ResNet

Deep Residual Learning for Image Recognition (CVPR 2016)

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

Deeper neural networks are more difficult to train. **We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously.** We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that **these residual networks are easier to optimize, and can gain accuracy from considerably increased depth.**

On the ImageNet dataset **we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets** but still having lower complexity. **An ensemble of these residual nets achieves 3.57% error on the ImageNet test set**. This result won the 1**st place on the ILSVRC 2015 classification task**. We also present analysis on CIFAR-10 with 100 and 1000 layers.

잔여 학습(residual learning) 제안

본 논문에서는 깊은 네트워크를 학습시키기 위한 방법으로 **잔여 학습**을 제안한다.

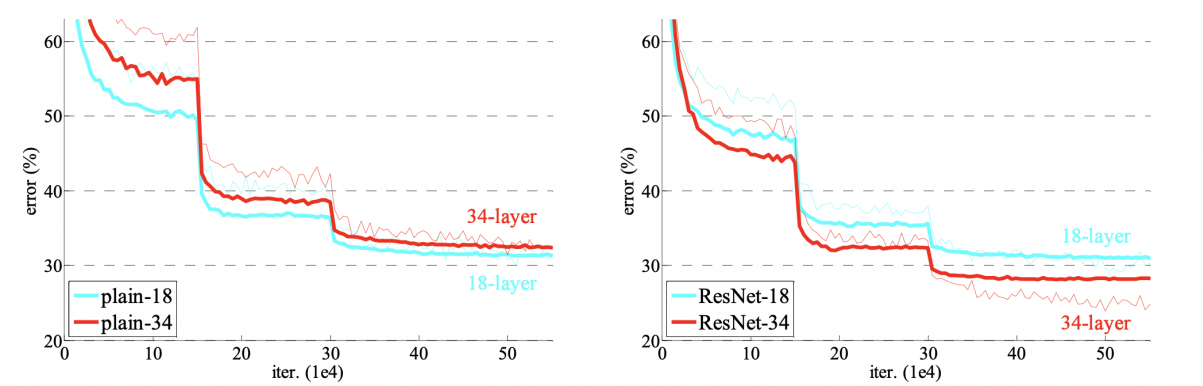


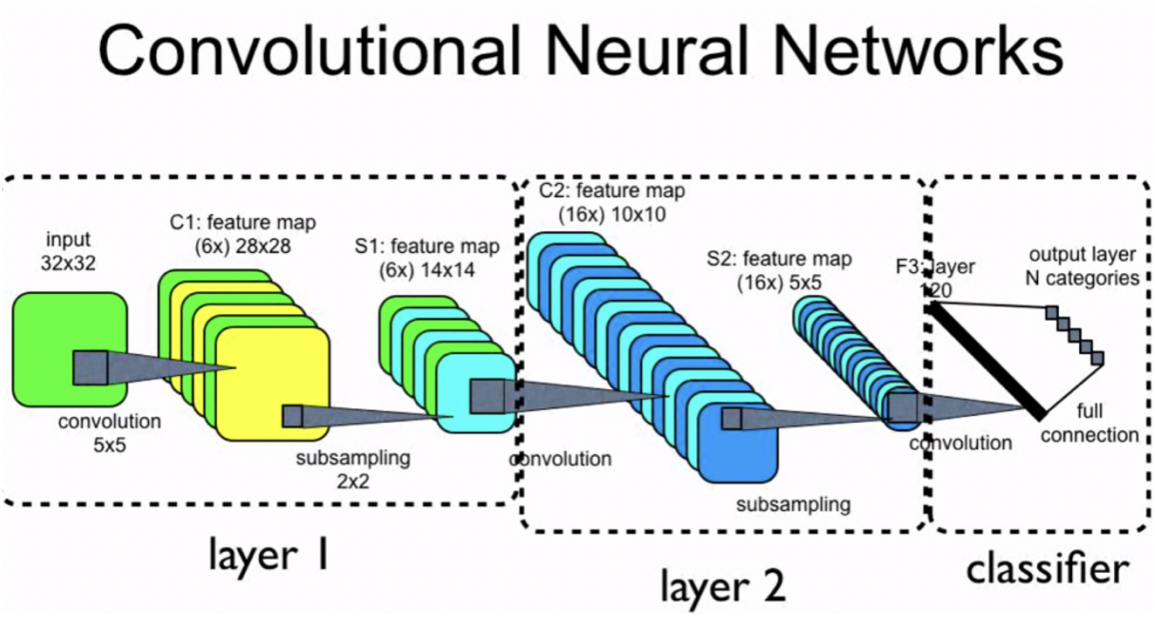
Figure 4. Training on ImageNet.

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.

