Normalization

CoE197Z/EE298Z (Deep Learning)

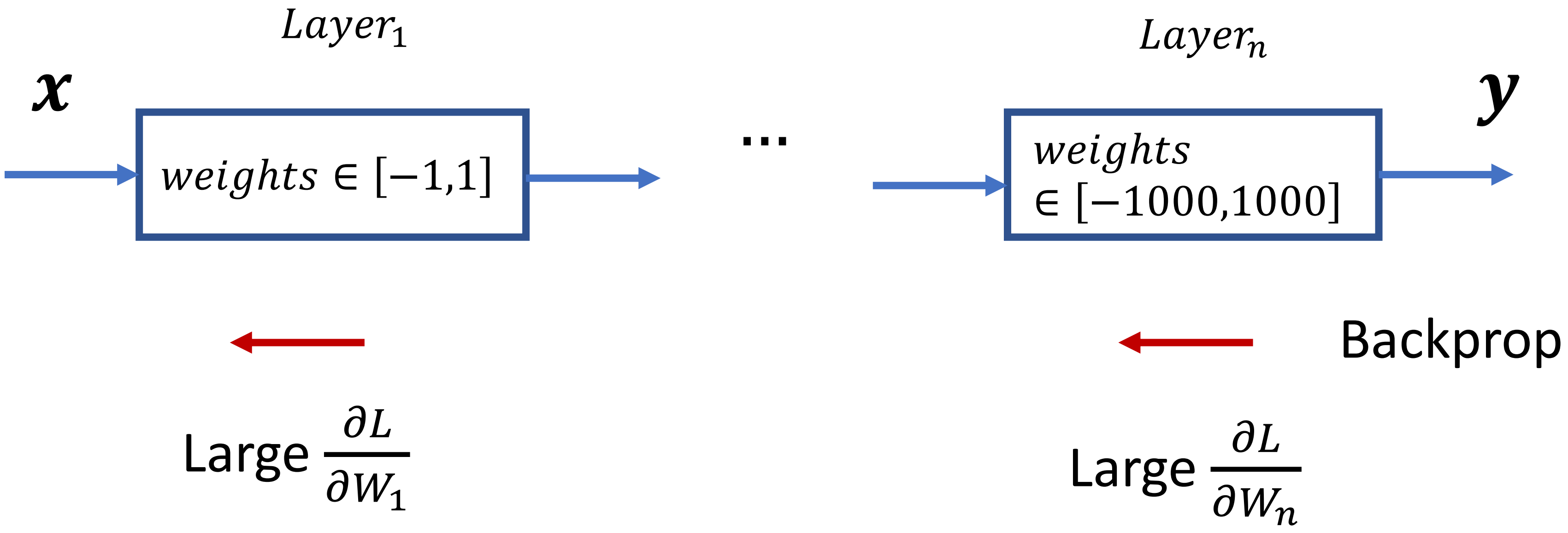
Rowel Atienza, Ph.D.

rowel@eee.upd.edu.ph

github.com/roatienza

Basic Idea

What if you have layers with different ranges of parameters?



