Group Normalization

ECCV 2018

Yuxin Wu, Kaiming He
Facebook AI Research (FAIR)

이주남
slack : 이주남_T1163
email : joonamm@naver.com

Contents

선행 지식

간단 정리

위 연구는 Computer Vision 관련 모델에 대한 성능평가 위주로 구성

- **정규화 (Normalization → Standardization)**

$$\frac{x_i - \mu}{\sigma} \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

- **정규화의 이점**

  각 차원 간의 크기와 분포를 동일하게 rescaling
  데이터의 분포를 일관 되게 바꿔 준다. 이로서 학습 시 수렴 속도가 올라가고 성능 개선이 가능하다.

- **Batch Normalization 의 도입**

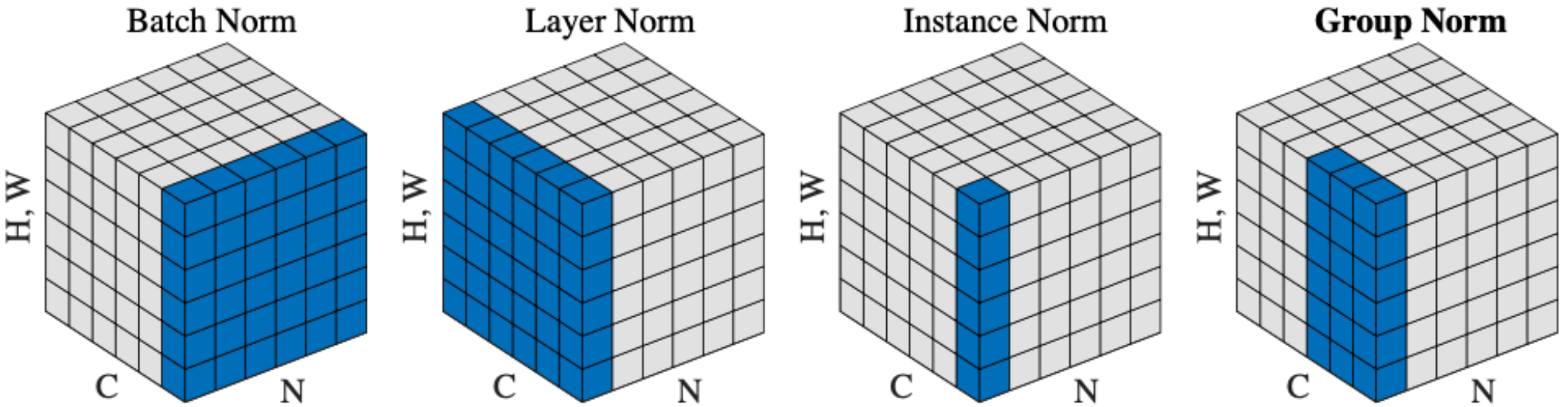
  최초 Input data를 정규화를 통해서 성능 향상을 볼 수 있다.
  하지만 Hidden Layer 에 Output들도 정규화 할 수 없을까?

$$\hat{x_i} = \frac{1}{\sigma_i}\left(x_i - \mu_i\right) \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

선행 지식

간단 정리

Hidden Layer를 정규화 하는 방법론들.



