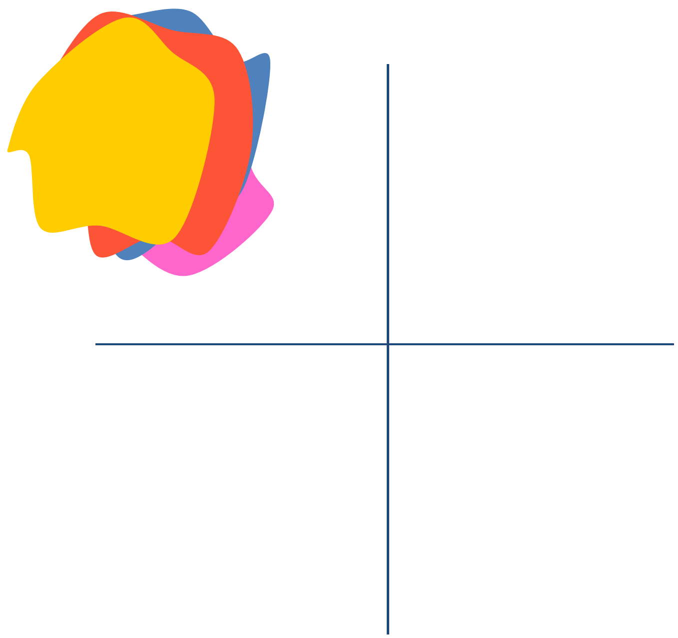
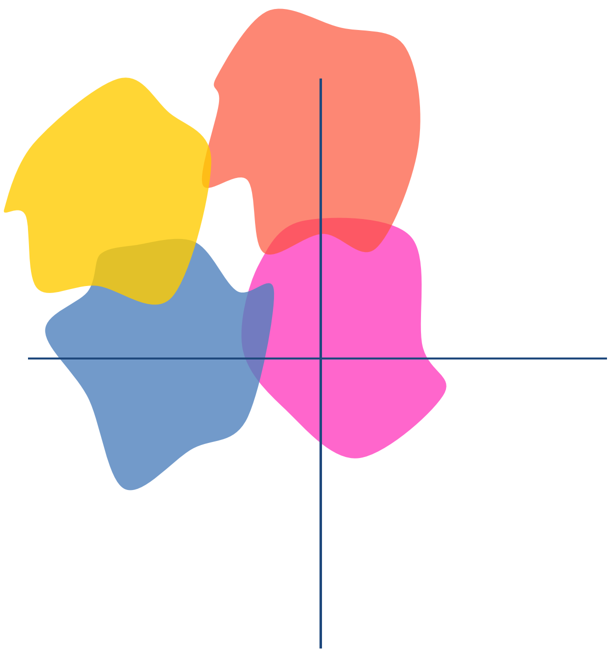
The problem of covariate shifts



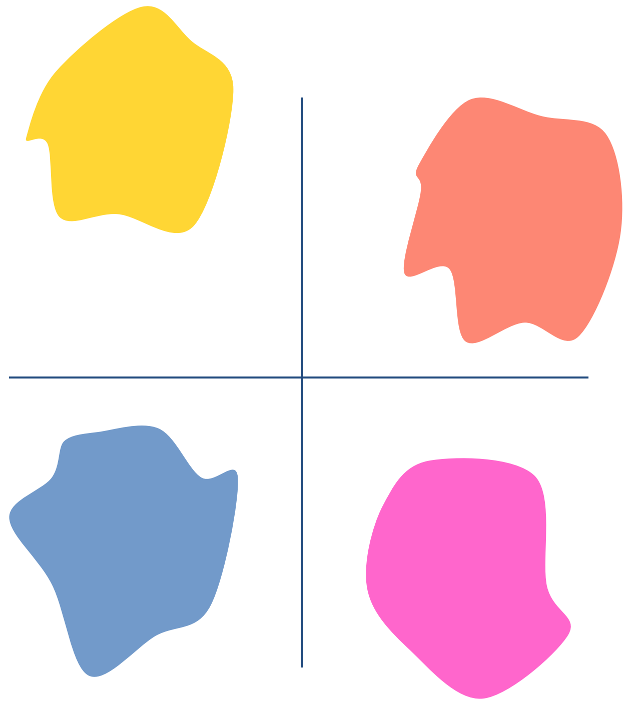
- Training assumes the training data are all similarly distributed
  - Minibatches have similar distribution

The problem of covariate shifts



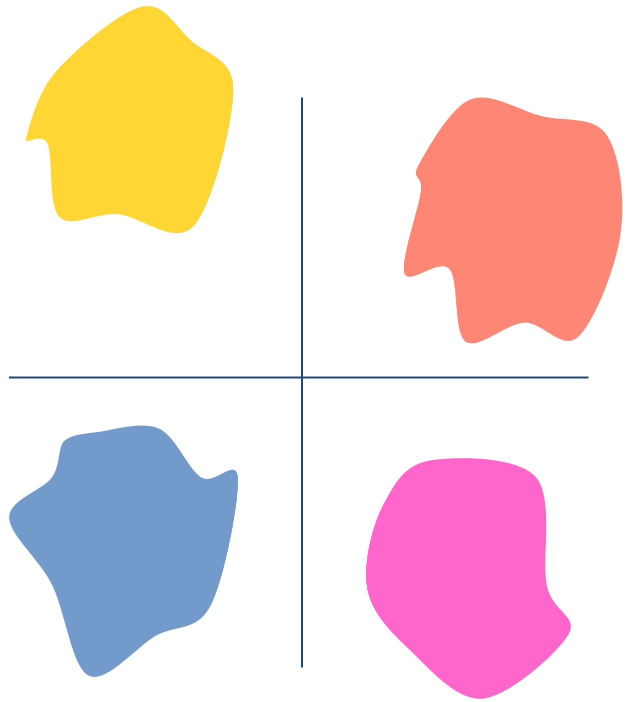
- Training assumes the training data are all similarly distributed
  - Minibatches have similar distribution
- In practice, each minibatch may have a different distribution
  - A "covariate shift"
  - Which may occur in *each* layer of the network

The problem of covariate shifts



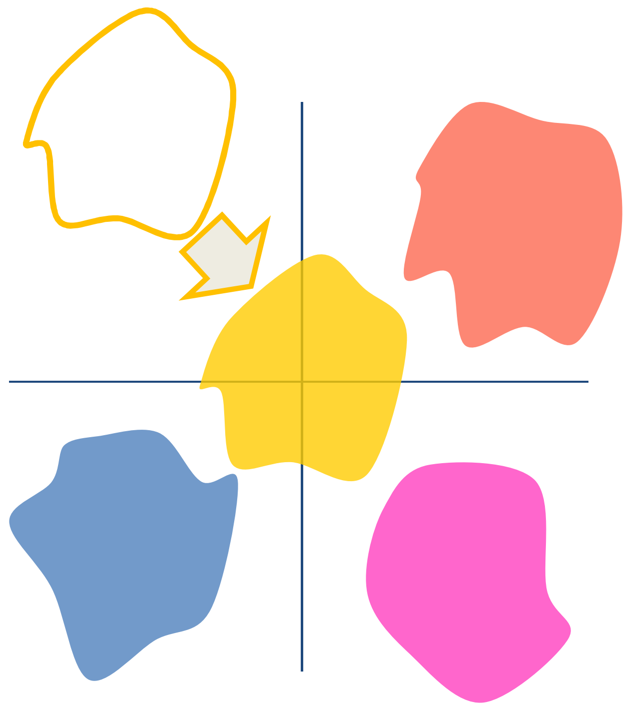
- Training assumes the training data are all similarly distributed
  - Minibatches have similar distribution
- In practice, each minibatch may have a different distribution
  - A “covariate shift”
- Covariate shifts can be large!
  - All covariate shifts can affect training badly

Solution: Move all subgroups to a "standard" location



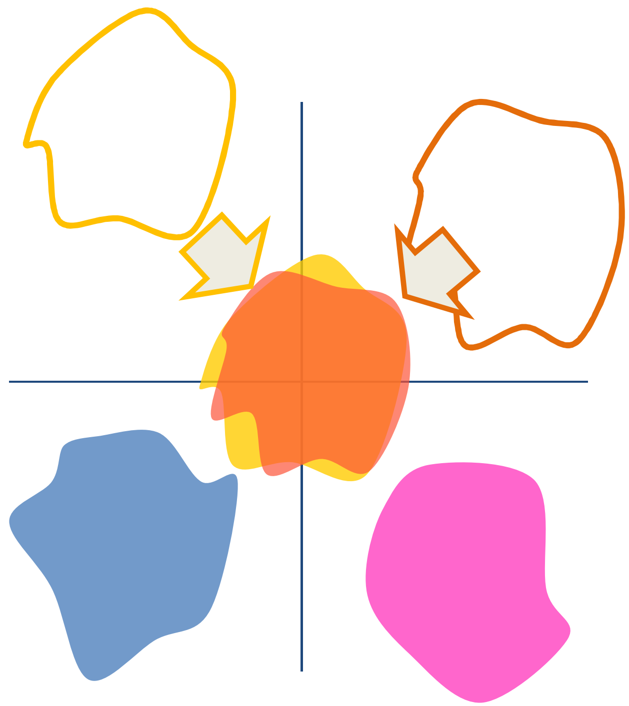
- "Move" all batches to have a mean of 0 and unit standard deviation
  - Eliminates covariate shift between batches

Solution: Move all subgroups to a "standard" location



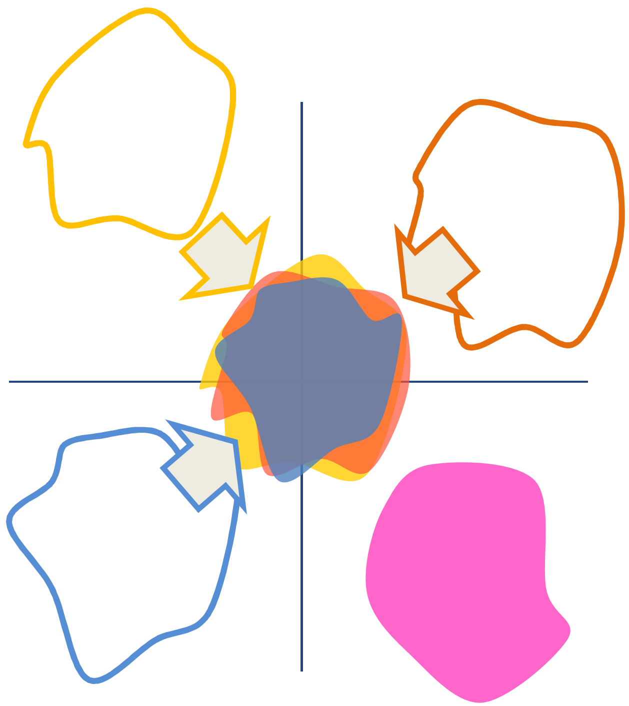
- "Move" all batches to have a mean of 0 and unit standard deviation
  - Eliminates covariate shift between batches

Solution: Move all subgroups to a "standard" location



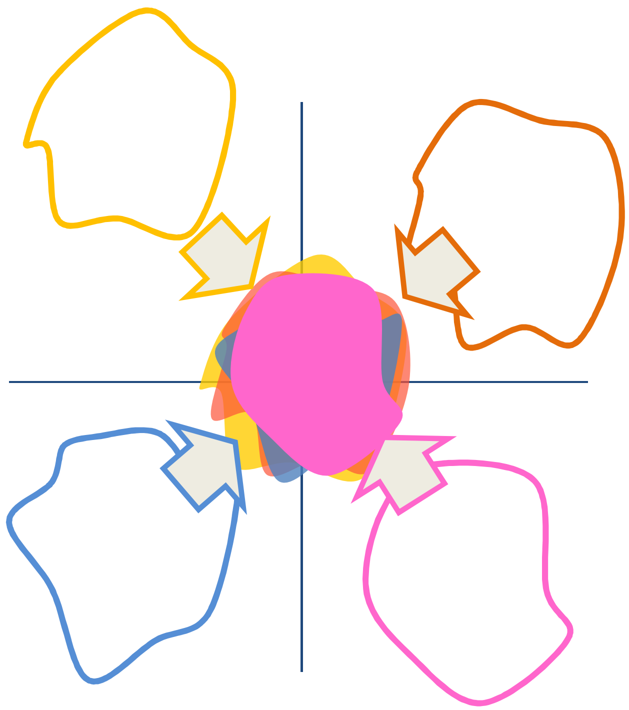
- "Move" all batches to have a mean of 0 and unit standard deviation
  - Eliminates covariate shift between batches

Solution: Move all subgroups to a "standard" location



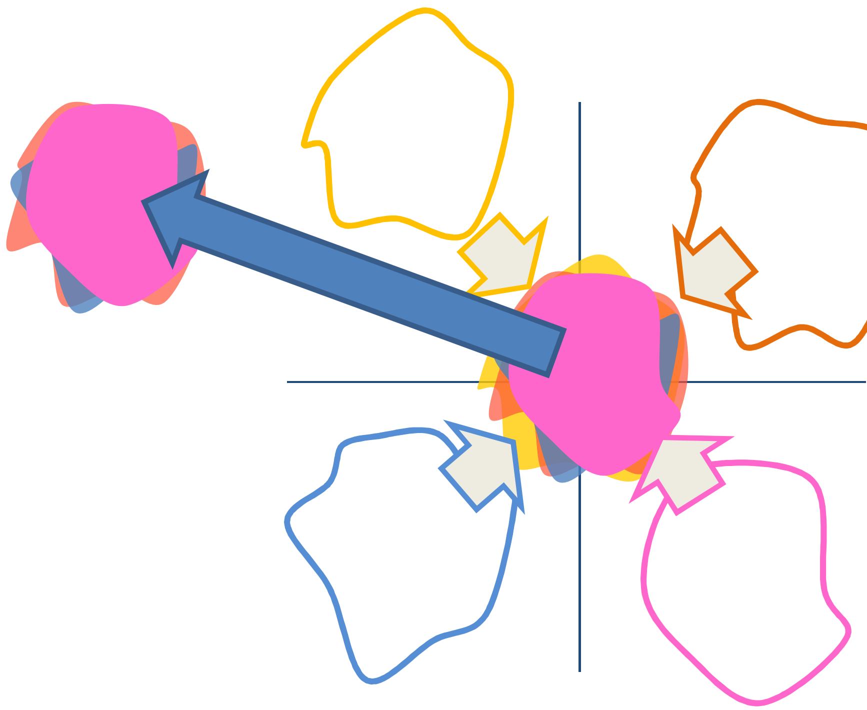
- "Move" all batches to have a mean of 0 and unit standard deviation
  - Eliminates covariate shift between batches

