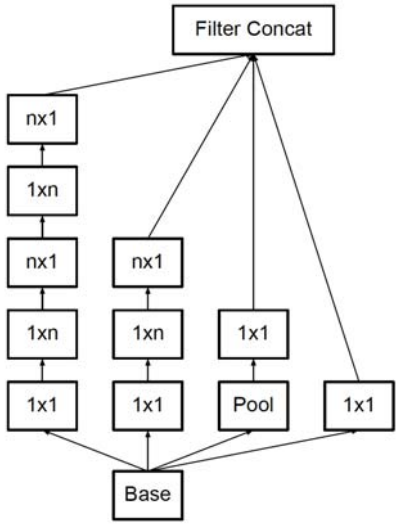
Batch Normalization (BN)

- Recap: Xavier/MSRA init are not directly applicable for multi-branch nets

- Optimizing multi-branch ConvNets largely benefits from BN
  - including all Inceptions and ResNets



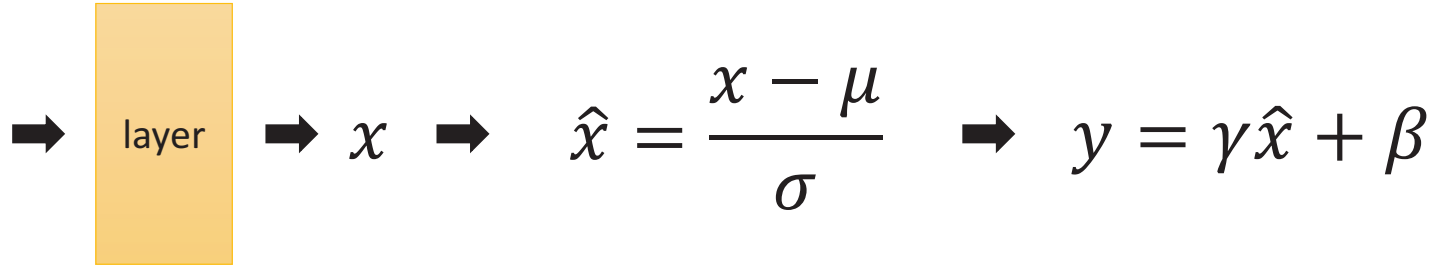
Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015.

Batch Normalization (BN)

- Recap: Normalizing image input (LeCun et al 1998 “Efficient Backprop”)
- Xavier/MSRA init: Analytic normalizing each layer
- BN: data-driven normalizing each layer, for each mini-batch
  - Greatly accelerate training
  - Less sensitive to initialization
  - Improve regularization

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015.

Batch Normalization (BN)

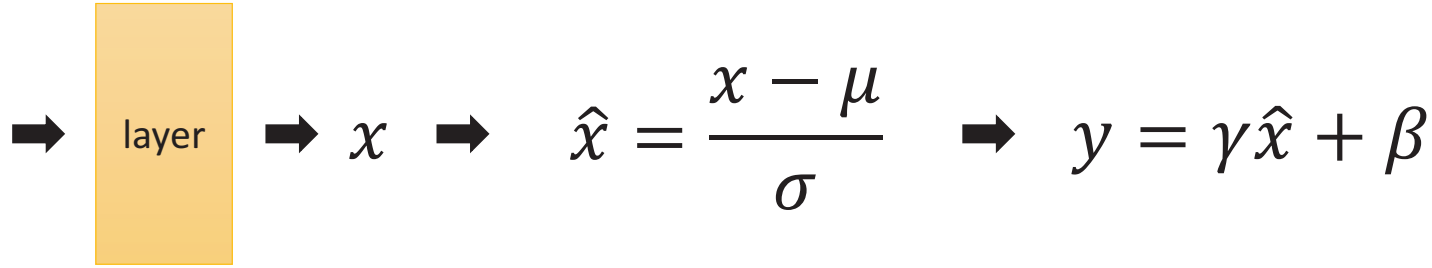


- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015.

Batch Normalization (BN)



2 modes of BN:

- Train mode:
  - μ, σ are functions of a batch of x
- Test mode:
  - μ, σ are pre-computed* on training set

Caution: make sure your BN usage is correct!
(this causes many of my bugs in my research experience!)

