Weight Standardization

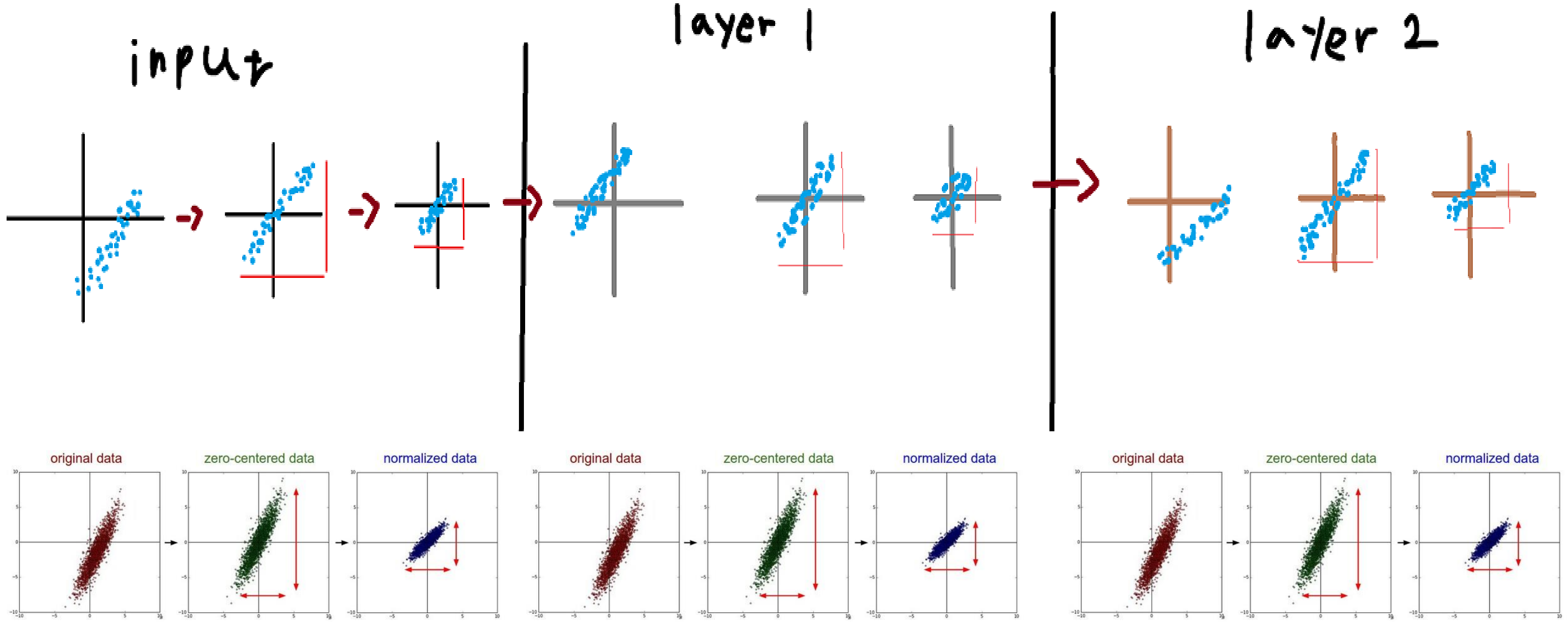
Micro-Batch Training with Batch-Channel Normalization and Weight Standardization(https://arxiv.org/abs/1903.10520)

