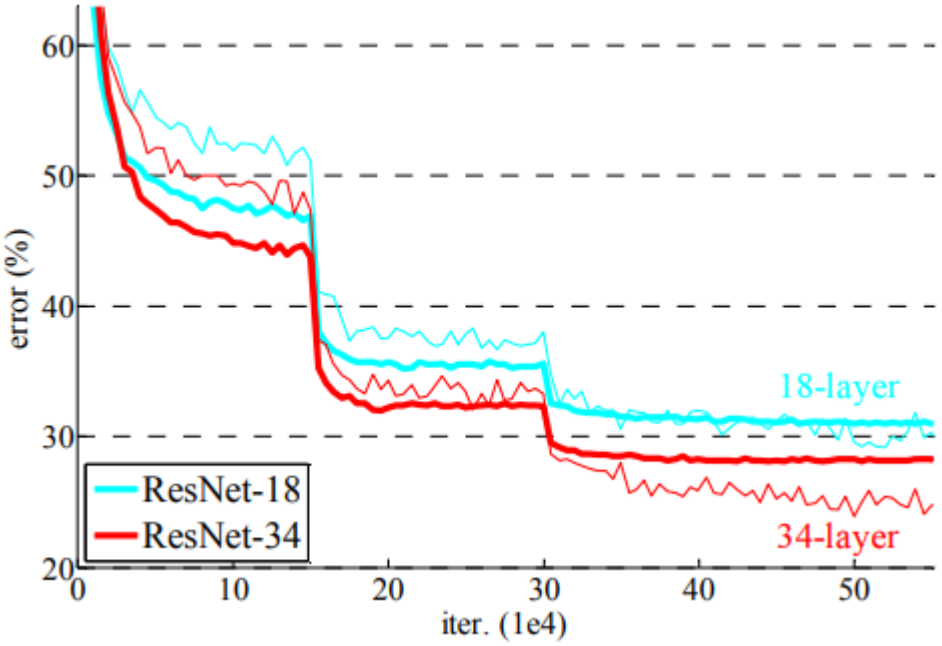
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

  - Plain Networks
    - 18 layer의 얕은 plain 모델에 비해 34 layer의 더 깊은 plain 모델에서 높은 Validation error가 나타 났다고 함
    - training / validation error 모두를 비교한 결과, 34 layer plain 모델에서 training error도 높았기 때문에 degradation 문제가 있다고 판단



- 이러한 최적화 문제는 Vanishing gradient 때문에 발생하는 것은 아니라 판단하였고 plain 모델은 Batch Normalization이 적용되어 순전파 과정에서 모든 신호의 variance는 0이 아니며, 역전파 과정에서의 기울기 또한 healthy norm을 보였기 때문
- 순전파, 역전파 신호 모두 사라지지 않았으며, 실제로 34-layer의 정확도는 경쟁력 있게 높은 수준이었음
- deep palin model은 exponentially low convergence rate를 가지기 때문에 training error의 감소에 좋지 못한 영향을 까쳤을것이라 추측

- Residual Networks
  - 18 layer 및 34 layer ReNet을 plain 모델과 비교
  - 모든 Shortcut은 identity mapping을 사용하고, 차원을 키우기 위해 zero padding을 사용하여 파라미터 수는 증가하지 않았음
    - residual learing으로 인해 상황이 역전되어 34-layer ResNet이 18-layer ResNet보다 2.8% 가량 우수한 성능을 보였음
      특히, 34-layer ResNet에서 상당히 낮은 training error를 보였고 이에 따라, validation 성능 또한 높아졌음. 이는 degradation 문제가 잘 해결되었으며, depth가 증가하더라도 좋은 정확도를 얻을 수 있음을 의미



- 34-layer ResNet의 top-1 error는 3.5%가량 줄었고, 이는 residual learing이 extremely deep system에서 매우 효과적임을 알 수 있음

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

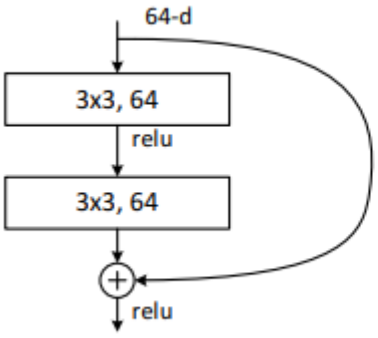
- 18-layer ResNet과 plainNet을 비교했을 때 성능이 거의 유사했지만, 18-layer의 ResNet이 더 빨리 수렴하였음
- 모델이 과도하게 깊지 않은 경우(18-layer), 현재의 SGD solver는 여전히 plainNet에서도 좋은 solution을 찾을 수 있지만, ResNet은 같은 상황에서 더 빨리 수렴할 수 있음

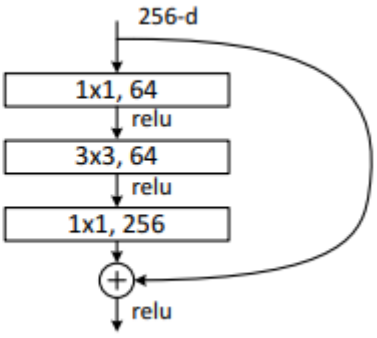
- Identidy vs Projection Shortcuts
  - projection shortcut
    1. zero-padding shortcut을 사용한경우 : (dimention matching시에 사용) 모든 shortcut은 parameter-free
    2. projection shortcut을 사용한경우 : (dimention을 키워줄 때에만 사용) 다른 모든 shortcut은 identity 함

