ACADENTIAL

Section 13. 표준화 (Normalization)

ACADENTIAL

목차

ACADENTIAL

Objective
학습 목표

- Normalization Layer의 정의
- Normalization Layer의 역할과 효과
- Normalization Layer의 종류
  - Batch Normalization
  - Layer Normalization
  - Instance Normalization
  - Group Normalization

ACADENTIAL

13-1. Normalization이란?

ACADENTIAL

Normalization

- “Normalization Layer”은 Neural Network에 “거의 필수불가결”한 요소
- 거의 모든 Deep Neural Network가 Normalization Layer을 포함한다고 봐도 무방!

  (일부 주관적일 수 있지만 Batch Normalization이 딥러닝에서 주요한 breakthrough들 중 하나로 생각된다!)

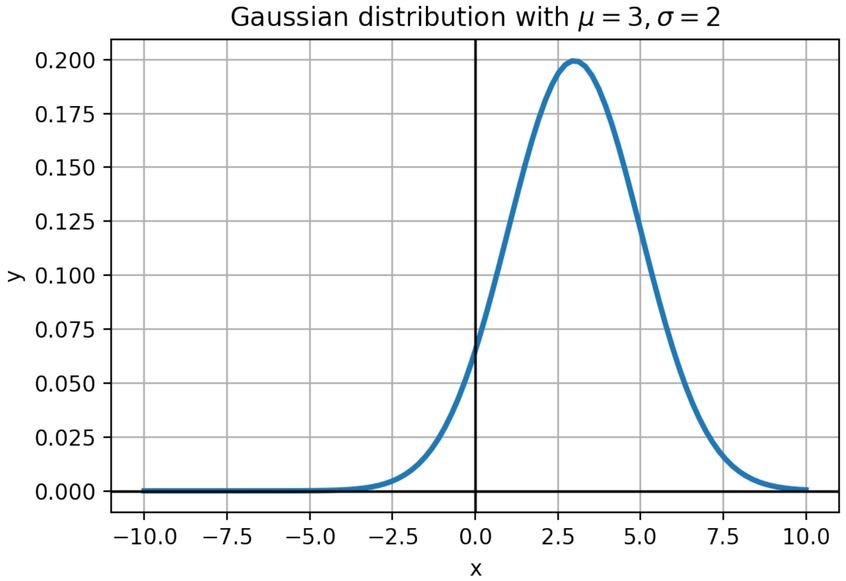
그래서 Normalization이 도대체 뭘까?

ACADENTIAL

Normalization (정규화)

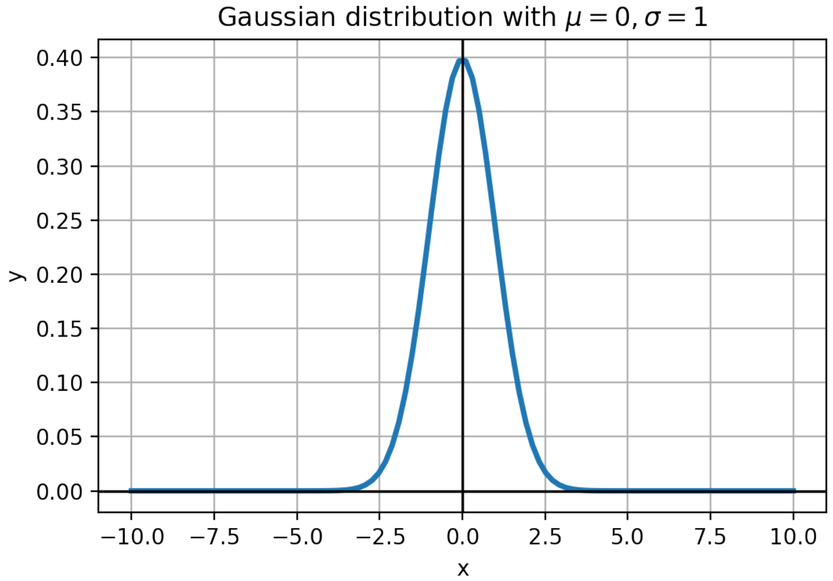
Normalization이란?

- Neural Network layer의 출력값의 분포를 정규화하는 것.
- “정규화”: 확률분포 $x \sim P(\mu, \sigma)$을 $z \sim P(0,1)$로 변환하는 것. (μ = 평균, σ = 표준편차)



$P(\mu = 5, \sigma = 2)$

$$z = \frac{x - \mu}{\sigma}$$



$P(\mu = 0, \sigma = 1)$

ACADENTIAL

Normalization

딥러닝에서 Normalization의 종류

1. Batch Normalization

   Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (Ioffe et al 2015)

2. Layer Normalization

   Layer Normalization (Lei Ba et al 2016)

3. Instance Normalization

   Instance Normalization: The Missing Ingredient for Fast Stylization (Ulyanov et al 2016)

4. Group Normalization

   Group Normalization (Wu et al 2018)

ACADENTIAL

Normalization

각 Normalization의 주된 사용처

1. Batch Normalization
    - CNN
2. Layer Normalization
    - Transformer
3. Instance Normalization
    - Style Transfer
4. Group Normalization
    - CNN

ACADENTIAL

13-2. BatchNorm의 Motivation

(처음에 제안되었을 때 motivation 하지만 알고보니 그게 아니었던 효과)

ACADENTIAL

Batch Normalization

Motivation

- Batch Normalization이 제안되었을때의 motivation:
  - 뉴럴넷의 **“Internal Covariate Shift”을 해결**하기 위해서 제안됨.
  - **실제로 실험적**으로 Batch Normalization을 사용한 것이 **아주 크게 도움이 됨!**

ACADENTIAL

Batch Normalization

Motivation

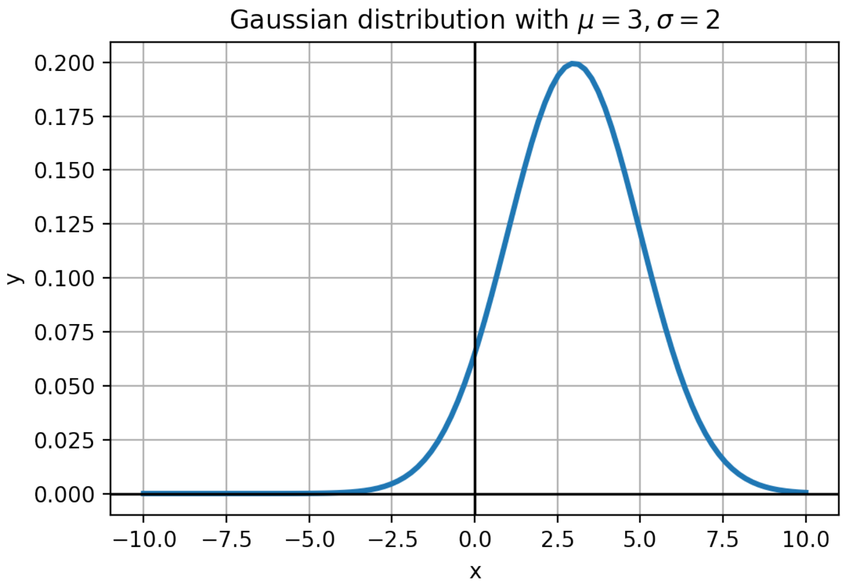
- Batch Normalization이 제안되었을때의 motivation:
  - 뉴럴넷의 **“Internal Covariate Shift”을 해결**하기 위해서 제안됨.
  - **실제로 실험적**으로 Batch Normalization을 사용한 것이 **아주 크게 도움이 됨!**
- **하지만** Batch Normalization가 **실제로 작동하는 원리**는
- Internal Covariate Shift을 해결해서가 아니라 **Loss surface을 더 smooth하게 만들어 주기 때문에 효과가 있는 것이다**
  - “How does Batch Normalization Help Optimization” (Santurkar et al, NeurIPS 2018)
- 먼저 Internal Covariate Shift가 무엇인지 살펴보자!

ACADENTIAL

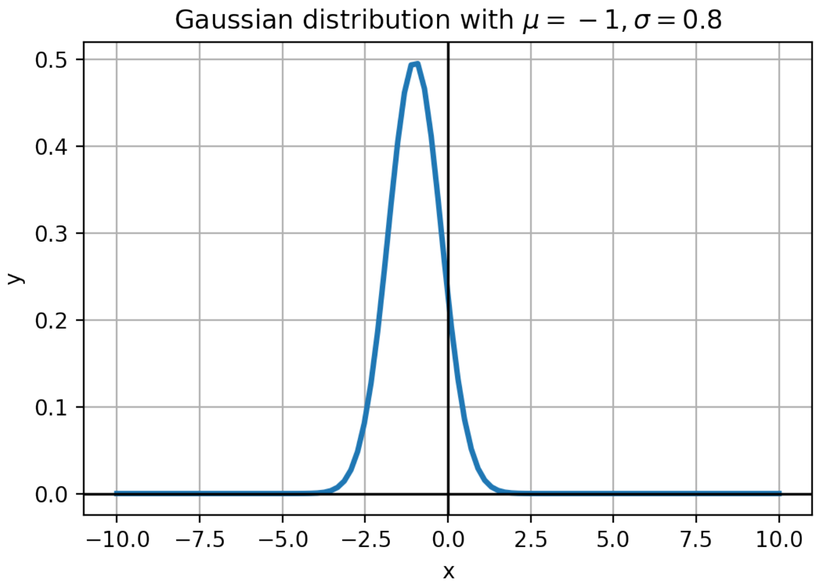
Batch Normalization

Internal Covariate Shift이란?

- Covariate shift: 입력값의 분포가 바뀌는 현상.



분포가 같지 않다!
Covariate Shift!



ACADENTIAL

Batch Normalization

Internal Covariate Shift이란?

Internal Covariate shift:

1. Neural Network에서는 각 layer들이 앞서는 layer의 출력값을 입력받는다.

ACADENTIAL

Batch Normalization

Internal Covariate Shift이란?

Internal Covariate shift:

1. Neural Network에서는 각 layer들이 앞서는 layer의 출력값을 입력받는다.
2. layer의 출력값은 해당 layer의 parameter (weight, bias)의 영향을 받는다.

ACADENTIAL

Batch Normalization

Internal Covariate Shift이란?

Internal Covariate shift:

1. Neural Network에서는 각 layer들이 앞서는 layer의 출력값을 입력받는다.
2. layer의 출력값은 해당 layer의 parameter (weight, bias)의 영향을 받는다.
3. 따라서 앞서는 layer의 parameter가 바뀌면 뒤따르는 layer의 입력은 covariate shift을 겪게된다.

ACADENTIAL

Batch Normalization

Internal Covariate Shift이란?

Internal Covariate shift:

1. Neural Network에서는 각 layer들이 앞서는 layer의 출력값을 입력받는다.
2. layer의 출력값은 해당 layer의 parameter (weight, bias)의 영향을 받는다.
3. 따라서 앞서는 layer의 parameter가 바뀌면 뒤따르는 layer의 입력은 covariate shift을 겪게된다.
4. Neural Network 안에서 발생하는 covariate shift을 internal covariate shift라고 한다.

ACADENTIAL

Batch Normalization

Internal Covariate Shift이란?