한성대학교 1971336 김태민

목차

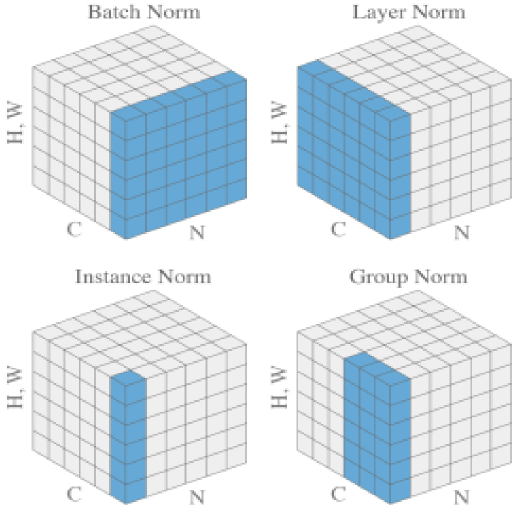
The need for normalizatiom

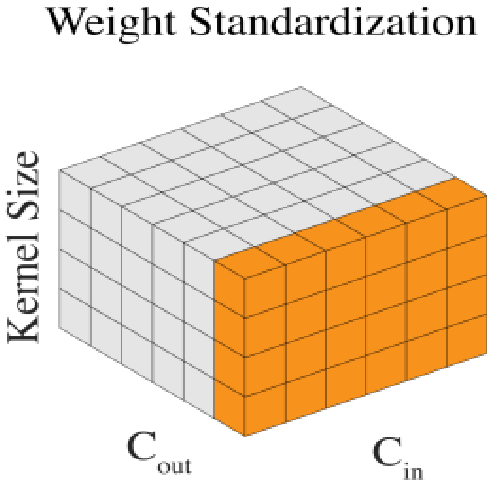
정규화의 이해

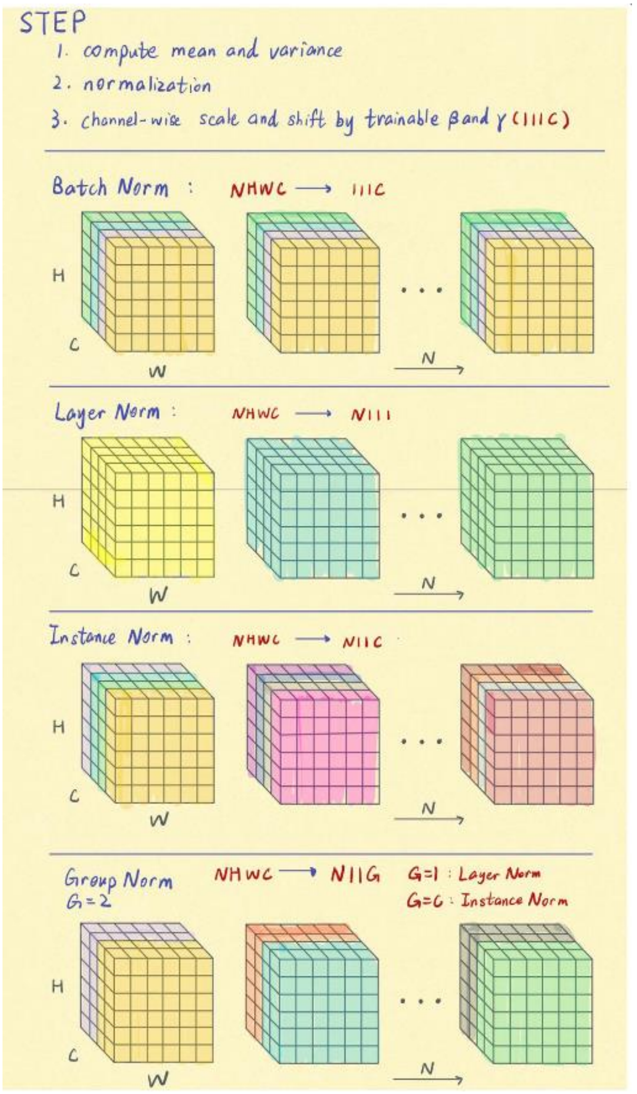


Type of Normalizatiom

각종 Normalization 방법



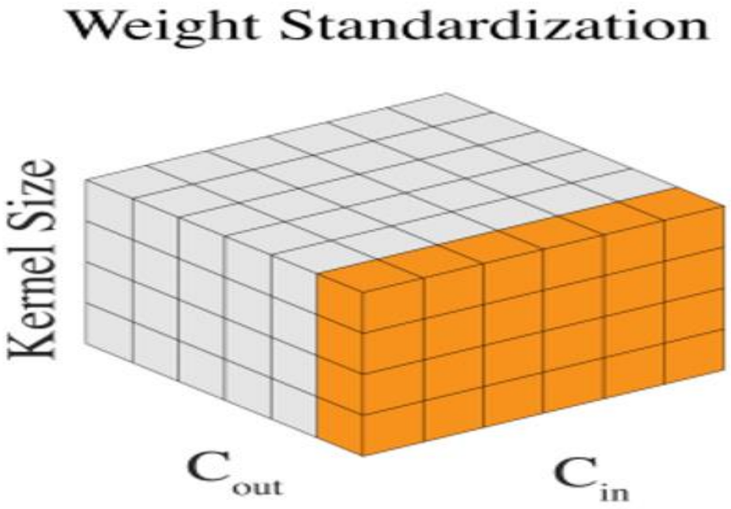




- Batch Norm
- 미니 배치 전체에 걸쳐 feature의 특정 채널을 모두 합쳐 normalize함
- [Batch Normalization(Sergey Ioffe, Christian Szegedy)]
- Group Norm
- 개별 데이터에서 나온 featur의 채널들을 N개의 그룹으로 묶어 normalize함
  [Group Normalization(Yuxin Wu, Kaiming He)]
- Weight Standardization
- 단순 하게 weight(convolution filter)을 대상을 normalization을 수행
- (Filter의 mean을, 0 variance를 1로 조정)
- [Micro-Batch Training with Batch-Channel Normalization and Weight Standardization]

Weight Standarization

Weight Standardization



Weight Standarization