End result: Unstable training

Notations

Batch size: N

Feature Map Width: W

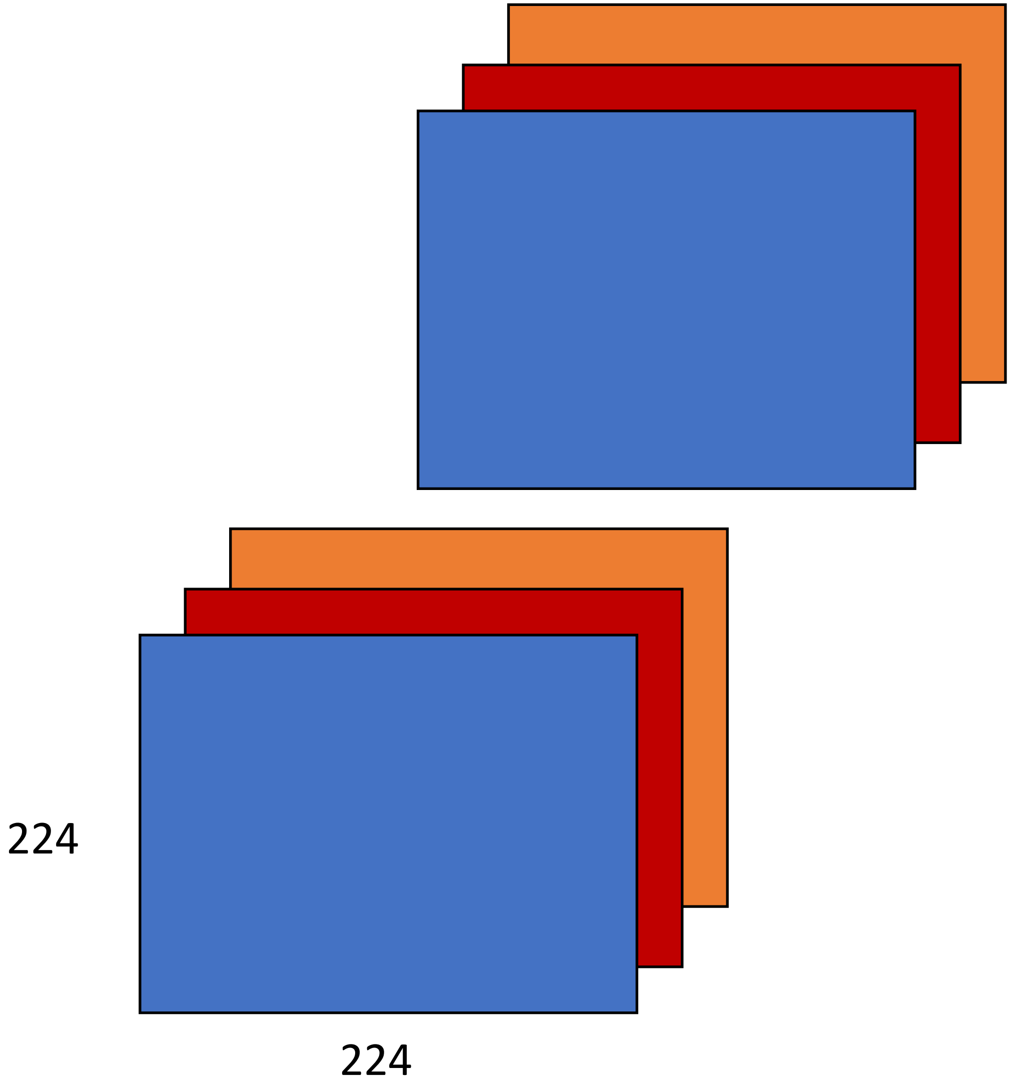
Feature Map Height: H

Feature Map Channels: C

Feature Map: $\boldsymbol{x}_{nchw}$

Example: $\boldsymbol{x}_{nchw}$

$$n = 1 \dots N = 2,$$
$$c = 1 \dots C = 3,$$
$$c = 1 \dots W = 224,$$
$$h = 1 \dots H = 224$$



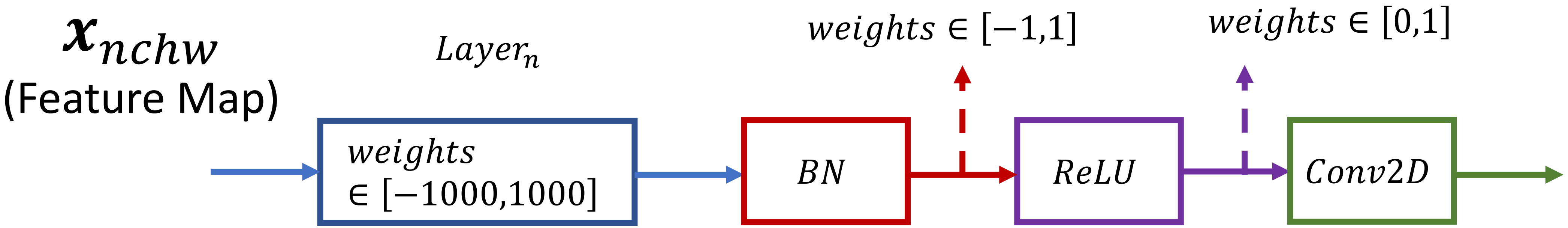
Batch Normalization (BN)

