Group Normalization

Yuxin Wu, Kaiming He

Presenters: Zhiyang Lin, Tianran Wang, Xueyan Zou

Introduction

1. Motivation of Normalization
2. Batch Normalization
3. Compare between Group Normalization, Layer Normalization and Instance Normalization
4. Experiment results on Group Normalization

Normalization

- Min-max normalization

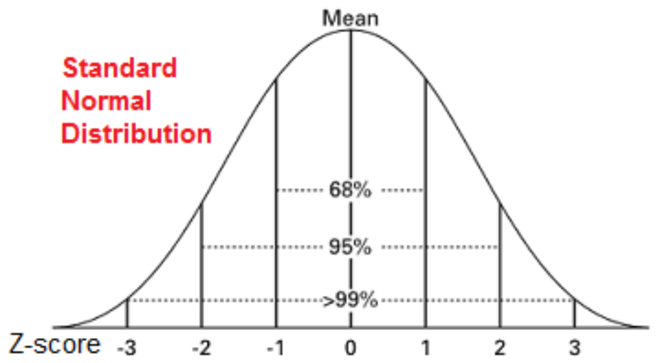
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

rescale the range to [0,1]

- Standardization

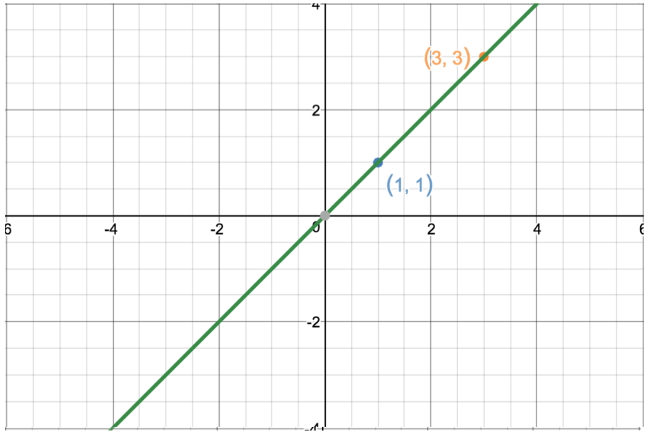
$$x' = \frac{x - \bar{x}}{\sigma}$$

zero mean and unit variance



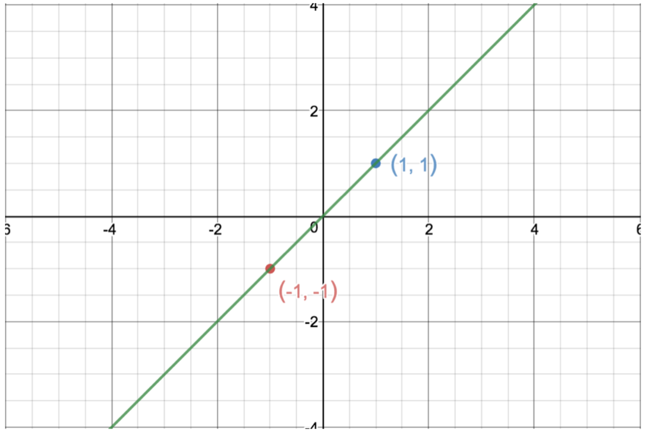
Motivation of Normalization - *Simplify Optimization Problem*

Ex.1 Using L2 loss to find a straight line fitting points (1, 1), (3, 3)



$$\underset{w,b}{\operatorname{argmin}} L(w,b) = \frac{1}{2}[(wx_1 + b - y_1)^2 + (wx_2 + b - y_2)^2]$$
$$= \frac{1}{2}[(w + b - 1)^2 + (2w + b - 3)^2]$$
$$= \frac{1}{2}(5w^2 + 2b^2 - 8b - 14w + 10)$$

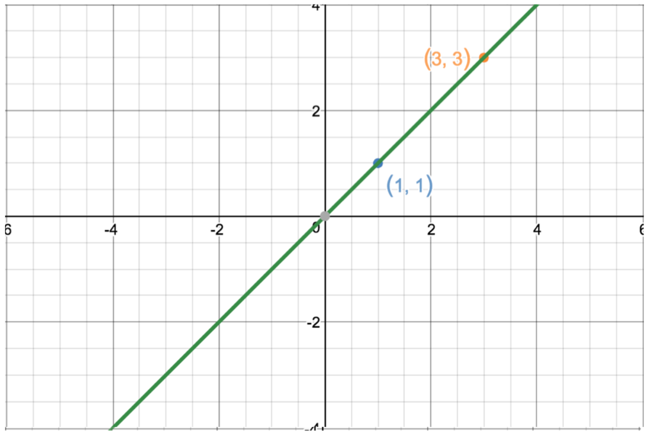
Normalize the data points to (-1, -1), (1, 1)



$$\underset{w,b}{\operatorname{argmin}} L(w,b) = \frac{1}{2}[(wx_1 + b - y_1)^2 + (wx_2 + b - y_2)^2]$$
$$= \frac{1}{2}[(w + b - 1)^2 + (-w + b + 1)^2]$$
$$= (w - 1)^2 + b^2$$

Motivation of Normalization - *Simplify Optimization Problem*

Ex.1 Using L2 loss to find a straight line fitting points (1, 1), (3, 3)



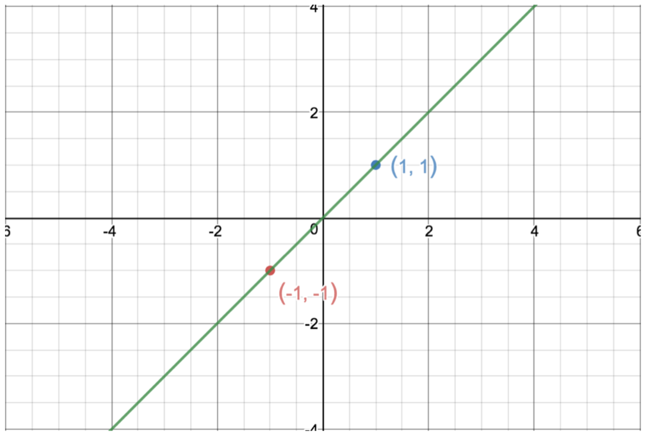
$$\underset{w,b}{\operatorname{argmin}} L(w,b) = \frac{1}{2}\left[(wx_1 + b - y_1)^2 + (wx_2 + b - y_2)^2\right]$$

$$= \frac{1}{2}\left[(w + b - 1)^2 + (2w + b - 3)^2\right]$$

$$\boxed{= \frac{1}{2}(5w^2 + 2b^2 - 8b - 14w + 10)}$$

Complex

Normalize the data points to (-1, -1), (1, 1)



$$\underset{w,b}{\operatorname{argmin}} L(w,b) = \frac{1}{2}\left[(wx_1 + b - y_1)^2 + (wx_2 + b - y_2)^2\right]$$