Weight Standardization

$$\hat{\boldsymbol{W}} = \left[\hat{\boldsymbol{W}}_{i,j} \mid \hat{\boldsymbol{W}}_{i,j} = \frac{\boldsymbol{W}_{i,j} - \mu_{\boldsymbol{W}_{i,\cdot}}}{\sigma_{\boldsymbol{W}_{i,\cdot}}}\right], \quad (5)$$

$$\boldsymbol{y} = \hat{\boldsymbol{W}} * \boldsymbol{x}, \quad (6)$$

where

$$\mu_{\boldsymbol{W}_{i,\cdot}} = \frac{1}{I}\sum_{j=1}^{I} \boldsymbol{W}_{i,j}, \quad \sigma_{\boldsymbol{W}_{i,\cdot}} = \sqrt{\frac{1}{I}\sum_{j=1}^{I} \boldsymbol{W}_{i,j}^2 - \mu_{\boldsymbol{W}_{i,\cdot}}^2 + \epsilon}. \quad (7)$$

Weight Standarization

Weight Standardization

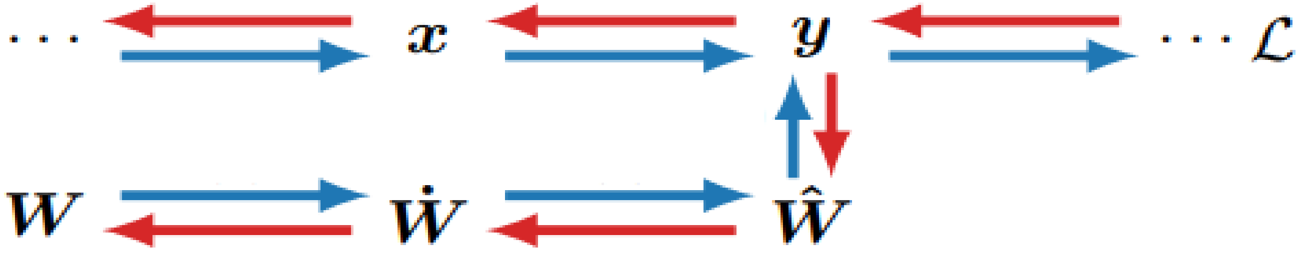
$$\dot{\boldsymbol{W}}_{c,\cdot} = \boldsymbol{W}_{c,\cdot} - \frac{1}{I}\mathbf{1}\langle \mathbf{1}, \boldsymbol{W}_{c,\cdot}\rangle,$$