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

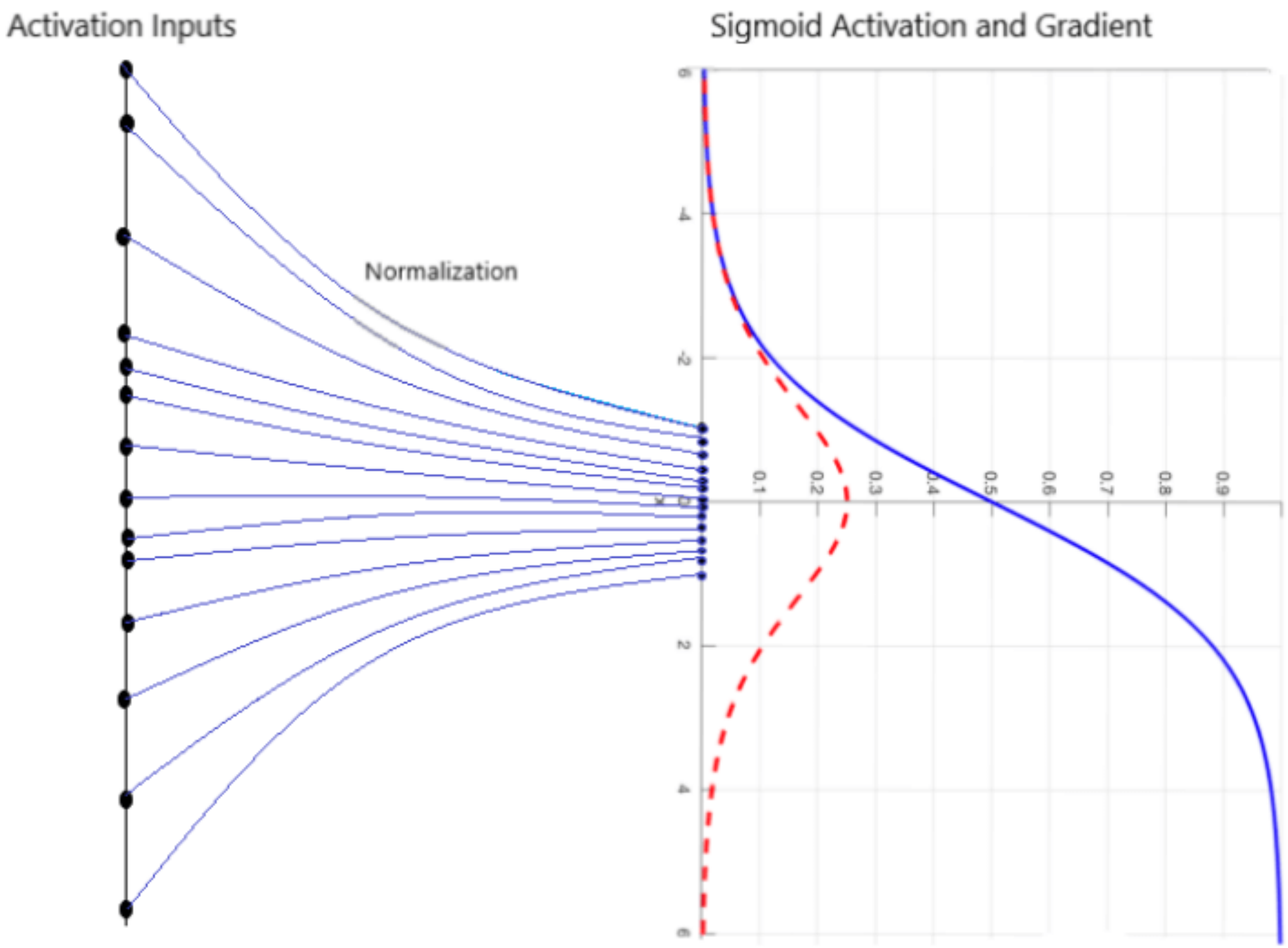
Batch Normalization (BN)

Batch Normalization normalizes the mean and standard deviation for each individual feature channel/map

Useful in deep networks (eg ResNet)