$$= \frac{1}{2}\left[(w + b - 1)^2 + (-w + b + 1)^2\right]$$

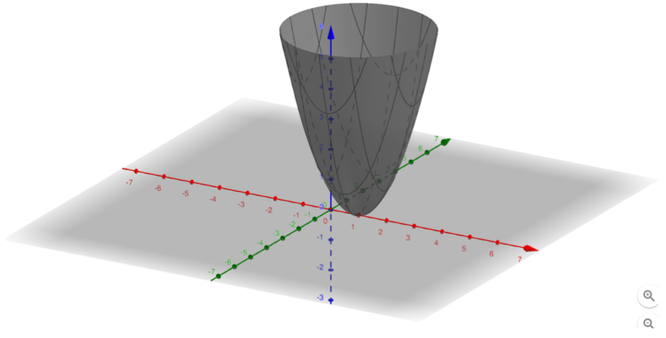
$$\boxed{= (w-1)^2 + b^2}$$

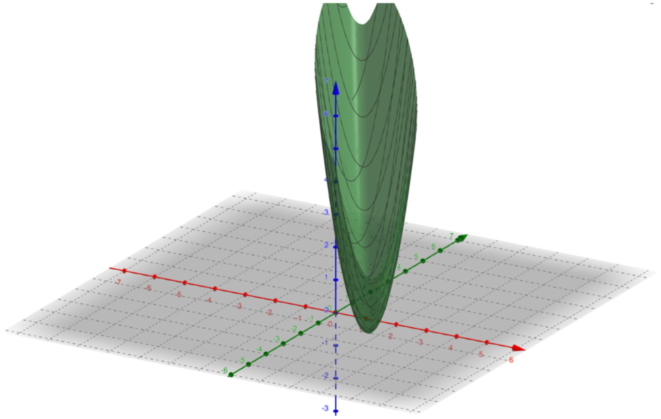
Simple

Motivation of Normalization - *Simplify Optimization Problem*

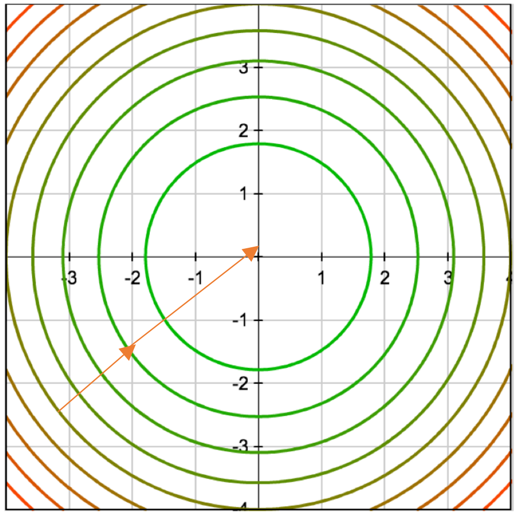
$$f(w,b) = (w-1)^2 + b^2$$

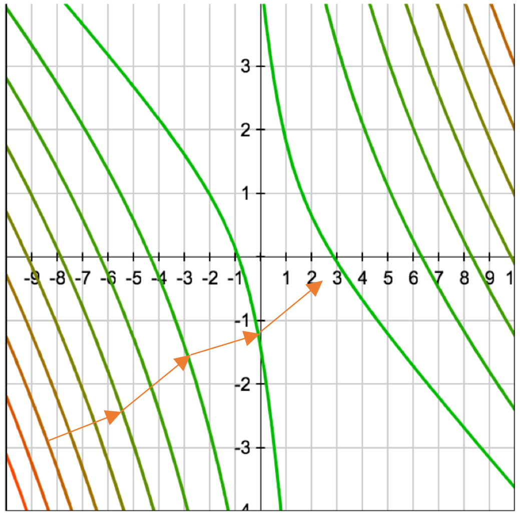
$$f(w,b) = \frac{1}{2}(5w^2 + 2b^2 - 8b - 14w + 10)$$





Surface is simpler :)

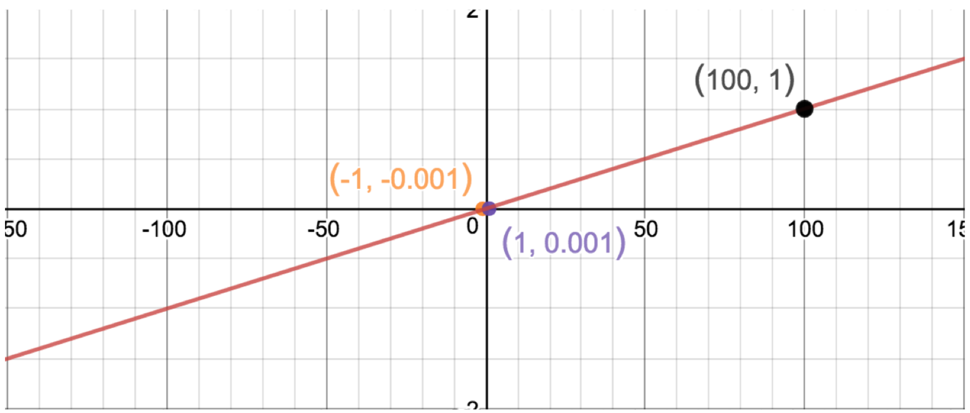




Simplify Optimization Problem

- Stabilize training procedure
- Larger Learning Rate

Motivation of Normalization – *Gradient Explode*



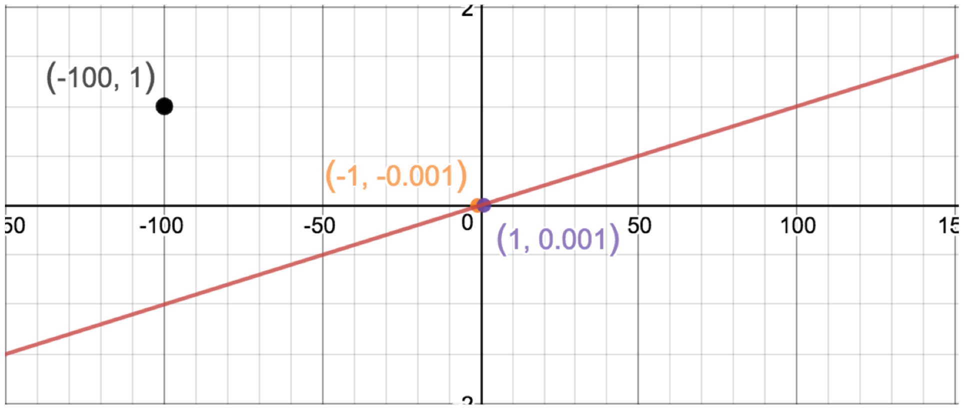
$$\underset{w,b}{\operatorname{argmin}} L(w,b) = \frac{1}{2}[(wx_1 + b - y_1)^2 + (wx_2 + b - y_2)^2]$$

$$\uparrow \frac{\partial L(w,b)}{\partial w} = x_1(wx_1\uparrow + b - y_1) + x_2(wx_2 + b - y_2)$$

$$\uparrow \frac{\partial L(w,b)}{\partial b} = (wx_1\uparrow + b - y_1) + (wx_2 + b - y_2)$$

$$\uparrow w = w - l\frac{\partial L(w,b)}{\partial w} \quad \uparrow b = b - l\frac{\partial L(w,b)}{\partial b}$$

Motivation of Normalization – *Gradient Explode*



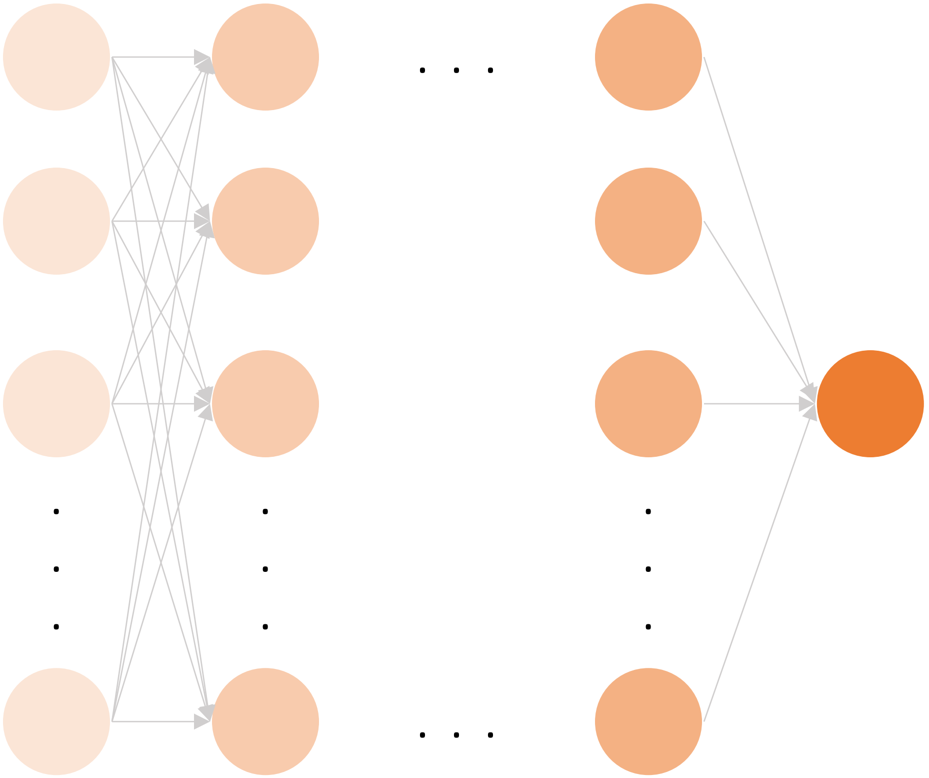
$$\underset{w,b}{\operatorname{argmin}} L(w,b) = \frac{1}{2}[(wx_1 + b - y_1)^2 + (wx_2 + b - y_2)^2]$$

$$\uparrow \frac{\partial L(w,b)}{\partial w} = x_1(wx_1\!\uparrow + b - y_1) + x_2(wx_2 + b - y_2)$$