Solution: Move all subgroups to a "standard" location



- "Move" all batches to have a mean of 0 and unit standard deviation
  - Eliminates covariate shift between batches

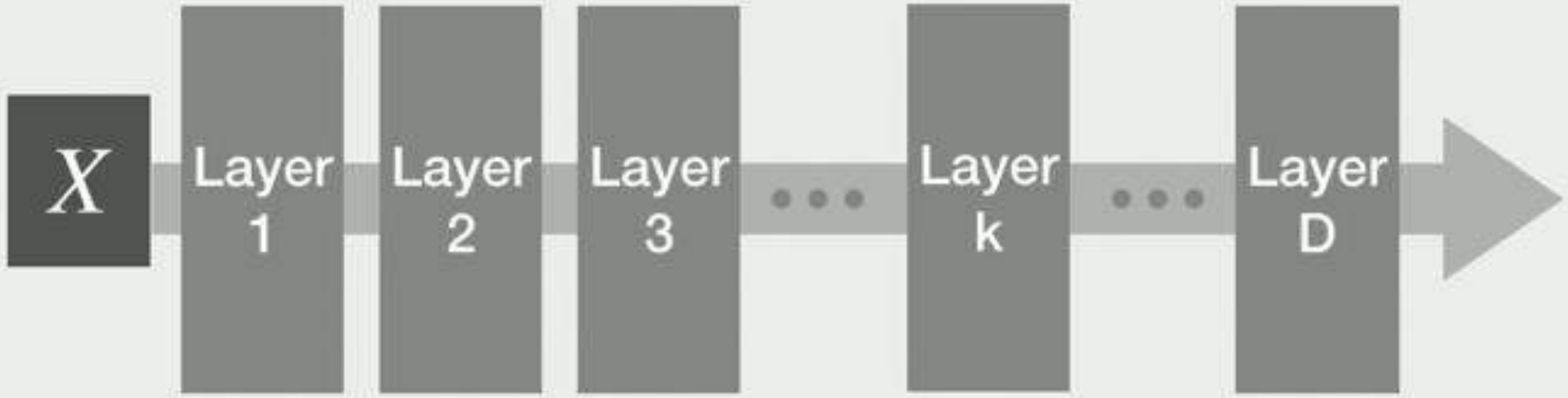
Solution: Move all subgroups to a "standard" location



- "Move" all batches to have a mean of 0 and unit standard deviation
  - Eliminates covariate shift between batches
  - Then move the entire collection to the appropriate location

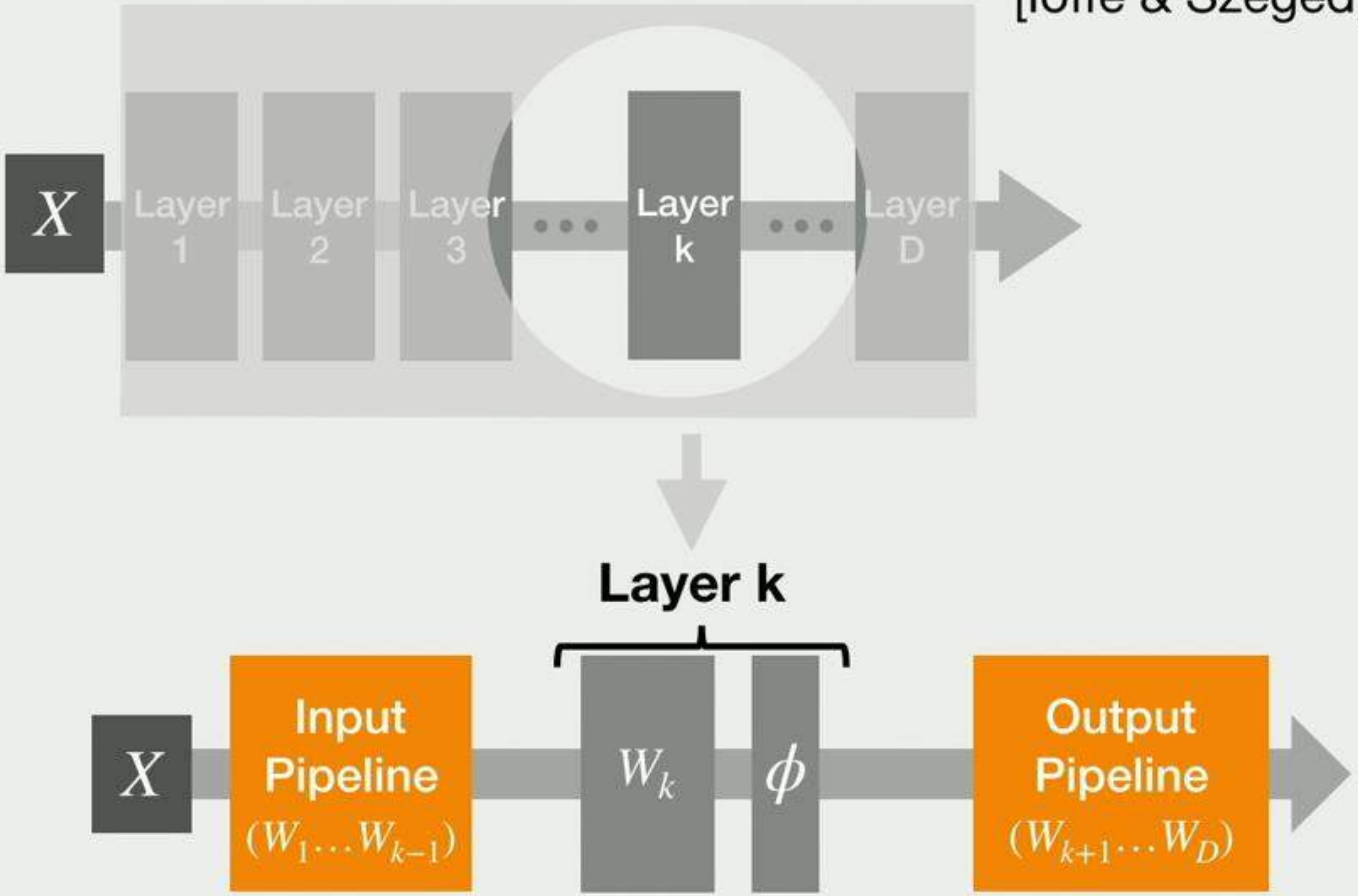
Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



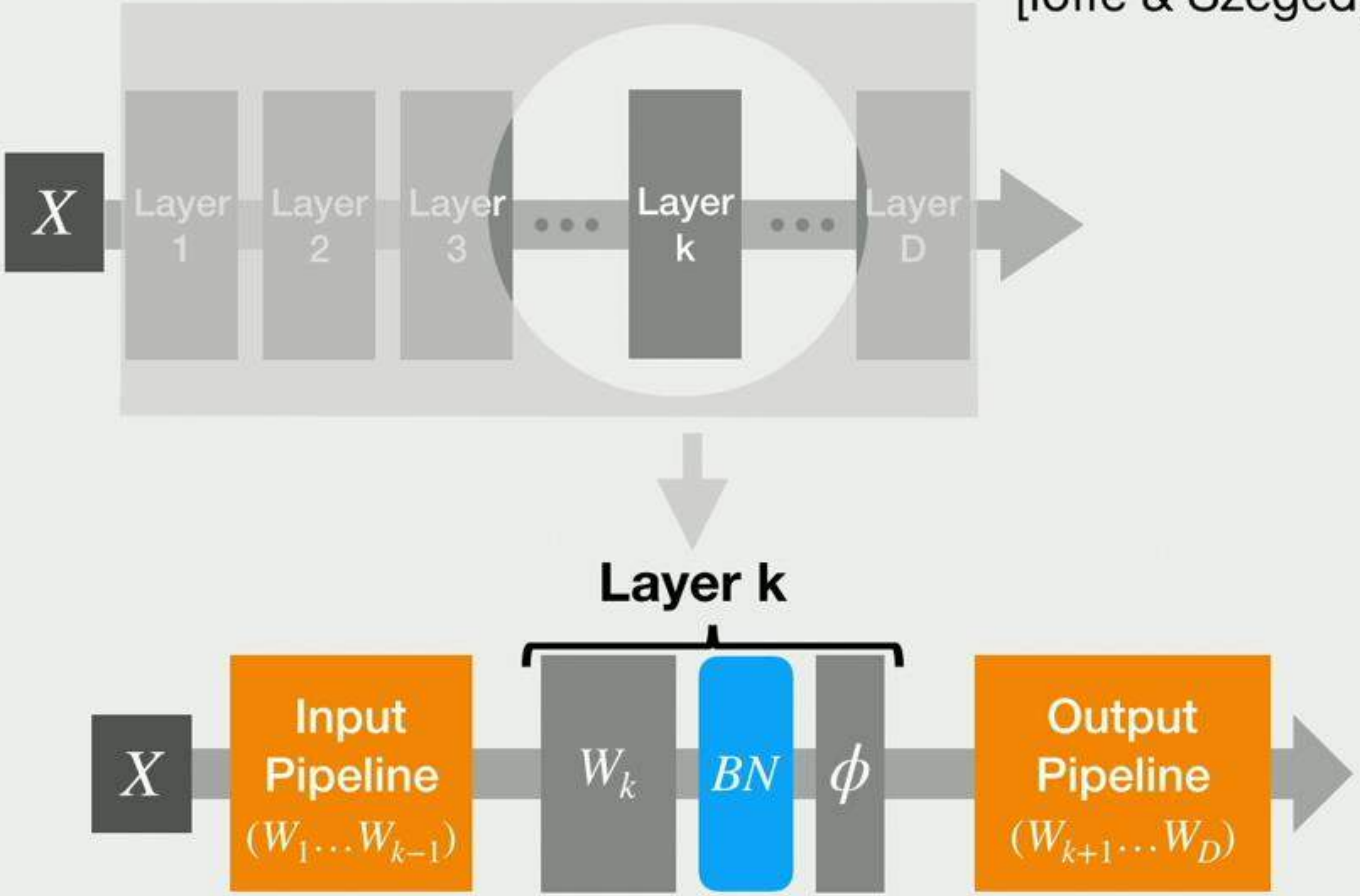
Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



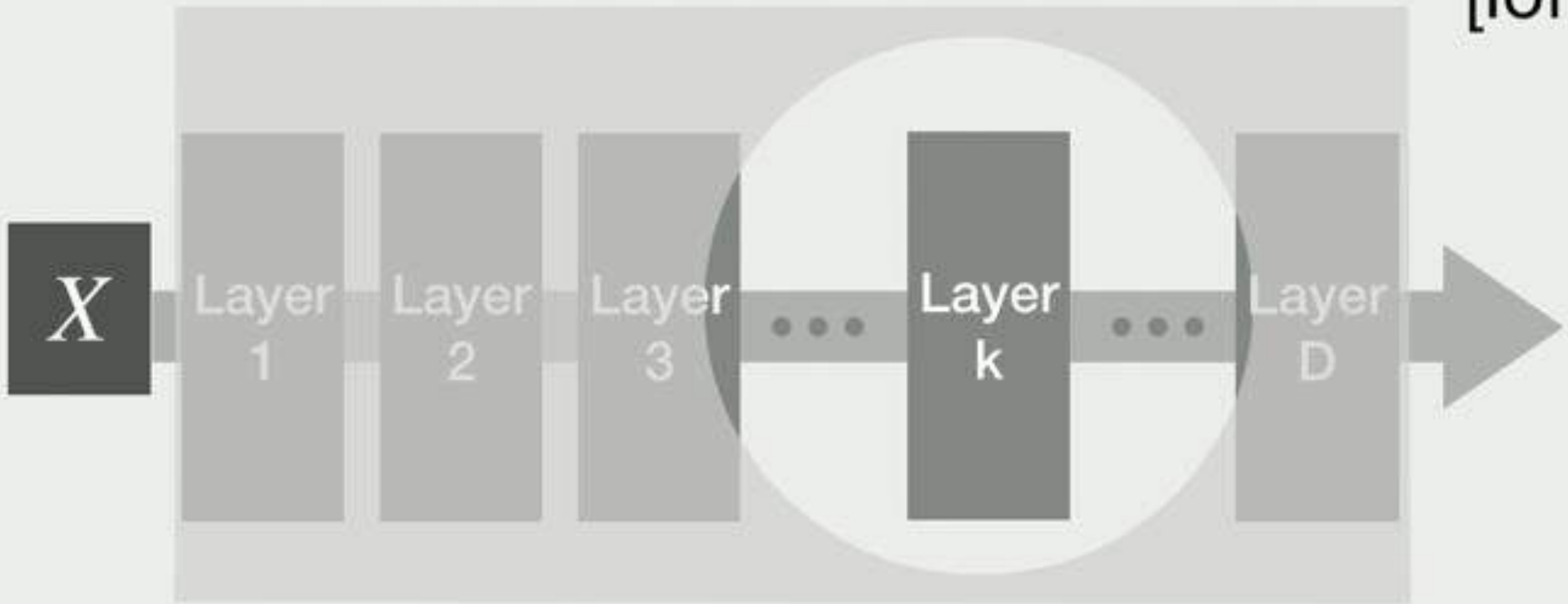
Batch Normalization (BatchNorm)