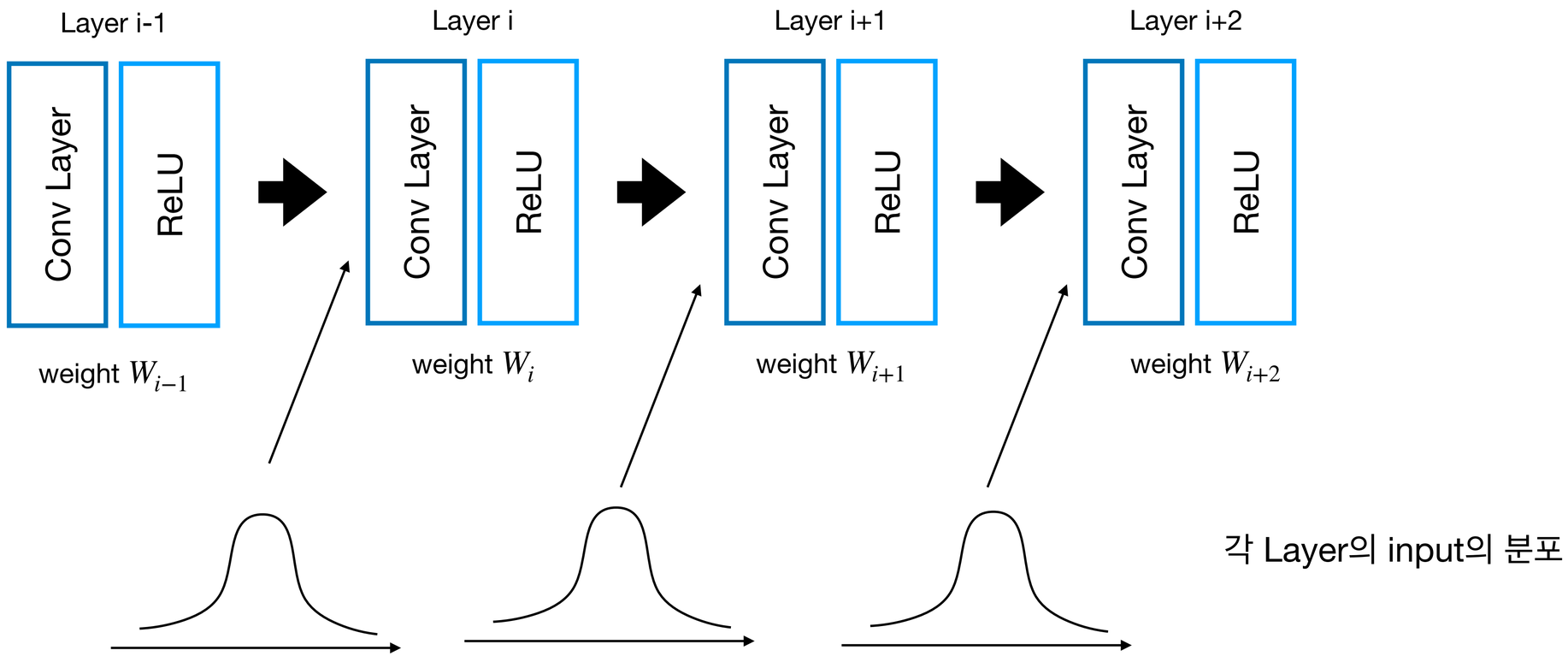
Internal Covariate shift:

1. Neural Network에서는 각 layer들이 앞서는 layer의 출력값을 입력받는다.
2. layer의 출력값은 해당 layer의 parameter (weight, bias)의 영향을 받는다.
3. 따라서 앞서는 layer의 parameter가 바뀌면 뒤따르는 layer의 입력은 covariate shift을 겪게된다.
4. Neural Network 안에서 발생하는 covariate shift을 internal covariate shift라고 한다.
5. 해당 internal covariate shift은 layer을 통과할수록 증폭된다.

ACADENTIAL

Batch Normalization

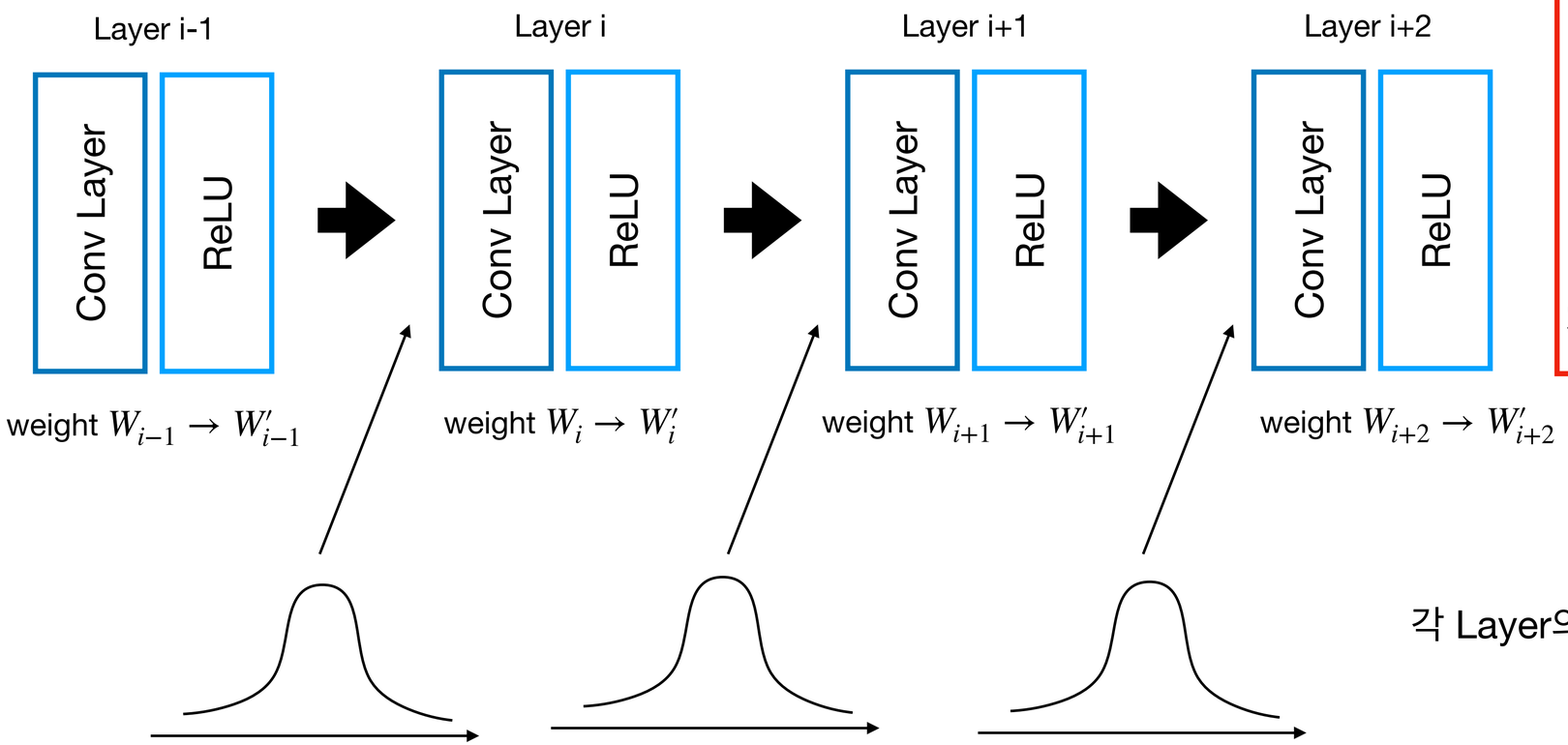
Internal Covariate Shift illustration



각 Layer의 input의 분포

Batch Normalization

Internal Covariate Shift illustration



각 Layer의 input의 분포

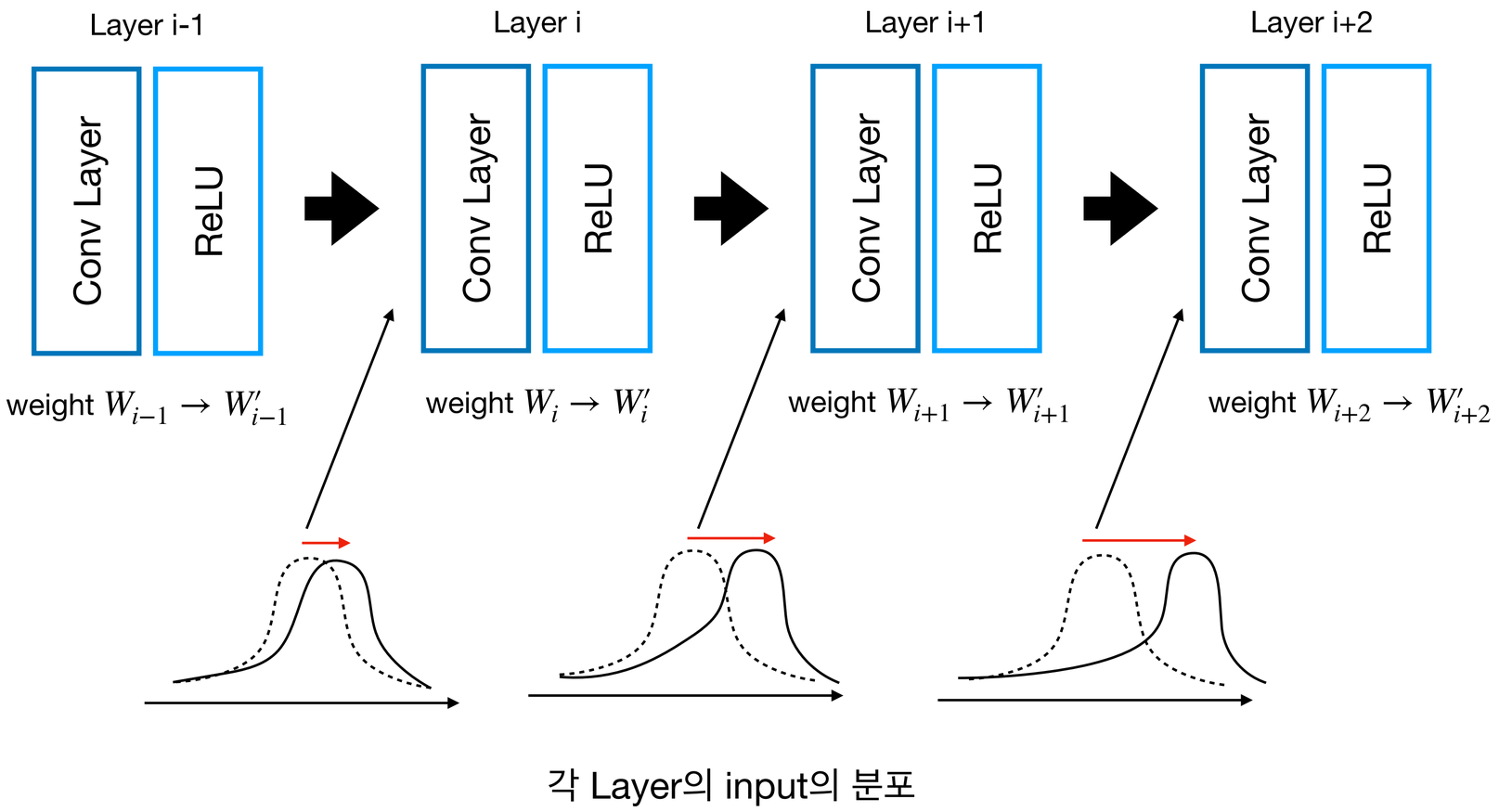
Gradient Descent에 의해서
weight들이 update되었다고
가정해보자:

$$W_i \rightarrow W'_i$$

각 Layer의 출력값도 바뀌게 된다!