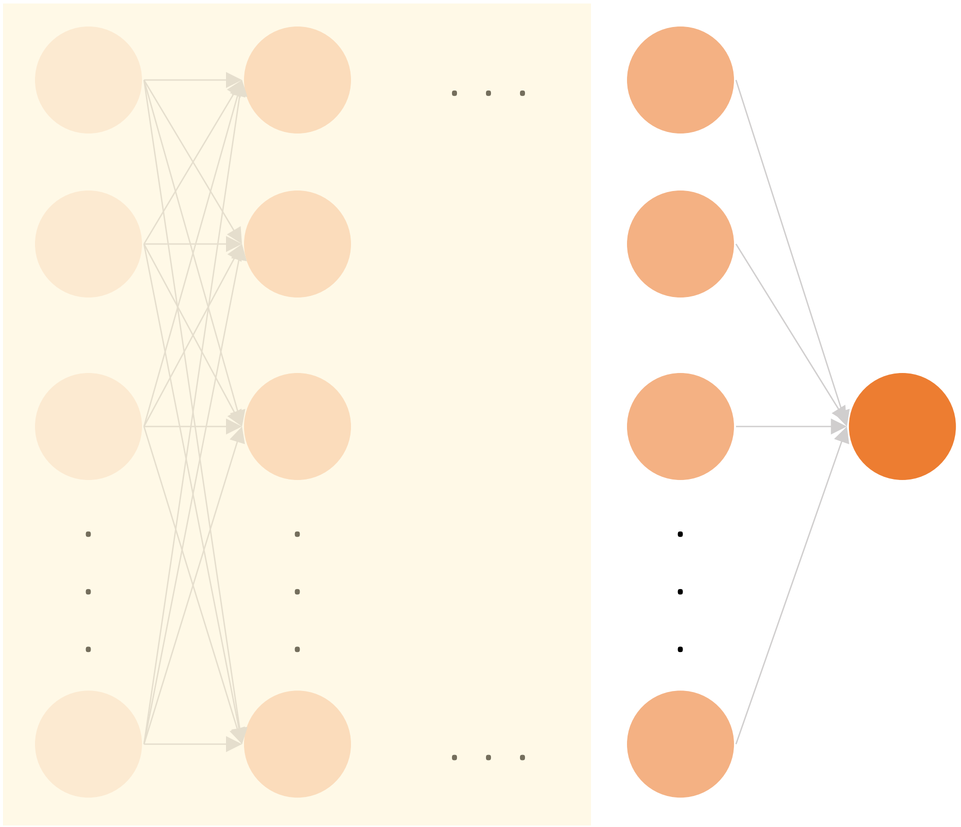
$$\uparrow \frac{\partial L(w,b)}{\partial b} = (wx_1\!\uparrow + b - y_1) + (wx_2 + b - y_2)$$

$$\uparrow w = w - l\frac{\partial L(w,b)}{\partial w} \quad \uparrow b = b - l\frac{\partial L(w,b)}{\partial b}$$

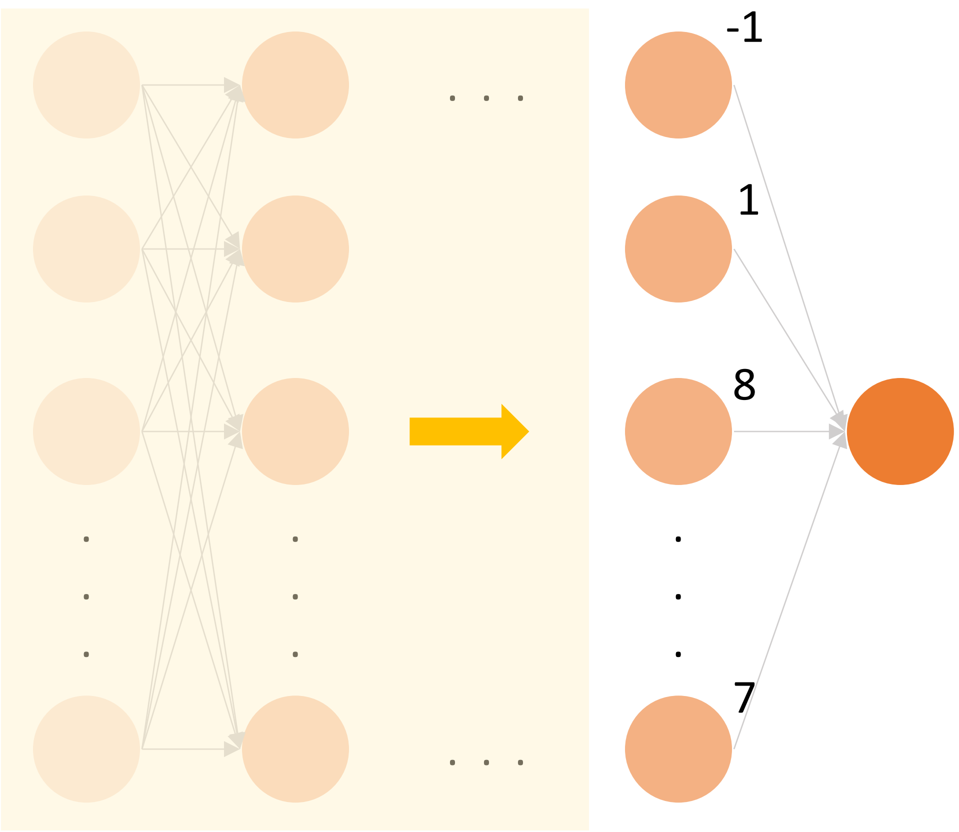
Motivation of Batch Normalization - *Covariance Shift*

