Batch Normalization

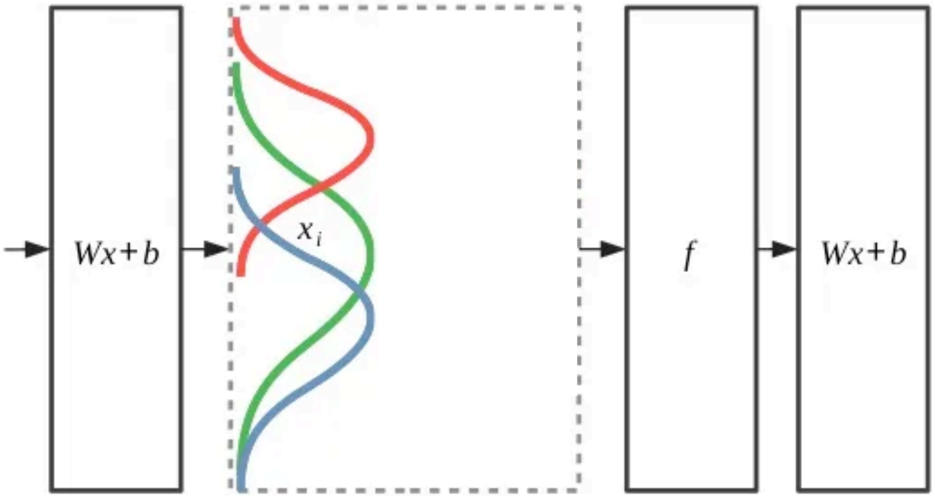
Sergey Ioffe & Christian Szegedy
Google Inc.
ICML 2015

DMAIS@CAU
Yeongon Kim

INDEX

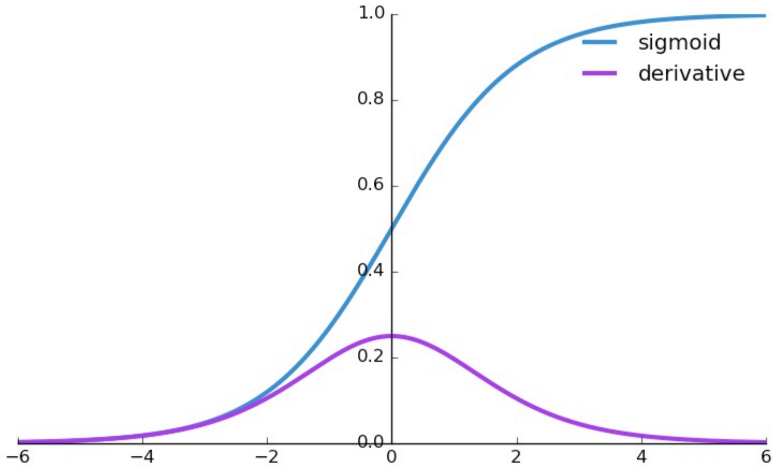
Internal Covariate Shift

- Changes in distribution of layer's inputs during training
- Slows down the training by requiring lower learning rates and careful parameter initialization



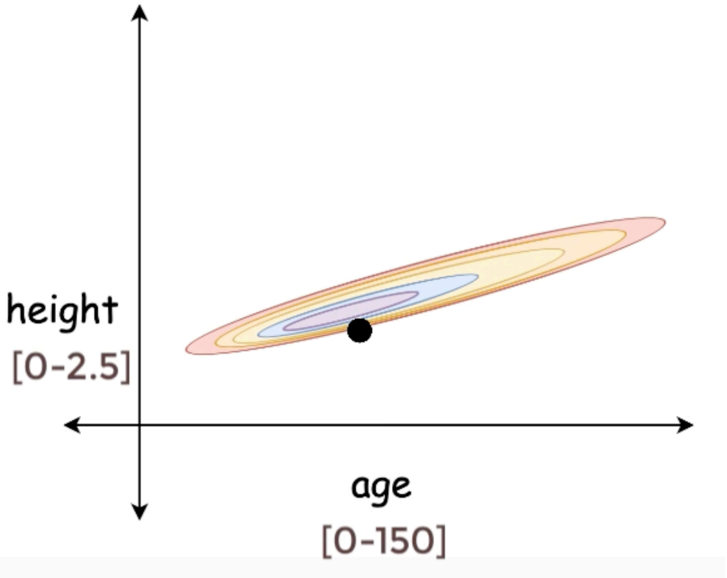
Internal Covariate Shift

- ❑ ICS makes it hard to train models with saturating nonlinearities
- ❑ As $|x|$ increases, derivative of sigmoid tends to zero
- ➡ ICS can move many dimensions of x into the saturated regime