Batch Normalization

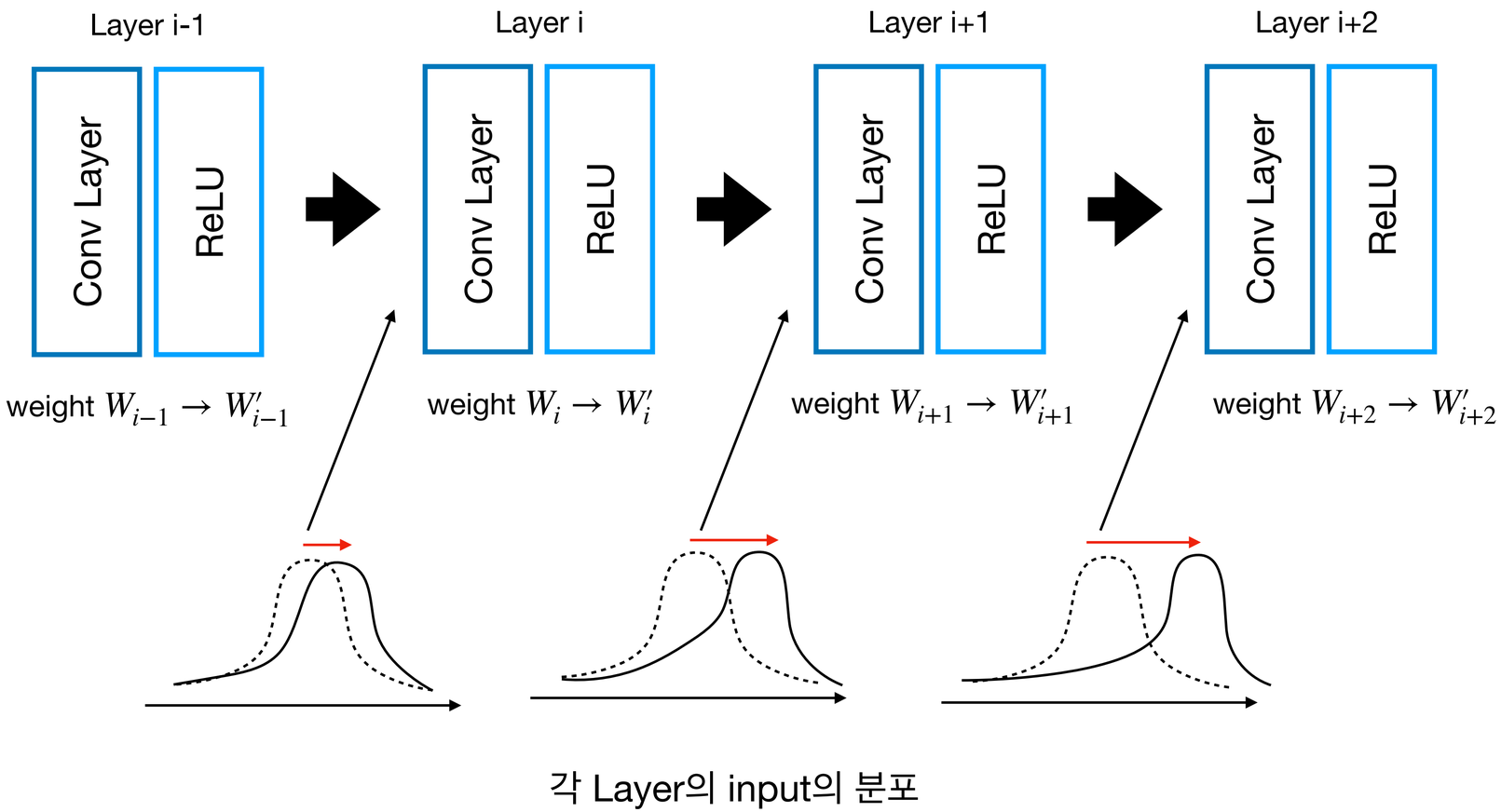
Internal Covariate Shift illustration



이 때 뒤에 위치한 **Layer일수록 input distribution의 covariate shift가 더 크다!**

Batch Normalization

Internal Covariate Shift illustration



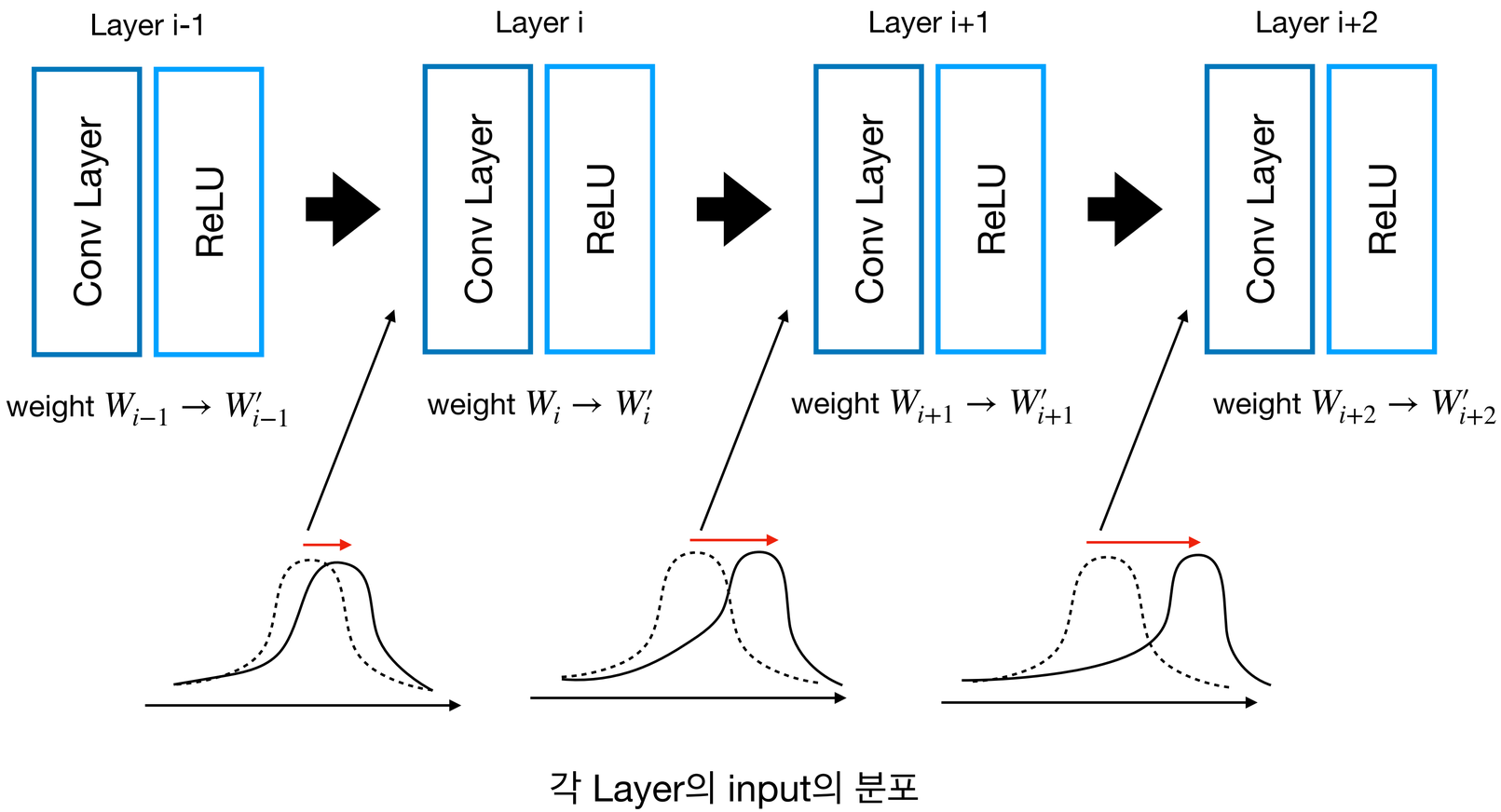
각 Layer의 input의 분포

이 때 뒤에 위치한 **Layer일수록 input distribution의 covariate shift가 더 크다!**

즉, **Internal Covariate Shift은 Neural Network의 layer을 통과할수록 증폭 (amplify)하게 된다!**

Batch Normalization

Internal Covariate Shift illustration



각 Layer의 input의 분포

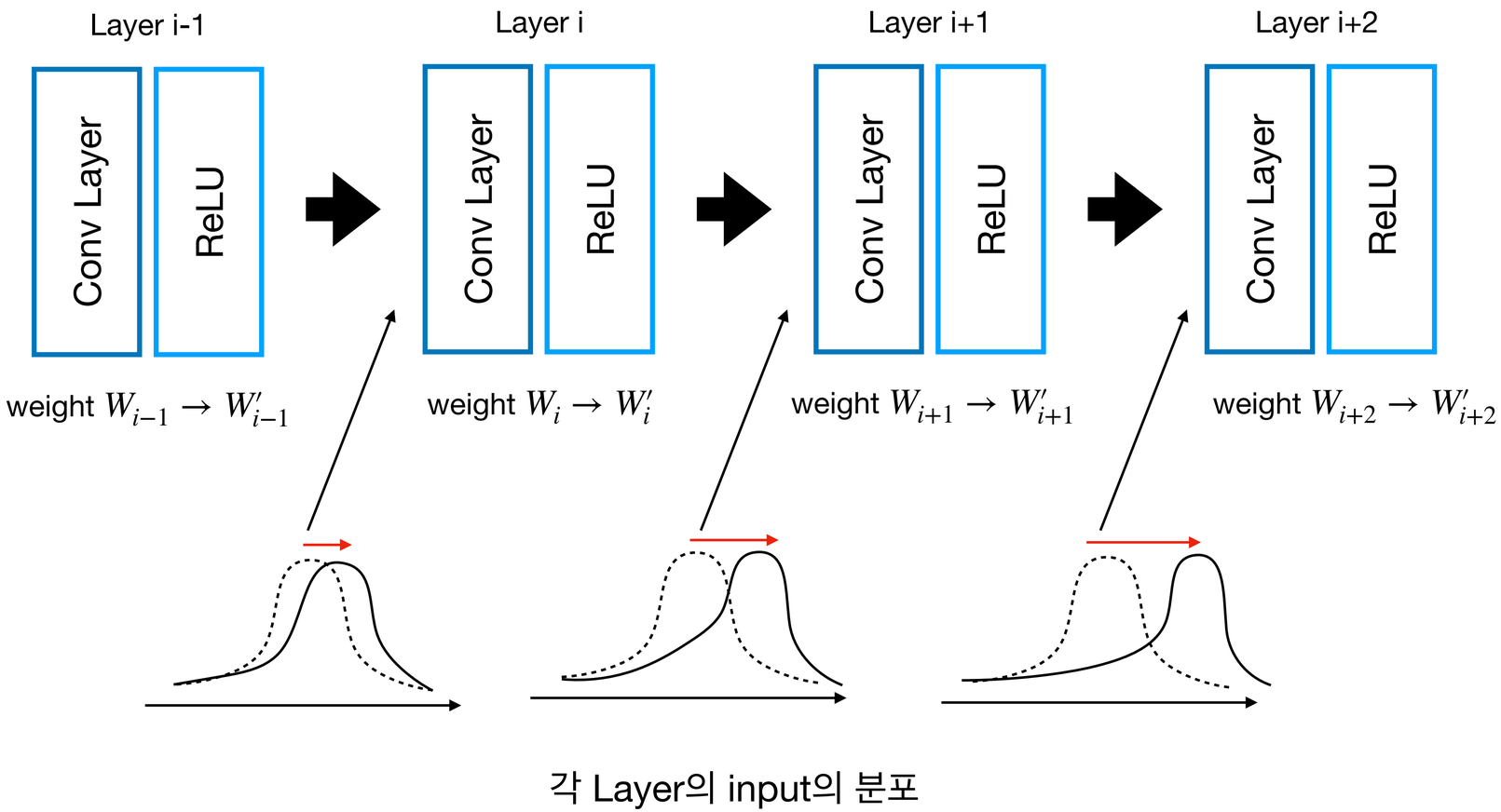
이 때 뒤에 위치한 **Layer일수록 input distribution의 covariate shift가 더 크다!**

즉, **Internal Covariate Shift은 Neural Network의 layer을 통과할수록 증폭 (amplify)하게 된다!**

Batch Normalization은 해당 문제를 해결하고자 한다! (Original Motivation)

Batch Normalization

Internal Covariate Shift illustration



각 Layer의 input의 분포

Batch Normalization은 해당 문제를 해결하고자 한다!
(Original Motivation)

하지만 사실 normalization의 실제 효과는 이것이 아니다!