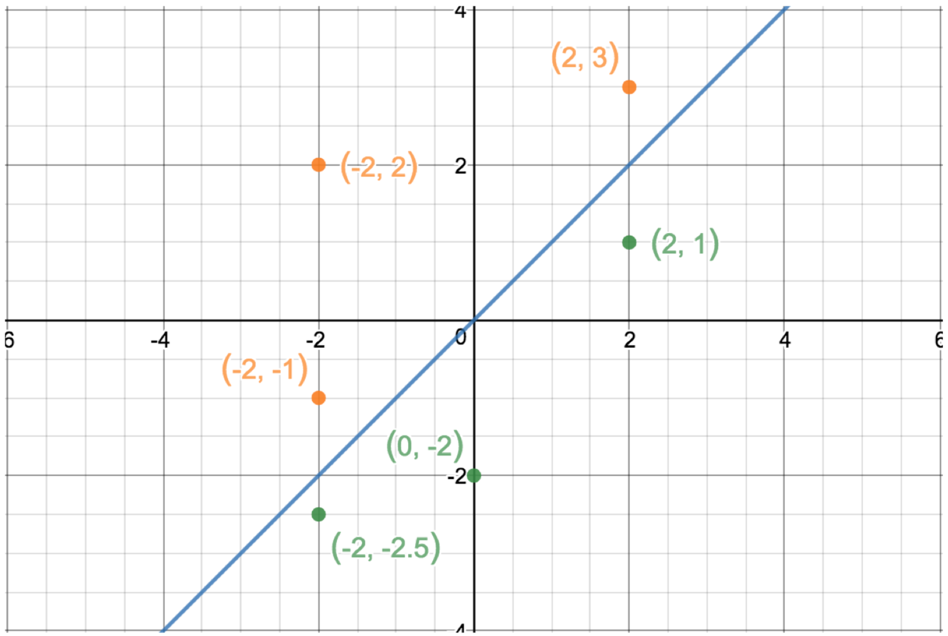


Motivation of Batch Normalization - *Covariance Shift*

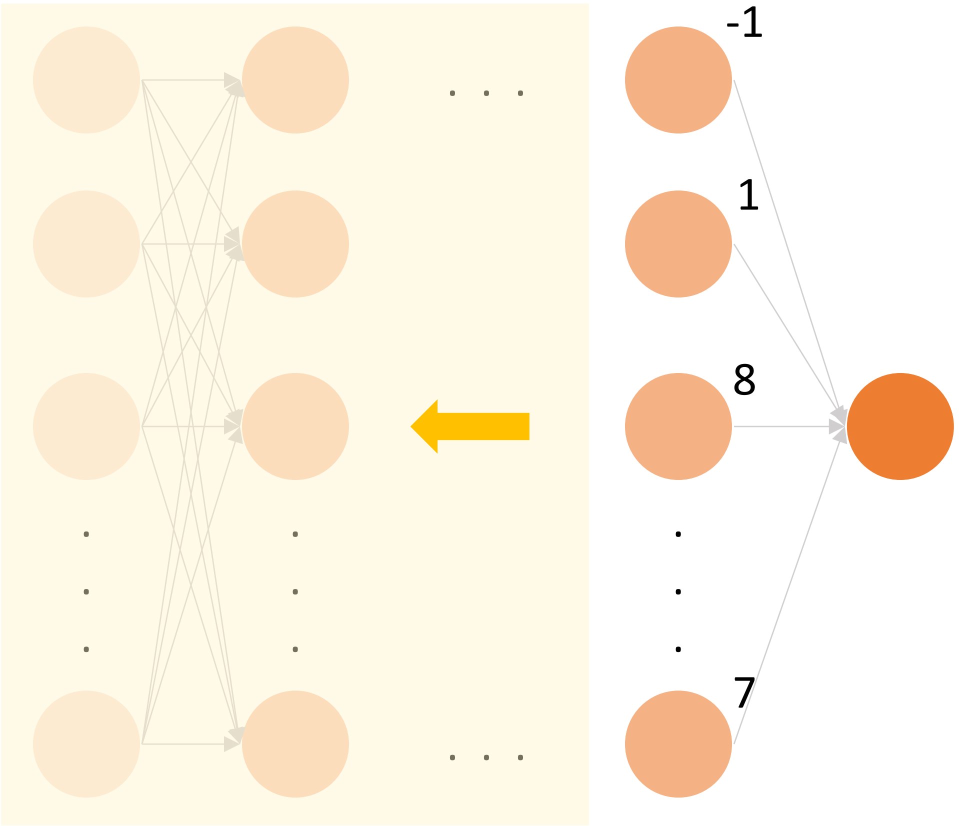


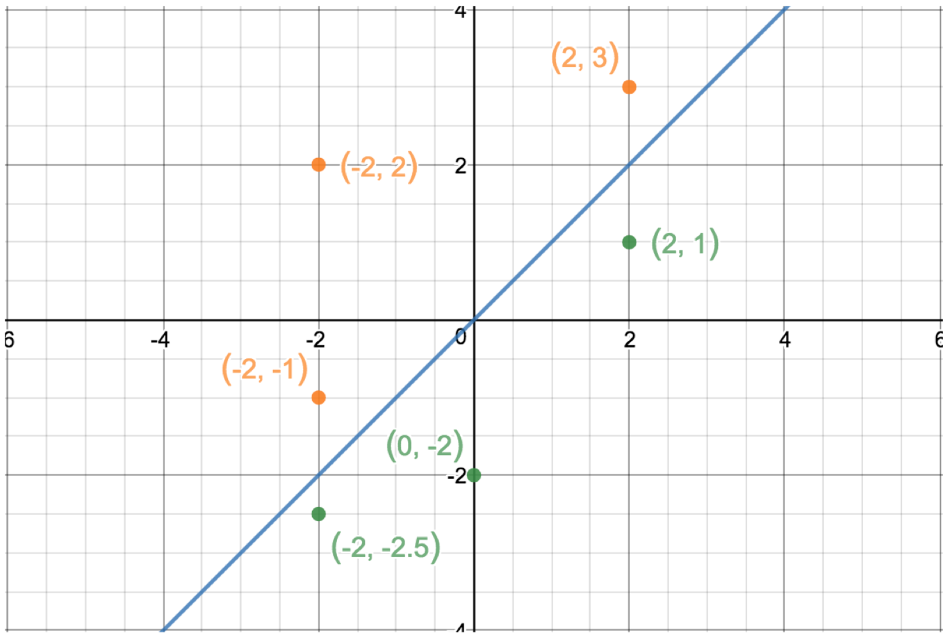
Motivation of Batch Normalization - *Covariance Shift*



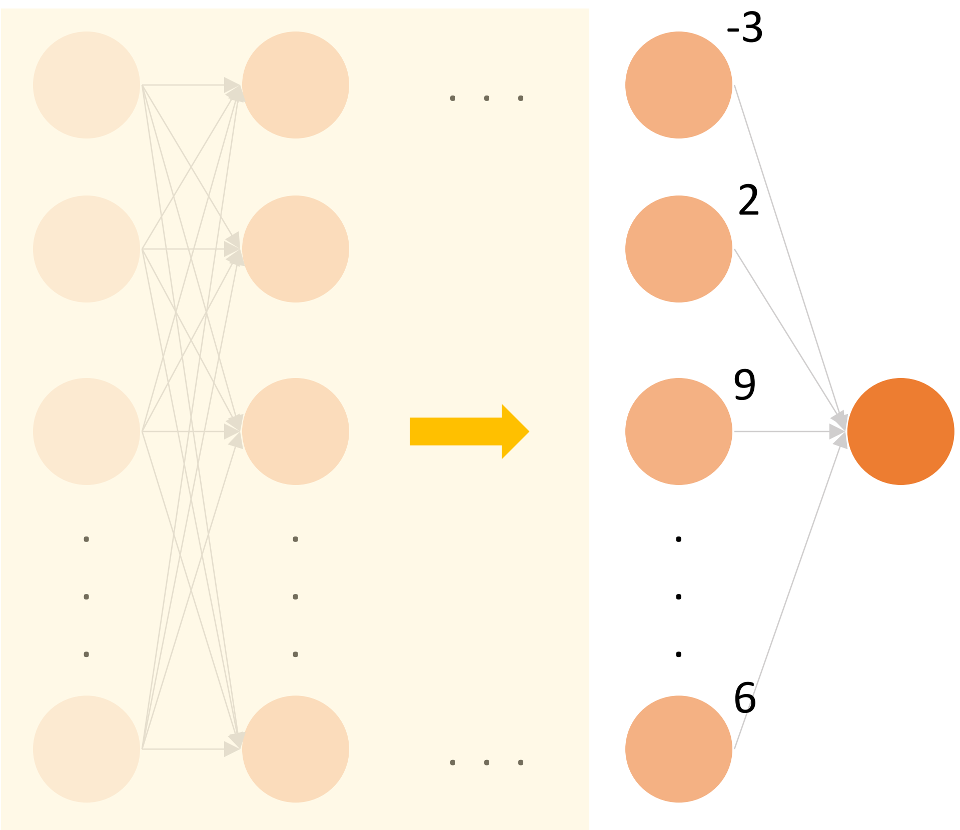


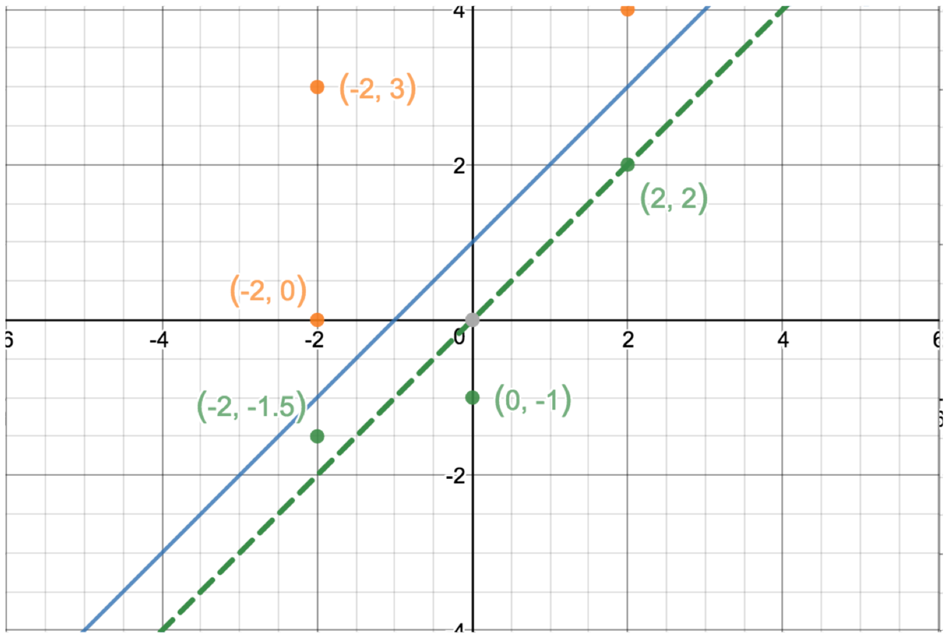
Motivation of Batch Normalization - *Covariance Shift*