BN is a mapping

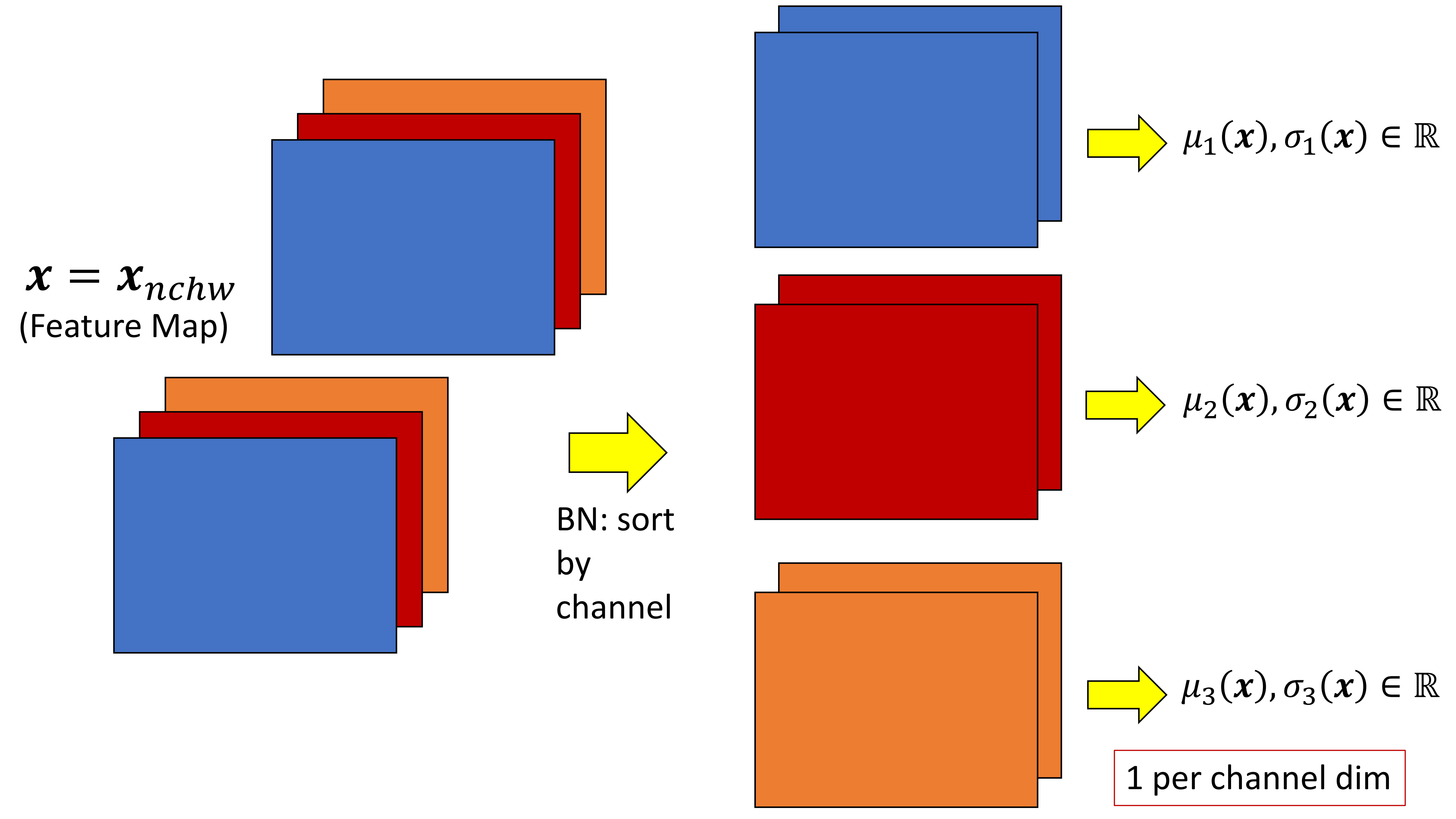


BN: The statistics are computed across the batch and the spatial dims

$$BN(\boldsymbol{x}_{nchw}) = \left(\frac{\boldsymbol{x}_{nchw} - \mu_c(\boldsymbol{x}_{nchw})}{\sigma_c(\boldsymbol{x}_{nchw})}\right)$$

$$\mu_c(\boldsymbol{x}_{nchw}) = \frac{1}{NHW}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{x}_{nchw}$$

$$\sigma_c(\boldsymbol{x}_{nchw}) = \sqrt{\frac{1}{NHW}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\boldsymbol{x}_{nchw} - \mu_c(\boldsymbol{x}_{nchw})\right)^2}$$



$\boldsymbol{x} = \boldsymbol{x}_{nchw}$
(Feature Map)
BN: sort by channel
$\mu_1(\boldsymbol{x}), \sigma_1(\boldsymbol{x}) \in \mathbb{R}$
$\mu_2(\boldsymbol{x}), \sigma_2(\boldsymbol{x}) \in \mathbb{R}$
$\mu_3(\boldsymbol{x}), \sigma_3(\boldsymbol{x}) \in \mathbb{R}$
1 per channel dim

Benefits of BN

BN accelerates the training of deep neural networks

Implicit regularization

Enhances gradient flow

Now ok to use of saturating non-linear functions (eg sigmoid or tanh)

Danger of BN

Small batch sizes leads to inaccurate batch statistics

Problematic with variable batch size

https://youtu.be/ZOabsYbmBRM : BN was thought to prevent internal covariate shift

Layer Normalization (LN)

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450 (2016).