ACADENTIAL

13-3. BatchNorm

ACADENTIAL

Batch Normalization

Definition

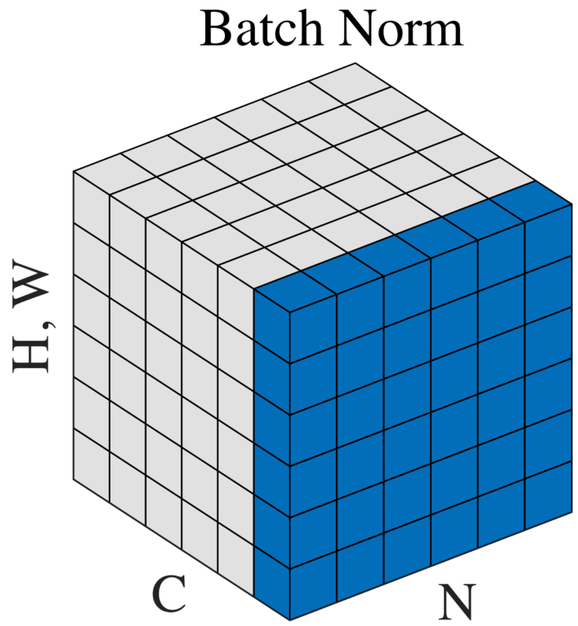
Formulation:

$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$



출처: Group Normalization (He et al, ECCV 2018)

ACADENTIAL

Batch Normalization
해석

Formulation:

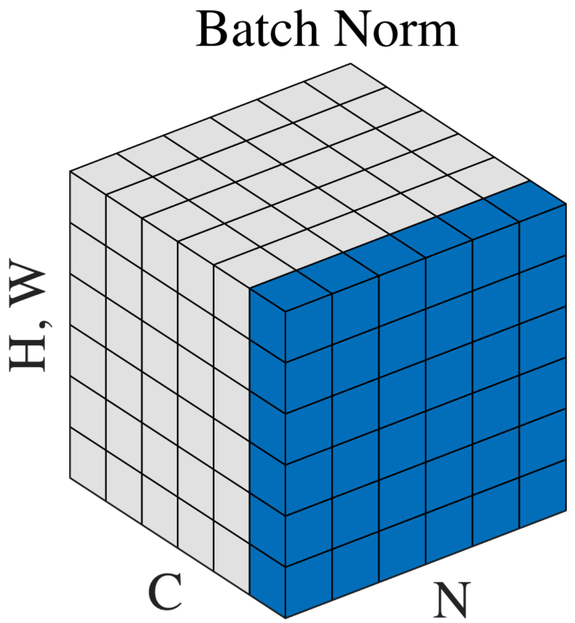
$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

- 각 channel $c \in C$ 마다 개별적인 평균과 표준편차 (μ_C, σ_C) 을 따른다고 본다.
- (N, H, W)을 묶어서 각 c에 대한 (μ_C, σ_C)을 계산한다.



출처: Group Normalization (He et al, ECCV 2018)

Batch Normalization

해석

각 Channel마다 개별적인 **평균** μ_c**, 표준편차** σ_c의 분포를 따른다는 것을 가정함.

Formulation:

$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

ACADENTIAL

Batch Normalization

해석

Formulation:

$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

Channel $c \in C$ 에 상관없이 Standardized $(\mu_c = 0, \sigma_c = 1)$된 분포를 가지도록 normalize해 주는 역할.

Batch Normalization
해석

Formulation:

$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

CLASS torch.nn.BatchNorm2d(*num_features*, *eps=1e-05*, *momentum=0.1*, *affine=True*, *track_running_stats=True*, *device=None*, *dtype=None*) [SOURCE]

affine=True로 사용할 경우, trainable한 γ_c, β_c parameter을 사용한다.

channel-wise fully connected layer을 적용하는 셈이다.

즉, c번째 channel을 평균 β_c, 표준편차 γ_c의 분포로 mapping해주는 셈이다.

Batch Normalization
해석

Formulation:

$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

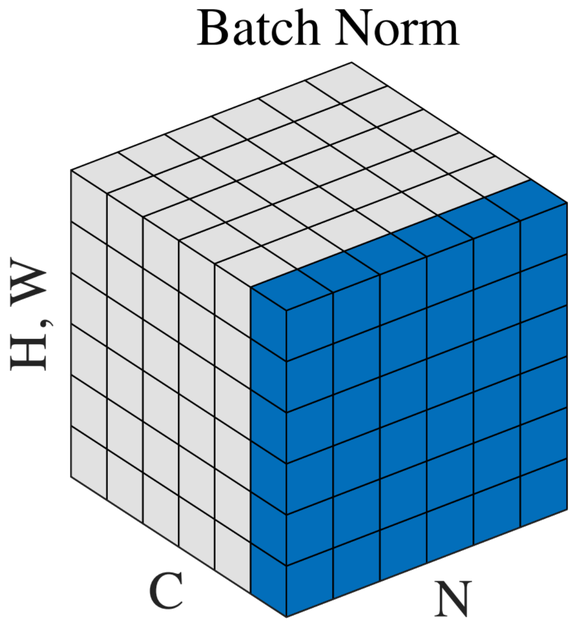
$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

μ_c와 σ_c은 학습하는 과정에서는 **mini-batch**의 **mean과 standard deviation (s.t.d)**을 사용!

추론하는 과정에서는 학습하는 과정에서 계산한 학습 데이터셋의 **running mean, running s.t.d.**을 사용!



출처: Group Normalization (He et al, ECCV 2018)

ACADENTIAL

Batch Normalization

해석

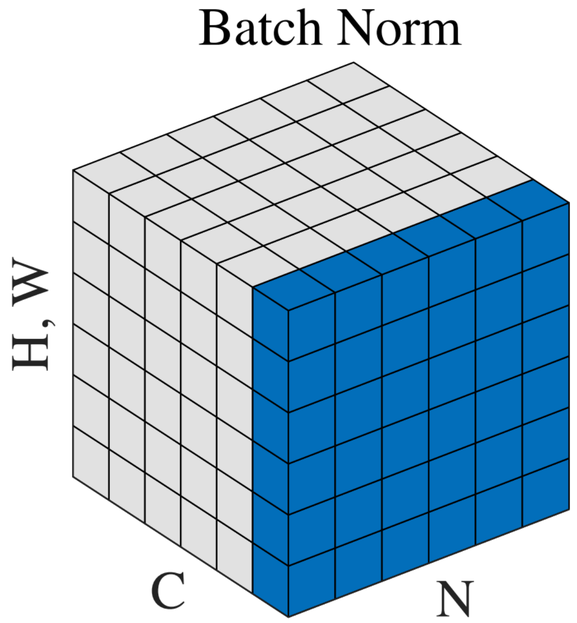
Formulation:

$$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_c)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$



γ_c와 β_c은 trainable한 parameter이다!
즉, batch-norm도 학습이 되는 Layer이다!

출처: Group Normalization (He et al, ECCV 2018)

ACADENTIAL

Batch Normalization

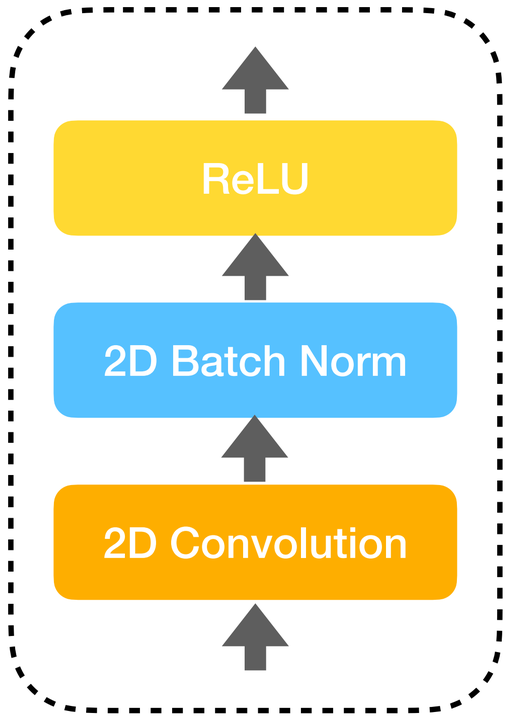
사용 예시

- 코드 상에서 model.train() 혹은 model.eval()을 하는데
- **model.train()**으로 설정할시:
  - Batch Norm에서 γ_c와 β_c을 학습함. (즉, requires_grad = True)
  - " " " μ_c와 σ_c은 각 mini-batch의 분포를 사용함.
- **model.eval()**로 설정할시:
  - Batch Norm에서 γ_c와 β_c을 학습하지 않음. (즉, requires_grad = False)
  - " " " μ_c와 σ_c은 training에서 계산한 running average된 분포를 사용함.

ACADENTIAL

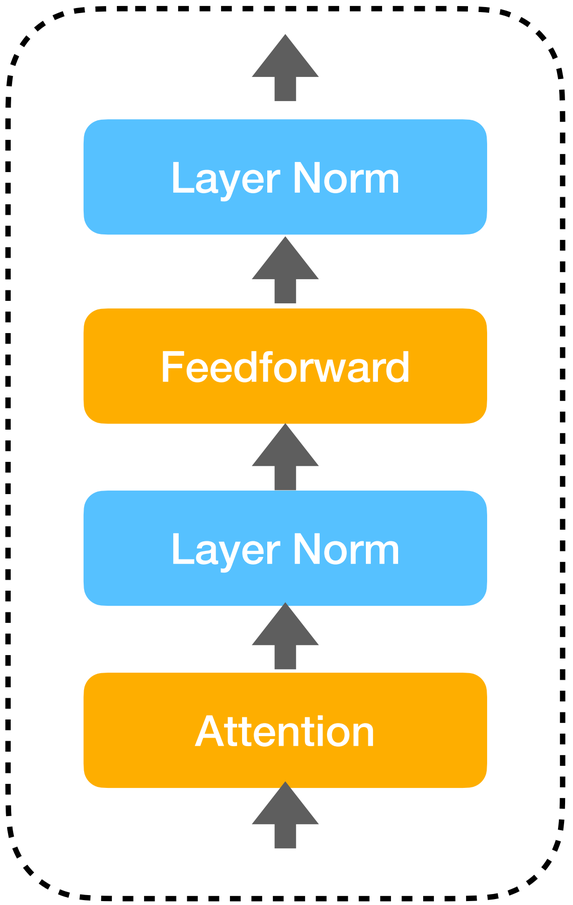
Batch Normalization

사용 예시



일반적인 CNN Block

(일반적으로) Normalization layer은 NN layer와 activation function의 사이에 위치함!