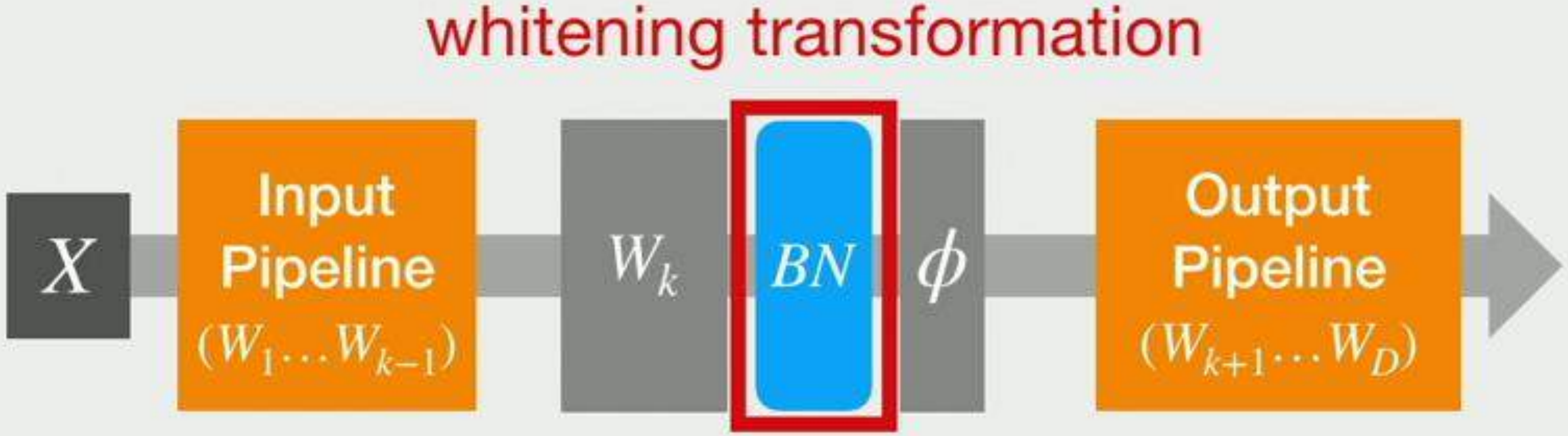
[Ioffe & Szegedy, 2015]



Batch Normalization (BatchNorm)

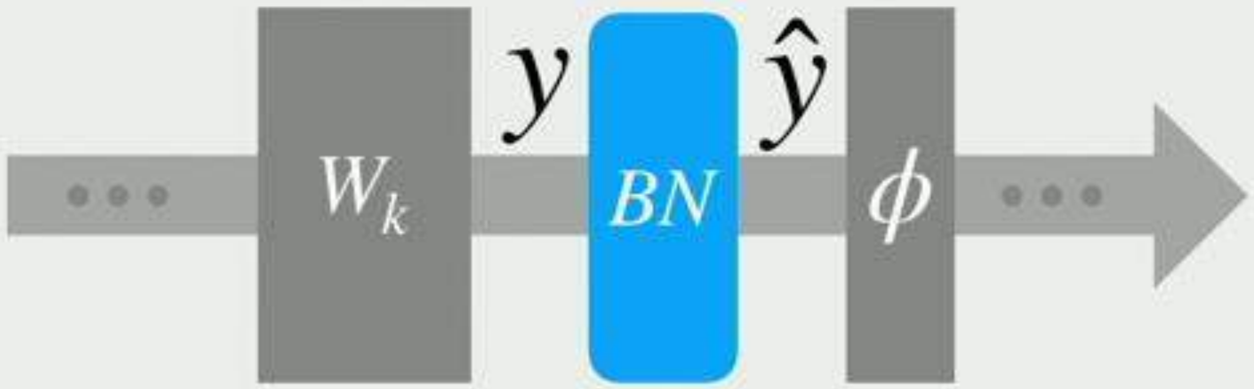
[Ioffe & Szegedy, 2015]





Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



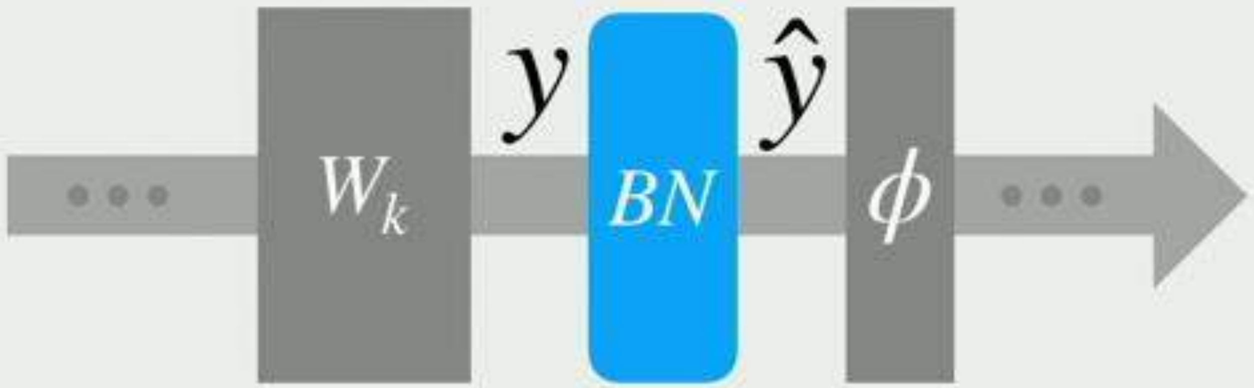
Make distribution "close" to **standard normal**

$$\hat{y} = \frac{y - \mu}{\sigma + \epsilon} \quad ,$$

where $\quad \mu = \mathbb{E}[y] \qquad \sigma^2 = Var(y)$

Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



Make distribution "close" to **standard normal**

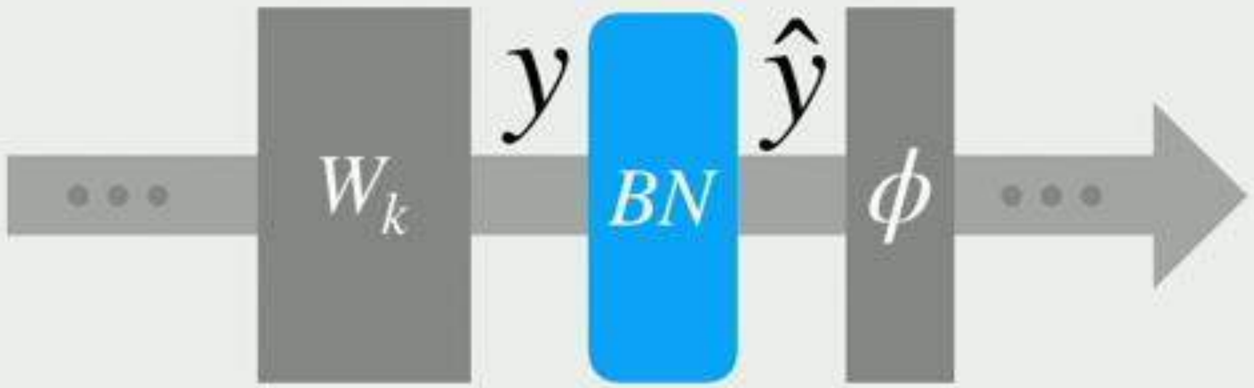
Batch Statistics

$$\hat{y} = \frac{y - \hat{\mu}}{\hat{\sigma} + \epsilon},$$

where $\quad \hat{\mu} = \frac{1}{B}\sum_{i=1}^{B} y_i \qquad \hat{\sigma}^2 = \frac{1}{B}\sum_{i=1}^{B} (y_i - \hat{\mu})^2$

Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



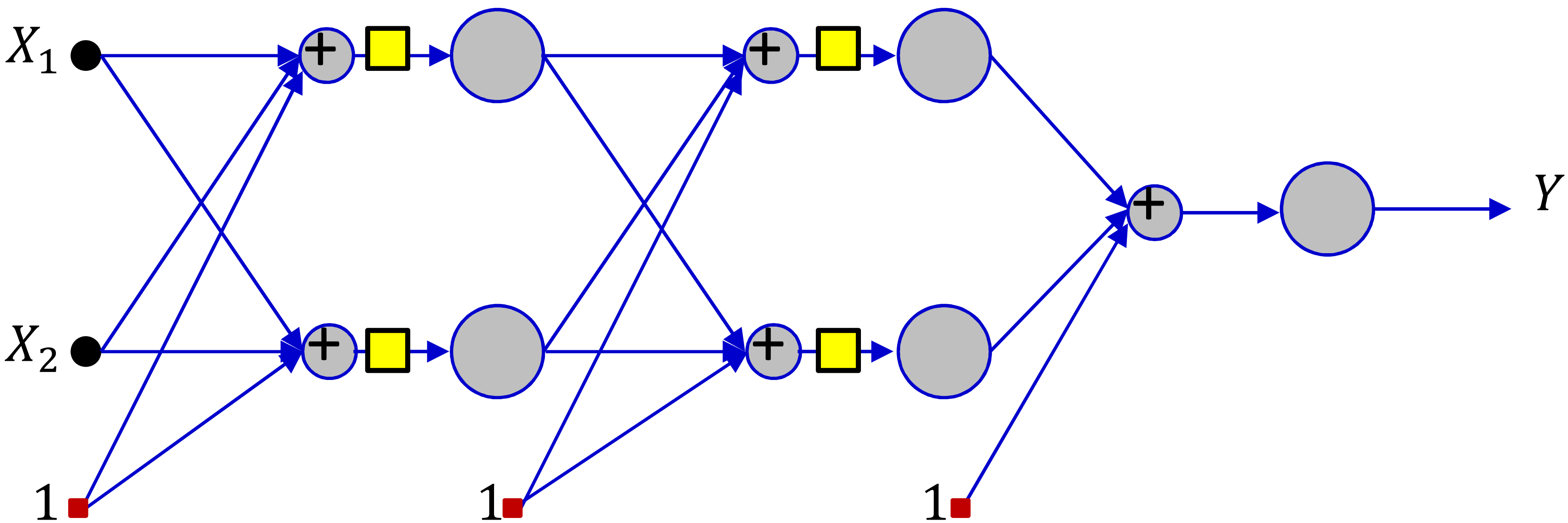
Make distribution "close" to **standard normal**

Learnable

$$\hat{y} = \boxed{\gamma} \frac{y - \hat{\mu}}{\hat{\sigma} + \epsilon} + \boxed{\beta},$$

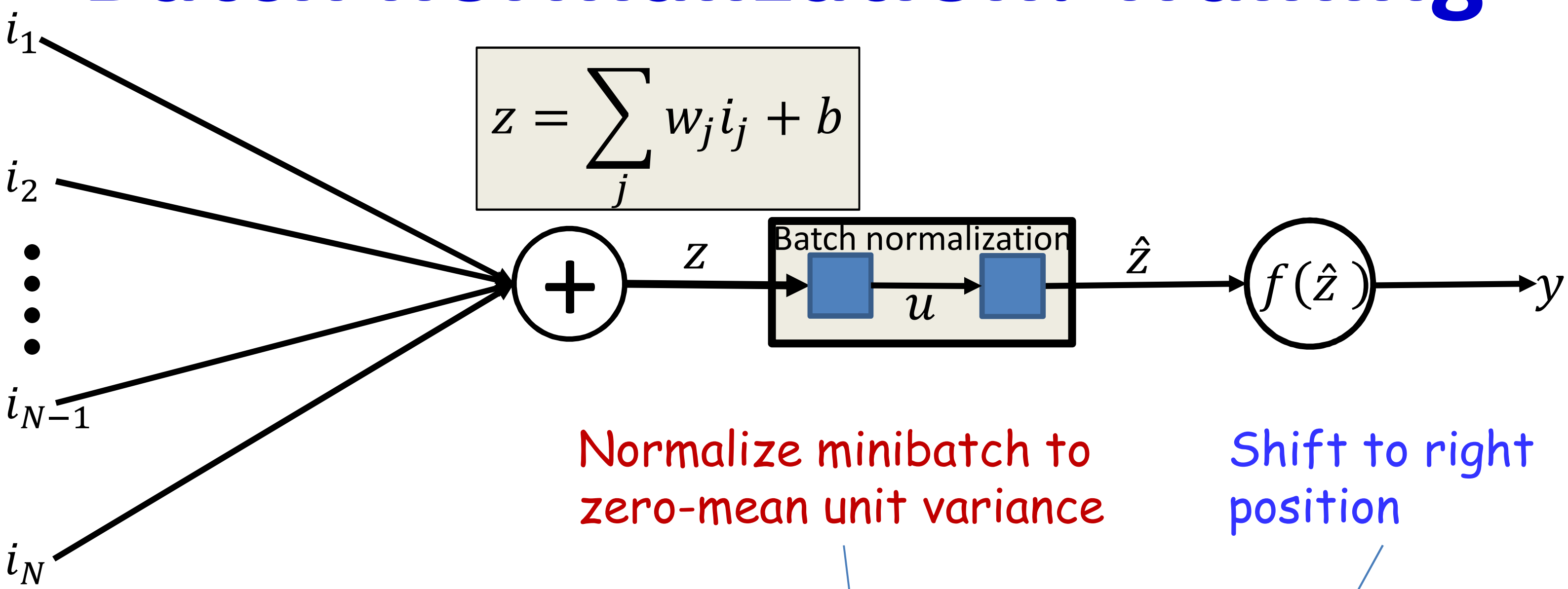
where $\hat{\mu} = \frac{1}{B} \sum_{i=1}^{B} y_i$ $\qquad \hat{\sigma}^2 = \frac{1}{B} \sum_{i=1}^{B} (y_i - \hat{\mu})^2$

