Identity Mappings in Deep Residual Networks

2019-02-10 | 이규희

Abstract

(1) skip connection과 (2) after-addition activation으로

Identity mapping을 사용할 때

forward & backward signal이 directly propagate됨을 보여준다.

새로운 Residual Unit을 선보인다.

Creating a "direct" path for propagating information
– not only within a residual unit, but **through the entire network**.

If both h(x) **and f(y) are identity mappings**, the signal could be directly propagated.

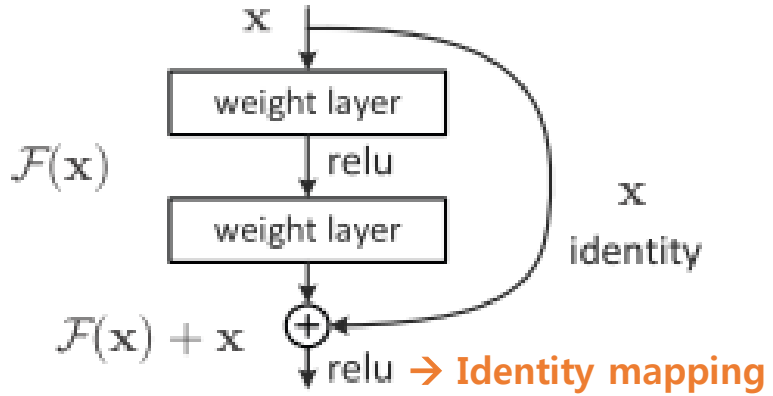
Skip -connection

F : Residual func.

$$\mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$$

$$\mathbf{x}_{l+1} = f(\mathbf{y}_l).$$

f : ReLU → **Identity mapping**



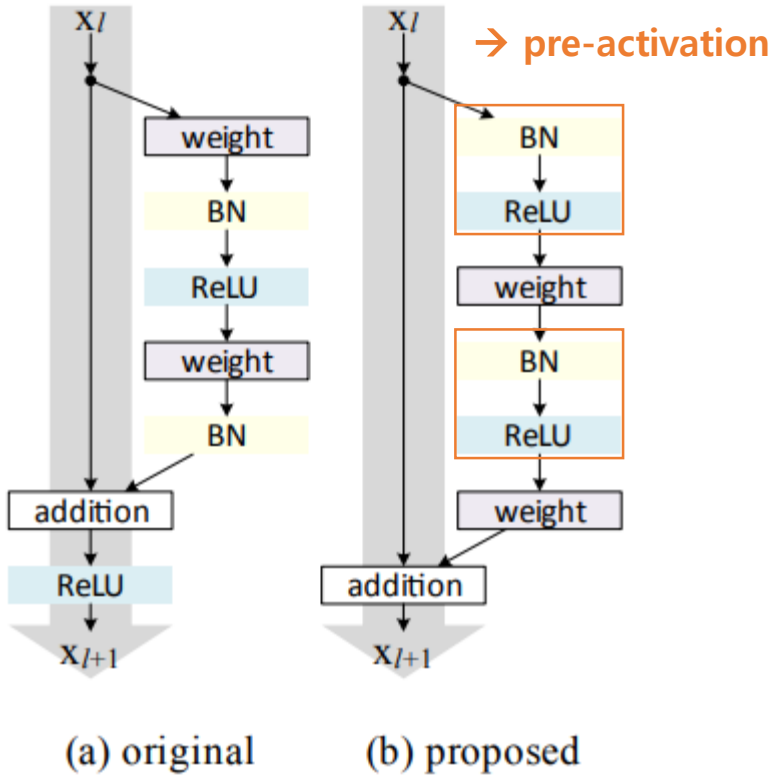
To construct an identity mapping,

we view the activation functions (ReLU & BN)

as **"pre-activation"** of the weight layers.

→ Easier to train and generalize

→ dimension of network depth ↑



2. Analysis of Deep Residual Networks

$$\mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$$
$$\mathbf{x}_{l+1} = f(\mathbf{y}_l).$$

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$$

Residual networks의 두 가지 성질을 도출할 수 있다.

(1) L-th unit = l-th unit + ∑ l~(L-1)th residual function

(2) 일반 network는 행렬"곱"인 반면,
이전 residual function의 "합"으로 표현된다.

$$(\mathbf{x}_{l+2} = \mathbf{x}_{l+1} + \mathcal{F}(\mathbf{x}_{l+1}, \mathcal{W}_{l+1}) = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) + \mathcal{F}(\mathbf{x}_{l+1}, \mathcal{W}_{l+1}),\ \text{etc.})$$

$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)$$

Addictive term (Xl)

- weight layer 고려할 필요없이 directly propagate
- Addictive term의 기울기는 미니배치 덕분에 cancel(vanish) 되지 않는다. 모든 미니배치 샘플에서 another term=-1일 가능성이 매우 낮기 때문이다.