Transformer Block

ACADENTIAL

13-4. BatchNorm의 실제 효과

ACADENTIAL

Batch Normalization

왜 효과가 있는가?

How Does Batch Normalization Help Optimization?

Shibani Santurkar* MIT shibani@mit.edu

Dimitris Tsipras* MIT tsipras@mit.edu

Andrew Ilyas* MIT ailyas@mit.edu

Aleksander Mądry MIT madry@mit.edu

Abstract

Batch Normalization (BatchNorm) is a widely adopted technique that enables faster and more stable training of deep neural networks (DNNs). Despite its pervasiveness, the exact reasons for BatchNorm's effectiveness are still poorly understood. The popular belief is that this effectiveness stems from controlling the change of the layers' input distributions during training to reduce the so-called "internal covariate shift". In this work, we demonstrate that such distributional stability of layer inputs has little to do with the success of BatchNorm. Instead, we uncover a more fundamental impact of BatchNorm on the training process: it makes the optimization landscape significantly smoother. This smoothness induces a more predictive and stable behavior of the gradients, allowing for faster training.

ACADENTIAL

Batch Normalization

왜 효과가 있는가?

How Does Batch Normalization Help Optimization?

Shibani Santurkar* MIT shibani@mit.edu
Dimitris Tsipras* MIT tsipras@mit.edu
Andrew Ilyas* MIT ailyas@mit.edu
Aleksander Mądry MIT madry@mit.edu

Abstract

Batch Normalization (BatchNorm) is a widely adopted technique that enables faster and more stable training of deep neural networks (DNNs). Despite its pervasiveness, the exact reasons for BatchNorm's effectiveness are still poorly understood. The popular belief is that this effectiveness stems from controlling the change of the layers' input distributions during training to reduce the so-called "internal covariate shift". In this work, we demonstrate that such distributional stability of layer inputs has little to do with the success of BatchNorm. Instead, we uncover a more fundamental impact of BatchNorm on the training process: it makes the optimization landscape significantly smoother. This smoothness induces a more predictive and stable behavior of the gradients, allowing for faster training.

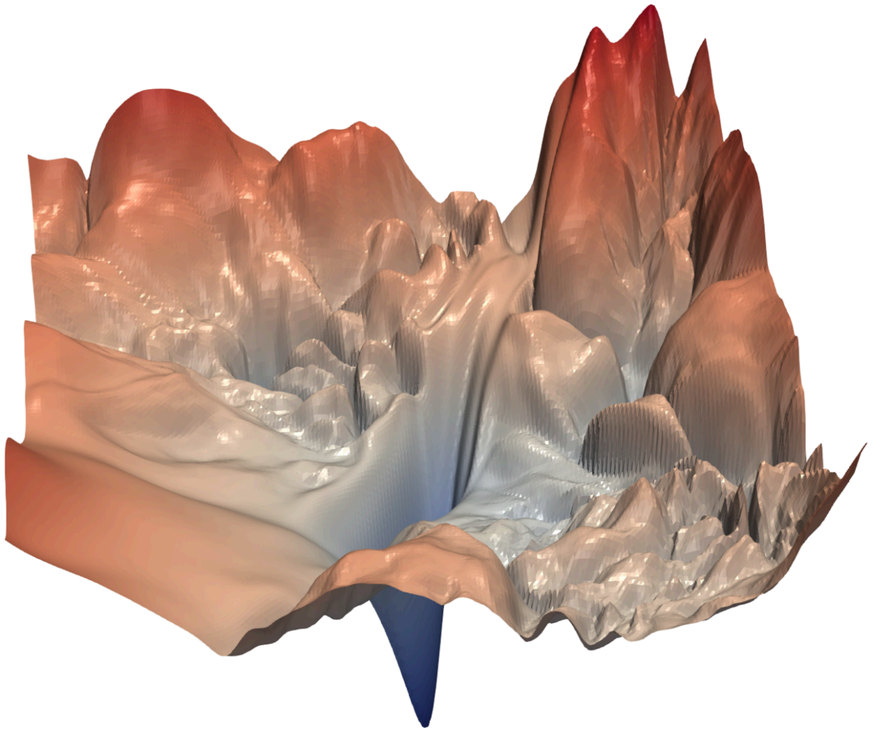
Batch Normalization이 도움이 되는 실제 원인은 바로 "Optimization Landscape"을 매끄럽게 만들어 주기 때문에!

ACADENTIAL

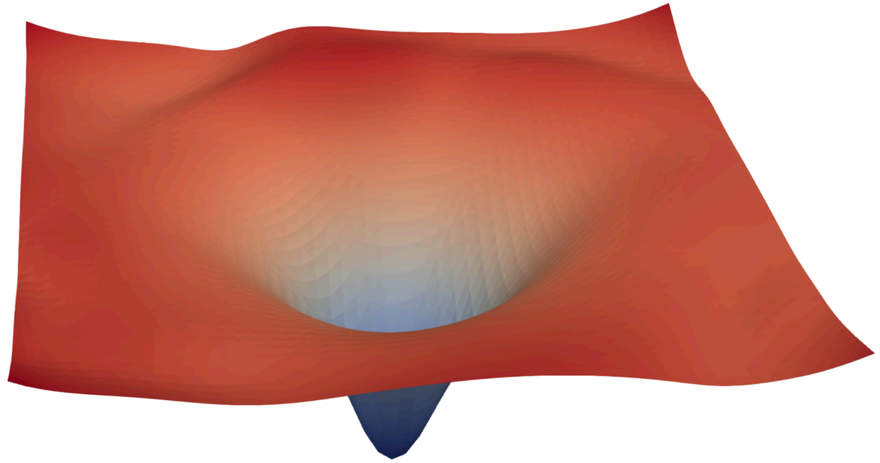
Batch Normalization

왜 효과가 있는가?

(대략적인 illustration)



Without Batch Normalization



With Batch Normalization

Visualizing the Loss Landscape of Neural Nets (NeurIPS 2018): Li et al.

ACADENTIAL

Batch Normalization

왜 효과가 있는가?

- Batch Normalization은 Loss landscape (Optimization landscape)을 매끄럽게 만들어준다.
- 따라서 학습이 더 안정적이고 더욱 빠르게 수렴하게 된다!

ACADENTIAL

13-5. InstanceNorm

ACADENTIAL

Instance Normalization

Definition

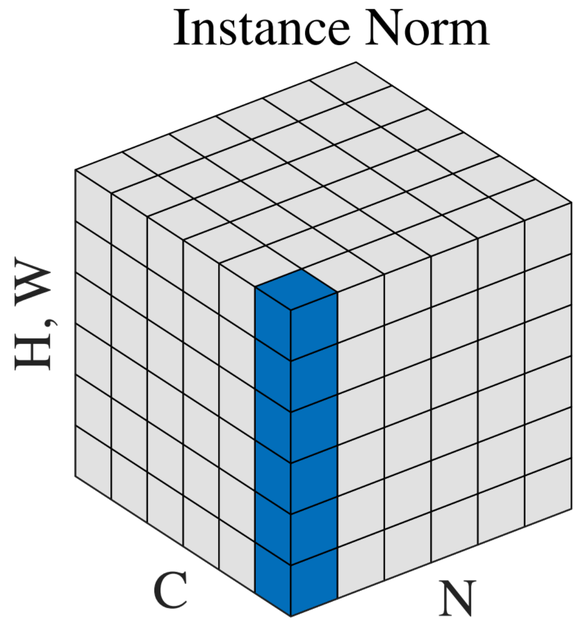
Formulation:

$$\mu_{nc} = \frac{1}{HW} \sum_{h}^{H} \sum_{w}^{W} x_{nhwc}$$

$$\sigma_{nc}^2 = \frac{1}{HW} \sum_{h}^{H} \sum_{w}^{W} (x_{hwc} - \mu_{nc})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$



출처: Group Normalization (He et al, ECCV 2018)

Instance Normalization

Definition

Formulation:

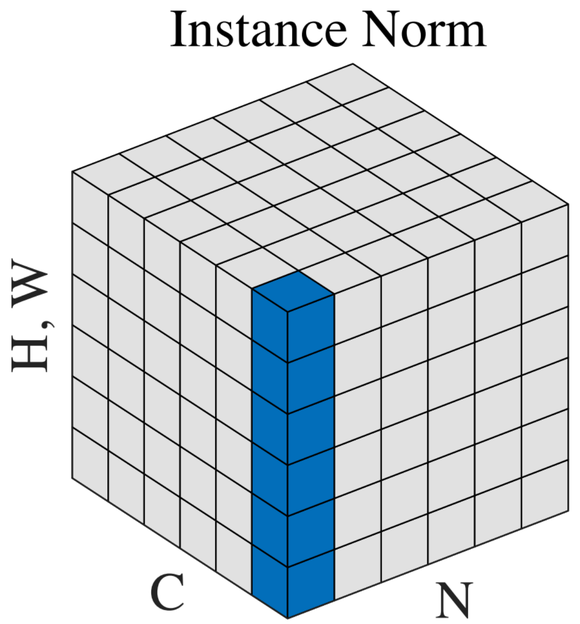
$$\mu_{nc} = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_{nc}^2 = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} (x_{hwc} - \mu_{nc})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

- 각 데이터와 channel ($n \in N, c \in C$) 마다 개별적인 평균 μ_{nc}과 표준편차 σ_{nc}을 따른다고 보는 셈.
- (H, W)만 묶어서 각 (n, c)에 대한 (μ_{nc}, σ_{nc})을 계산한다.



ACADENTIAL

Instance Normalization

Definition

Formulation:

$$\mu_{nc} = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_{nc}^2 = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} (x_{hwc} - \mu_{nc})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

데이터 $n \in N$, $c \in C$에 상관없이 standardized
$(\mu_{nc} = 0, \sigma_{nc} = 1)$된 분포를 따르도록 normalize하는 것.

ACADENTIAL

Instance Normalization

Definition

Formulation:

$$\mu_{nc} = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_{nc}^2 = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} (x_{hwc} - \mu_{nc})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

affine=True로 사용할 경우, trainable한 γ_c, β_c parameter을 사용한다.

channel-wise fully connected layer을 적용하는 셈이다.

즉, c번째 channel을 평균 β_c, 표준편차 γ_c의 분포로 mapping해주는 셈이다.

ACADENTIAL

Instance Normalization

사용 예시

- “StyleNet: Generating Attractive Visual Captions with Styles”에서 처음 제안됨.
- **각 sample의 분포**를 사용하여 **normalization을 각 sample마다 따로 따로** 적용하는 것.
  - Batch norm은 각 sample아니라 **mini-batch의 분포를 계산**하여, **mini-batch의 샘플들을 일괄적으로** normalize한다.
- 스타일 변환 (Style Transfer)에서 주로 활용된다.