*: by running average, or post-processing after training

Ioffe & Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015.

Batch Normalization (BN)

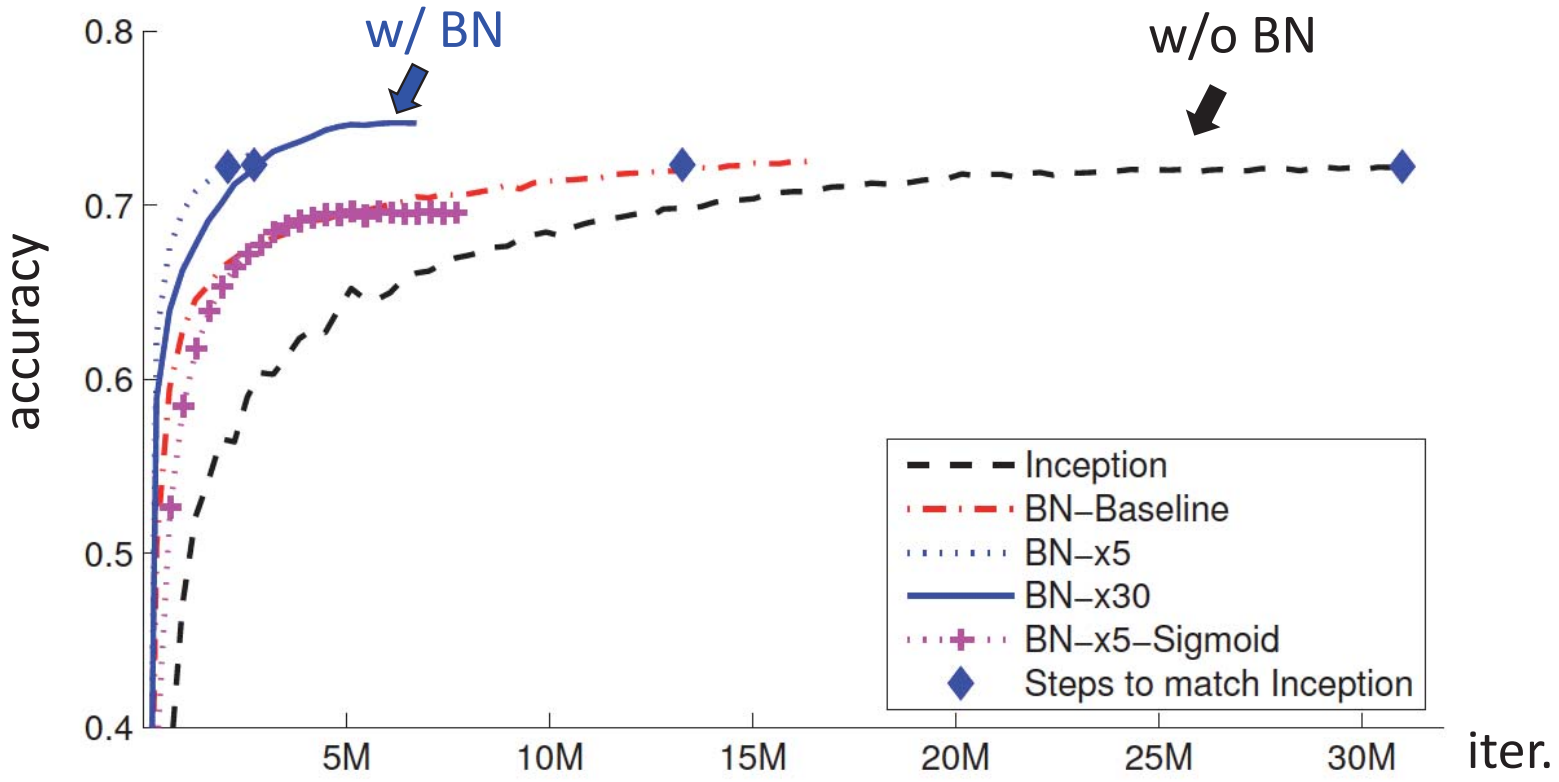


Figure credit: Ioffe & Szegedy

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015.