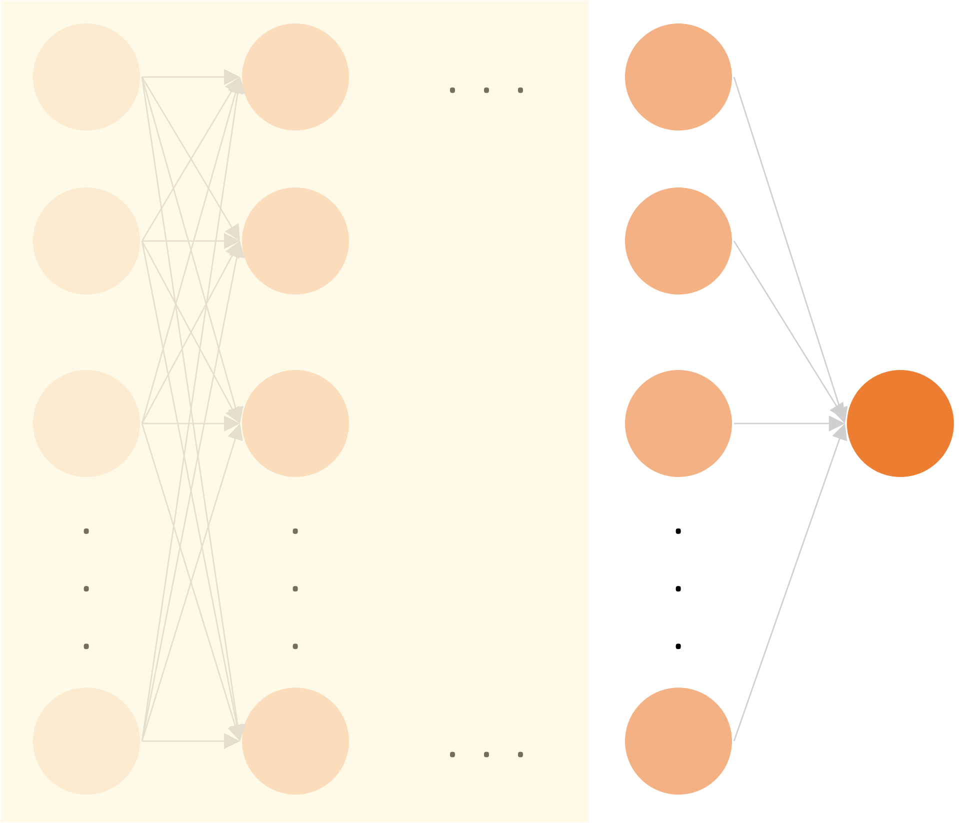


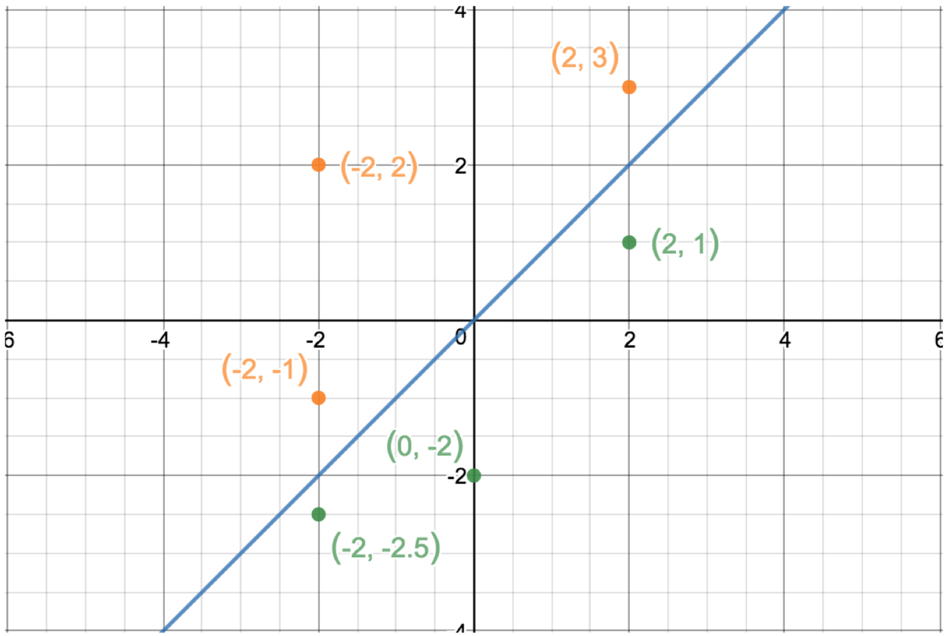
Motivation of Batch Normalization - *Covariance Shift*





Motivation of Batch Normalization - *Covariance Shift*





Batch Normalization - *Training Phase*

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

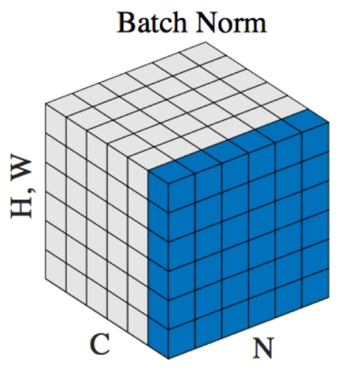
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

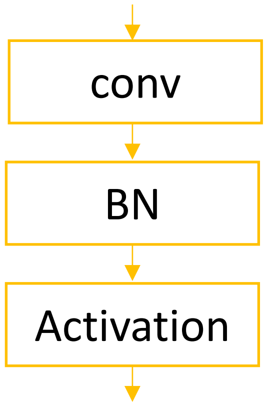
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$





Batch Normalization - *Training Phase*

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

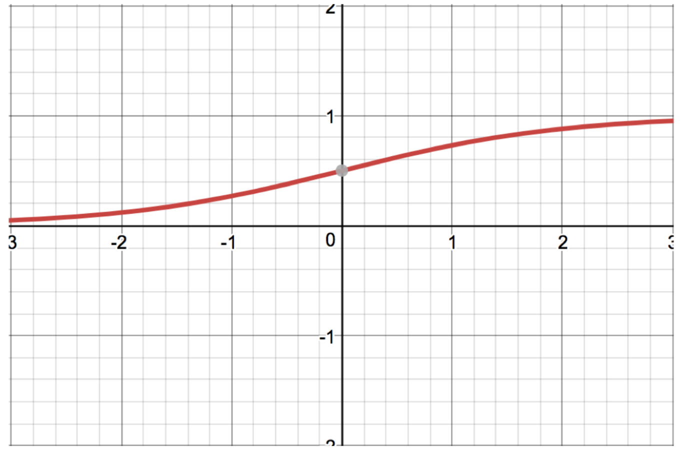
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Rescale γ



Rescale β

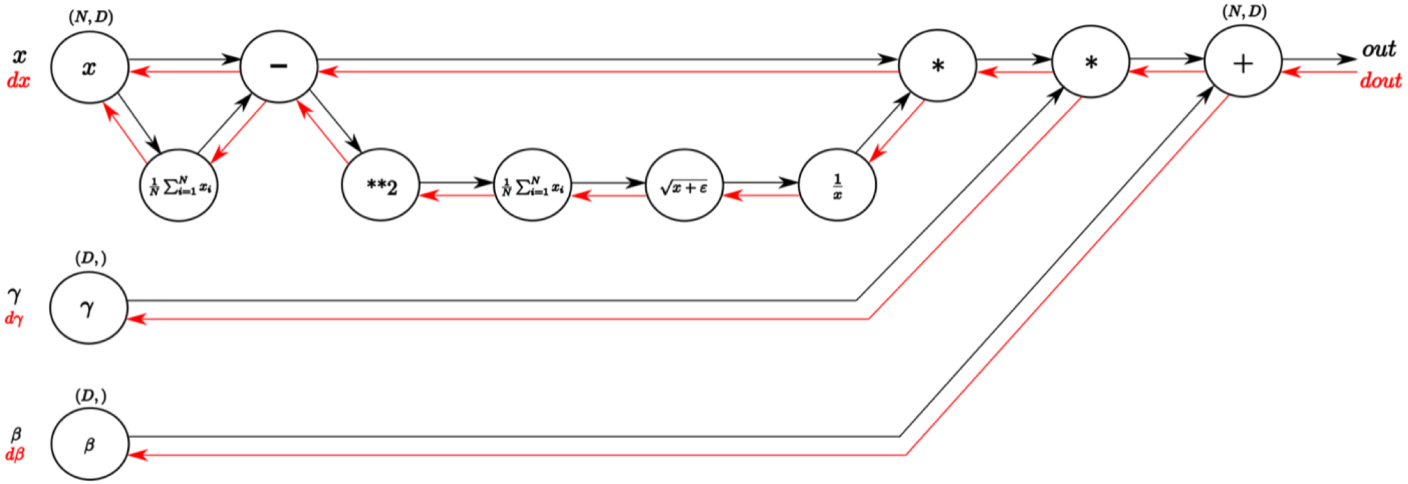
$$(kx_1 + b), (kx_2 + b\ )$$

$$\mu = \frac{1}{2}(kx_1 + b + kx_2 + b)$$

$$\left(\frac{1}{2}kx_1 - \frac{1}{2}kx_2\right), \left(\frac{1}{2}kx_2 - \frac{1}{2}kx_1\right)$$