Batch normalization



- Batch normalization is a covariate adjustment unit that happens after the weighted addition of inputs but before the application of activation
  – Is done independently for each unit, to simplify computation
- **Training: The adjustment occurs over individual minibatches**

Batch normalization: Training



$$\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$$

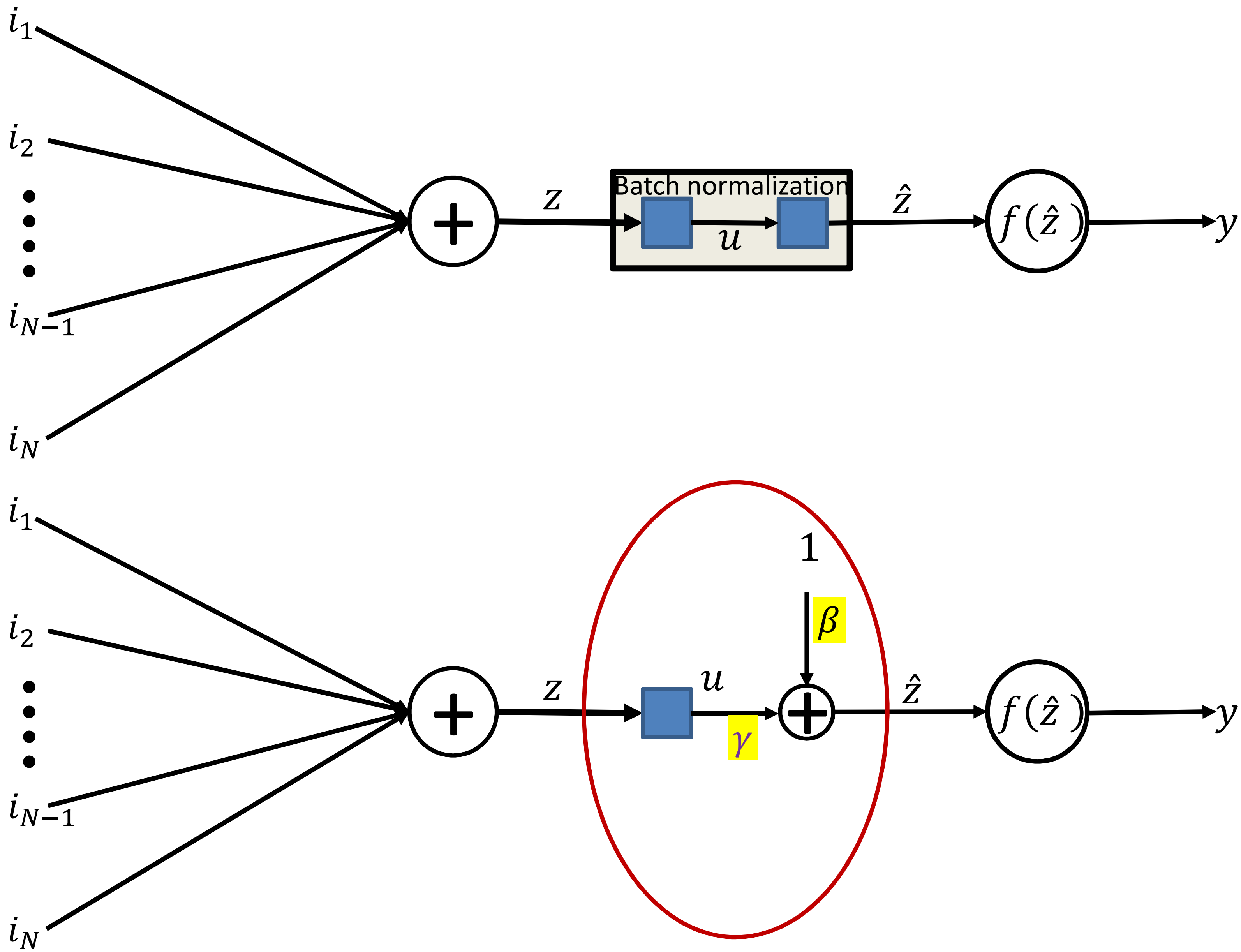
$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B} (z_i - \mu_B)^2$$

$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\hat{z}_i = \gamma u_i + \beta$$

- BN aggregates the statistics over a minibatch and normalizes the batch by them
- Normalized instances are “shifted” to a *unit-specific* location

A better picture for batch norm



A note on derivatives

- In conventional learning, we attempt to compute the derivative of the divergence for *individual* training instances w.r.t. parameters

- This is based on the following relations

$$Div(minibatch) = \frac{1}{B}\sum_t Div(Y_t(X_t), d_t(X_t))$$

$$\frac{dDiv(minibatch)}{dw_{i,j}^{(k)}} = \frac{1}{T}\sum_t \frac{dDiv(Y_t(X_t), d_t(X_t))}{dw_{i,j}^{(k)}}$$

- If we use Batch Norm, the above relation gets a little complicated

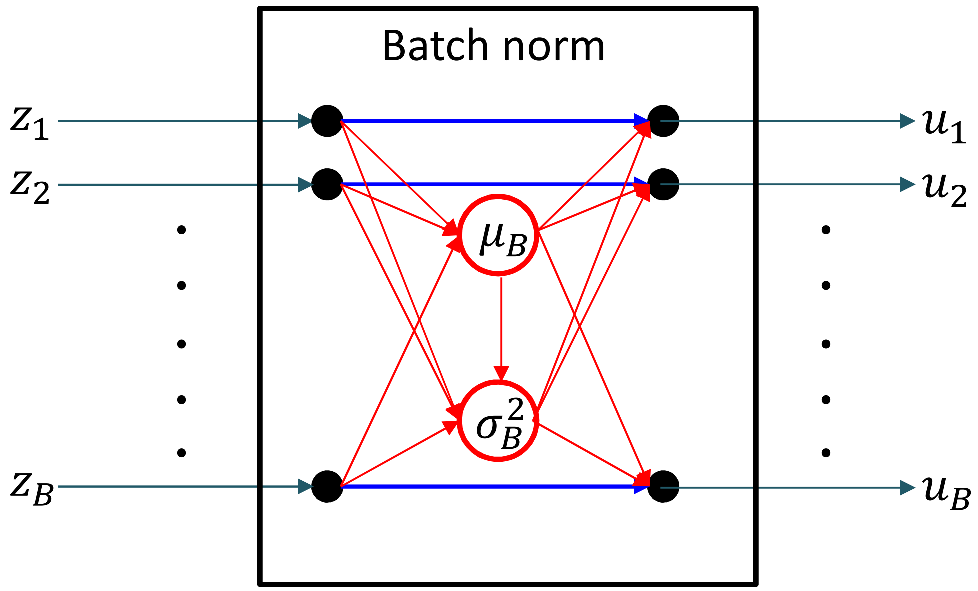
A note on derivatives

- The outputs are now functions of μ_B and σ_B^2 which are functions of the entire minibatch

$$Div(MB) = \frac{1}{B}\sum_t Div(Y_t(X_t, \mu_B, \sigma_B^2), d_t(X_t))$$

- The Divergence for each Y_t depends on *all* the X_t within the minibatch
- Specifically, within each layer, we get the relationship in the following slide

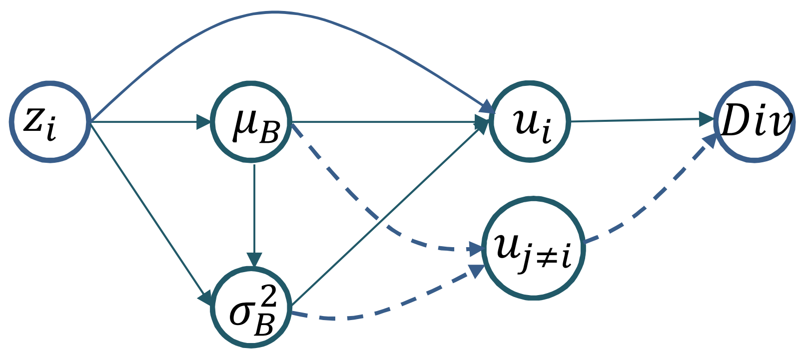
Batchnorm



- The complete dependency figure for Batchnorm
- Note : inputs and outputs are different *instances* in a minibatch
  - The diagram represents BN occurring at a *single neuron*
- You can use vector function differentiation rules to compute the derivatives
  - But the equations in the following slides summarize them for you
  - The actual derivation uses the simplified diagram shown in the next slide, but you could do it directly off the figure above and arrive at the same answers

Batchnorm

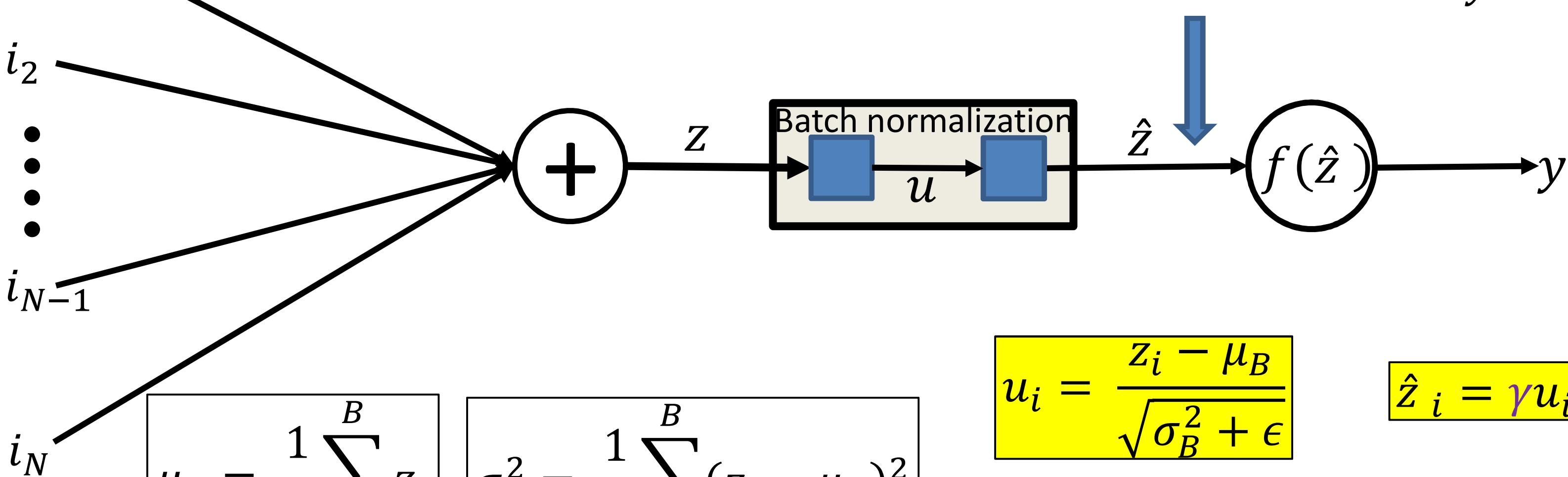
Influence diagram



- Simplified diagram for a *single* input in a minibatch

Batch normalization: Backpropagation

$$\frac{dDiv}{d\hat{z}} = f'(\hat{z})\frac{dDiv}{dy}$$



$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\hat{z}_i = \gamma u_i + \beta$$