CNN 모델의 특징 맵 (Feature Map)

일반적으로 CNN에서 레이어가 깊어질수록 채널의 수가 많아지고 너비와 높이는 줄어든다

컨볼루션 레이어의 서로 다른 필터들을 각각 적절한 특징(feature) 값을 추출하도록 학습된다

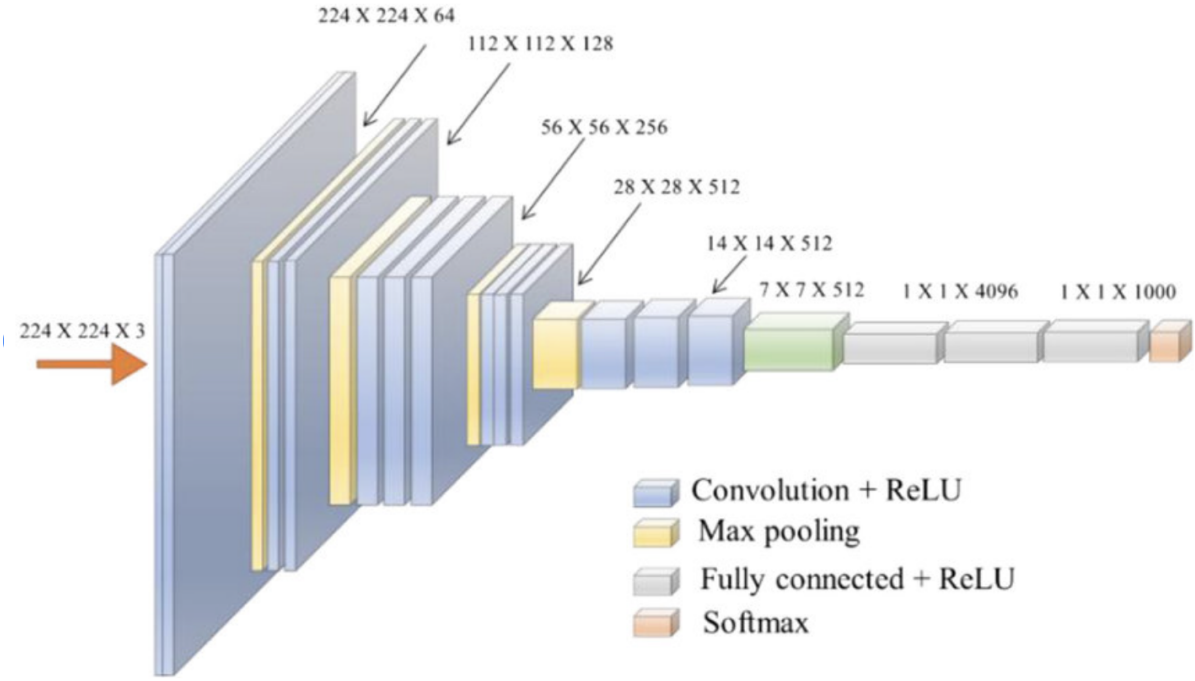


예시) VGG Network

Very Deep Convolutional Networks for Large-Scale Image Recognition (ICLR 2015)

VGG 네트워크는 작은 크기의 3x3 컨볼루션 필터를 이용해 레이어의 깊이를 늘려 우수한 성능을 보인다

https://arxiv.org/pdf/1409.1556.pdf



When deeper networks are able to start converging, a **degradation problem** has been exposed: with the network depth increasing, a**ccuracy gets saturated (which might be unsurprising) and then degrades rapidly.** Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher train- ing error, as reported in and thoroughly verified by our experiments

잔여 블록 (Residual Block)

In this paper, we address the degradation problem by **introducing a deep residual learning framework**. In- stead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. 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. We 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.