Layer Normalization

The statistics are computed across all channels and spatial dims

The statistics are independent of the batch

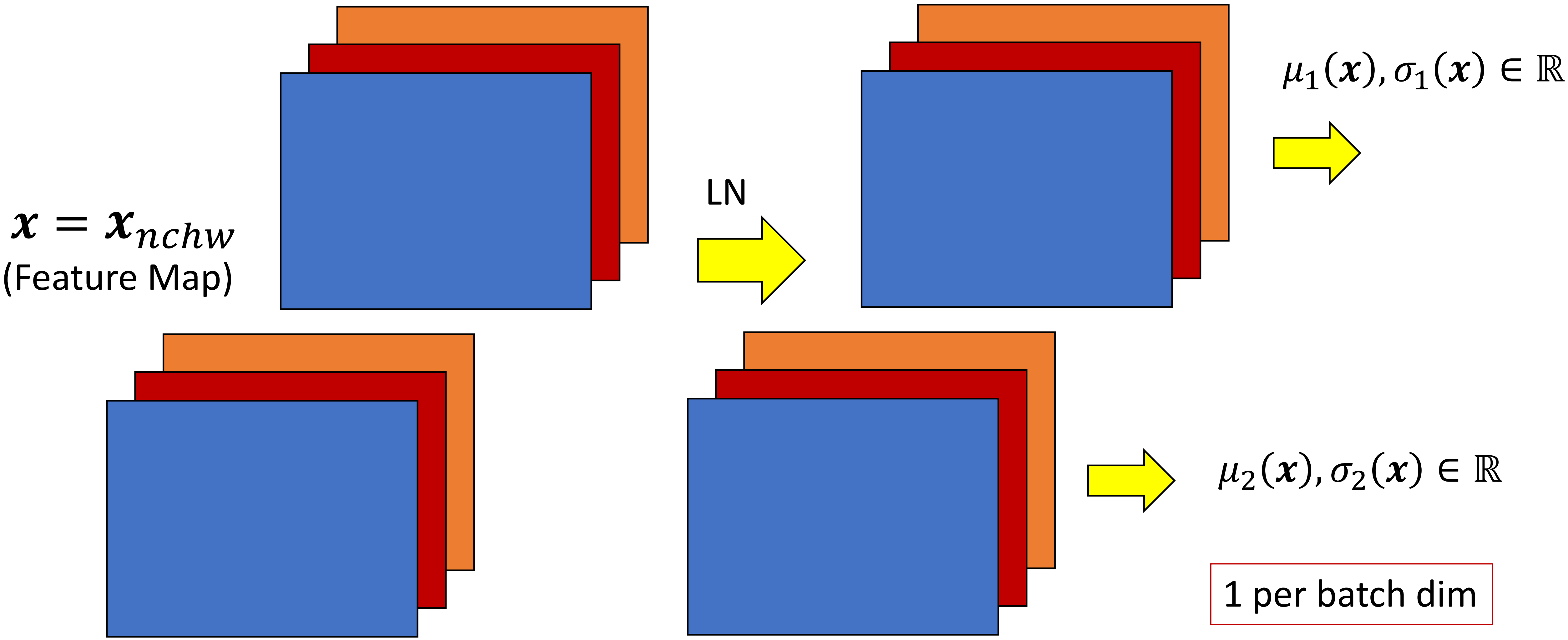
Useful in transformers

LN: The statistics are computed across all channels and the spatial dims

$$LN(\boldsymbol{x}_{nchw}) = \left(\frac{\boldsymbol{x}_{nchw} - \mu_n(\boldsymbol{x}_{nchw})}{\sigma_n(\boldsymbol{x}_{nchw})}\right)$$

$$\mu_n(\boldsymbol{x}_{nchw}) = \frac{1}{CHW}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{x}_{nchw}$$

$$\sigma_n(\boldsymbol{x}_{nchw}) = \sqrt{\frac{1}{CHW}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\boldsymbol{x}_{nchw} - \mu_n(\boldsymbol{x}_{nchw})\right)^2}$$



$\boldsymbol{x} = \boldsymbol{x}_{nchw}$
(Feature Map)
LN
$\mu_1(\boldsymbol{x}), \sigma_1(\boldsymbol{x}) \in \mathbb{R}$
$\mu_2(\boldsymbol{x}), \sigma_2(\boldsymbol{x}) \in \mathbb{R}$
1 per batch dim

Instance Normalization (IN)

Instance Normalization (IN)

Instance Normalization (IN) is computed only across the features' spatial dimensions