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right)$$

3. On the Importance of Identity Skip Connections

$$\mathbf{x}_{l+1} = \lambda_l \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l)$$

$$\mathbf{x}_L = (\prod_{i=l}^{L-1} \lambda_i)\mathbf{x}_l + \sum_{i=l}^{L-1} \hat{\mathcal{F}}(\mathbf{x}_i, \mathcal{W}_i)$$

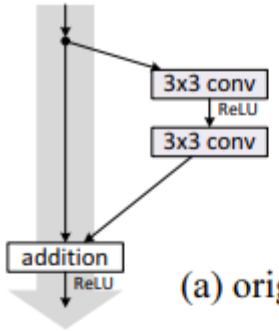
$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left((\prod_{i=l}^{L-1} \lambda_i) + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \hat{\mathcal{F}}(\mathbf{x}_i, \mathcal{W}_i) \right)$$

Identity shortcut이 아닌,
lambda만큼 scaling된 형태라고 생각해보자.

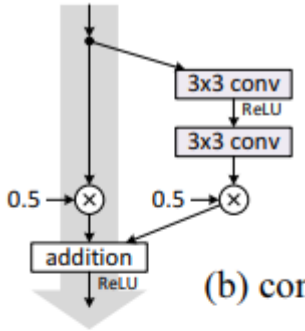
미분하면 lambda^N이 살아남는데,
만일 lambda > 1이면 발산, < 1이면 0에 수렴해서
최적화가 어렵다.

Scaling이 아닌 다른 형태더라도,
그 미분형태의 곱은 최적화를 어렵게 한다.

3.1 Experiments on Skip Connections

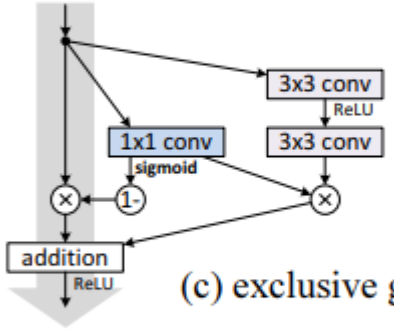


(a) original



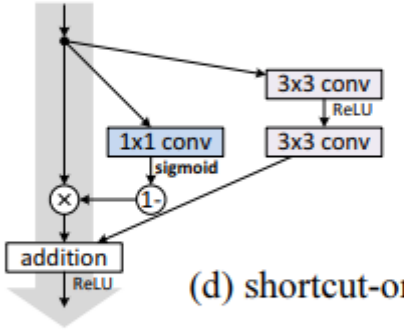
(b) constant scaling

- Not converged well
- Higher training error
- Higher test error



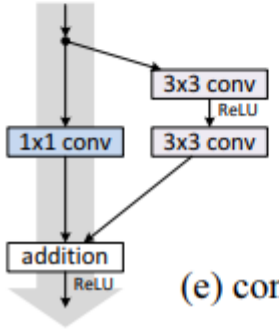
(c) exclusive gating

- Sigmoid b의 초깃값에 성능이 좌우됨.
- Shortcut 1-g & g ...



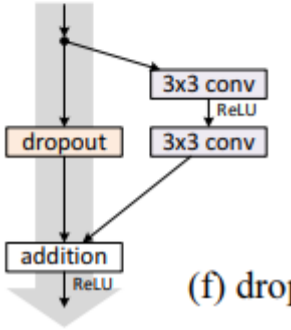
(d) shortcut-only gating

- Similar or worse test error



(e) conv shortcut

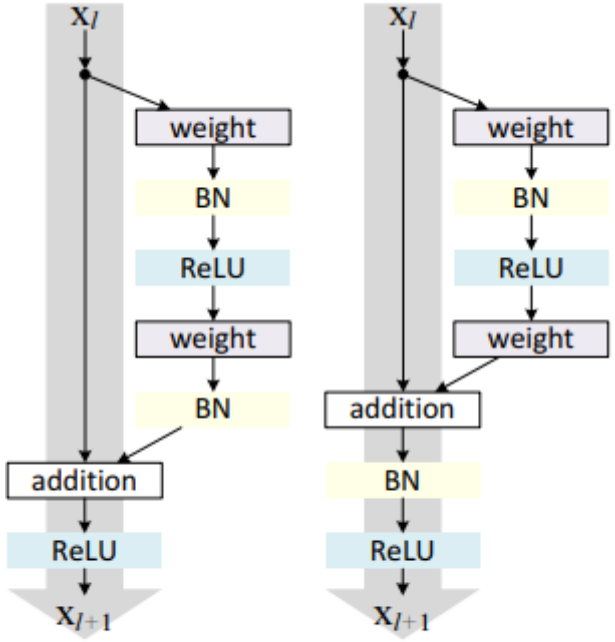
- residual unit이 많아지면 성능 저조.