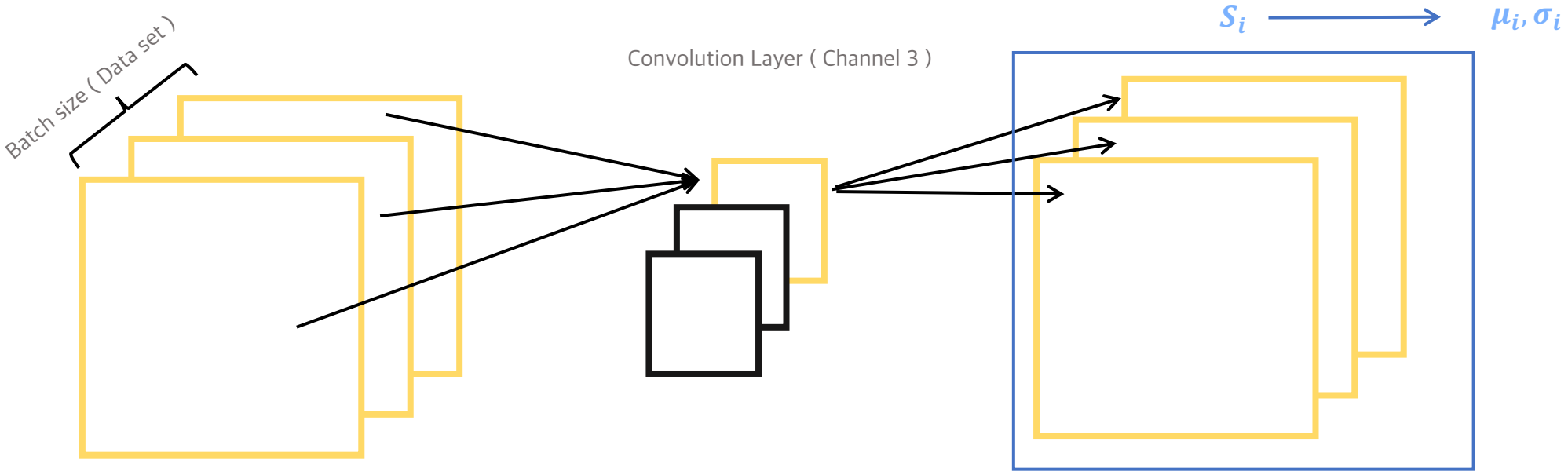
Batch normalization

Batch size 만큼의 데이터를 입력으로 **하나의 Channel(= Node)**를 통해서 나온 출력값을 묶어서 정규화 수행

$$\widehat{x_i} = \frac{1}{\sigma_i}\left(x_i - \mu_i\right) \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

$$S_i = \{k \mid k_C = i_C\}$$

Batch normalization



Layer normalization

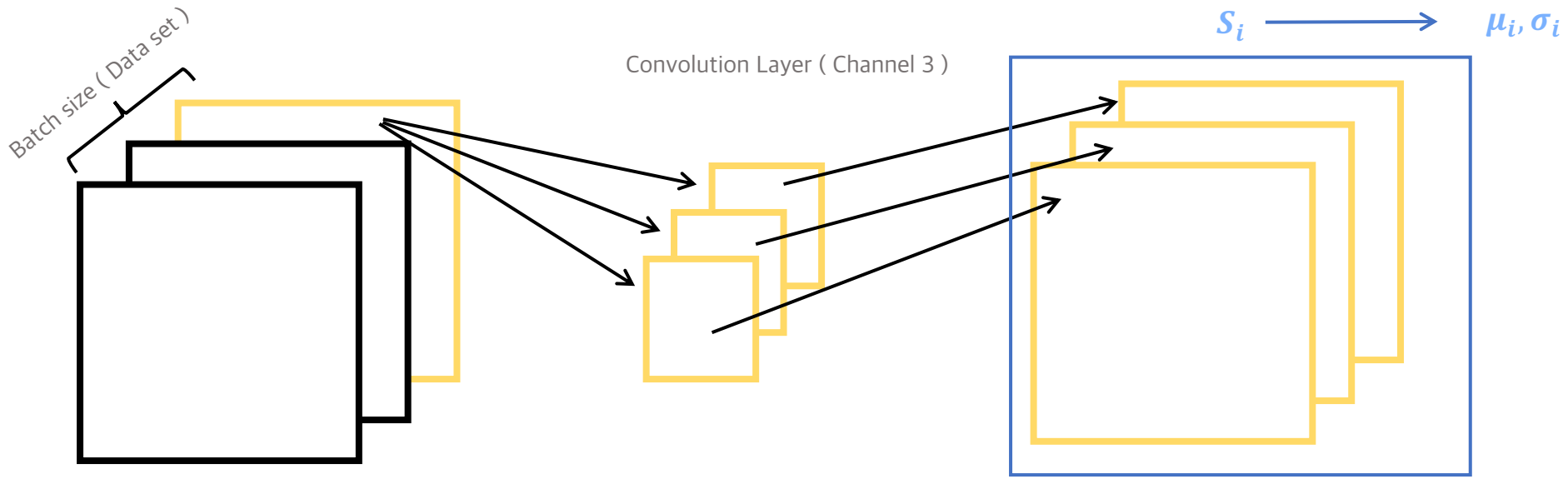
한개의 데이터를 입력으로 **모든 Channel을** 통해서 나온 출력 값들을 묶어서 정규화 수행