ACADENTIAL

13-6. LayerNorm

ACADENTIAL

Layer Normalization

Definition

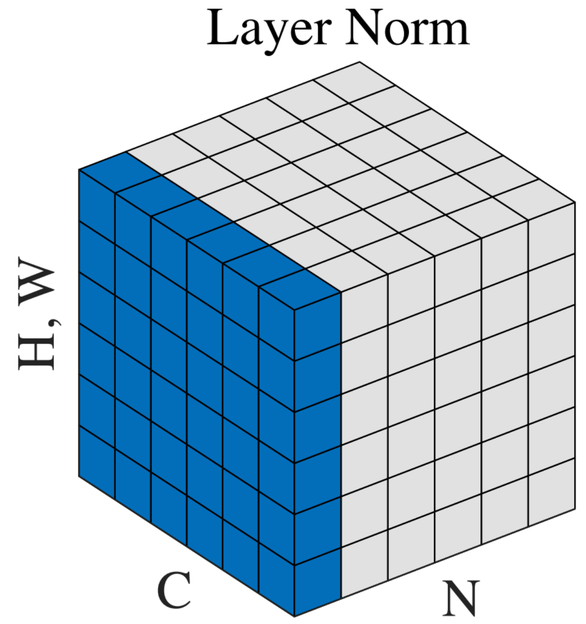
Formulation:

$$\mu_n = \frac{1}{CHW} \sum_{c}^{C} \sum_{h}^{H} \sum_{w}^{W} x_{nhwc}$$

$$\sigma_n^2 = \frac{1}{CHW} \sum_{c}^{C} \sum_{h}^{H} \sum_{w}^{W} (x_{nhwc} - \mu_n)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}$$

$$y_{nhwc} = \gamma \hat{x}_{nhwc} + \beta$$



출처: Group Normalization (He et al, ECCV 2018)

ACADENTIAL

Layer Normalization

Definition

Formulation:

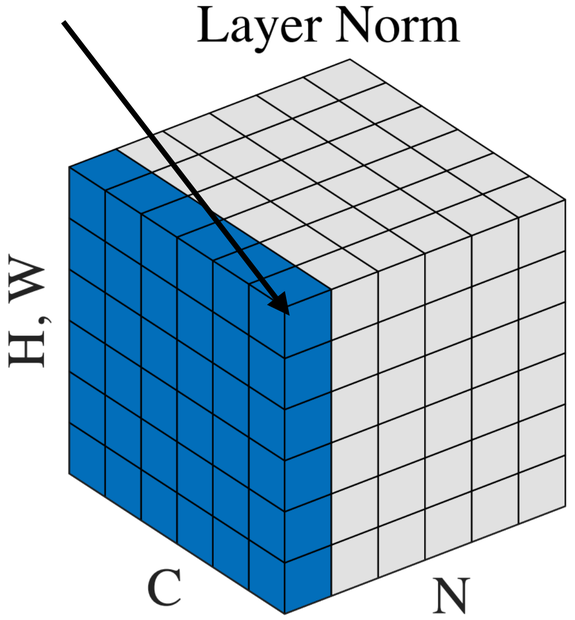
$$\mu_n = \frac{1}{CHW}\sum_{c}^{C}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_n^2 = \frac{1}{CHW}\sum_{c}^{C}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_n)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}$$

$$y_{nhwc} = \gamma \hat{x}_{nhwc} + \beta$$

- 각 데이터 $n \in N$ 마다 개별적인 평균 μ_n과 표준편차 σ_n을 따른다고 보는 셈.
- (C, H, W)을 묶어서 각 n에 대한 (μ_n, σ_n)을 계산한다.



출처: Group Normalization (He et al, ECCV 2018)

ACADENTIAL

Layer Normalization

Definition

Formulation:

$$\mu_n = \frac{1}{CHW}\sum_{c}^{C}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$$

$$\sigma_n^2 = \frac{1}{CHW}\sum_{c}^{C}\sum_{h}^{H}\sum_{w}^{W} (x_{nhwc} - \mu_n)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}$$

$$y_{nhwc} = \gamma\hat{x}_{nhwc} + \beta$$

데이터 $n \in N$ 에 상관없이 standardized

$(\mu_n = 0, \sigma_n = 1)$된 분포를 따르도록 normalize하는 것.

ACADENTIAL

Layer Normalization

Definition

Formulation:

$$\mu_n = \frac{1}{CHW} \sum_{c}^{C} \sum_{h}^{H} \sum_{w}^{W} x_{nhwc}$$

$$\sigma_n^2 = \frac{1}{CHW} \sum_{c}^{C} \sum_{h}^{H} \sum_{w}^{W} (x_{nhwc} - \mu_n)^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}$$

element-wise affine transformation

$$y_{nhwc} = \gamma \hat{x}_{nhwc} + \beta$$

γ, β은 scalar 값들이다.

ACADENTIAL

Layer Normalization
사용 예시

- 동일한 층의 뉴런들 간에 normalization을 해준다.
- 다음에 대해서 robust하다:
  - 입력 데이터의 scale
  - 가중치 행렬의 scale과 shift
- 주로 RNN, Self-attention, Transformer 등등에 사용됨.
- CNN에서는 Batch Norm이 더 잘 작동된다.

ACADENTIAL

13-6. GroupNorm

ACADENTIAL

Group Normalization

Definition

Formulation:

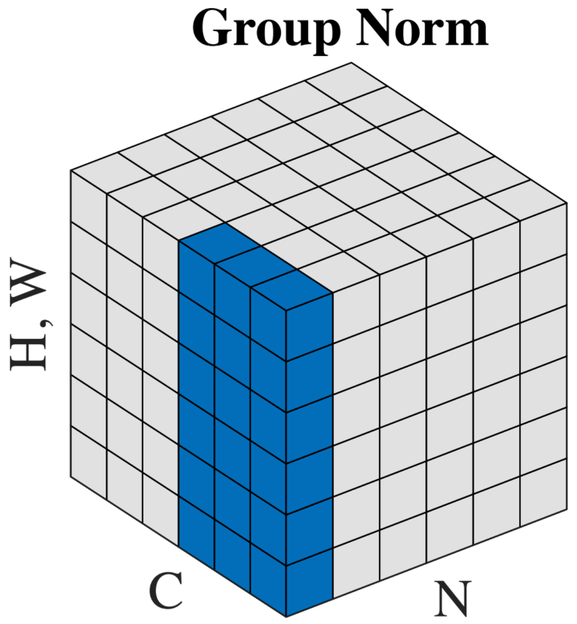
$$\mu_{ni} = \frac{1}{S_i HW} \sum_{c \in S_i} \sum_{h}^{H} \sum_{w}^{W} x_{nhwc}$$

$$\sigma_{ni}^2 = \frac{1}{S_i HW} \sum_{c \in S_i} \sum_{h}^{H} \sum_{w}^{W} (x_{nhwc} - \mu_{ni})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{ni}}{\sqrt{\sigma_{ni}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

- 각 데이터와 그룹 $(n \in N, i \in I)$에 따라 개별적인 평균 μ_n과 표준편차 σ_n을 따른다고 보는 셈.
- (S_i, H, W)을 묶어서 각 (n, i)에 대한 (μ_{ni}, σ_{ni})을 계산한다.



출처: Group Normalization (He et al, ECCV 2018)

ACADENTIAL

Group Normalization

Definition

Formulation:

$$\mu_{ni} = \frac{1}{S_i HW} \sum_{c \in S_i} \sum_{h}^{H} \sum_{w}^{W} x_{nhwc}$$

$$\sigma_{ni}^2 = \frac{1}{S_i HW} \sum_{c \in S_i} \sum_{h}^{H} \sum_{w}^{W} (x_{nhwc} - \mu_{ni})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{ni}}{\sqrt{\sigma_{ni}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

데이터와 그룹 $n \in N, i \in I$ 에 상관없이 standardized $(\mu_{ni} = 0, \sigma_{ni} = 1)$된 분포를 따르도록 normalize하는 것.

Group Normalization

Definition

Formulation:

$$\mu_{ni} = \frac{1}{S_i HW} \sum_{c \in S_i} \sum_{h}^{H} \sum_{w}^{W} x_{nhwc}$$

$$\sigma_{ni}^2 = \frac{1}{S_i HW} \sum_{c \in S_i} \sum_{h}^{H} \sum_{w}^{W} (x_{nhwc} - \mu_{ni})^2$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_{ni}}{\sqrt{\sigma_{ni}^2 + \epsilon}}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c$$

affine=True로 사용할 경우, trainable한 γ_c, β_c parameter을 사용한다.

Group Normalization
사용 예시

- Batch 사이즈가 극도로 작은 상황에서 Batch Norm 대신 사용하면 더 효과적이다.
  - Object Detection과 같은 경우.
- 왜냐하면:
  - Mini-batch가 충분히 큰 경우에 mini-batch의 분포가 전체 데이터셋의 분포를 어느정도 잘 '대표'할 수 있다는 가정 성립한다.
  - 하지만 mini-batch가 너무 작을 경우, mini-batch의 평균과 분산이 매 iteration마다 fluctuate한다.
  - Group Norm은 각 채널을 N개의 그룹으로 나누어 정규화함 (HOG와 SIFT 같은 전통적인 영상처리 검출 방법에서 착안됨)

ACADENTIAL

13-8. PyTorch로 구현해보는 Normalisation

ACADENTIAL

13-9. Section 13 요약

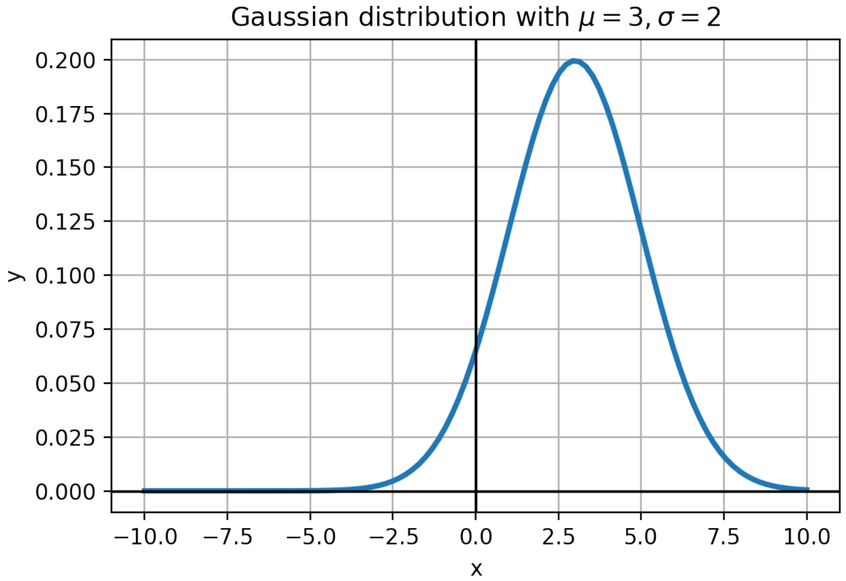
ACADENTIAL

Section Summary

Normalization - 정의

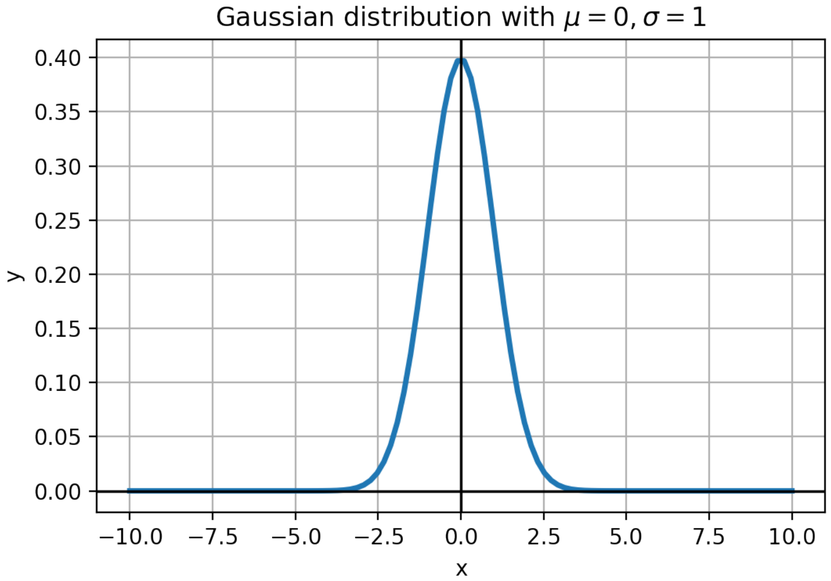
"정규화 (Normalization)":