Batch Normalization - *Backward*

$$
\begin{aligned}
\frac{\partial \ell}{\partial \widehat{x}_i} &= \frac{\partial \ell}{\partial y_i} \cdot \gamma \\
\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2} (\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2} \\
\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} &= \left(\sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\mathcal{B}})}{m} \\
\frac{\partial \ell}{\partial x_i} &= \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m} \\
\frac{\partial \ell}{\partial \gamma} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \cdot \widehat{x}_i \\
\frac{\partial \ell}{\partial \beta} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i}
\end{aligned}
$$



https://kratzert.github.io/2016/02/12/understanding-the-gradient-flow-through-the-batch-normalization-layer.html

Batch Normalization – *Testing Phase*

Input: Network N with trainable parameters Θ; subset of activations $\{x^{(k)}\}_{k=1}^{K}$

for $k = 1 \dots K$ **do**

// For clarity, $x \equiv x^{(k)}$, $\gamma \equiv \gamma^{(k)}$, $\mu_\mathcal{B} \equiv \mu_\mathcal{B}^{(k)}$, etc.
Process multiple training mini-batches $\mathcal{B}$, each of size m, and average over them:

$$\mathrm{E}[x] \leftarrow \mathrm{E}_\mathcal{B}[\mu_\mathcal{B}]$$
$$\mathrm{Var}[x] \leftarrow \frac{m}{m-1}\mathrm{E}_\mathcal{B}[\sigma_\mathcal{B}^2]$$

In $N_{\mathrm{BN}}^{\mathrm{inf}}$, replace the transform $y = \mathrm{BN}_{\gamma,\beta}(x)$ with $y = \frac{\gamma}{\sqrt{\mathrm{Var}[x]+\epsilon}} \cdot x + \left(\beta - \frac{\gamma\,\mathrm{E}[x]}{\sqrt{\mathrm{Var}[x]+\epsilon}}\right)$

end for

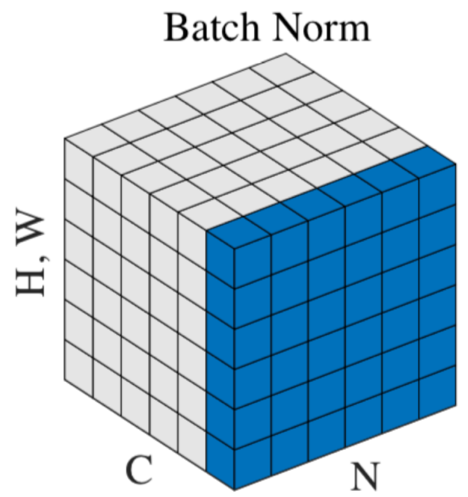
In Practice

```
# BatchNorm training forward propagation
h2, bn2_cache, mu, var = batchnorm_forward(h2, gamma2, beta2)
bn_params['bn2_mean'] = .9 * bn_params['bn2_mean'] + .1 * mu
bn_params['bn2_var'] = .9 * bn_params['bn2_var'] + .1 * var
```

Set for Batch Norm

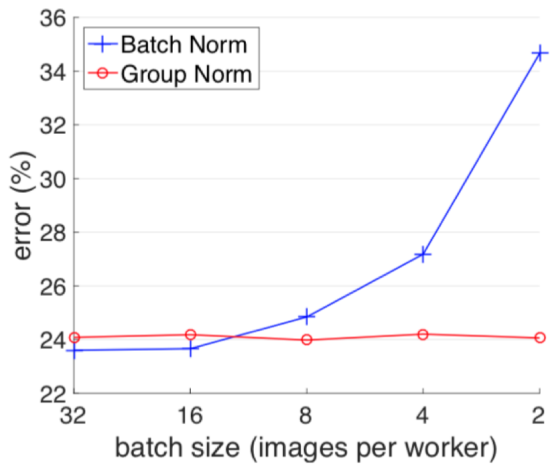
(3) $\mathcal{S}_i = \{k \mid k_C = i_C\},$

one channel(filter), all samples(images)



Problem with Small Batches

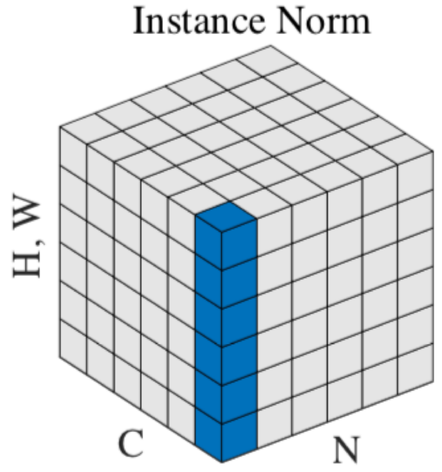
- have to use small batches due to memory limit
- high error rate using small batches
- calc μ & σ from small batches
- norm is not independent from batch axis



Instance Norm

(5) $\mathcal{S}_i = \{k \mid k_N = i_N, k_C = i_C\}.$

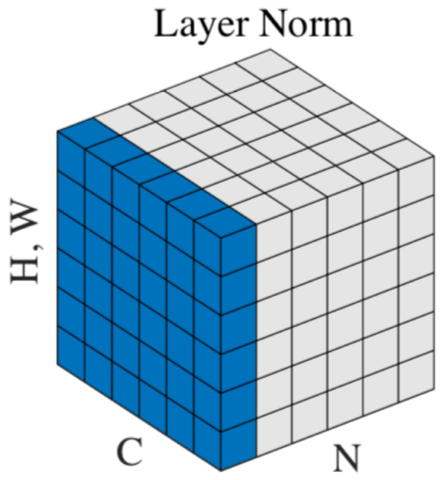
- one sample(image), one channel(filter)
- used in style transfer tasks
- do not use the relation of channels



Layer Norm

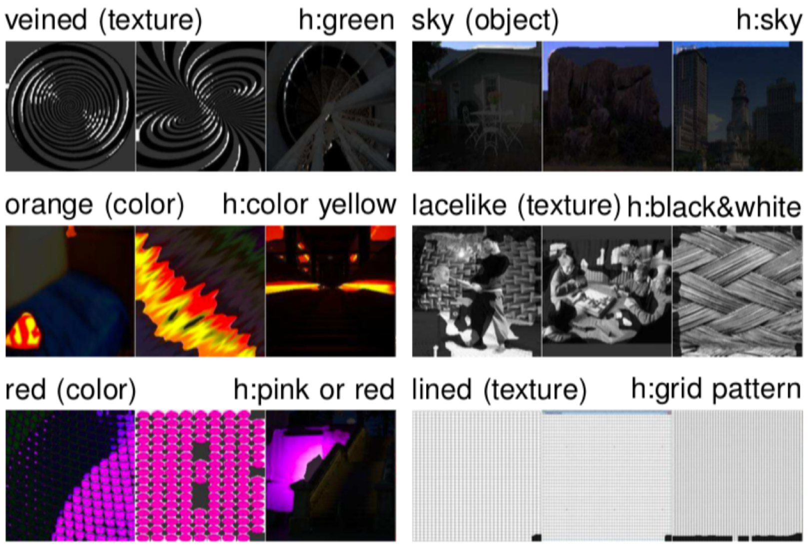
(4) $\mathcal{S}_i = \{k \mid k_N = i_N\},$

- one image, all channels
- used in RNNs and GANs
- assume that all channels make similar contributions
- calc μ & σ among all channels doesn't make sense



Group Norm

- channels have factors like color, texture, object, shape, etc.
- independent among factors
- interrelated within same factor
- divide channels into groups
- group with different channels has higher loss
- guide the network to put similar channels together

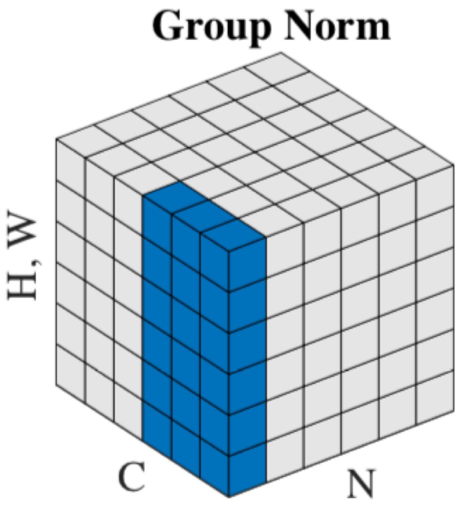


(7) $$\mathcal{S}_i = \{k \mid k_N = i_N, \lfloor \frac{k_C}{C/G} \rfloor = \lfloor \frac{i_C}{C/G} \rfloor \}.$$