(f) dropout shortcut

- scaling과 유사하게 잘 수렴하지 않음.

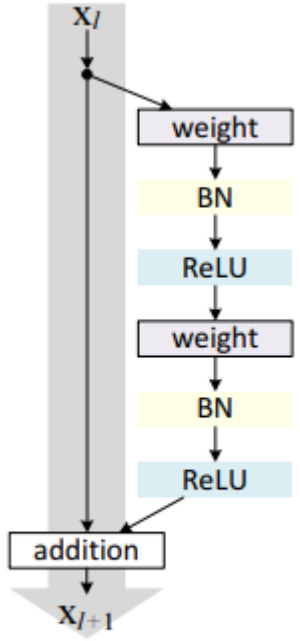
4.1 Experiments on Activation



(a) original

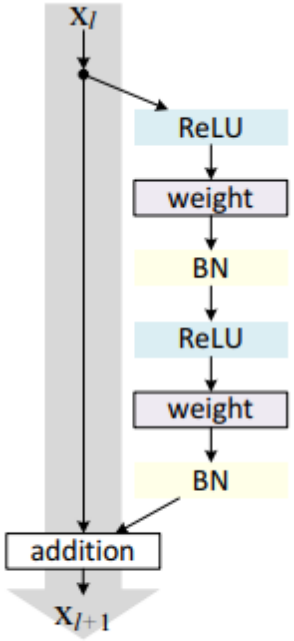
(b) BN after addition

- BN이 shortcut으로 온 값을 바꿔버린다.
- 학습 초반에 training error의 감소 속도가 느림.



(c) ReLU before addition

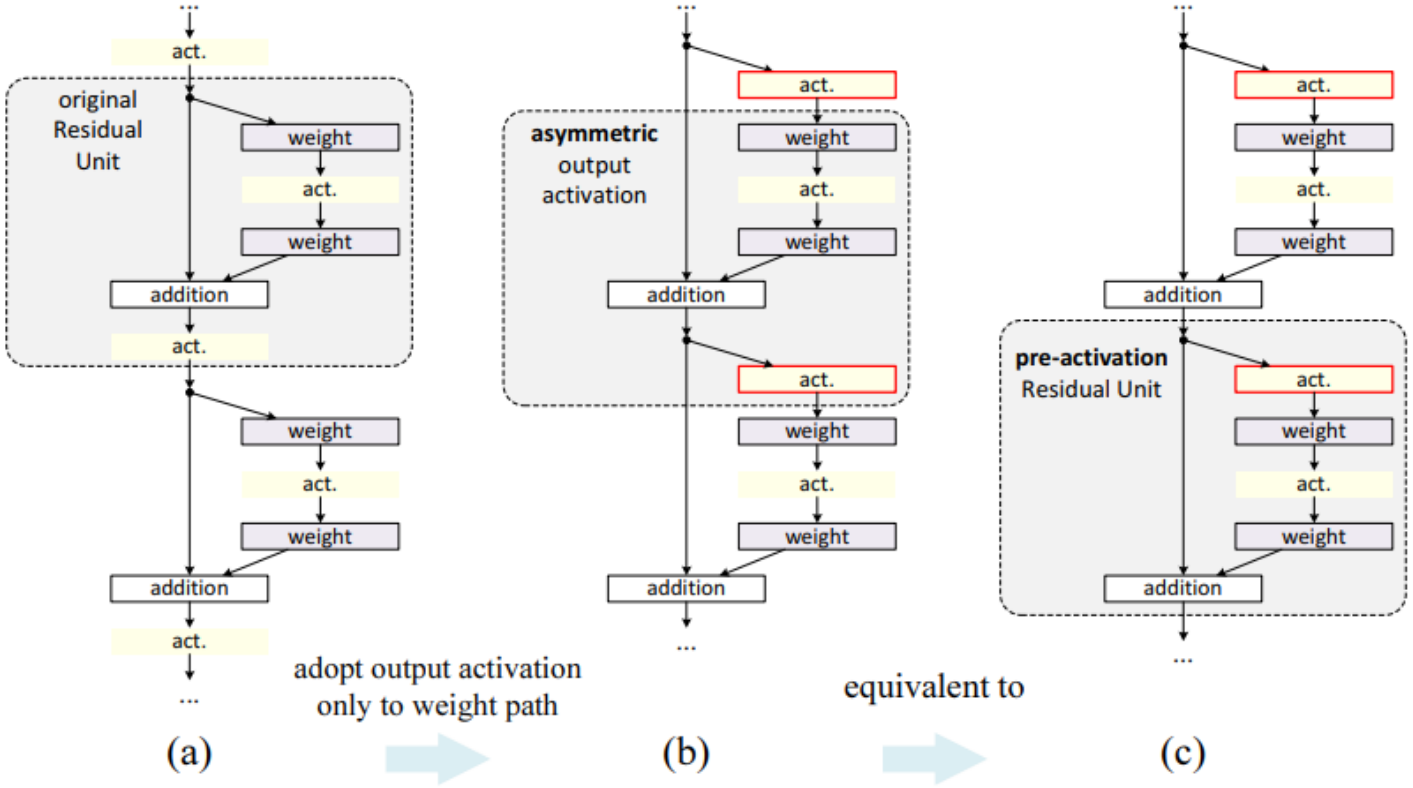
- 일반적으로 residual func. 값이 ReLU 때문에 0 이상의 값만 나오는 문제가 있었고, 그 결과 순전파 시에 값이 계속 증가함.
- representational ability에 영향을 주고, baseline보다 결과가 나쁨.