$$\hat{\boldsymbol{W}}_{c,\cdot} = \dot{\boldsymbol{W}}_{c,\cdot} / \left(\sqrt{\frac{1}{I}\langle \mathbf{1}, \dot{\boldsymbol{W}}^{\circ 2}_{c,\cdot}\rangle}\right), \text{ we set } \epsilon = 0$$

$$\boldsymbol{y}_c = \boldsymbol{x}_c \hat{\boldsymbol{W}}_{c,\cdot},$$

Weight Standarization

Backpropagation of Weight Standardization



forward propagation

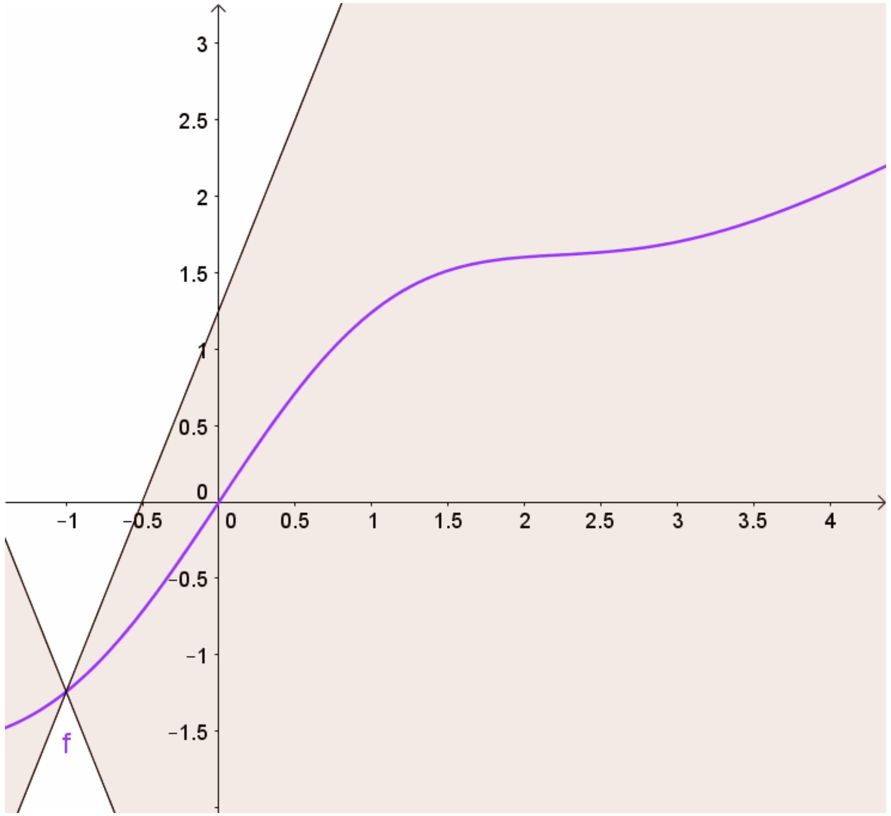
$$\dot{\boldsymbol{W}}_{c,\cdot} = \boldsymbol{W}_{c,\cdot} - \frac{1}{I}\mathbf{1}\langle \mathbf{1}, \boldsymbol{W}_{c,\cdot}\rangle,$$
$$\hat{\boldsymbol{W}}_{c,\cdot} = \dot{\boldsymbol{W}}_{c,\cdot}/\left(\sqrt{\frac{1}{I}\langle \mathbf{1}, \dot{\boldsymbol{W}}_{c,\cdot}^{\circ 2}\rangle}\right),$$
$$\boldsymbol{y}_c = \boldsymbol{x}_c \hat{\boldsymbol{W}}_{c,\cdot},$$