3. 모든 shortcut으로 projection shortcut을 사용한 경우

- 세가지 옵션 모두 plain model보다 좋은 성능을 보였고, 그 순위는 1 < 2 < 3 순이였음
  - A < B는 zero-padded 차원이 residual learning을 수행하지 않기 때문
  - B < C는 projection shortcut에 의해 파라미터가 추가 되었기 때문
- 3가지 옵션의 성능차가 미미했기에 projection shortcut이 degradation 문제를 해결하는데 필수적이지 않다는 것을 확인할 수 있음
- memory / time complexity와 model size를 줄이기 위해 이 논문에서는 C 옵션을 사용하지 않음
- 특히 Identity shortcut은 bottleneck 구조의 복잡성을 높이지 않는 데에 매우 중요하기 때문

- Deeper Bottleneck Architectures
  - ImageNet에 대하여 학습을 진행할 때 training time이 매우 길어질 것 같아 building block을 bottleneck design으로 수정하였다고 함
  - 각각의 residual function인 F는 3-layer stack 구조로 바뀌었는데, 이는 1 x 1, 3 × 3, 1 × 1 conv로 구성되어 있음
    - 이 때 1 x 1은 dimension을 줄이거나 늘리는데 사용되어 3 x 3 layer의 input / output 차원을 줄인 bottleneck 구조를 만들어 줌





  - parameter-free한 identity shortcut은 bottleneck구조에서 특히 중요함
  - 만약 identity shortcut이 projection shortcut으로 대체되면, shortcut이 2개의 high-dimensional 출력과 연결되어 time complexity와 model size가 2배로 늘어난다.

(위 그림에서 64-d가 256-d로 늘어난건, identity shortcut을 유지하기 위해 zero-padding을 통해 차원을 눌려준것으로 생각)

- 따라서 identity shortcut은 bottleneck design을 더 효율적인 모델로 만들어 줌

- 50-layer ResNet
  - 34-layer ResNet의 2-layer block을 3-layer bottleneck block으로 대체하여 50-layer ResNet을 구성, 이 때 dimension matching을 위해 2옵션을 사용

- 101-layer and 152-layer ResNet
  - 더 많은 3-layer block을 사용하여 101-layer 및 152-layer ResNet을 구성
  - depth가 상당히 증가하였음에도 VGG-16 / 19 모델보다 더 낮은 복잡성을 가졌으며,
    degradation 문제없이 상당히 높은 정확도를 보임

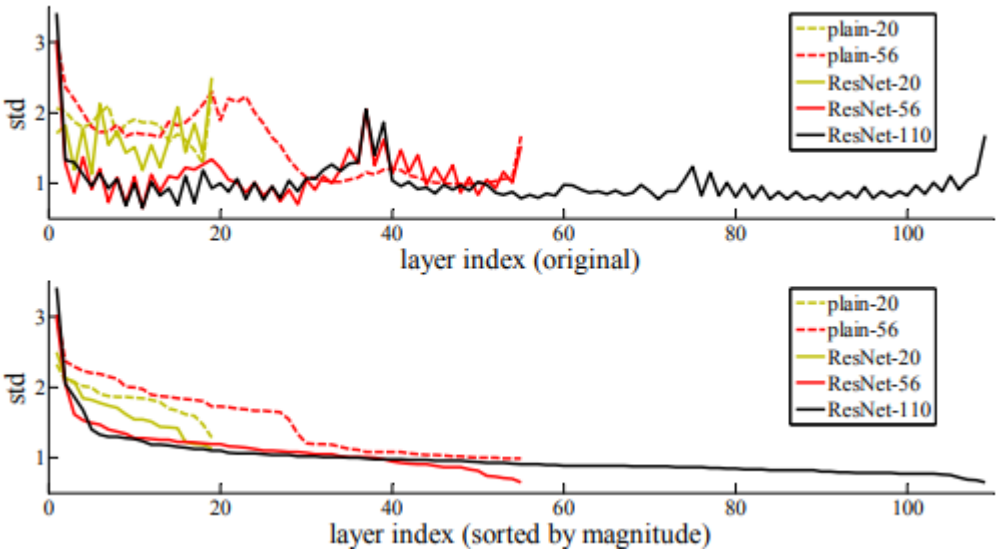
method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

- Comparisons with State-of-the-art Methods

        - ResNet의 single 모델(앙상블 적용 안 한 모델)은 앙상블이 적용된 이전의 다른 모델을 능가하였고, 앙상블을 적용했을 경우, 무려 top-5 error 3.57%를 달성할 수 있었음

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

- CIFAR-10 and Analysis
    - Analysis of Layer Responses
        - ResNet의 response가 plainNet보다 상대적으로 많이 낮게 나왔는데, 이는 residual function이 non-residual function보다 일반적으로 0에 가까울 것이라는 주장을 뒷받침 해줌
            - depth가 깊어짐에 따라 response가 작아지는 것은 각 layer가 학습 시 signal이 변화하는 정도가 작다는 것을 의미



- Exploring Over 1000 layers
  - 1,000개 이상의 layer가 사용된 모델은 110-layer ResNet과 raining error가 비슷했지만, test결과는 좋지 못하였는데 이는 오버 피팅 때문인 것으로 판단되었음