It is independent for each channel and sample

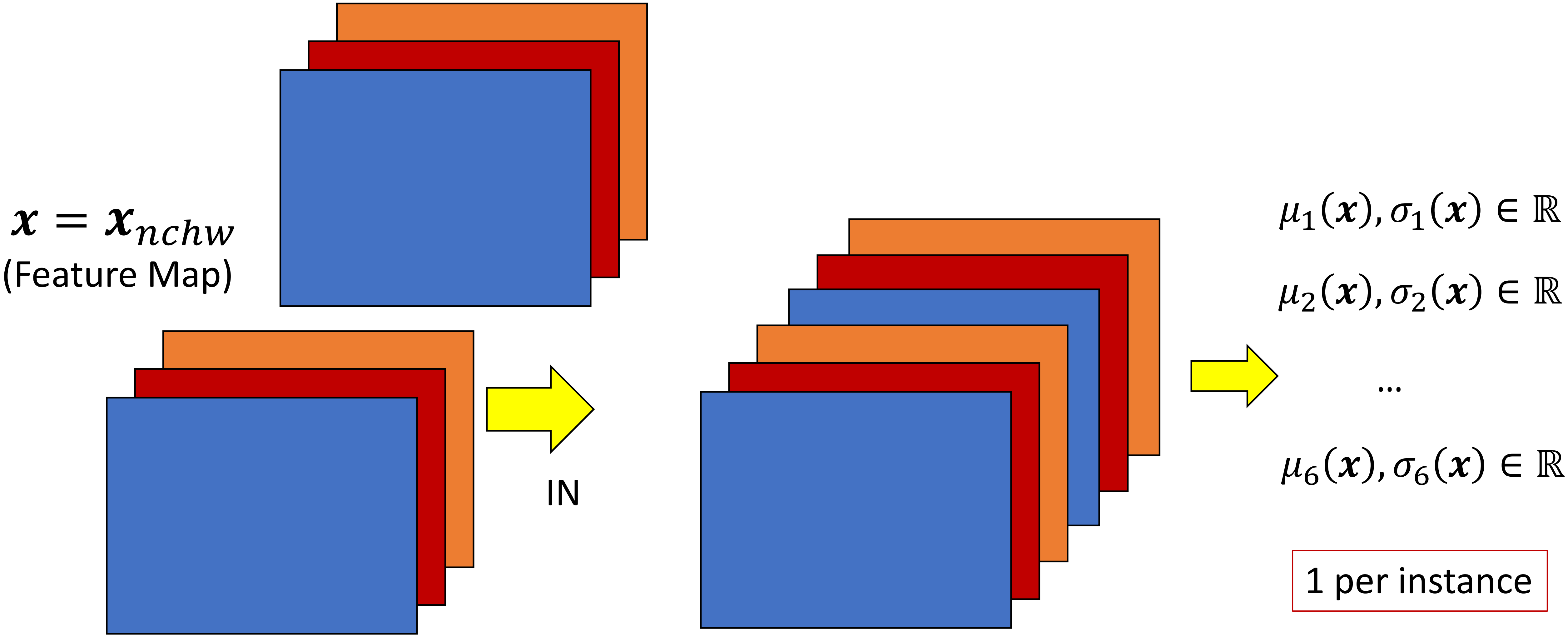
Useful in style transfer

IN: The statistics are computed across spatial dims of each feature map

$$IN(\boldsymbol{x}_{nchw}) = \left(\frac{\boldsymbol{x}_{nchw} - \mu_{nc}(\boldsymbol{x}_{nchw})}{\sigma_{nc}(\boldsymbol{x}_{nchw})}\right)$$

$$\mu_{nc}(\boldsymbol{x}_{nchw}) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{x}_{nchw}$$

$$\sigma_{nc}(\boldsymbol{x}_{nchw}) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\boldsymbol{x}_{nchw} - \mu_{nc}(\boldsymbol{x}_{nchw})\right)^2}$$