Batch normalization 의 Batch size 에 따른 의존성을 제거 하기 위해 도입된 방법.

$$\hat{x_i} = \frac{1}{\sigma_i}\left(x_i - \mu_i\right) \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

$$S_i = \{k \mid k_N = i_N\}$$

Layer normalization



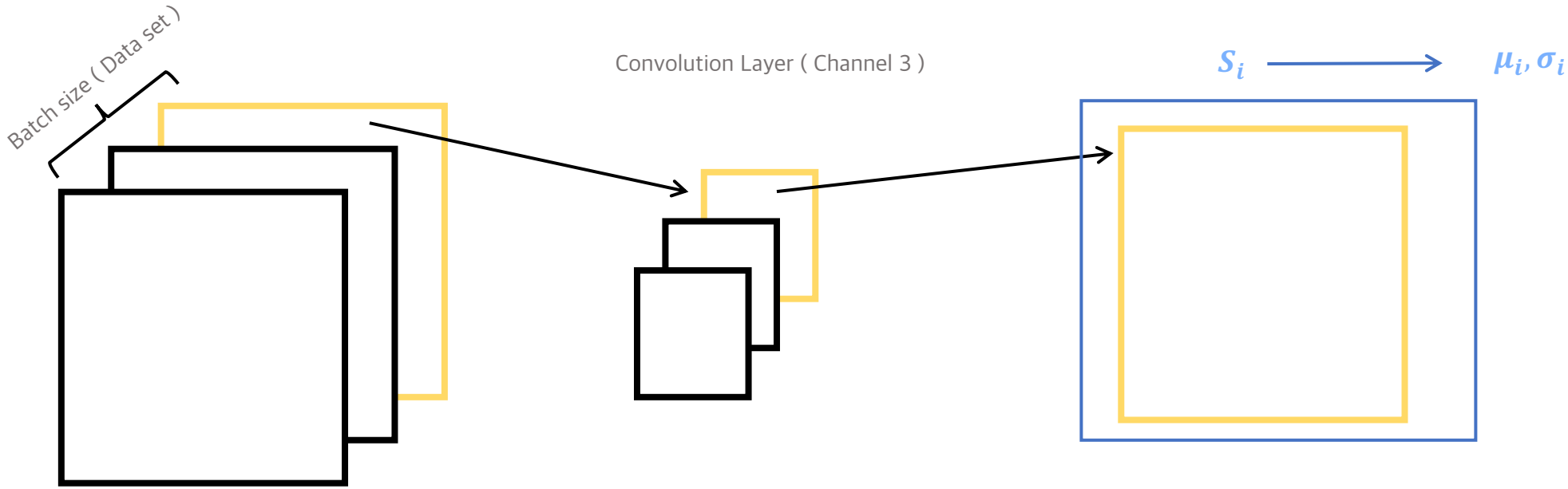
Instance normalization

한개의 데이터를 입력으로 **한개의 Channel을** 통해서 나온 출력 값을 묶어서 정규화 수행

style transfer을 위해 고안된 방법이기 때문에 style transfer에서 BN을 대체하여 많이 사용하고
real-time generation에 효과적

$$\hat{x_i} = \frac{1}{\sigma_i}\left(x_i - \mu_i\right) \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