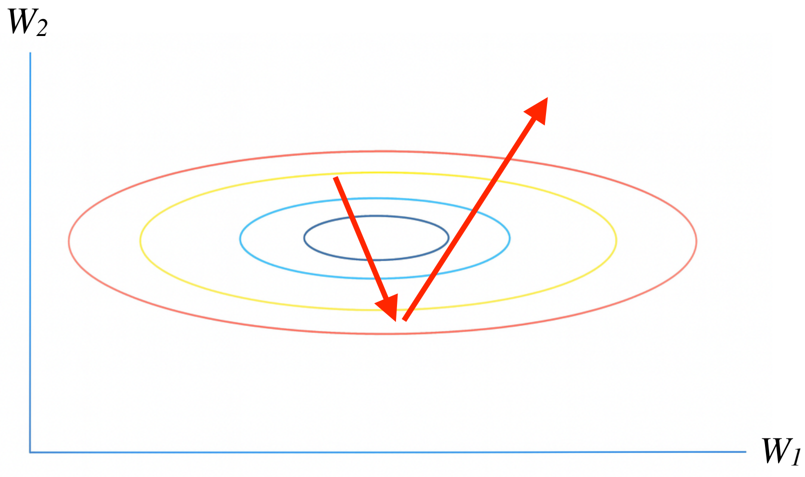
Valley-Shaped Loss Surface

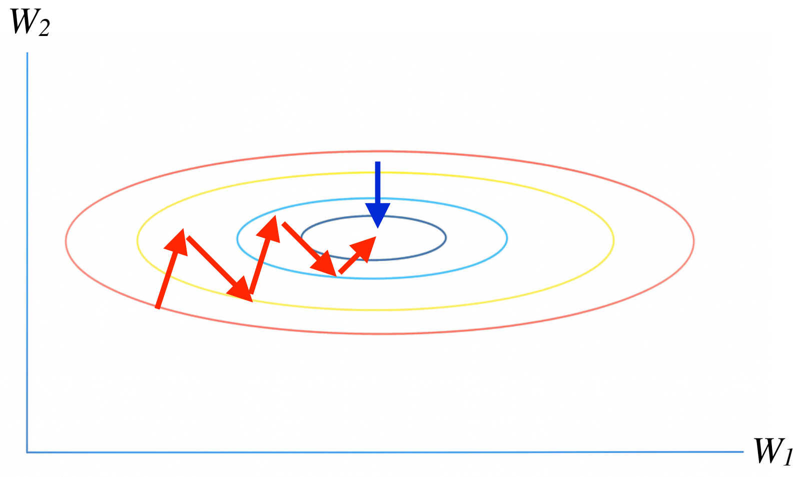
- Large-scale input features cause significant output changes even with small weight updates



Valley-Shaped Loss Surface

- Loss is more sensitive to the weight whose input feature has the larger scale
- Requiring lower learning rates and careful parameter initialization





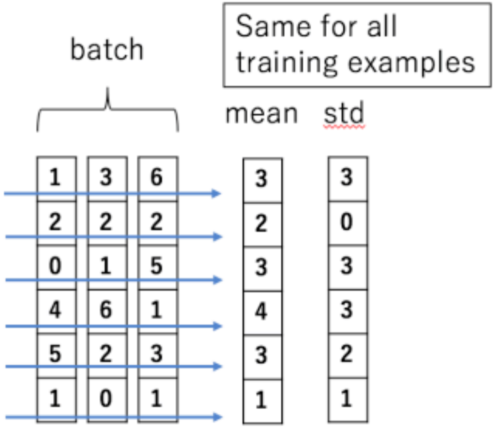
INDEX

- Introduction
  - Internal Covariate Shift
  - Valley-Shaped Loss Surface
- Batch Normalization
- Benefits of Batch Normalization
- Experiment
- Conclusion

Batch Normalization

- Each mini-batch produces estimates of the mean and variance of each activation
- Normalize each scalar feature independently