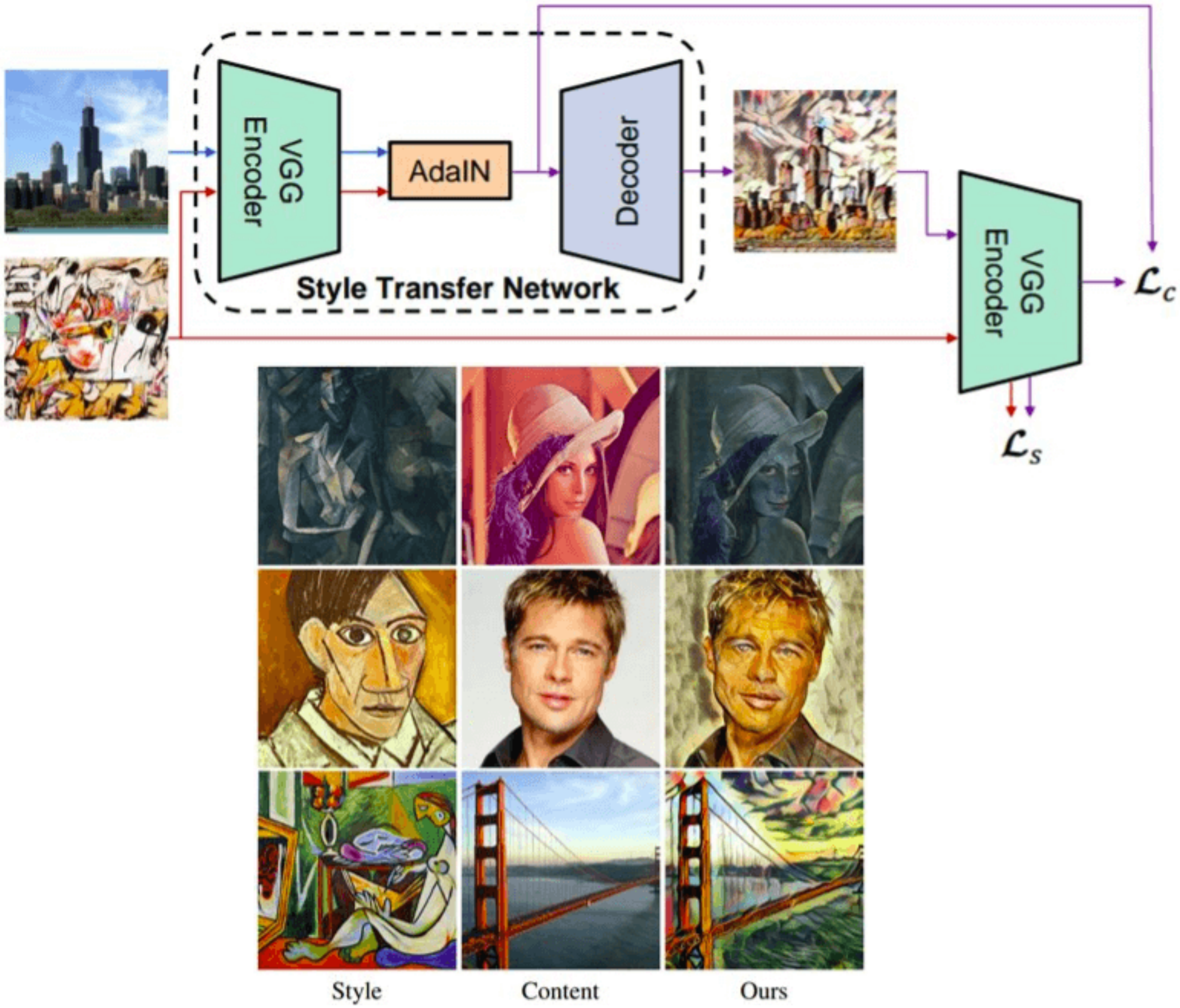

$\boldsymbol{x} = \boldsymbol{x}_{nchw}$
(Feature Map)
IN
$\mu_1(\boldsymbol{x}), \sigma_1(\boldsymbol{x}) \in \mathbb{R}$
$\mu_2(\boldsymbol{x}), \sigma_2(\boldsymbol{x}) \in \mathbb{R}$
…
$\mu_6(\boldsymbol{x}), \sigma_6(\boldsymbol{x}) \in \mathbb{R}$
1 per instance

Adaptive Instance Normalization (AIN)

Adaptive Instance Normalization (AIN)

Adaptive Instance Normalization (AdaIN) receives an input image $\boldsymbol{x}$ (content) and a style input $\boldsymbol{y}$, and simply aligns the channel-wise mean and variance of $\boldsymbol{x}$ to match those of $\boldsymbol{y}$.

$$AIN(\boldsymbol{x}_{nchw}) = \sigma_{nc}(\boldsymbol{y}_{nchw})\left(\frac{\boldsymbol{x}_{nchw} - \mu_{nc}(\boldsymbol{x}_{nchw})}{\sigma_{nc}(\boldsymbol{x}_{nchw})}\right) - \mu_{nc}(\boldsymbol{y}_{nchw})$$



VGG Encoder
AdaIN
Decoder
Style Transfer Network
VGG Encoder
$\mathcal{L}_c$
$\mathcal{L}_s$
Style
Content
Ours

Group Normalization

Group Normalization

Group normalization (GN) divides the channels into groups and computes the first-order statistics within each group.