Instance normalization $\quad \mathcal{S}_i = \{k \mid k_N = i_N, k_C = i_C\}$



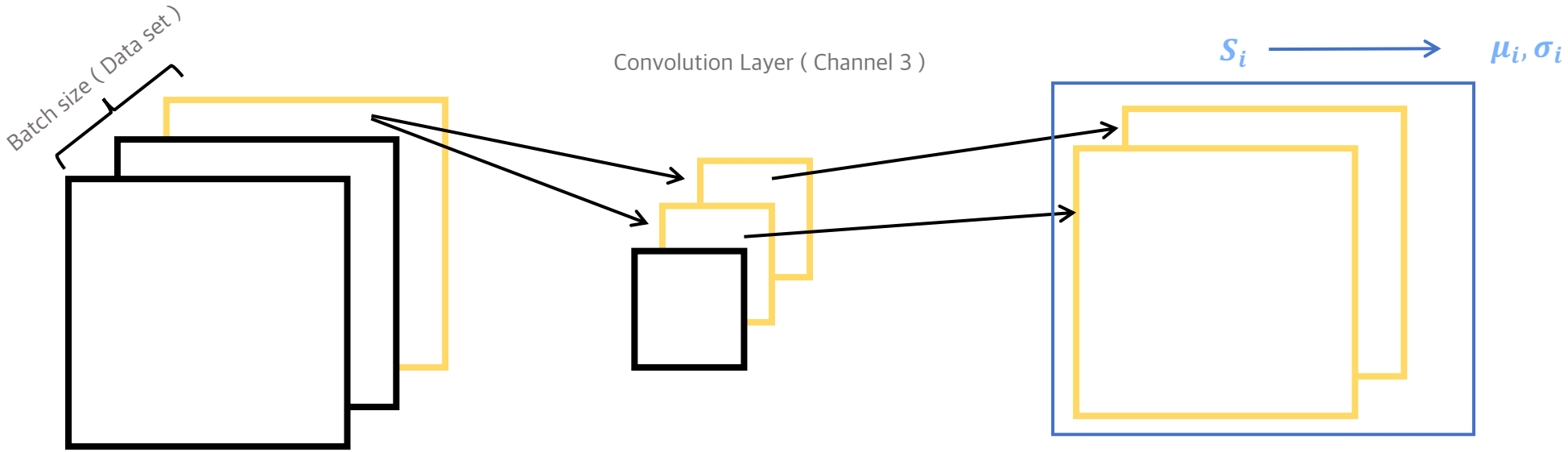
Group normalization

한개의 데이터를 입력으로 **임의의 n개의 Channel을** 통해서 나온 출력 값들을 묶어서 정규화 수행

$$\hat{x_i} = \frac{1}{\sigma_i}\left(x_i - \mu_i\right) \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

$$\mathcal{S}_i = \{k \mid k_N = i_N, \lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor\}.$$

Group normalization



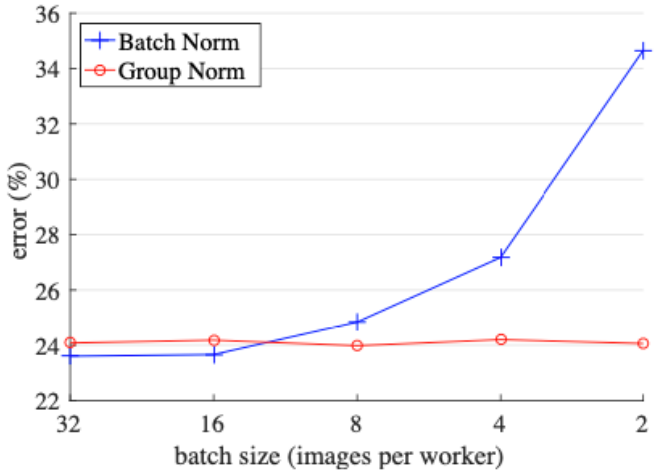
Batch normalization (BN) 의 한계

성능은 Group Normalization에 비해서 월등하다

BN 은 Batch size에 **상당히 의존적**
Batch size 크기에 따라서 학습 성능 차이가 현저하게 나타남

Image Detection, Segmentation 같은 Large model의 경우
Batch size를 줄여 메모리 할당을 최소화 하는 것이 필연적 이에 따른 BN의 한계

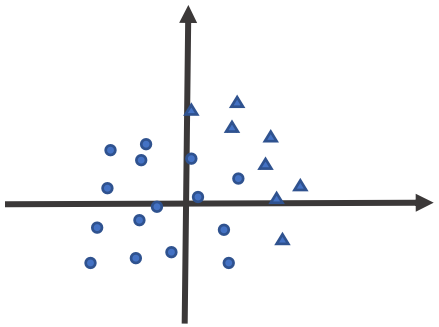
ImageNet classification error *vs.* batch sizes. (ResNet-50)



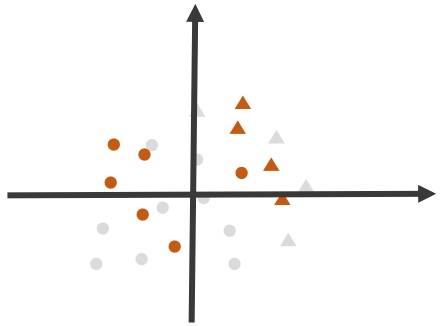
Why?