Backpropagation

$$\nabla_{\dot{\boldsymbol{W}}_{c,\cdot}} \mathcal{L} = \frac{1}{\sigma_{\boldsymbol{W}_{c,\cdot}}}\left(\nabla_{\hat{\boldsymbol{W}}_{c,\cdot}} \mathcal{L} - \frac{1}{I}\langle \hat{\boldsymbol{W}}_{c,\cdot}, \nabla_{\hat{\boldsymbol{W}}_{c,\cdot}} \mathcal{L}\rangle \hat{\boldsymbol{W}}_{c,\cdot}\right), \quad (14)$$
$$\nabla_{\boldsymbol{W}_{c,\cdot}} \mathcal{L} = \nabla_{\dot{\boldsymbol{W}}_{c,\cdot}} \mathcal{L} - \frac{1}{I}\mathbf{1}\langle \mathbf{1}, \nabla_{\dot{\boldsymbol{W}}_{c,\cdot}} \mathcal{L}\rangle. \quad (15)$$

Weight Standarization Proof

Lipschitzness



$$||f(x) - f(y)|| \leq K||x - y||$$

- 립시츠 연속성
- 아래와 같은 식을 립시츠 연속성이라고 부른다.
- (f의 기울기가 색칠된 영역 안에 존재)

$$||f(x) - f(y)|| \leq K||x - y||$$

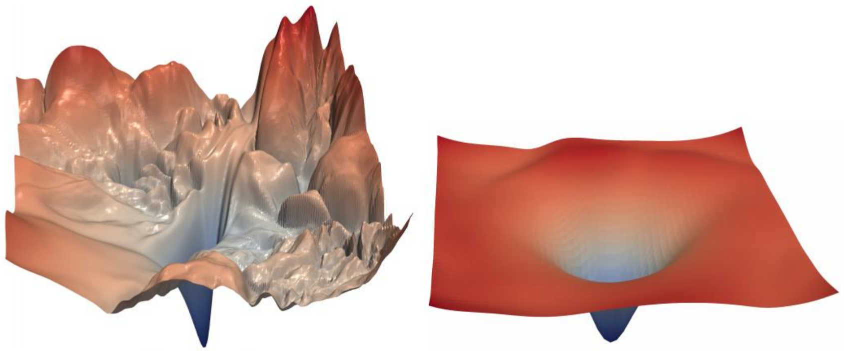
$$\frac{||f(x) - f(y)||}{||x - y||} \leq K$$

- 설명 : 주어진 구간의 함수의 두 점을 이은 직선의 기울기가 K보다 작다.

Weight Standarization Proof

Lipschitzness

$$\forall x_1, x_2 : |f(x_1) - f(x)| \leq L \, \|x_1 - x_2\|$$



(a) without skip connections　　(b) with skip connections

Visualizing Loss landscape

- L을 Lipxchitz constant라고 부른다.
  L이 작을수록 f가 좀 더 smooth해진다.

- Lipxchitz constant는 gradient의 ‘크기’에 좌우되므로
- (Gradient크기 얼마나 빠르게 해당방향으로 증가)

- Loss(L)의 landscape를 smooth 해야하기 위해
- Loss gradient를 줄여야 한다.

- gradient의 landscape를 smooth하게 만들기 위해서는
- Gradient의 gradient인 Hessian(H) 을 줄여야한다.

- 결론 :
- Hessian(H)과 loss의 gradient 줄이는 문제로 재정의

Weight Standarization Proof

Lipschitzness

By Eq. 14, we rewrite the second term:

$$\frac{1}{I}\langle \mathbf{1}, \nabla_{\dot{\boldsymbol{W}}_{c,\cdot}}\mathcal{L}\rangle^2 = \frac{1}{I\cdot\sigma^2_{W_{c,\cdot}}}\Big(\langle \mathbf{1}, \nabla_{\hat{\boldsymbol{W}}_{c,\cdot}}\mathcal{L}\rangle - \frac{1}{I}\langle \hat{\boldsymbol{W}}_{c,\cdot}, \nabla_{\hat{\boldsymbol{W}}_{c,\cdot}}\mathcal{L}\rangle \cdot \langle \mathbf{1}, \hat{\boldsymbol{W}}_{c,\cdot}\rangle\Big)^2.$$

