Abstract

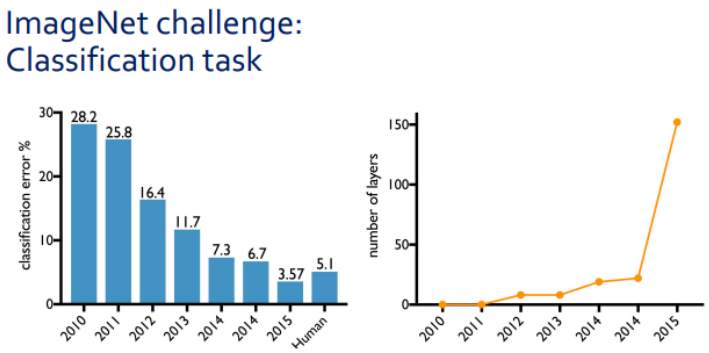

Figure 1: EEE3314.01-00 Lecture note #20

ILSVRC(ImageNet Large Scale Visual Recognition Challenge) [6]는 2010년에 시작되었다. 2010, 2011년에 우승을 한 알고리즘은 shallow architecture를 가졌었다. 하지만 2012년 CNN 기반의 딥러닝 알고리즘인 AlexNet [3]은 이전 대회의 모델에 비해 classification error를 약 10% 낮추었다. 이후 CNN기반의 모델이 나오면서 ILSVRC14에서는 기존 model보다 더 많은 convolution layer를 쌓아올렸고 Network의 깊이가 성능과 관계가 있음을 증명하였다. 2015년 ResNet에 이르러서는 사람의 인식 수준을 뛰어넘는 모델이 탄생하였다. 이 레포트는 준우승을 한 VGGNet [7]과 ILSVRC15에서 우승을 한 ResNet [1]에 대한 extended abstract이다. 각 모델의 성능과 layer의 수(깊이, Depth)를 Figure 1에서 확인할 수 있다.

VGGNet

1. Architecture

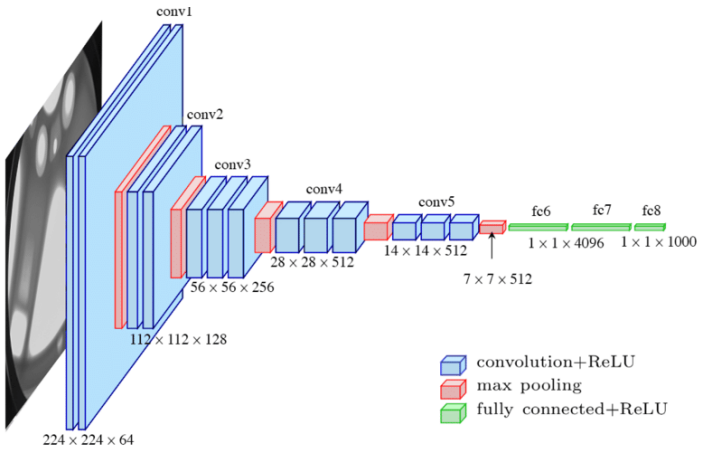

Figure 2: **VGGNet16**, Source: Researchgate.net

Figure2는 VGGNet의 대표적인 VGGNet16 model이다. VGGNet의 Architecture는 다음과 같다.

- Input으로 224×224 RGB image를 사용
- 모든 conv. layer에서 3×3 filter, convolution stride 1pixel, padding 1pixel 사용
  ↪ 입력 이미지의 spatial resolution이 유지
- 5번의 max pooling
- Activation function : ReLU
- feature extraction 후 두 개의 4096 channels Fully-Connected layers, 마지막으로 한 개의 1000 channels FC layer

2. Architecture in details

이때, VGG16 model에서는 1×1 filter가 사용되는데, 1×1 filter는 이미지의 행과 열 크기 변화 없이 Channel의 수를 조절할 수 있으며, 비선형성을 증가시킬 수 있다.
또한, 모든 conv. layer에서 3×3 filter를 사용한다는 점은 ILSVRC-2012, ILSVRC-2013에서 사용된 최고 성능의 configurations과 다르다. 이들은 첫 번째 conv. layer에서 상대적으로 큰 receptive fields의 7×7, 11×11 filter를 사용하였다. 3×3 conv layers의 stack은 5×5의 receptive fields효과, three stacks는 7×7의 receptive fields효과를 갖는다고 볼 수 있다. 하지만 3×3 conv layers의 stack을 사용함으로써 비선형성을 증가시킬 수 있고, 파라미터 수를 감소시킬 수 있다. [7]에 나온 예시로, 만약 입력과 출력의 채널 수를 C라 하면 3×3 conv. layer는 $3 \times (3 \times 3) \times C^2$, 7×7 conv. layer는 $7 \times 7 \times C^2$으로 7×7 filters가 81% 더 많은 파라미터가 필요하다.