Batch size가 작을수록 Batch의 **평균과 분산이 불안정.**

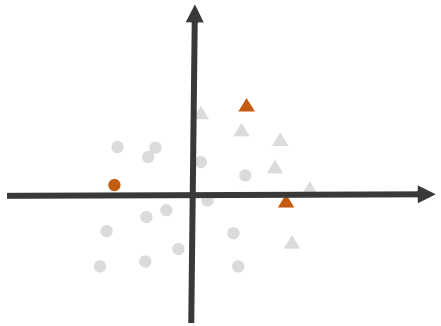
Data set



Many batch



Small batch



Batch normalization (BN) 의 대안

Batch size 에 독립적으로 할 순 없을까? : Instance Normalization, Layer Normalization

Batch size 의 크기를 줄여도 학습이 떨어지는 것을 막을 수 없을까? : Batch Renormalization

Batch Renormalization

기존의 Batch Normalization에서 Batch 의 평균과 분산을 제한하는 Parameter를 부여 하는 방식

하지만 Batch size 에 영향을 받음

분산처리 (Multiple GPU)

분산 처리를 통한 Batch size의 크기를 줄이지 못하게 제한하는 방법

하드웨어 구간 까지 고려하며 학습해야 하기에 전체적인 비용이 큼
Large scale distributed deep networks. In: NIPS. (2012)

Group normalization

Group

Hidden layer의 channel 묶음

If c = 10, g = 2 → 5개 의 channel 을 하나의 그룹으로 묶음

한개의 데이터를 입력으로 **임의의 n개의 Channel을** 통해서 나온 출력 값들을 묶어서 정규화 수행

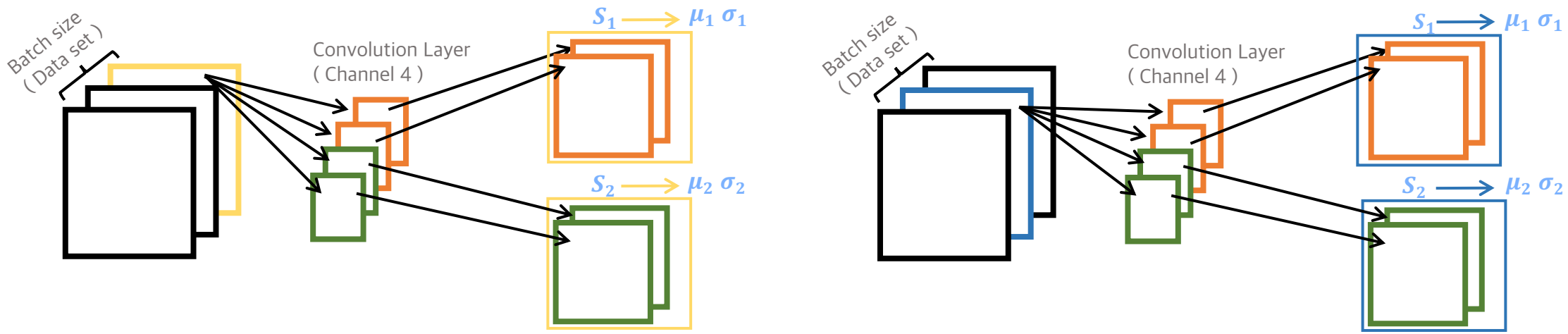
$$\hat{x_i} = \frac{1}{\sigma_i}(x_i - \mu_i) \qquad \mu_i = \frac{1}{m}\sum_{k \in S_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon},$$

$$S_i = \{k \mid k_N = i_N, \lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor\}.$$