Since $\langle \mathbf{1}, \hat{\boldsymbol{W}}_{c,\cdot}\rangle = 0$, we have

$$\left\|\nabla_{\boldsymbol{W}_{c,\cdot}}\mathcal{L}\right\|^2 = \left\|\nabla_{\dot{\boldsymbol{W}}_{c,\cdot}}\mathcal{L}\right\|^2 - \frac{1}{I\cdot\sigma^2_{W_{c,\cdot}}}\langle \mathbf{1}, \nabla_{\hat{\boldsymbol{W}}_{c,\cdot}}\mathcal{L}\rangle^2.$$

$$\begin{aligned}\|\boldsymbol{H}\|_F &= \sum_{i=1}^{I}\sum_{j=1}^{I}\boldsymbol{H}^2_{i,j}\\ &= \|\dot{\boldsymbol{H}}\|_F + \frac{1}{I^2}\Big(\sum_{i=1}^{I}\sum_{j=1}^{I}\dot{\boldsymbol{H}}_{i,j}\Big)^2\\ &\quad - \frac{1}{I}\sum_{i=1}^{I}\Big(\sum_{j=1}^{I}\dot{\boldsymbol{H}}_{i,j}\Big)^2 - \frac{1}{I}\sum_{j=1}^{I}\Big(\sum_{i=1}^{I}\dot{\boldsymbol{H}}_{i,j}\Big)^2\\ &\leq \|\dot{\boldsymbol{H}}\|_F - \frac{1}{I^2}\Big(\sum_{i=1}^{I}\sum_{j=1}^{I}\dot{\boldsymbol{H}}_{i,j}\Big)^2\end{aligned}$$

$$\dot{\boldsymbol{W}}_{c,\cdot} = \boldsymbol{W}_{c,\cdot} - \frac{1}{I}\mathbf{1}\langle \mathbf{1}, \boldsymbol{W}_{c,\cdot}\rangle, \tag{11}$$

$$\hat{\boldsymbol{W}}_{c,\cdot} = \dot{\boldsymbol{W}}_{c,\cdot} / \Big(\sqrt{\frac{1}{I}\langle \mathbf{1}, \dot{\boldsymbol{W}}^{\circ 2}_{c,\cdot}\rangle}\Big), \tag{12}$$