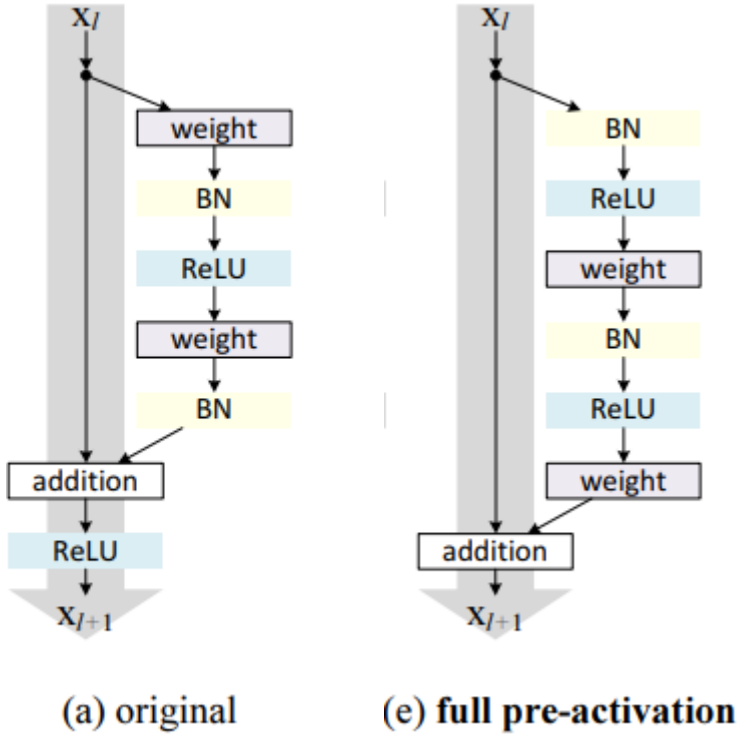
(d) ReLU-only pre-activation

- Similar to the baseline
- Not enjoy the benefits of BN

4.1 Experiments on Activation



- 처음엔 Relu만 썼을때는 baseline과 비슷하거나 조금 낮아졌지만, BN의 효과를 덜봤다는 생각에 Relu앞에 BN을 써줬더니 효과가 좋았다.



4.2 Analysis

(1) Ease of optimization

- Identity mapping이면 층이 깊어져도 training loss가 빨리 줄어들고 제일 작아진다.

- Depth가 낮을 때는 Relu도 괜찮다.
- 관찰 결과, 학습 초기단계에서 weight이 일정 status가 되고, Relu 이전 addition 값 >= 0 인 경우가 많기 때문에 truncate할 것도 없다. (residual function의 값이 정말 큰 음수가 아닌 이상, 이전 단계의 Relu로 인해 input값이 non-negative가 된다.)
- 하지만 depth가 깊어지면 truncate을 자주 하게 되더라.

(2) Reducing overfitting (Regularization from BN)

- pre-activation은 higher training loss & lower test error. 이건 아마도 BN의 효과.

- 기존 residual unit에도 BN이 있었지만, 곧장 shortcut에 더해지기 때문에, 더해진 signal이 normalized 되지 않음.