$$\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$$

$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2$$

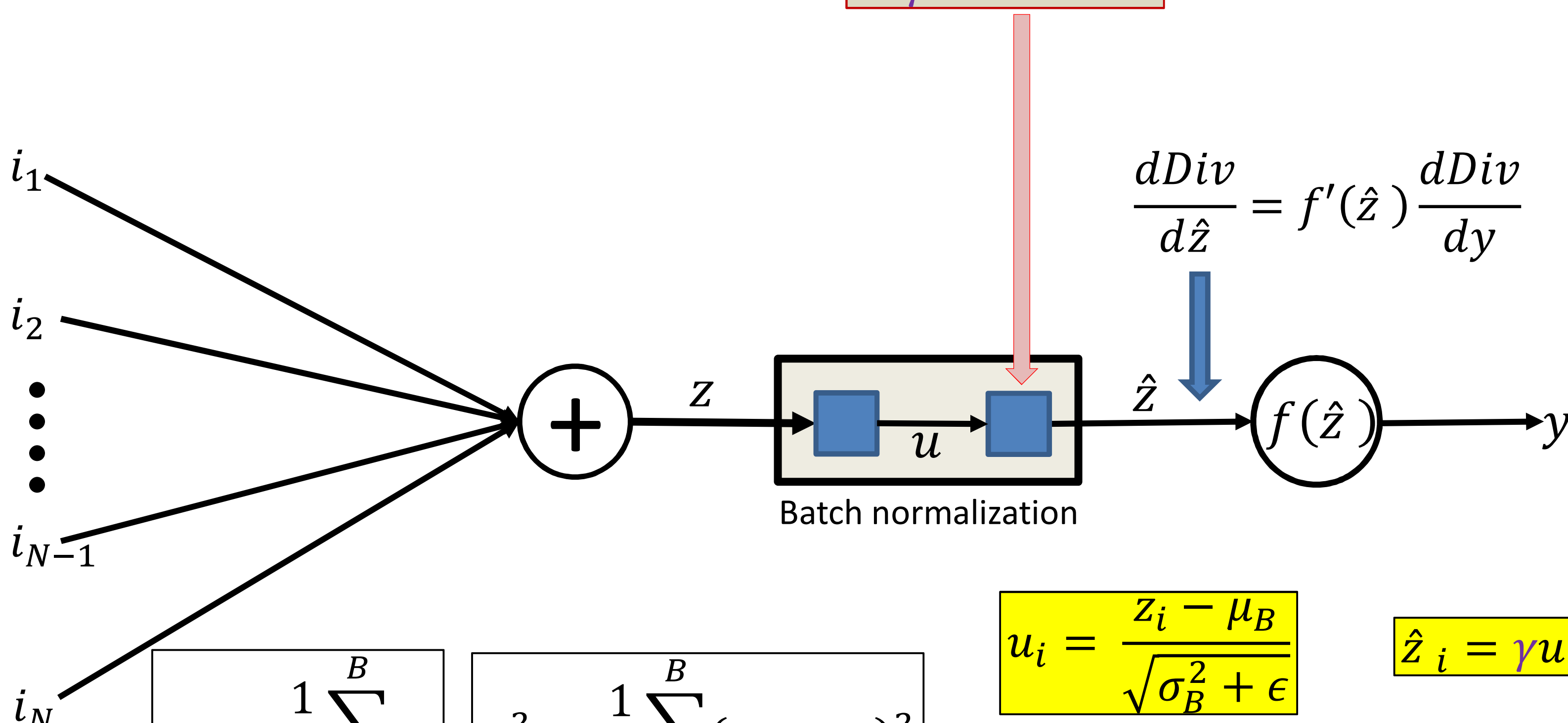
Batch normalization: Backpropagation

$$\frac{dDiv}{d\beta} = \frac{dDiv}{d\hat{z}}$$

$$\frac{dDiv}{d\gamma} = u\frac{dDiv}{d\hat{z}}$$

Parameters to be learned

$$\frac{dDiv}{d\hat{z}} = f'(\hat{z})\frac{dDiv}{dy}$$



$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\hat{z}_i = \gamma u_i + \beta$$

$$\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$$

$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2$$

Batch normalization: Backpropagation

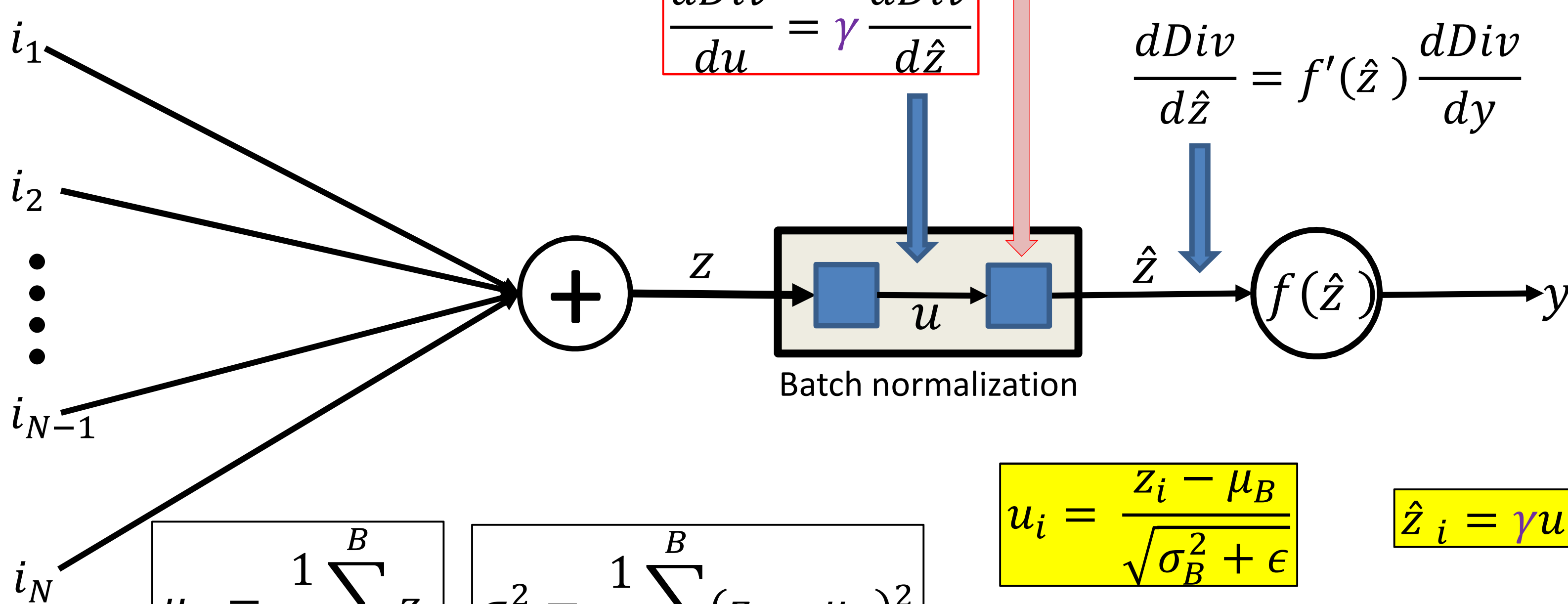
$$\frac{dDiv}{d\beta} = \frac{dDiv}{d\hat{z}}$$

$$\frac{dDiv}{d\gamma} = u\frac{dDiv}{d\hat{z}}$$

Parameters to be learned

$$\frac{dDiv}{du} = \gamma\frac{dDiv}{d\hat{z}}$$

$$\frac{dDiv}{d\hat{z}} = f'(\hat{z}\,)\frac{dDiv}{dy}$$



$$\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$$

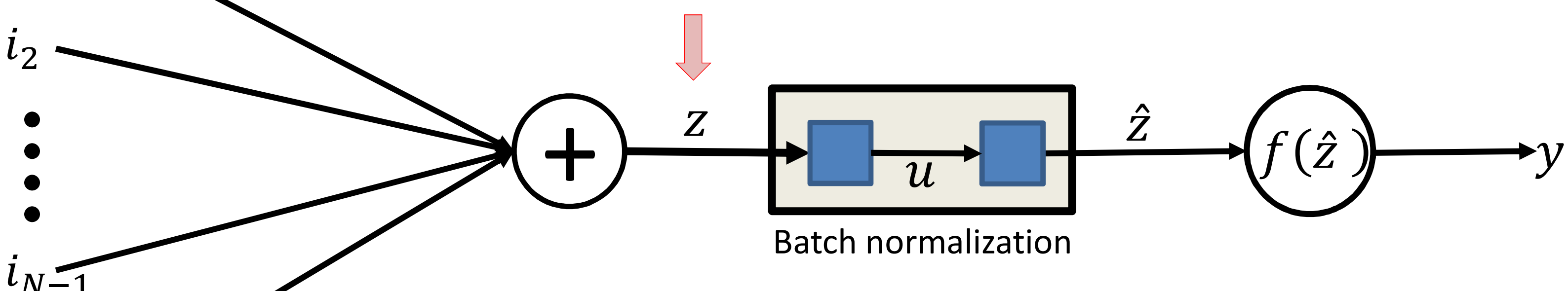
$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2$$

$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\hat{z}_{\ i} = \gamma u_i + \beta$$

Batch normalization: Backpropagation

- Final step of backprop: compute $\frac{\partial Div}{\partial z_i}$



$$\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$$

$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2$$

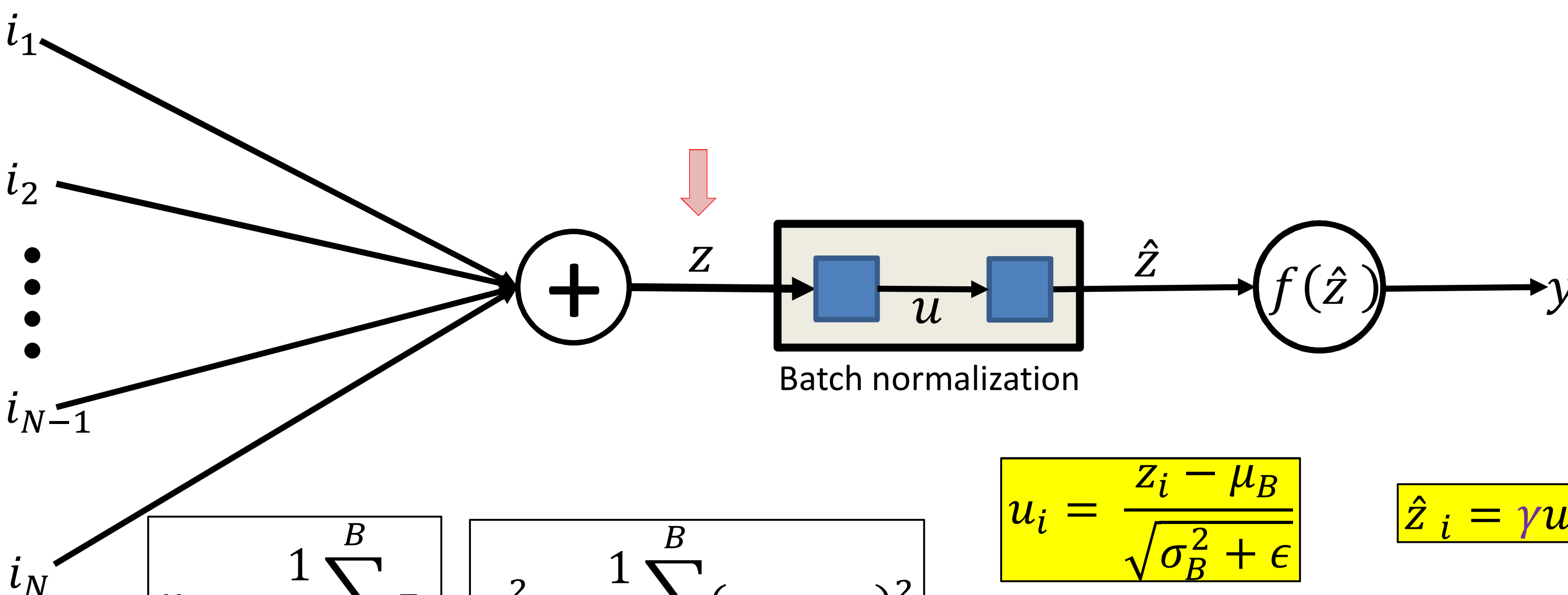
$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\hat{z}_i = \gamma u_i + \beta$$

Batch normalization: Backpropagation

$$Div = function(u_i, \mu_B, \sigma_B^2)$$

$$\boxed{\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}}$$



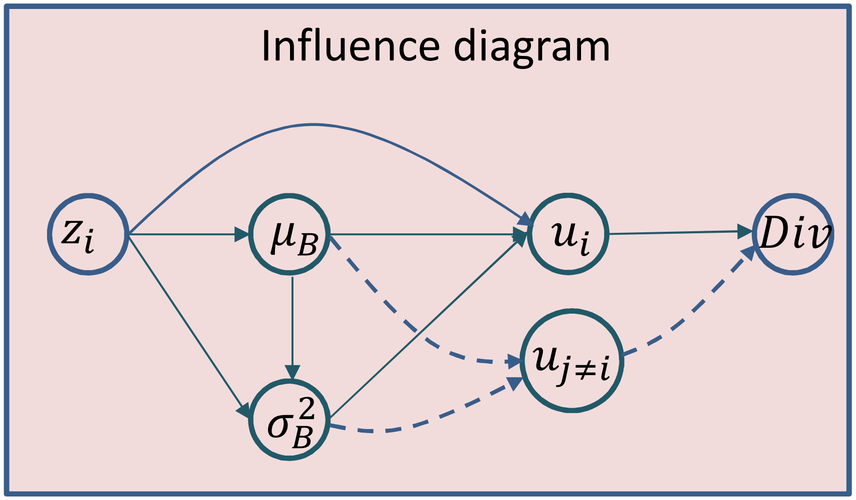
$$\boxed{u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}}$$