3. Conclusion

VGGNet은 visual representations에서 model의 depth가 classification accuracy에 유익함을 보여준 것에 의의가 있다. VGGNet 이전의 이미지 분류에서 퍼포먼스가 좋은 Convolution Network를 활용한 모델에서 사용한 7×7, 11×11 filter를 사용하지 않고 비교적 작은 receptive field를 갖는 3×3의 작은 filter만을 사용하였다. 또한 모든 covolution layer에서 3×3 filter만 사용함으로써 overfitting되지 않고 이전보다 더 깊은 conv. layer를 쌓아 16-19 layers 신경망 모델을 성공적으로 학습할 수 있었다.

ResNet

1. Main Concept : Identity Mapping

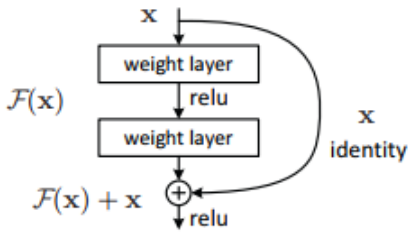

Figure 3: block : Shortcut Connection

ResNet에서 다루는 가장 중요한 concept은 "Identity mapping"이다. ResNet은 기존의 mapping방법을 취하지 않고 새로운 mapping을 이용하였다. Figure3의 block을 통과한 출력을 $H(x)$라고 할 때, ResNet에서 $H(x)$는 다음과 같이 표현할 수 있다 $H(x) = F(x, W_i) + W_s x$. F는 residual function을 의미하고, 함수F의 인자인 x는 입력, 함수F 외부의 x는 Identity mapping이다. W_i는 multiple convolution layers i번 째 weights이며 W_s는 함수F와 x의 dimension을 맞추기 위한 projection이다. Identity mapping은 앞의 Input을 그대로 받아오기 때문에 추가적인 파라미터가 필요하지 않다. 또한 글이 진행되면서 Identity mapping을 통해 기존의 model보다 ResNet의 향상된 성능을 확인할 수 있다.

This is an extended abstract. The full paper is available at the 1409.1556, 1512.03385

2. Performance of ResNet compared with previous one

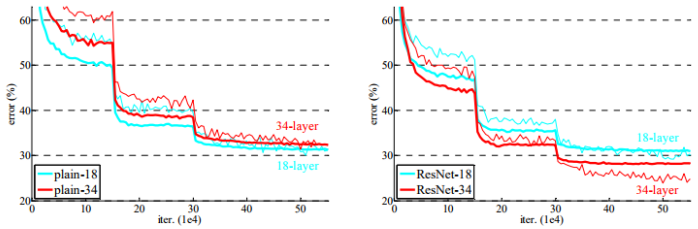

Figure 4: Graph, **Left**: plain network, **Left**: residual network (training error)

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

Figure 5: Table, **Left**: plain network, **Right**: residual network (top-1 error on ImageNet validation

Figure4와 Figure5는 이전 model과 비교하여 ResNet의 퍼포먼스를 가장 잘 보여주는 그래프와 표이다. Figure4의 왼쪽 그래프는 VGG의 형태를 따르는 CNN, 오른쪽은 잔여 학습(residual learning)을 적용한 CNN의 training error이다. VGG Network에서는 layers를 16, 19층으로 깊게 쌓아 풍부한 특징을 데이터로 추출해 낼 수 있었지만 더 깊게 layer를 쌓으면 training error가 증가하는 것을 확인할 수 있다. ResNet의 paper에서는 이 문제를 언급하고 개선할 수 있는 방안으로 residual Network를 제안한다. 또한 validation error에서 좋은 성능을 보여주는 것을 Figure5에서 확인할 수 있다.