- one image, one group

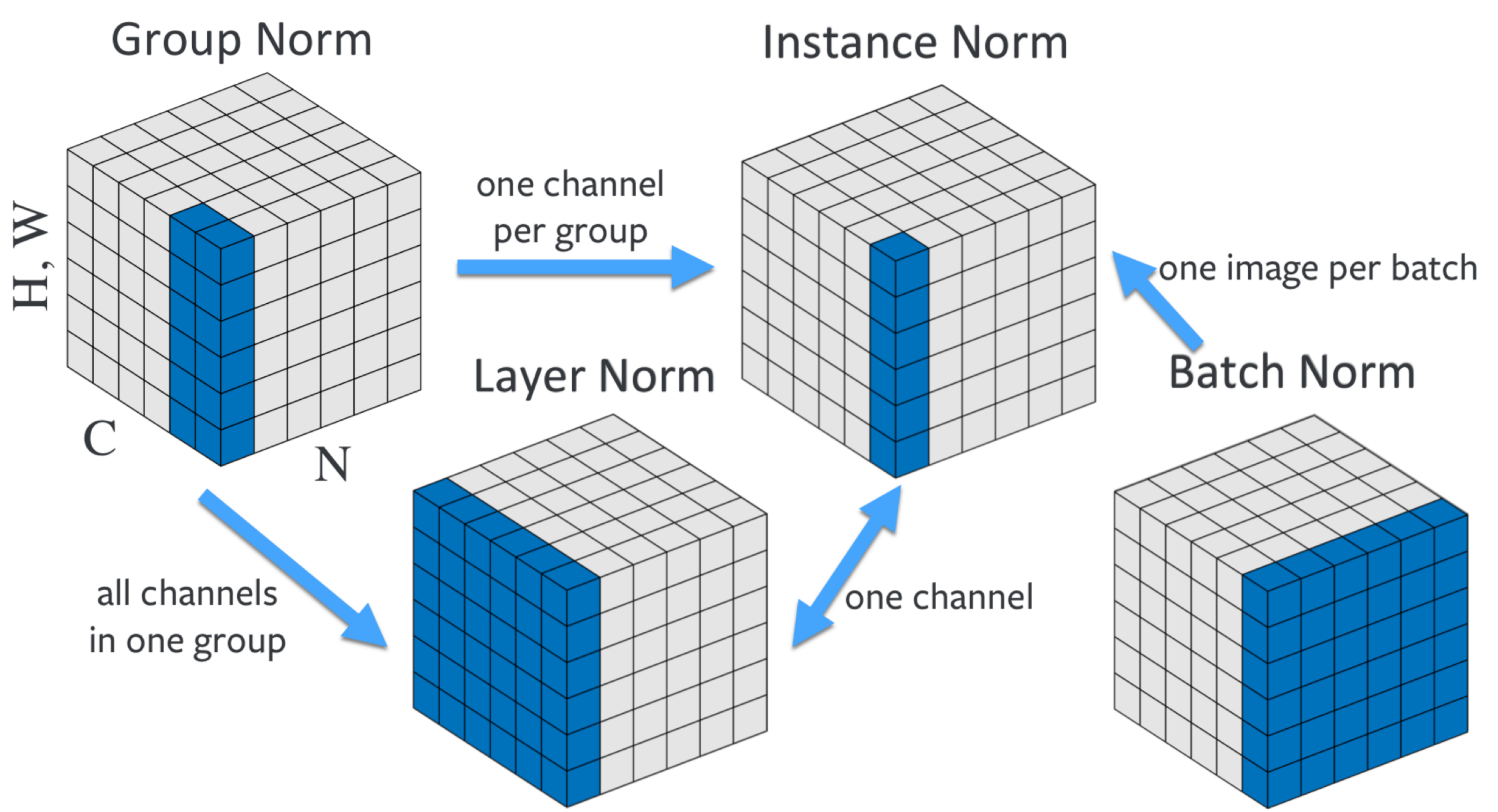
number of groups
pre-defined hyper-parameter

number of channels per group
groups are stored sequentially along the C axis



https://arxiv.org/abs/1704.05796 Network Dissection: Quantifying Interpretability of Deep Visual Representations

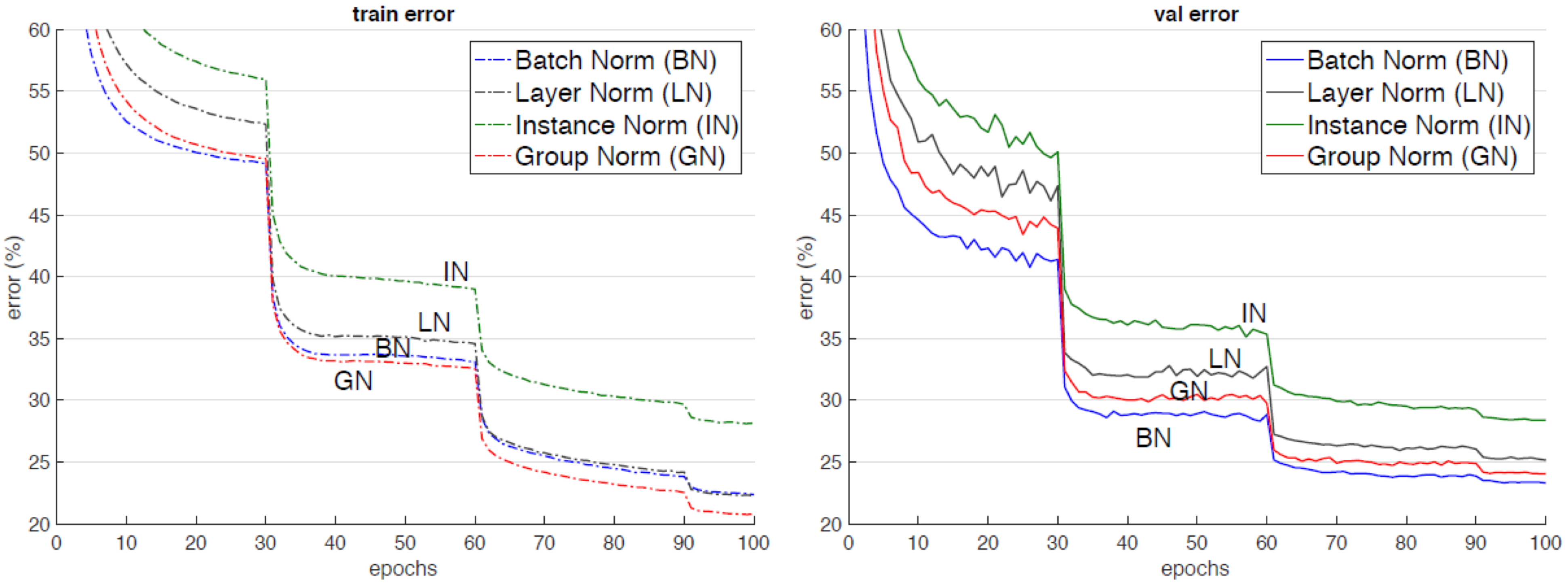
Relationships between Normalization Methods