$$\boxed{\hat{z}_i = \gamma u_i + \beta}$$

$$\boxed{\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i}$$

$$\boxed{\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2}$$

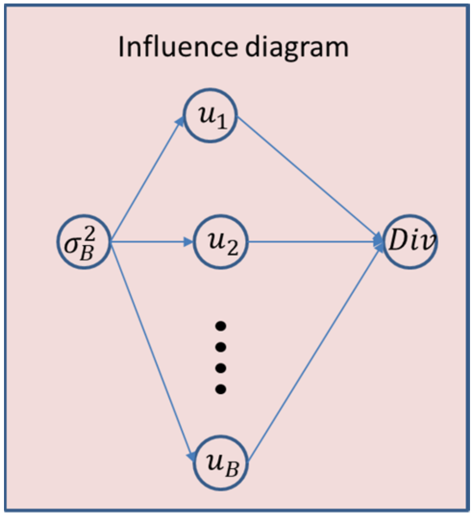
Batch normalization: Backpropagation

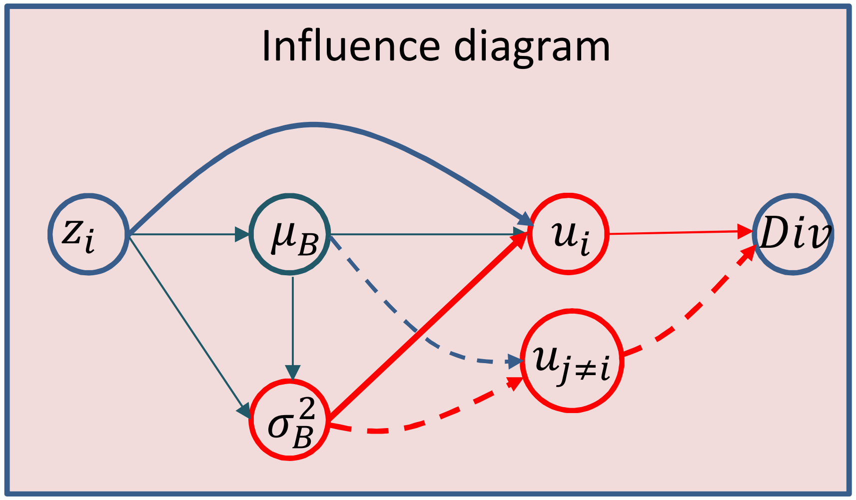


Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

Batch normalization: Backpropagation





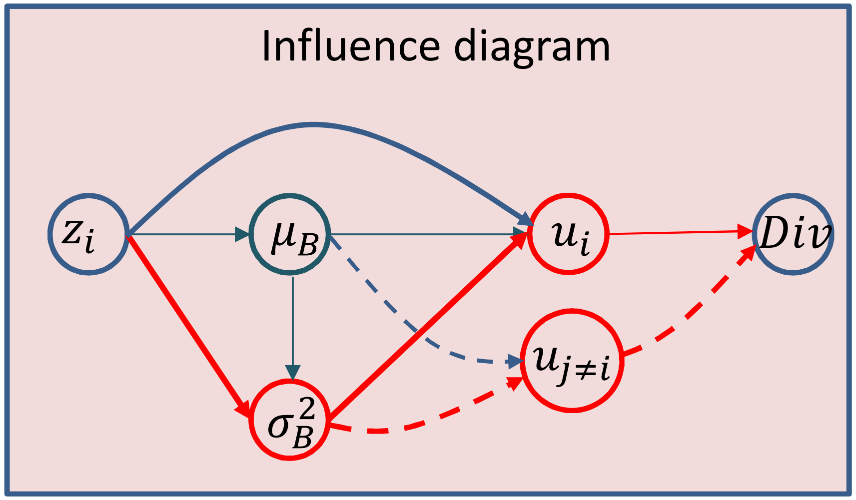
Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\frac{\partial Div}{\partial \sigma_B^2} = \frac{-1}{2}\left(\sigma_B^2 + \epsilon\right)^{-3/2} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}\left(z_i - \mu_B\right)$$

Batch normalization: Backpropagation



Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i}\cdot\frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2}\cdot\frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B}\cdot\frac{\partial \mu_B}{\partial z_i}$$

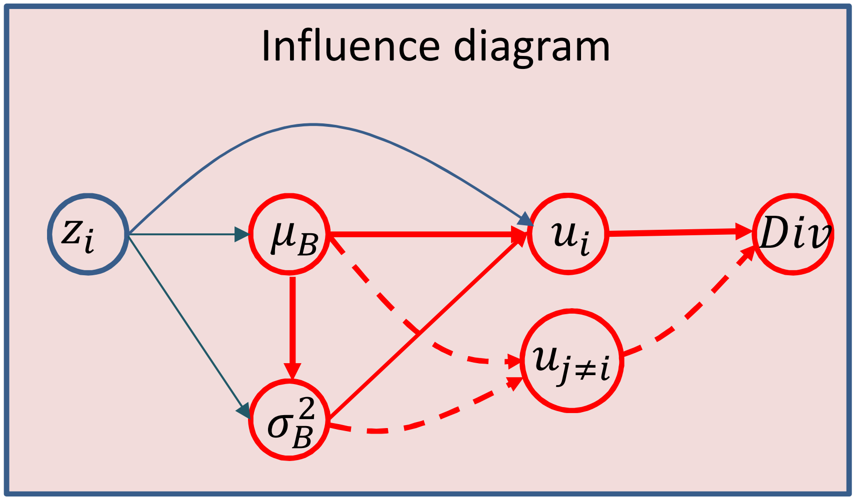
$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\frac{\partial Div}{\partial \sigma_B^2} = \frac{-1}{2}\left(\sigma_B^2 + \epsilon\right)^{-3/2}\sum_{i=1}^{B}\frac{\partial Div}{\partial u_i}(z_i - \mu_B)$$

$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2$$

$$\frac{\partial \sigma_B^2}{\partial z_i} = \frac{2(z_i - \mu_B)}{B}$$

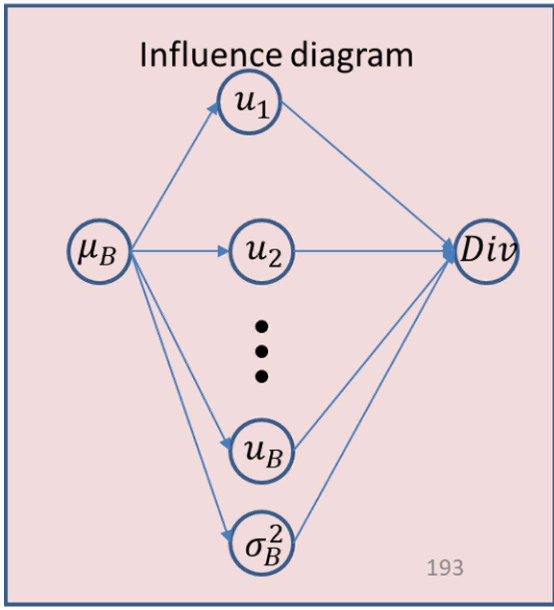
Batch normalization: Backpropagation

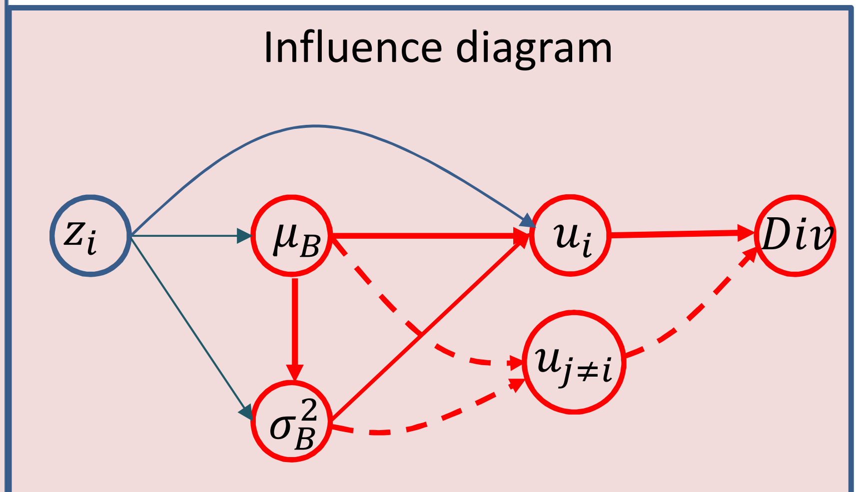


Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

Batch normalization: Backpropagation





Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

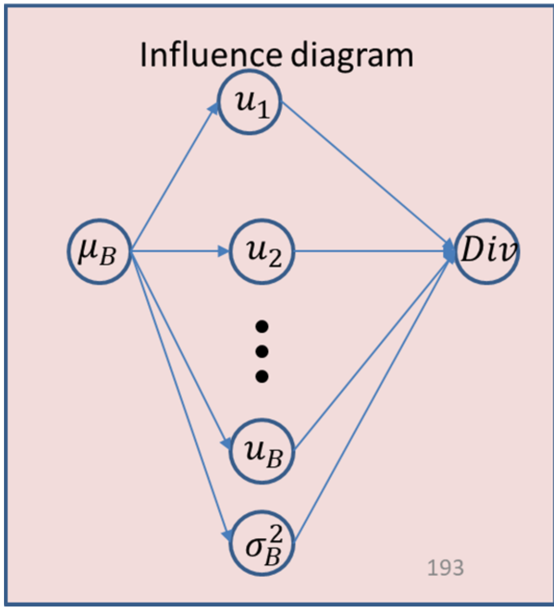
Second term goes to 0

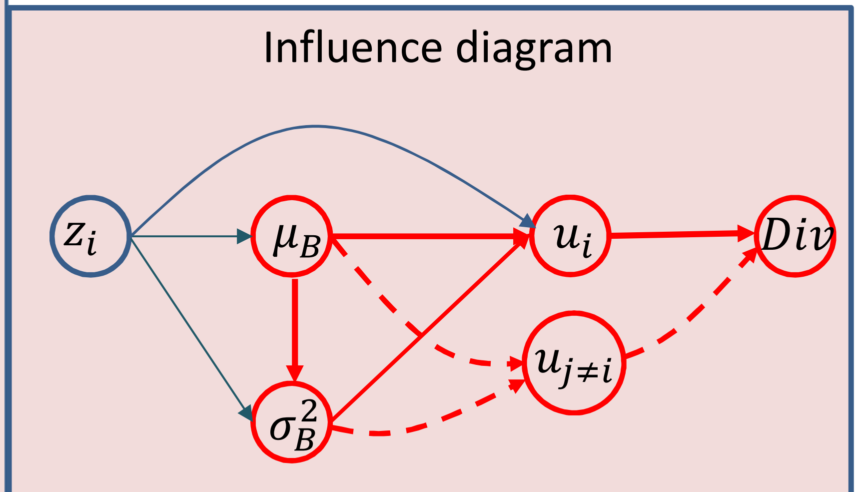
$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\frac{\partial Div}{\partial \mu_B} = \left(\sum_{i=1}^{B} \frac{\partial Div}{\partial u_i} \cdot \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\sum_{i=1}^{B} -2(z_i - \mu_B)}{B}$$

$$\sigma_B^2 = \frac{1}{B} \sum_{i=1}^{B} (z_i - \mu_B)^2$$

Batch normalization: Backpropagation





Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

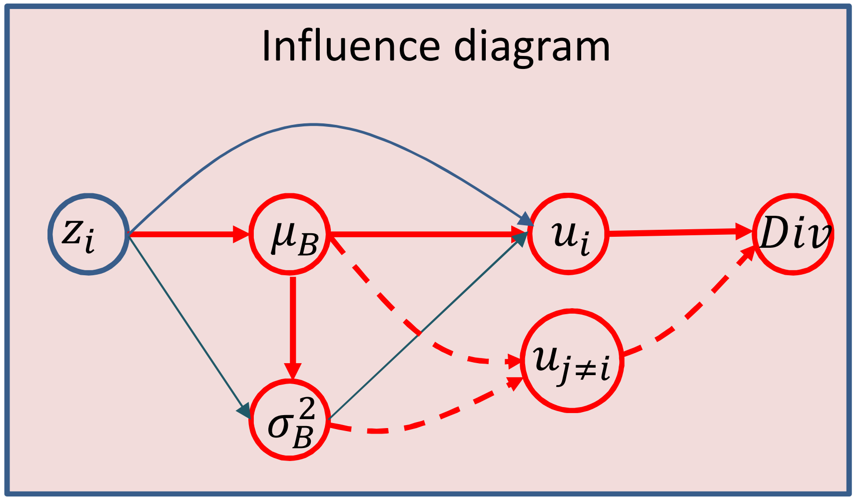
$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\frac{\partial Div}{\partial \mu_B} = \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}$$

$$\sigma_B^2 = \frac{1}{B} \sum_{i=1}^{B} (z_i - \mu_B)^2$$

Batch normalization: Backpropagation



Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

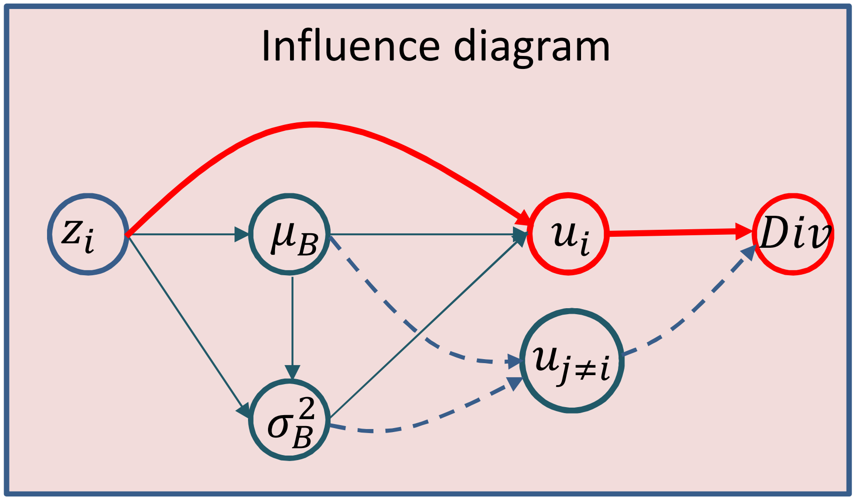
$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

$$\frac{\partial \mu_B}{\partial z_i} = \frac{1}{B}$$

$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\frac{\partial Div}{\partial \mu_B} = \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}$$

Batch normalization: Backpropagation



Dotted lines show dependence through other u_js because Divergence is computed over a minibatch

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{\partial u_i}{\partial z_i} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial z_i}$$