Independent of batch sizes

Accuracy is more stable than BN in a wide range of batch sizes

Groups = number of channels : Equivalent to instance normalization

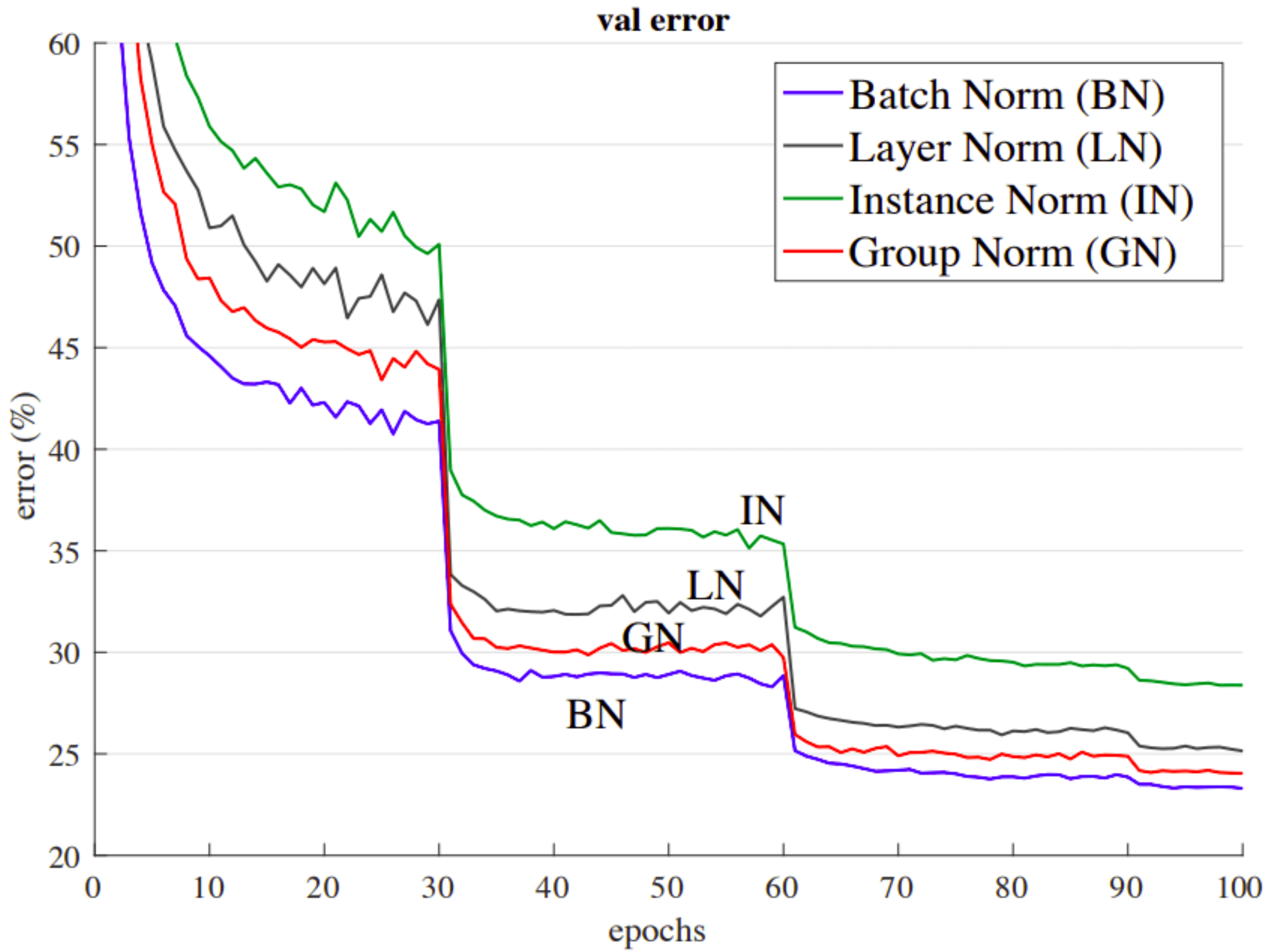
Groups = 1: Equivalent to layer normalization

Group Normalization

ECCV 2018, Munich

Yuxin Wu, Kaiming He
Facebook AI Research (FAIR)

https://youtu.be/m3TN9FFmqsI



Comparison using a batch size of 32 images per GPU in ImageNet. Validation error VS the numbers of training epochs is shown. The model is ResNet-50. Source: [Group Normalization]

In Summary: Different problems require different normalizations

https://youtu.be/UFffxcCQMPQ