Why good results in ResNet?

Reference

- Identity Mappings in Deep Residual Networks, kaiming He et al.

Intro

이번에 리뷰할 논문은 Identity Mappings in Deep Residual Networks로, 이 논문은 ResNet이 왜 좋은 성능을 가지는지 검증하는 성격을 가진다. 논문에서는 이를 propagation 관점에서 접근하며 다른 경우와 비교도 하면서 identity mapping이 좋은 이유를 설명한다.

들어가기에 앞서, ResNet을 간략하게 리뷰해보자. ResNet은 아래의 Residual Units을 쌓아 올린다.

$$y_l = h(x_l) + \mathcal{F}(x_l, \mathcal{W}_l) \tag{1}$$

$$x_{l+1} = f(y_l) \tag{2}$$

x_{l+1},x_l은 각각 l번째 residual unit의 output, input이다. $\mathcal{F}$는 residual function으로 두 개의 3×3 conv를 가리키고 f는 ReLU function이다. ResNet의 key idea는 residual function과 h를 identity function으로 보아 이 둘로 신경망을 구성하는 것이다

본 논문에서는 h, f가 identity mapping일 때 signal이 한 unit에서 다른 unit으로 forward, backward process 모두에서 직접적으로 propagate됨을 보인다. 또한 skip connection인 h의 역할에 대해 이해하기 위해 다양한 형태의 h를 살펴본다.

Analysis of Deep Residual Networks

논의를 더 진행하기 위해 activation function f를 identity mapping으로 두자. 즉,

$$x_{l+1} \equiv y_l \tag{3}$$

위와 같이 두면 (1)은 아래와 같이 쓸 수 있다.

$$x_{l+1} = x_l + \mathcal{F}(x_l, \mathcal{W}_l) \tag{4}$$

반복적으로 $x_{l+2} = x_{l+1} + \mathcal{F}(x_{l+1}, \mathcal{W}_l) = x_l + \mathcal{F}(x_l, \mathcal{W}_l) + \mathcal{F}(x_{l+1}, \mathcal{W}_l)$를 이용하면

$$x_L = x_l + \sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i) \tag{5}$$

도출한 특징 (5)는 nice properties를 가진다. 어떠한 깊은 L번째 unit이라도 그보다 얕은 l번째 feature x_l과 residual function의 합으로 표현할 수 있다. 또한 $l = 0$을 대입하면 x_L은 그 이전의 모든 residual function과 x_0으로 구성됨을 알 수 있다. 즉, ***합***으로 표현된다는 것인데, 이는 기존의 plain network인 $\prod_{i=1}^{L-1} W_i x_0$의 곱 형태랑 비교해본다면 큰 이점을 가진다. 즉, vanishing gradient 문제를 해결할 수 있다는 것이다.

(5)는 backward propagation에서도 큰 이점을 가진다. loss function을 L이라고 두면 chain rule에 의해서

$$\frac{\partial L}{\partial x_l} = \frac{\partial L}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial L}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1}\mathcal{F}(x_i, \mathcal{W}_i)\right) \tag{6}$$

(6)으로부터 $\frac{\partial L}{\partial x_l}$은 두 부분으로 쪼개짐을 알 수 있다.

- $\frac{\partial L}{\partial x_L}$은 weight layers ($\mathcal{W}_i$)의 걱정 없이 정보를 직접적으로 propagates하는 term이다.
- $\frac{\partial L}{\partial x_L}\left(\frac{\partial}{\partial x_l}\sum_{i=l}^{L-1}\mathcal{F}(x_i,\mathcal{W}_i)\right)$은 wieght layers을 통해 propagates하는 term이다.

결국 (5)와 (6)으로부터 signal이 한 unit에서 다른 unit으로 각각 직접적으로 forward, backward propagate됨을 확인할 수 있다.

On the Importance of Identity Skip Connections

그러면 identity mapping말고 다른 mapping, $h(x_l)=\lambda_l x_l$을 생각해보자. identity mapping과 동일한 논리로 아래를 도출할 수 있다.

$$x_{l+1}=x_l+\mathcal{F}(x_l,\mathcal{W}_l) \tag{7}$$

$$x_L=\prod_{i=l}^{L-1}\lambda_i x_l+\sum_{i=l}^{L-1}\hat{\mathcal{F}}(x_i,\mathcal{W}_i) \tag{8}$$

$$\frac{\partial L}{\partial x_l}=\frac{\partial L}{\partial x_L}\left(\prod_{i=l}^{L-1}\lambda_i+\frac{\partial}{\partial x_l}\sum_{i=l}^{L-1}\hat{\mathcal{F}}(x_i,\mathcal{W}_i)\right) \tag{9}$$

여기서 $\hat{\mathcal{F}}$은 scalars λ을 포함한다.
(8)은 factor $\prod_{i=l}^{L-1}\lambda_i$가 있는데, L이 매우 큰 신경망에서 이 factor는 매우 커지거나 vanish할 것이다. 따라서 shortcut으로 가는 것을 막고 weight layers을 통해 가도록 한다. 이는 최적화에 나쁜 영향을 줄 것이다.