$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

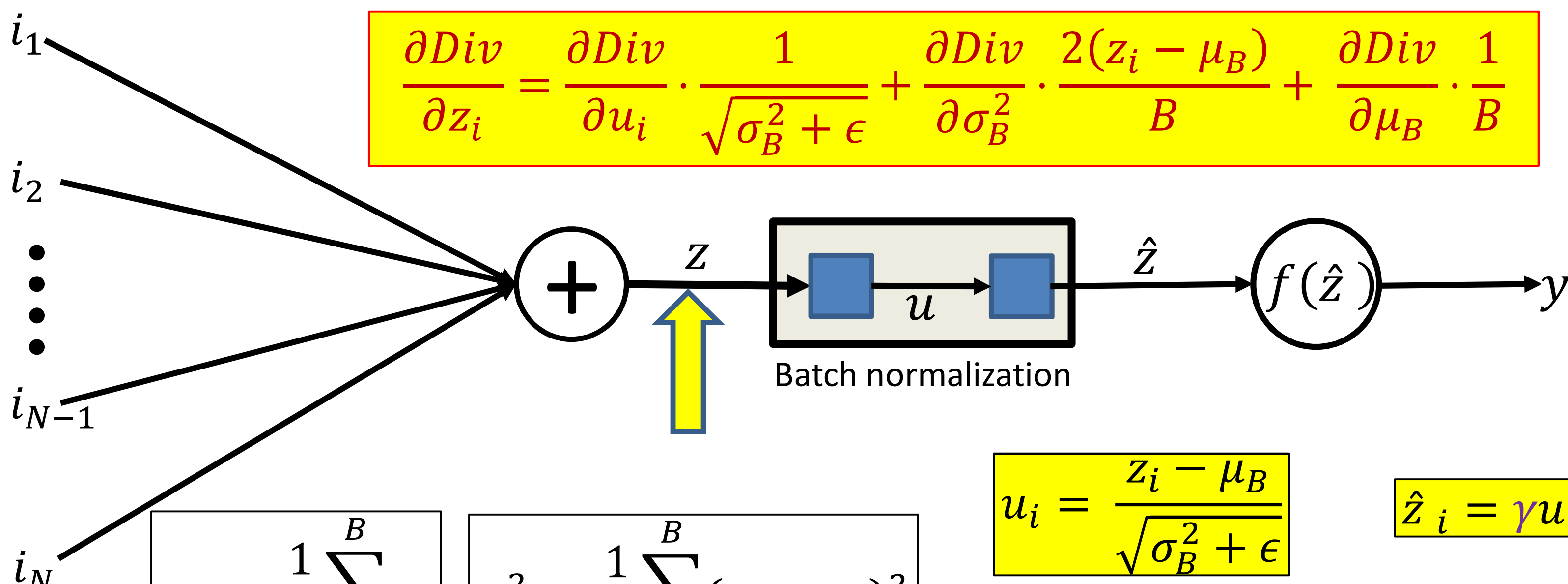
$$\frac{\partial Div}{\partial u_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}}$$

Batch normalization: Backpropagation

$$\frac{\partial Div}{\partial \sigma_B^2} = \frac{-1}{2}\left(\sigma_B^2 + \epsilon\right)^{-3/2} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}\left(z_i - \mu_B\right)$$

$$\frac{\partial Div}{\partial \mu_B} = \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}$$

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{2(z_i - \mu_B)}{B} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{1}{B}$$



$$u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\hat{z}_i = \gamma u_i + \beta$$

$$\mu_B = \frac{1}{B} \sum_{i=1}^{B} z_i$$

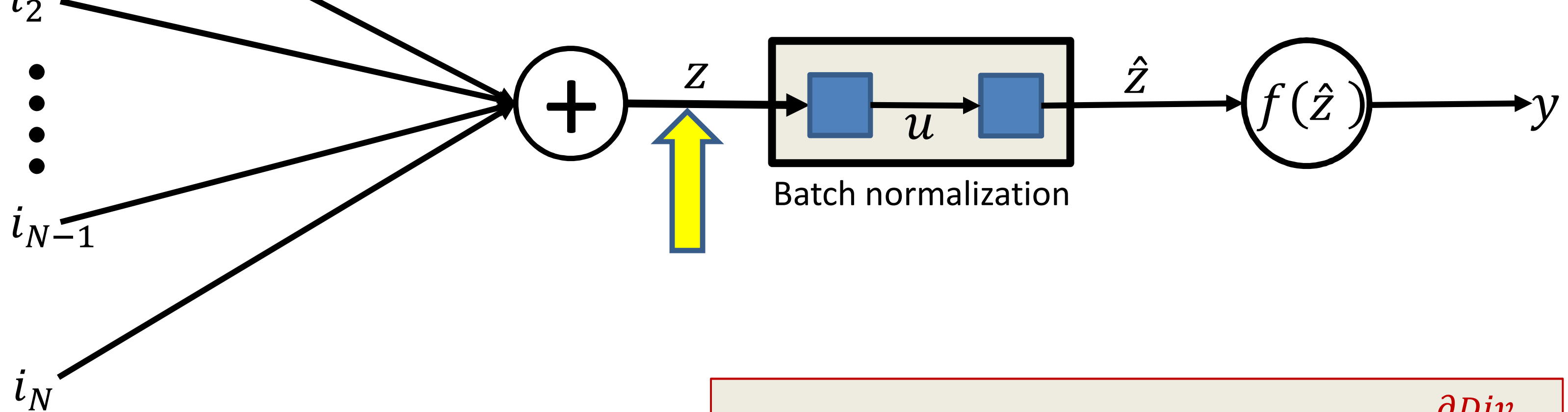
$$\sigma_B^2 = \frac{1}{B} \sum_{i=1}^{B} \left(z_i - \mu_B\right)^2$$

Batch normalization: Backpropagation

$$\frac{\partial Div}{\partial \sigma_B^2} = \frac{-1}{2}(\sigma_B^2 + \epsilon)^{-3/2} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}(z_i - \mu_B)$$

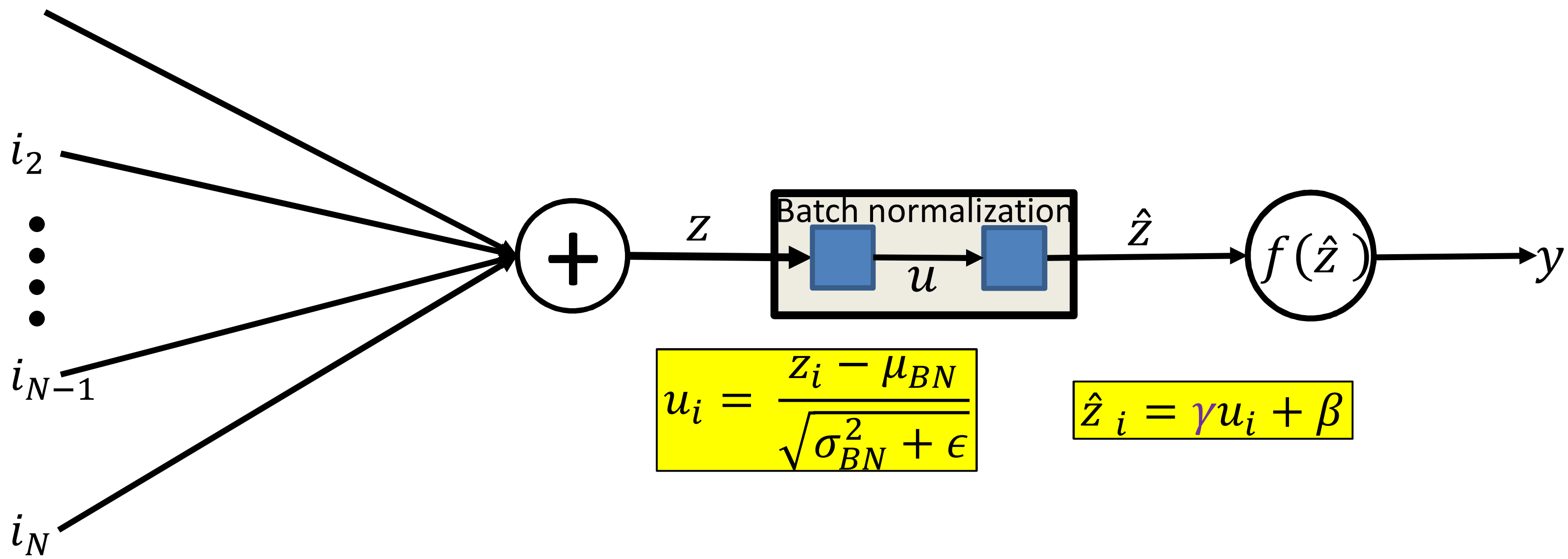
$$\frac{\partial Div}{\partial \mu_B} = \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} \sum_{i=1}^{B} \frac{\partial Div}{\partial u_i}$$

$$\frac{\partial Div}{\partial z_i} = \frac{\partial Div}{\partial u_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial Div}{\partial \sigma_B^2} \cdot \frac{2(z_i - \mu_B)}{B} + \frac{\partial Div}{\partial \mu_B} \cdot \frac{1}{B}$$



The rest of backprop continues from $\frac{\partial Div}{\partial z_i}$

Batch normalization: Inference



$$u_i = \frac{z_i - \mu_{BN}}{\sqrt{\sigma_{BN}^2 + \epsilon}}$$

$$\hat{z}_i = \gamma u_i + \beta$$

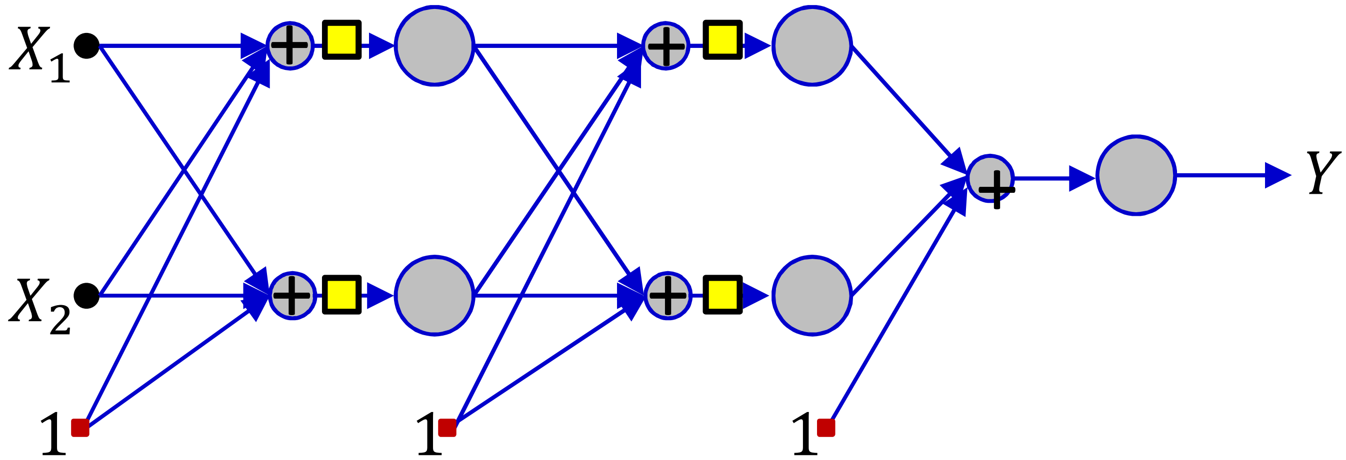
- On test data, BN requires μ_B and σ_B^2.
- We will use the *average over all training minibatches*

$$\mu_{BN} = \frac{1}{Nbatches} \sum_{batch} \mu_B(batch)$$

$$\sigma_{BN}^2 = \frac{B}{(B-1)Nbatches} \sum_{batch} \sigma_B^2(batch)$$

- Note: these are *neuron-specific*
  - $\mu_B(batch)$ and $\sigma_B^2(batch)$ here are obtained from the *final converged network*
  - The $B/(B-1)$ term gives us an unbiased estimator for the variance

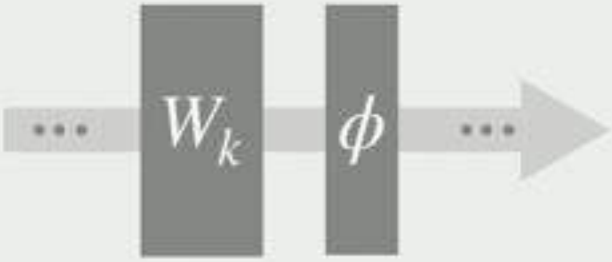
Batch normalization



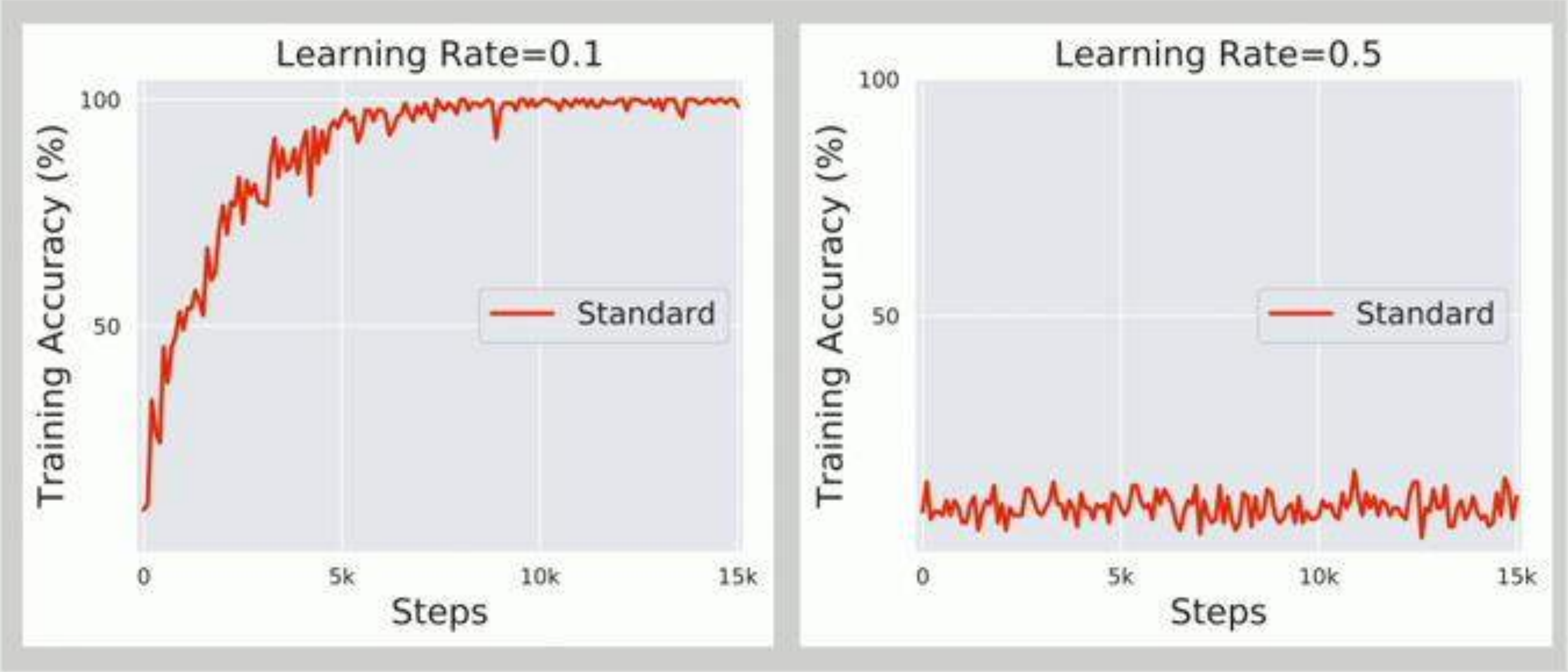
- Batch normalization may only be applied to *some* layers
  – Or even only selected neurons in the layer
- Improves both convergence rate and neural network performance
  – Anecdotal evidence that BN eliminates the need for dropout
  – To get maximum benefit from BN, learning rates must be increased and learning rate decay can be faster
    - Since the data generally remain in the high-gradient regions of the activations
  – Also needs better randomization of training data order

Why do we use BatchNorm?