$$\boldsymbol{y}_c = \boldsymbol{x}_c\hat{\boldsymbol{W}}_{c,\cdot}, \tag{13}$$

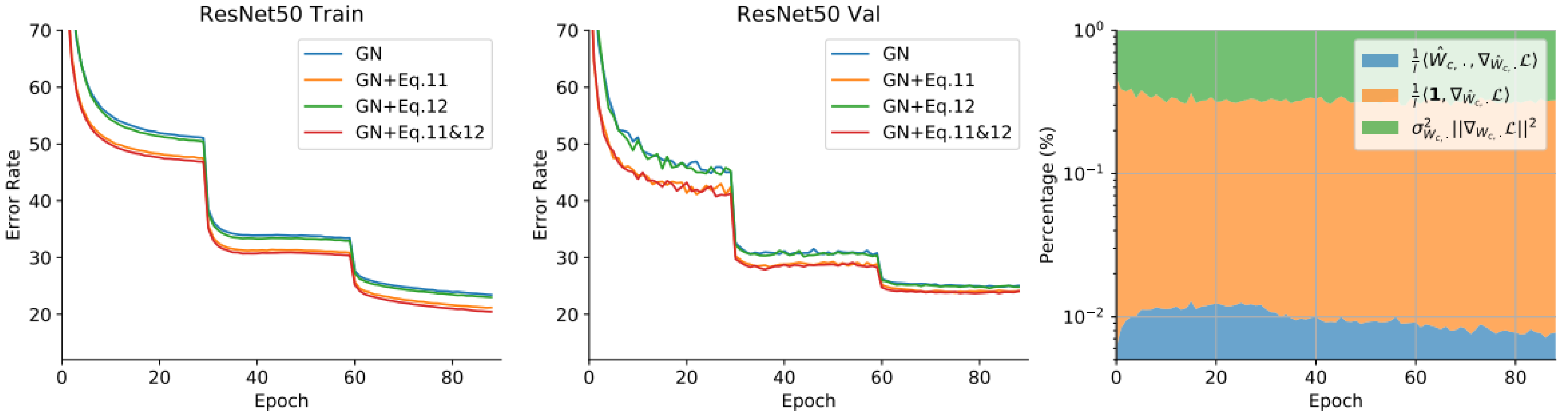
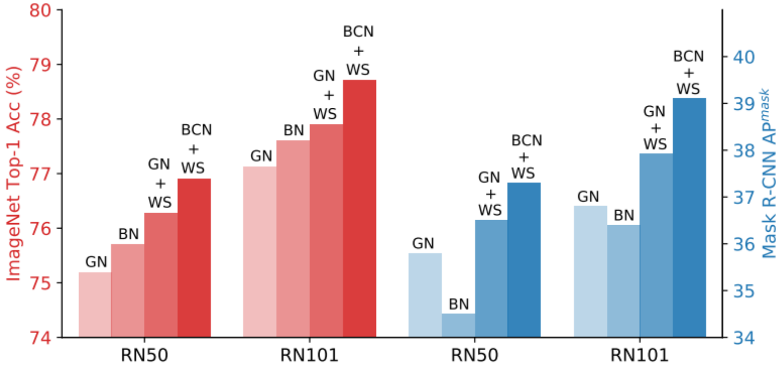


Fig. 4: Training ResNet-50 on ImageNet with GN, Eq. 11 and 12. The left and the middle figures show the training dynamics. The right figure shows the reduction percentages on the Lipschitz constant. Note that the y-axis of the right figure is in **log** scale.

Result

Result



- 왼쪽 그림은 ImageNet과 Mask R-CNN의 비교
- ImageNet의 BN과 BCN은 큰 Batch-size 사용
- 나머지는 1개의 GPU당 1개의 이미지 할당

- 우측은 BN과 BCN은 micro batch-size 사용

Method – Batch Size	BN [3] – 64 / 32		SN [39] – 1		GN [6] – 1		BN+WS – 64 / 32		GN+WS – 1	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ResNet-50 [2]	24.30	7.19	25.00	–	24.81	7.46	23.76	7.13	23.72	6.99
ResNet-101 [2]	22.44	6.21	–	–	22.87	6.51	21.89	6.01	22.10	6.07

TABLE 2: Error rates of ResNet-50 and ResNet-101 on ImageNet. ResNet-50 models with BN are trained with batch size 64 per GPU, and ResNet-101 models with BN are trained with 32 images per GPU. The others are trained with 1 image per GPU.

Result

Result

Dataset	Model	GN	BN	WS	BCN	mIoU
VOC Val	RN101	✓				74.90
VOC Val	RN101	✓		✓		77.20
VOC Val	RN101		✓			76.49
VOC Val	RN101		✓	✓		77.15
VOC Val	RN101			✓	✓	78.22

TABLE 11: Comparisons of semantic segmentation performance of DeepLabV3 [53] trained with different normalizations on PASCAL VOC 2012 [13] validation set. Output stride is 16, without multi-scale or flipping when testing.

Model	#Frame	GN	BN	WS	BCN	Top-1	Top-5
RN50	8	✓				42.07	73.20
RN50	8	✓		✓		44.26	75.51
RN50	8		✓			44.30	74.53
RN50	8		✓	✓		46.49	76.46
RN50	8			✓	✓	45.27	75.22

TABLE 12: Comparing video recognition accuracy of TSM [55] on Something-SomethingV1 [12].

Code

code

- https://github.com/ThomasEhling/Weight_Standardization/blob/master/Weight_Standardization_analysis.pdf


Q&A
SESSION