확률분포 $x \sim P(\mu, \sigma)$을 $z \sim P(0,1)$로 변환하는 것. (μ = 평균, σ = 표준편차)



$$P(\mu = 5, \sigma = 2)$$

$$z = \frac{x - \mu}{\sigma}$$

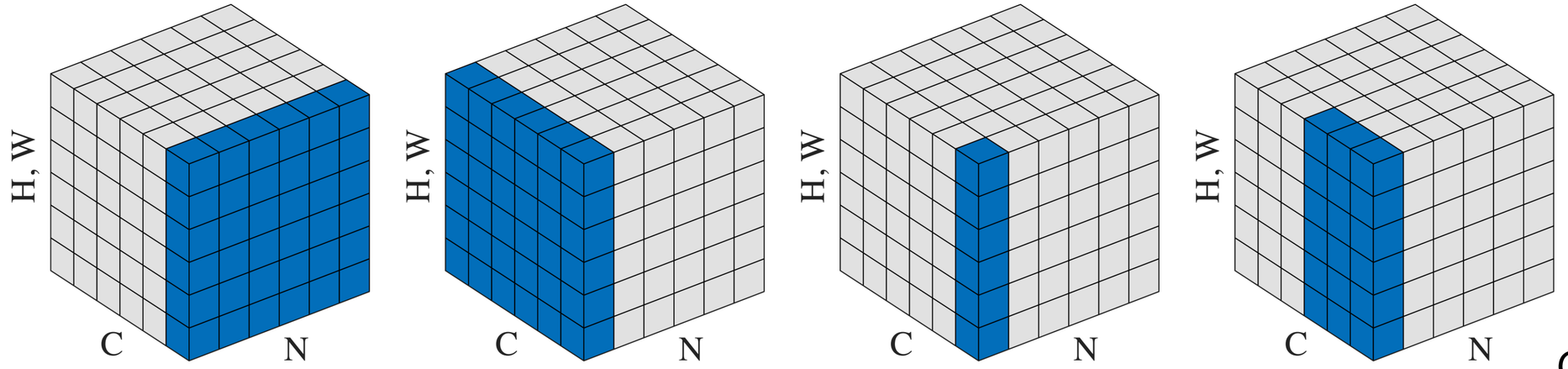


$$P(\mu = 0, \sigma = 1)$$

Overview of Normalizations

ACADENTIAL

- Batch Norm:
  - **mini-batch을 묶어서 평균**을 내고, **각 channel에 대해서는 평균을 각각 따로 계산**한다.
- Layer Norm:
  - **channel을 묶어서** 평균을 내고, 각 데이터에 대해서는 평균을 각각 따로 계산한다.
- Instance Norm:
  - **channel과 데이터**에 대해서 **평균을 각각 따로** 계산한다.
- Group Norm:
  - **같은 그룹에 속한 channel들끼리 묶어서** 평균을 내고, **각 데이터에 대해서는 평균을 따로** 계산한다.



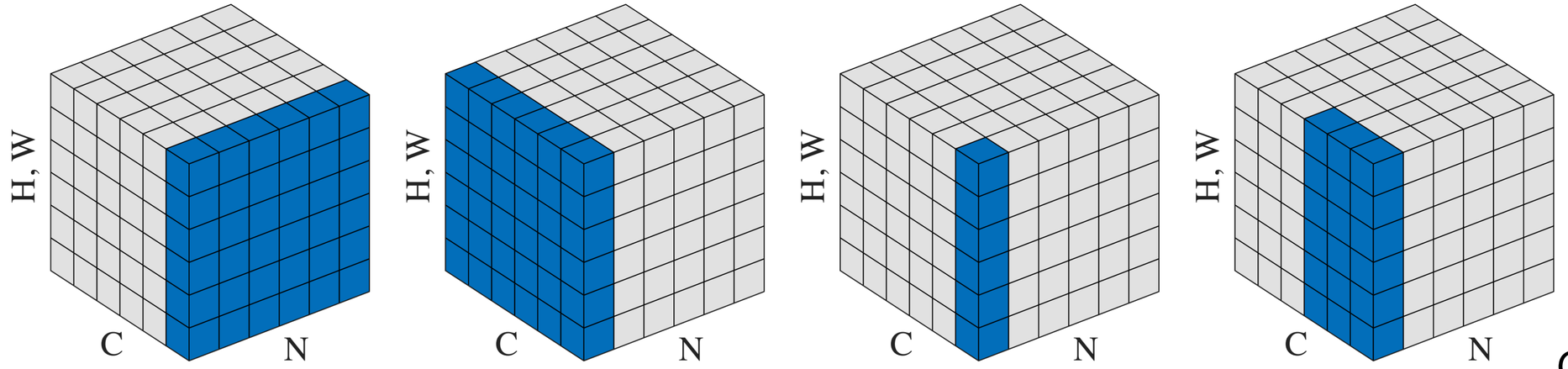
Batch Norm Layer Norm Instance Norm Group Norm

출처: Group Normalization (He et al, ECCV 2018)

Overview of Normalizations

ACADENTIAL

	Batch Norm	Layer Norm	Instance Norm	Group Norm
“묶어서 평균을 내는 기준”	Batch, Width, Height	Channel, Width, Height	Width, Height	Channels in each group Width, Height
“개별적으로 평균을 내는 기준”	Channel	Batch	Batch, Channel	Batch, Group



Batch Norm Layer Norm Instance Norm Group Norm

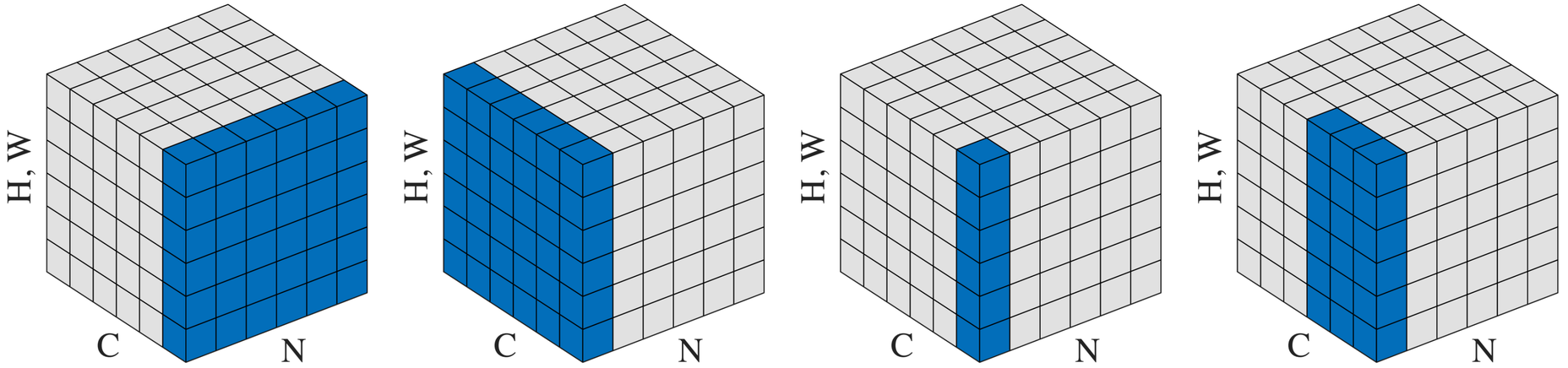
출처: Group Normalization (He et al, ECCV 2018)

정규화 (Normalization)의 종류

ACADENTIAL

Feature의 어떤 dimension을 정규화하는지에 따라 Normalization의 종류가 구별된다!

	Batch Norm	Layer Norm	Instance Norm	Group Norm
“묶어서 평균을 내는 기준”	Batch, Width, Height	Channel, Width, Height	Width, Height	Channels in each group Width, Height
“개별적으로 평균을 내는 기준”	Channel	Batch	Batch, Channel	Batch, Group



ACADENTIAL

	Batch Norm	Layer Norm	Instance Norm	Group Norm
사용처	CNN	Transformer	StyleNet (Style Transfer)	CNN
수식	$\mu_c = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$ $\sigma_c^2 = \frac{1}{NHW}\sum_{n}^{N}\sum_{h}^{H}\sum_{w}^{W}(x_{nhwc}-\mu_c)^2$ $\hat{x}_{nhwc} = \frac{x_{nhwc}-\mu_c}{\sqrt{\sigma_c^2+\epsilon}}$ $y_{nhwc} = \gamma_c\hat{x}_{nhwc} + \beta_c$	$\mu_n = \frac{1}{CHW}\sum_{c}^{C}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$ $\sigma_n^2 = \frac{1}{CHW}\sum_{c}^{C}\sum_{h}^{H}\sum_{w}^{W}(x_{nhwc}-\mu_n)^2$ $\hat{x}_{nhwc} = \frac{x_{nhwc}-\mu_n}{\sqrt{\sigma_n^2+\epsilon}}$ $y_{nhwc} = \gamma_c\hat{x}_{nhwc} + \beta_c$	$\mu_{nc} = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$ $\sigma_{nc}^2 = \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W}(x_{hwc}-\mu_{nc})^2$ $\hat{x}_{nhwc} = \frac{x_{nhwc}-\mu_{nc}}{\sqrt{\sigma_{nc}^2+\epsilon}}$ $y_{nhwc} = \gamma_c\hat{x}_{nhwc} + \beta_c$	$\mu_{ni} = \frac{1}{S_iHW}\sum_{c\in S_i}\sum_{h}^{H}\sum_{w}^{W} x_{nhwc}$ $\sigma_{ni}^2 = \frac{1}{S_iHW}\sum_{c\in S_i}\sum_{h}^{H}\sum_{w}^{W}(x_{nhwc}-\mu_{ni})^2$ $\hat{x}_{nhwc} = \frac{x_{nhwc}-\mu_{ni}}{\sqrt{\sigma_{ni}^2+\epsilon}}$ $y_{nhwc} = \gamma_c\hat{x}_{nhwc} + \beta_c$
“묶어서 평균을 내는 기준”	Batch, Width, Height	Channel, Width, Height	Width, Height	Channels in each group Width, Height
“개별적으로 평균을 내는 기준”	Channel	Batch	Batch, Channel	Batch, Group

ACADENTIAL

Section Summary

Batch Normalization의 효과

(잘못 알려졌던 효과)

Batch Normalization은 Internal Covariate Shift 문제를 해소시켜 모델 성능에 도움을 준다.