BatchNorm's Role in Optimization

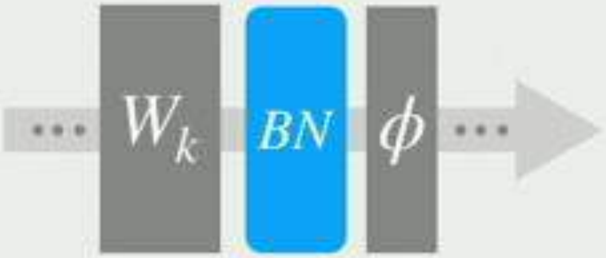


Network **without** BatchNorm

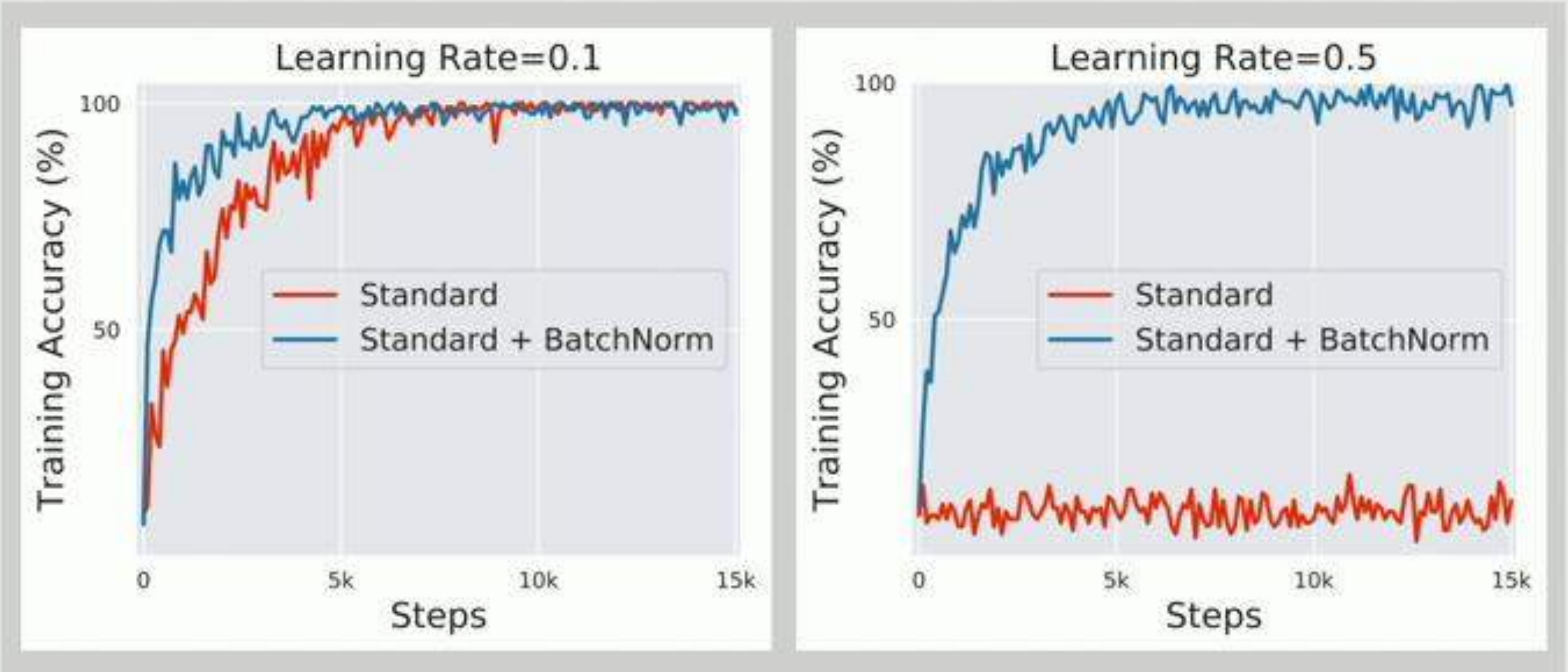


[Simoyan and Zisserman, 2014] [Krizhevsky and Hinton, 2009]

BatchNorm's Role in Optimization



Network **with** BatchNorm

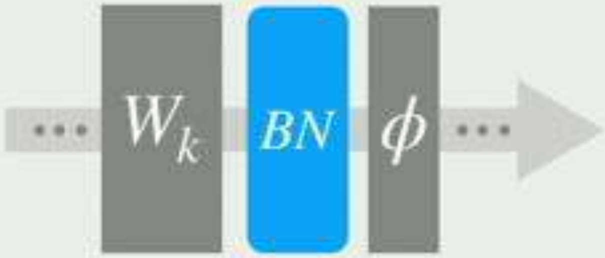


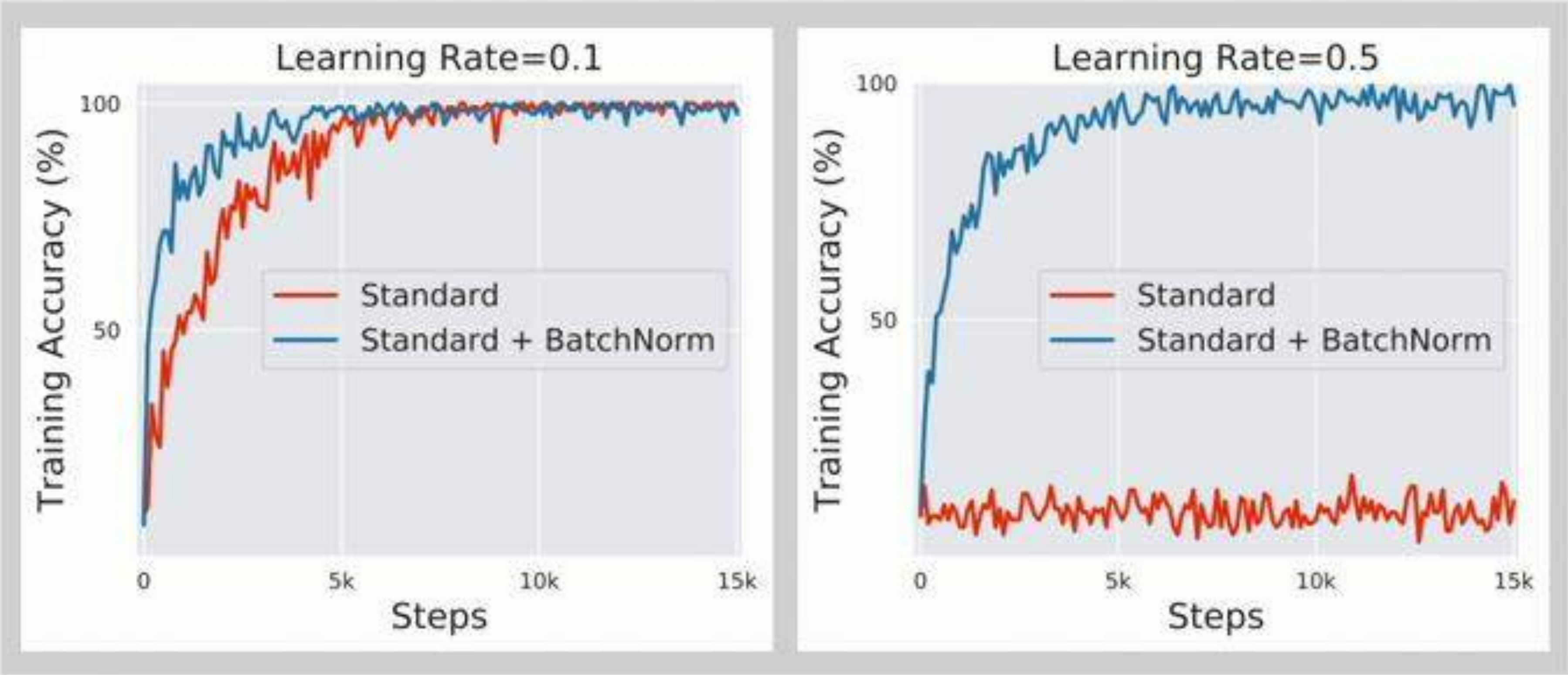
Faster Convergence

Robust to Hyperparameters

BatchNorm’s Role in Optimization

Network **with** BatchNorm





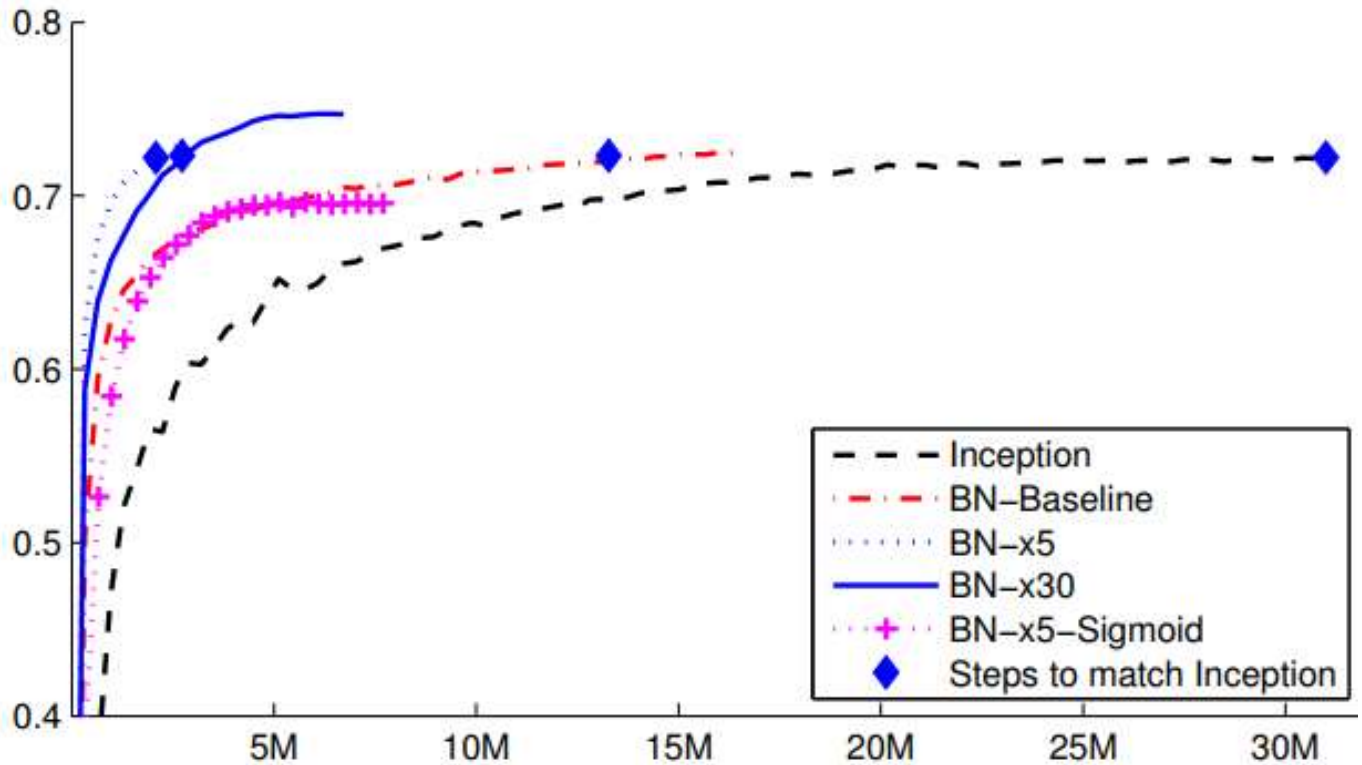
Faster Convergence

Robust to Hyperparameters

One of the most influential techniques in DNN training

Default in almost all standard architectures

Batch Normalization: Typical result



- Performance on Imagenet, from Ioffe and Szegedy, JMLR 2015