3. Deep Residual Learning : Solution of *degradation*

Convoultion layers를 깊게 쌓음으로써 low, mid, high level의 feature가 추출되어 더욱 풍부한 특징으로 모델을 학습할 수 있다. 하지만, layer의 깊이가 깊어질수록 gradient가 사라져 학습이 제대로 이루어지지 못하는 vanishing gradient와 network가 깊어질수록 accuracy가 빠르게 감소하는 degradation문제가 발생하였다.
vanishing gradient 문제는 normalized initialization([5]), intermediate normalization layers([2]), stochastic gradient descent([4]) 등의 다양한 방법으로 개선되어 왔다. 그리고 degradation문제에 대해서는 1. Main Concept에 나와 있는 Deep Residual learning 프레임워크를 통해 해결할 수 있다.

4. Difference of concurrent Work

Residual 방법은 이미지 인식 분야 뿐만 아니라 low-level vision, 컴퓨터 그래픽스 등의 분야에서도 사용되어 왔고, 이 방법은 최적화에서 좋은 방안이라고 할 수 있다. 또한 Shortcut Connection은 오랜 기간동안 연구되어 왔고 많은 연구에서 사용되었다.
위 두 방법을 사용하여, Residual Network와 동시에 진행된 highway networks [8]에 대한 연구가 진행되었지만 ResNet만의 특징이 존재한다. ResNet은 parameter-free인 identity shortcuts을 가지며, 항상 residual function을 통해 학습한다는 것이다. 다른 큰 장점으로는 Network 구성에서 100 layers 이상의 depth를 쌓은 것이다.

5. Deeper Bottleneck Architectures

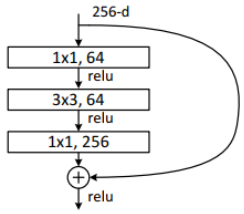

Figure 6: **bottleneck** building block for deeper design

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

Figure 7: Architecture of ResNet

Figure7에서 50, 101, 152 layer의 모양이 위의 18,34 layer와 다른 것을 확인할 수 있고. Figure6가 다른 모양의 layer인 BottleNeck design이다. BottleNeck design은 3개의 layers로 1×1과 3×3, 1×1 convolution을 순서대로 진행한다. 1×1 layers는 dimensions를 줄이거나 늘리는 역할을 한다. Convolution의 Parameters는 $ConvolutionParameters = KernelSize \times KernelSize \times InputChannel \times OutputChannel$로 계산이 되는데, 1×1 Convolution은 $1\times1\times InputChannel \times OutputChannel$으로 연산량이 작아 Output Channel을 줄이거나 키울 때 용이하다고 할 수 있다. 3×3 convolution은 동일한 채널을 갖는 1×1 convoultion 보다 연산량이 9배 많기 때문에, 1×1 convolution으로로 채널 수를 줄여 3×3 layer을 통과하고 다시 1×1conv. 으로 채널을 증가시킨 것이다. 또한 BottleNeck design역시 Identity mapping을 사용한다. 이로써 50, 101, 152와 같은 layer의 Network를 구성하였고, 152-layer ResNet이 11.3billion FLOPs, 16/19 layer의 VGGNet이 각각 15.3/19.6 billion FLOPs로 더 낮은 복잡도를 가지며 성능 역시 이전의 model보다 대폭 향상되었다.

FLOPs는 딥러닝 모델에서 계산 복잡도를 나타내기 위한 척도, 절대적인 연산량의 횟수를 의미한다.

* Implementation

ImageNet Classification에 대해 ResNet에서 사용된 최적화 알고리즘과 전처리는 아래와 같다.

- scale augmentation : [256, 480]의 이미지를 224×224size로 랜덤하게 자른다. 이때 horizontal flip된 이미지도 사용된다.
- 각 픽셀에 픽셀의 평균을 substract.
- convolution 후 batch normalization
- SGD with a mini-batch size of 256
- 초기 learning rate를 0.1로 설정한 후 점점 줄여나간다.
- weight decay : 0.0001, momentum : 0.9

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[4] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[5] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. PMLR, 2015.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, corr abs/1409.1556. *arXiv preprint arXiv:1409.1556*, 2015.

[8] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.