(실제 효과)

Batch Normalization은 Optimization Landscape을 매끄럽게 만들어줘서 모델의 학습을 안정화한다!

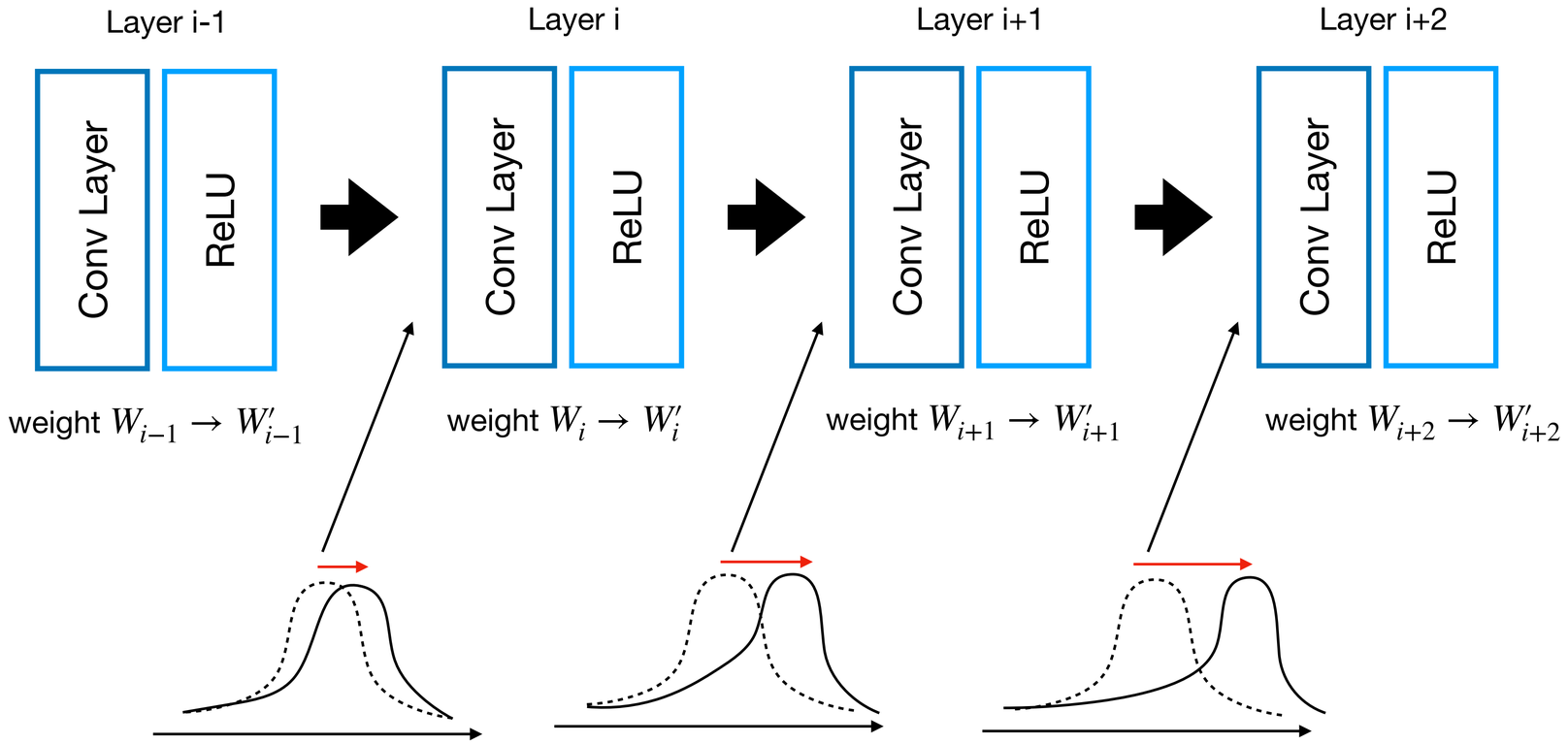
ACADENTIAL

Section Summary

Batch Normalization의 효과

Internal Covariate Shift

Batch Normalization은 이것에 도움을 주는게 아니라

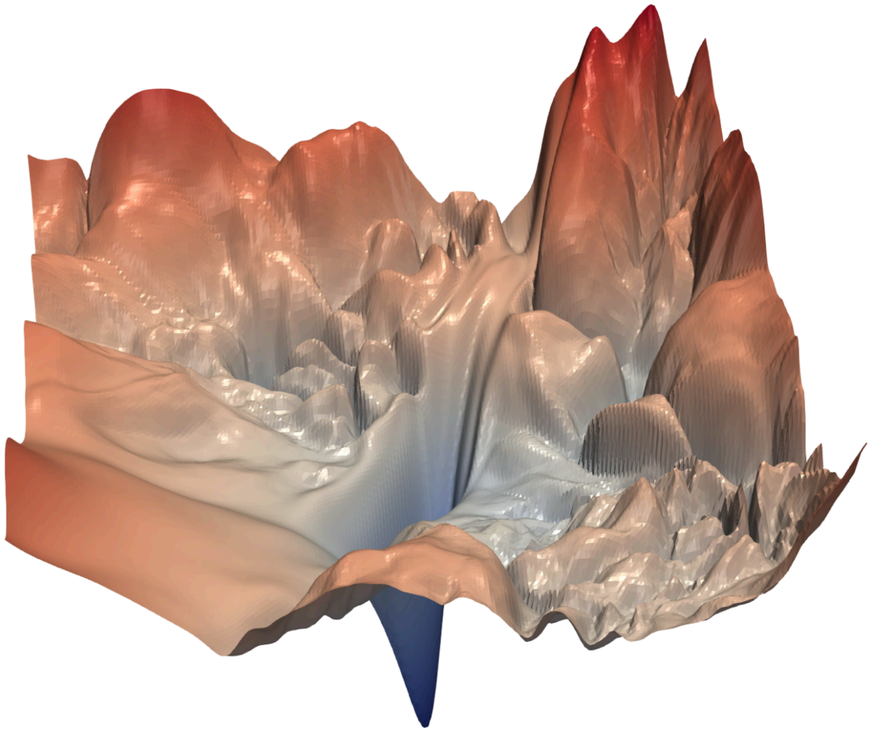


각 Layer의 input의 분포

ACADENTIAL

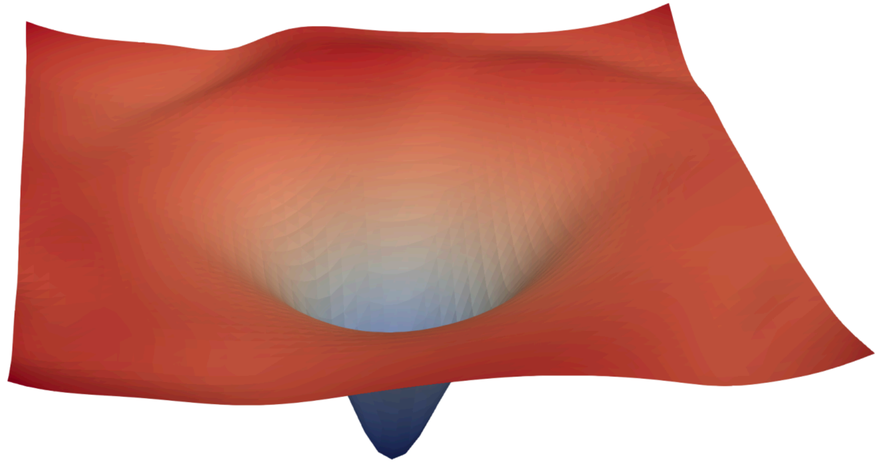
Section Summary

Batch Normalization의 효과



Without Batch Normalization

이거에 도움을 주는 것이다!



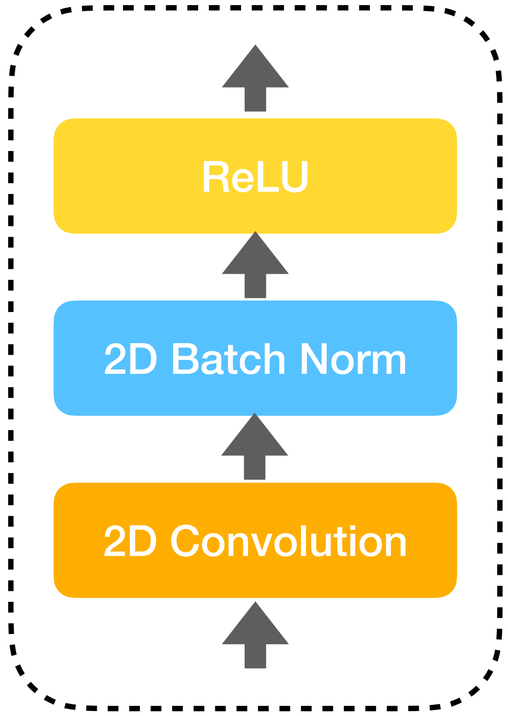
With Batch Normalization

Visualizing the Loss Landscape of Neural Nets (NeurIPS 2018): Li et al.

ACADENTIAL

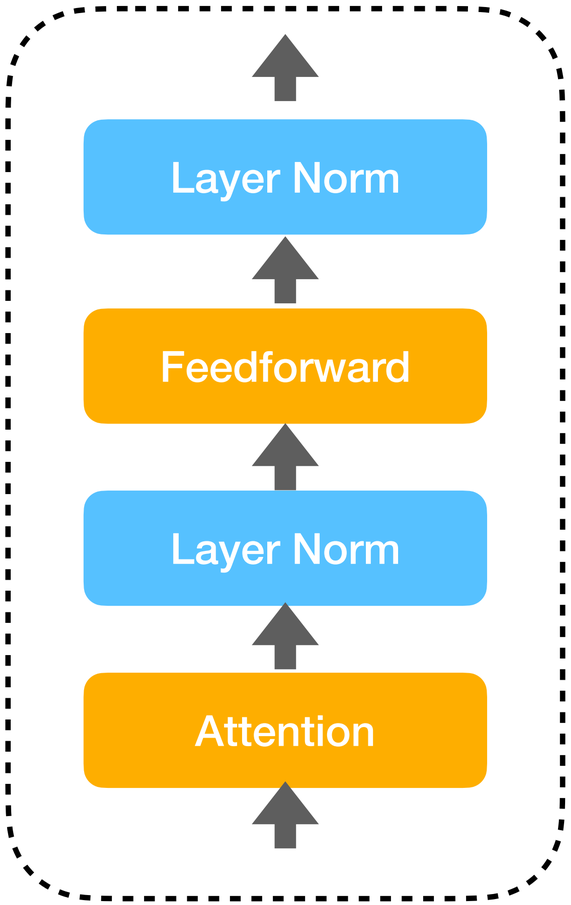
Section Summary

Normalization Layer의 위치



일반적인 CNN Block

(일반적으로) Normalization layer은 NN layer와 activation function의 사이에 위치함!



Transformer Block

ACADENTIAL

Next Up!