G : Group 의 수 k : grouping data의 집합 N : Batch size

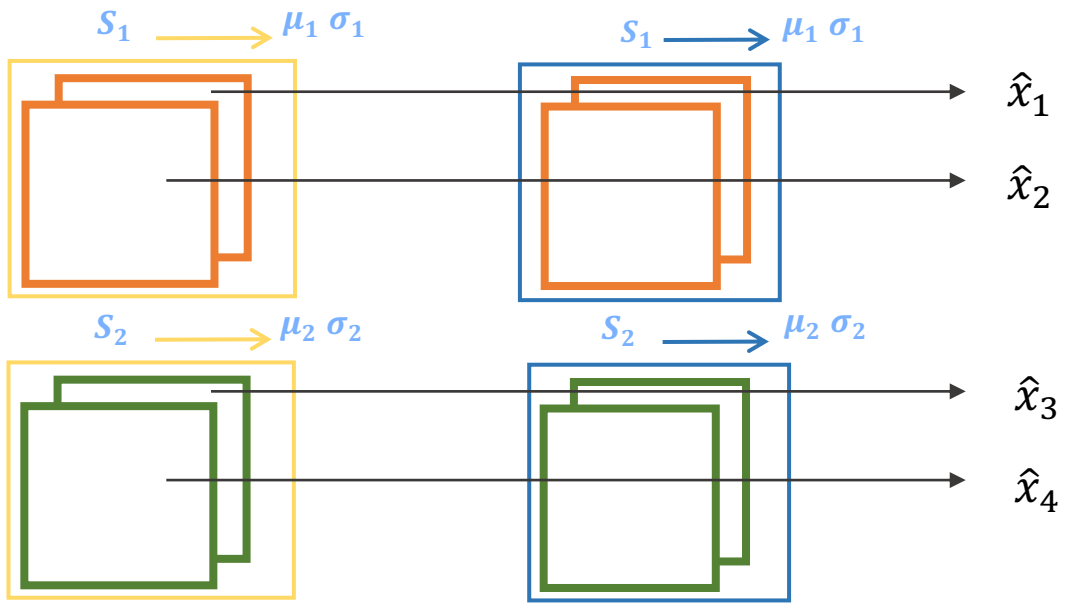
C : Channel i : Input

Group normalization (Group 2)



Group normalization

Normalization 이후의 과정은 Batch normalization과 동일하다.



$$y_i = \gamma\ \hat{x}_i + \beta$$

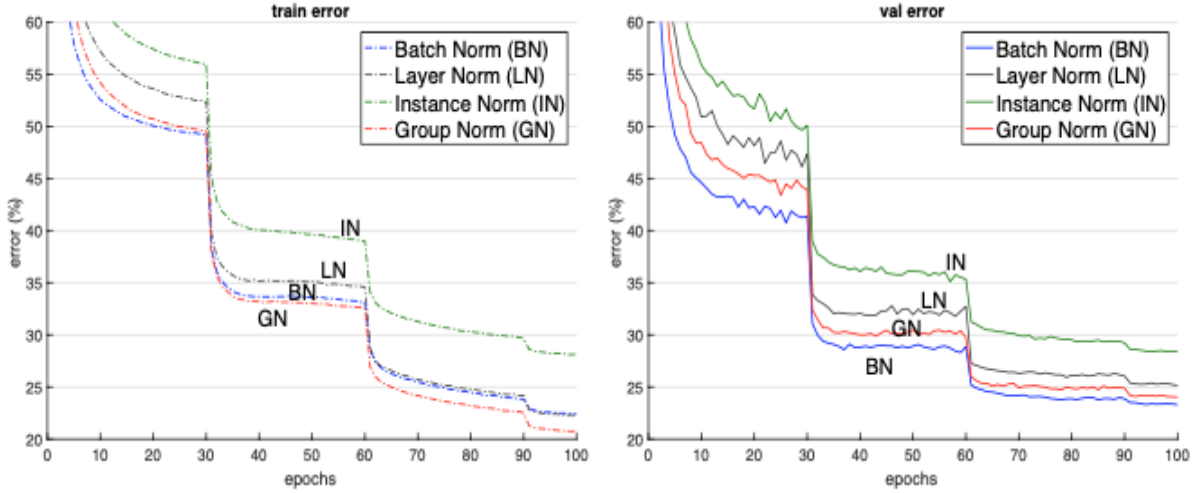
Group Normalization

Why?

Why Group Normalization?

Batch normalization과 비교할 때 Batch size에 **독립적**

error curves with a batch size of 32 images/GPU (ResNet-50.)



Layer Normalization 과 Instance normalization 의 사이

Layer normalization, Instance normalization

Layer Normalization (LN)

Layer normalization의 가정 : 각각의 channel 들을 동일한 기여를 한다.