잔여 블록을 이용해 네트워크의 최적화 난이도를 낮춘다

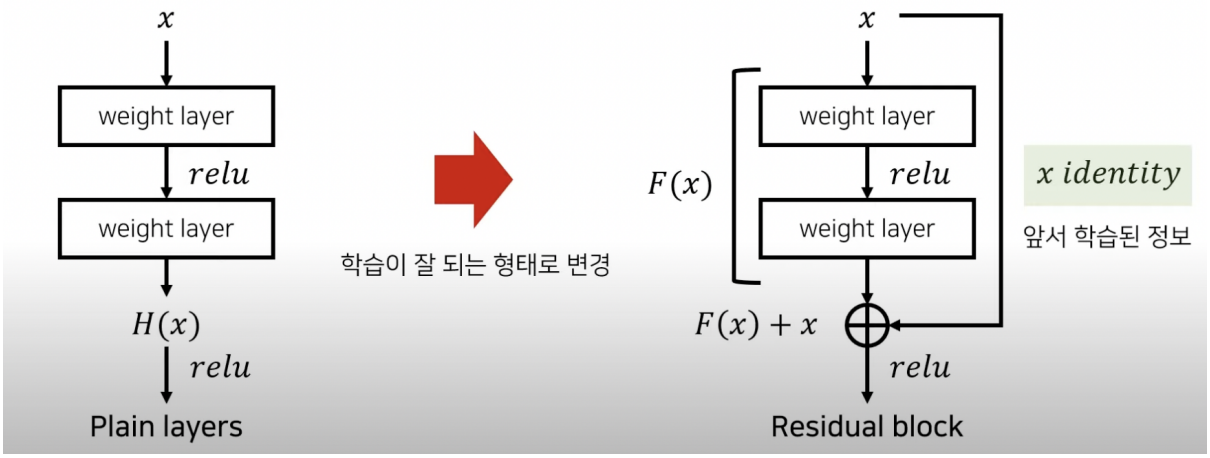
실제로 내재한 mapping인 H(x)를 곧바로 학습하는 것은 어려우므로 대신 F(x) = H(x) - x 를 학습한다

일반적으로 컨볼루션 연산을 통해 relu와 같은 activation function을 거쳐서 전체 네트워크 전체가 non-linear한 동작을 수행하도록 만든다

그 다음 연속적으로 컨볼루션 레이어를 거치는 형태로 네트워크를 구성한다

데이터 x를 입력했을 때 우리가 의도하는 매핑을 H(x)라고 가정해보자

이상적으로 동작하는 함수의 학습을 어렵기 때문에 학습이 잘 되는 형태인 F(x)를 이용하자



앞서 학습된 정보를 그대로 가져오고, 추가적으로 F(x)를 더해주는 것이다

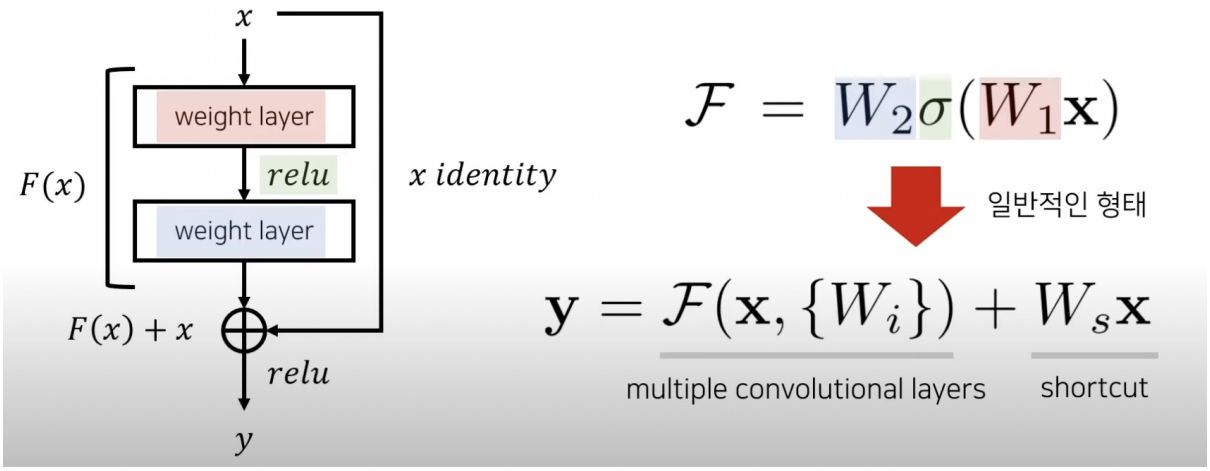
잔여한 정보인 F(x)만 추가적으로 학습할 수 있는 형태로 만들어주는 것이 더욱 쉽다

H(x)는 각각의 weight layer에 대해 가중치값을 개별적으로 모두 학습을 진행해야 한다

당연히 수렴 난이도가 더욱 높아지고 이는 레이어가 깊어질수록 심각해진다

반면에, F(x)는 기존의 학습했던 정보는 그대로 가져오고, 추가적으로 필요한 정보만 따로 학습을 진행한 후 더해주는 것이다

이를 수식적으로 자세하게 알아보자



F 함수에 입력값 x가 들어오면, w1을 곱하고 relu를 씌워준 후 그 결과에 w2를 곱해준다

멀티플 컨볼루션 레이어를 통해 가중치값을 여러번 사용할 수 있음을 보여준다

추가적으로, shortcut connection을 이용하여 기존의 입력값을 그대로 가져와서 결과에 더해준다

Top-1 error (%, 10-crop testing) on ImageNet validation

	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.

Error rates (%, 10-crop testing) on ImageNet validation

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**