http://kaiminghe.com/eccv18gn/group_norm_yuxinwu.pdf

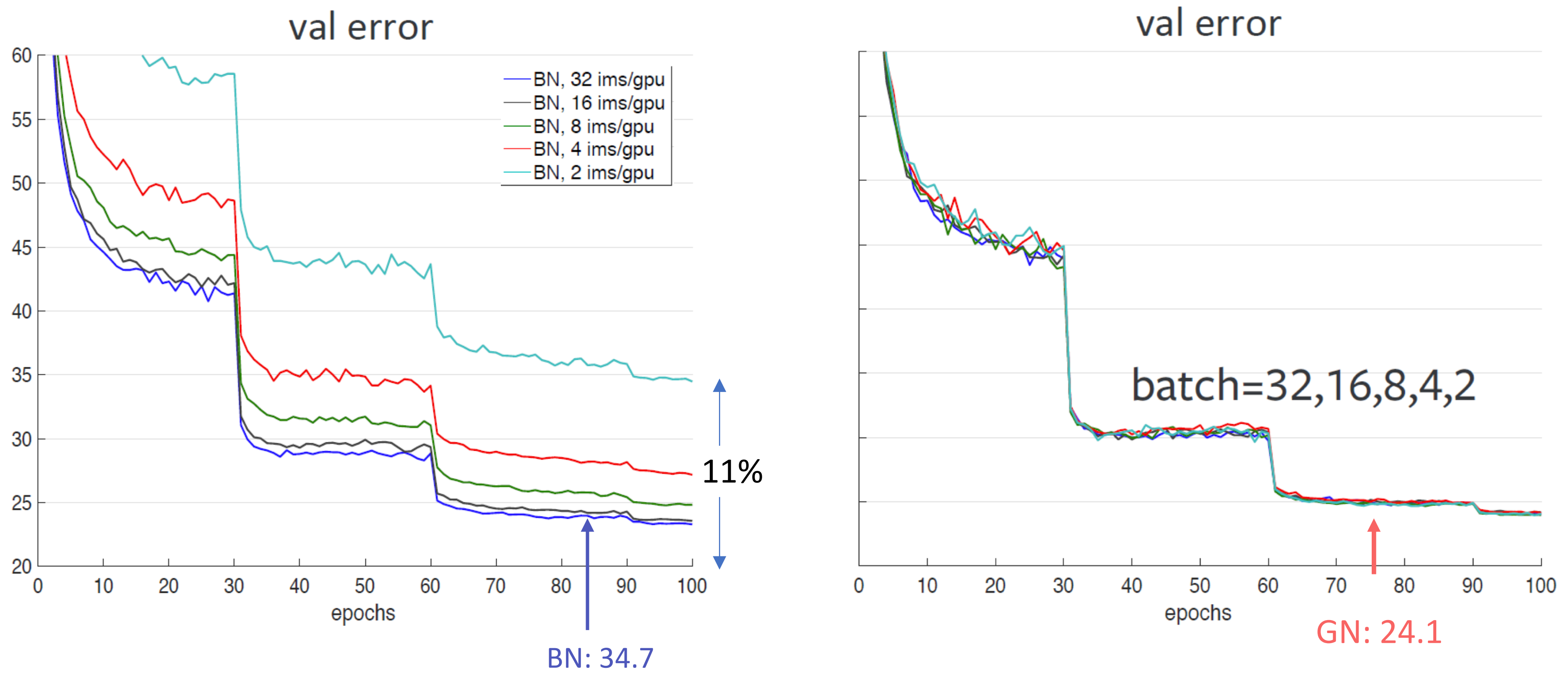
Imagenet Classification

Batch size = 32
Model : ResNet-50



Imagenet Classification

Batch size = 32
Model : ResNet-50



Imagenet Classification

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**
$\triangle$	*0.5*	*0.5*	*-0.8*	*-3.1*	*-10.6*

Imagenet Classification

ResNet-50's validation error (%)

Group Number: 32

# 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*

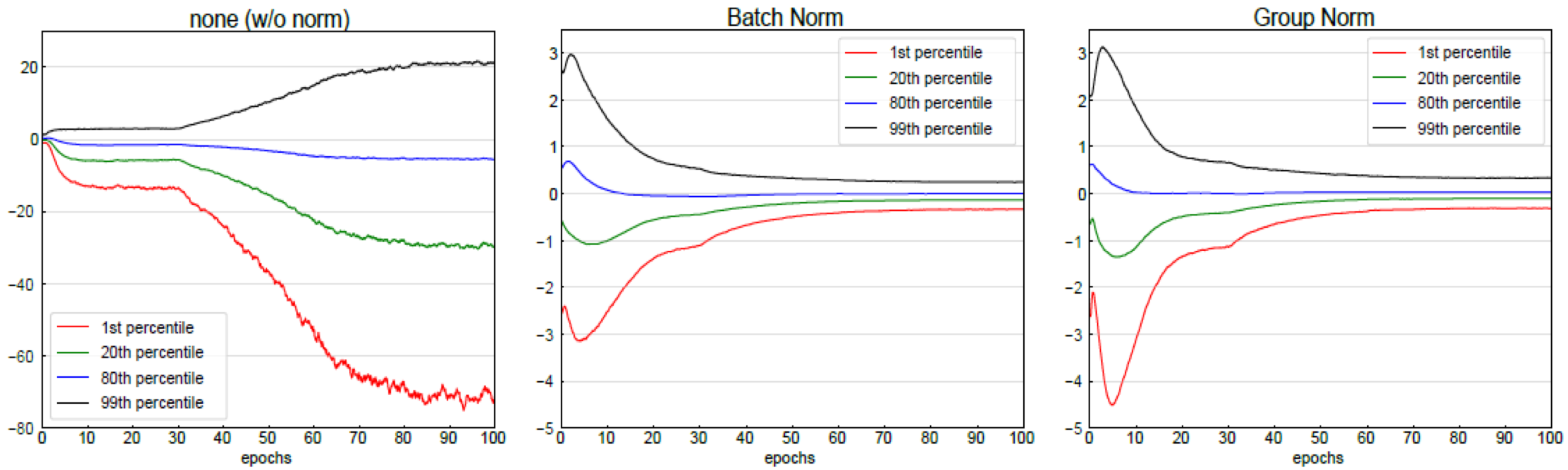
fixing the number of channels per group

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*

fixing the group number

Imagenet Classification

Model : VGG-19



	err.
none	29.2
BN	28.0
GN	**27.6**

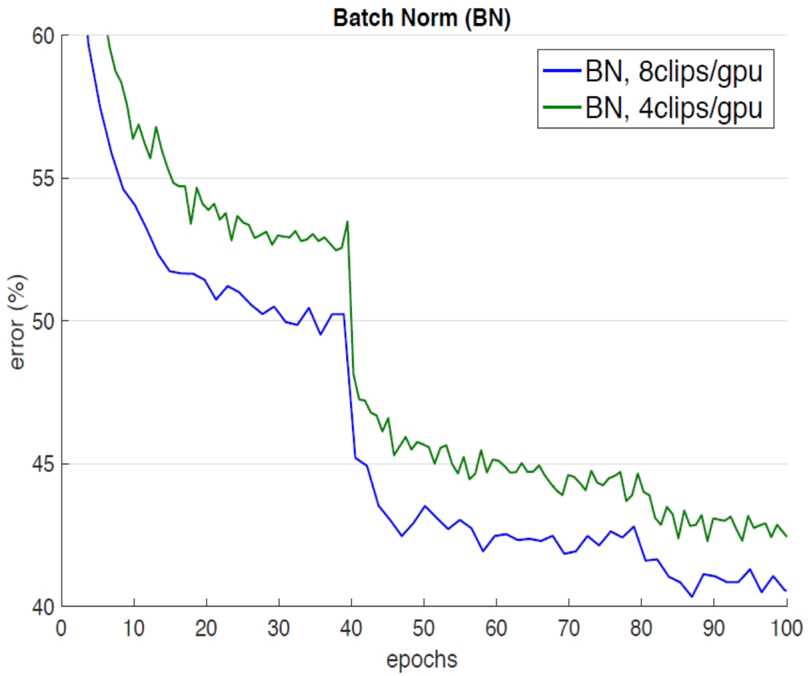
Object Detection and Segmentation in COCO

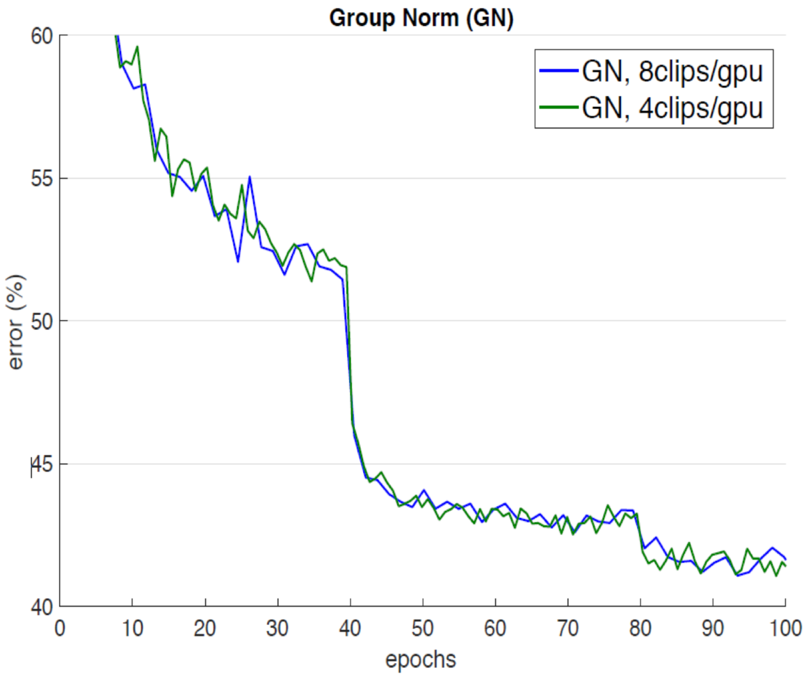
Method: Mask R-CNN
Model: ResNet-50

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**

Video Classification in Kinetics

Model: ResNet-50 I3D





Conclusion

- Normalization is an effective component in deep learning
- Batch is not always ideal
- channels can be grouped
- GN is a strong alternative of BN

Slides: https://github.com/MaureenZOU/ECS269_presentation/tree/master

Contact: zxyzou@ucdavis.edu
lzylin@ucdavis.edu
tirwang@ucdavis.edu