Layer normalization. 저자 Et al. (2016)

Cf) Fully Connected Layer

이미지 처리 과정 마지막 단에 Fully Connected Layer를 수행하는 이유는 각각 channel에 따른 특성들을 뽑아 내기 위함
하지만 모든 Channel 이 동등한 기여를 한다고 가정한다면. 이는 어찌 본다면 Convolution 을 사용하는 목적 부재.

Instance normalization (IN)

Layer normalization과는 다르게 하나의 채널에 대해서만 평균과 분산을 구함.
하지만 이는 다른 채널에 대한 영향을 아예 받지 않기 때문에 다른channel 들의 특징을 공유하지 못함.

Group Normalization 은 LN, IN 과 같이 극단적으로 channel의 묶거나 나누지 않음.
이는 채널들 간의 중요성을 나타낼 수 있게 보정
또한 각 channel간의 특징도 공유하기에 성능 개선에 효과를 보임

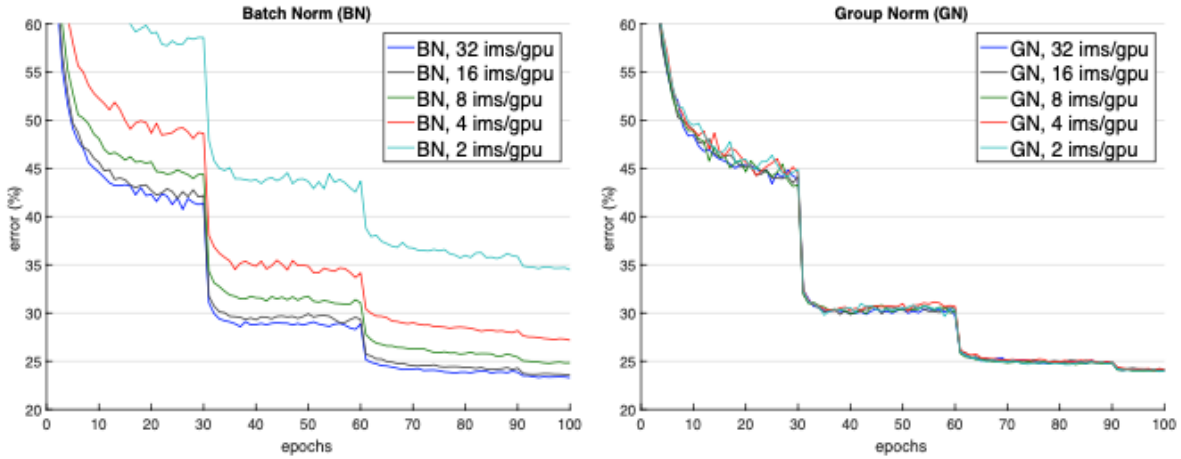
연구 성과

수식 정리, 도식도

이미지 분류 에선 BN보단 성능이 미약하게 안좋지만,
Batch size 에 대한 민감도 가 낮다.

ResNet-50's validation error using a batch size of 32, 16, 8, 4, and 2 images/GPU

batch size	32	16	8	4	2
BN	**23.6**	**23.7**	24.8	27.3	34.7
GN	24.1	24.2	**24.0**	**24.2**	**24.1**
△	*0.5*	*0.5*	*-0.8*	*-3.1*	*-10.6*



Segmentation 데이터 셋인 COCO에선 GN 더 좋은 성능을 보임

Detection and segmentation results in COCO, using Mask R-CNN with the ResNet-50 C4 backbone

backbone	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
C4, BN*	37.7	57.9	40.9	32.8	54.3	34.7
C4, GN	**38.8**	**59.2**	**42.2**	**33.6**	**55.9**	**35.4**

하이퍼 파라미터 G (Group의 수)

Group 의 사이즈를 고정(오른쪽)
Group 의 사이즈를 유동적(왼쪽)

ResNet-50's ImageNet validation error

# groups (G)						
64	32	16	8	4	2	1 (=LN)
24.6	**24.1**	24.6	24.4	24.6	24.7	25.3
0.5	-	*0.5*	*0.3*	*0.5*	*0.6*	*1.2*

channels per group						
64	32	16	8	4	2	1 (=IN)
24.4	24.5	**24.2**	24.3	24.8	25.6	28.4
0.2	*0.3*	-	*0.1*	*0.6*	*1.4*	*4.2*