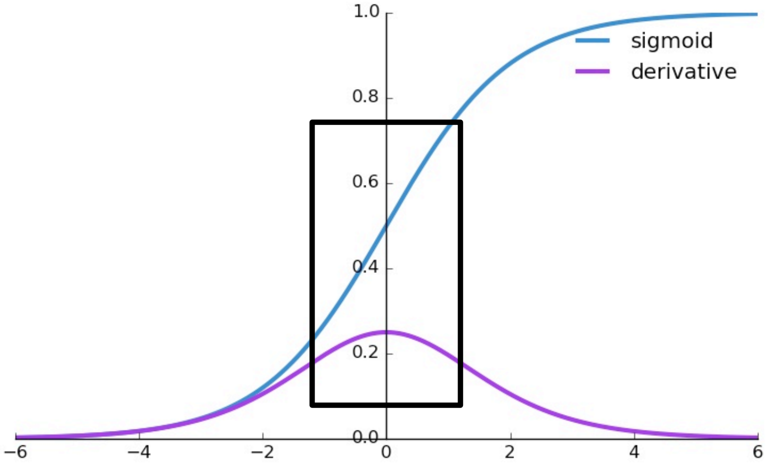
$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$



Batch Normalization

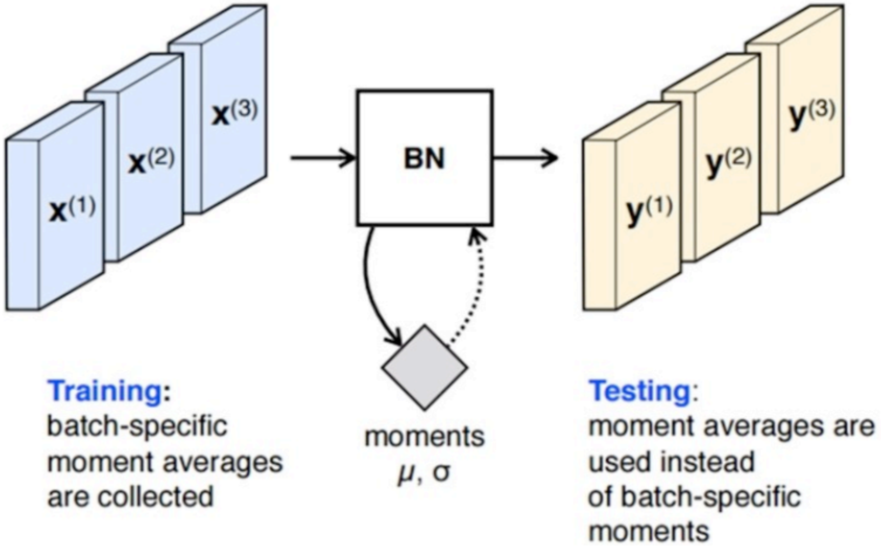
- Simply normalizing each input of layer may change what the layer can represent
  - ➡ Learnable parameters are inserted in each activation

$$y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)}$$



Batch Normalization

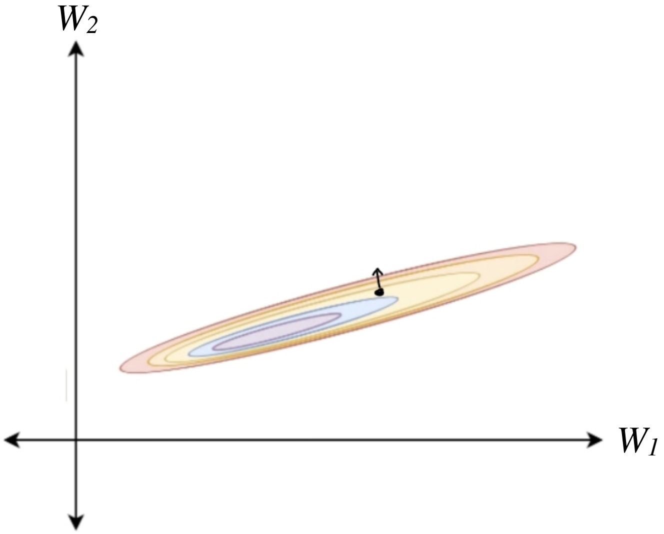
- Inference phase
  - Uses the running mean/variance computed during training
  - Enables reliable normalization even with small batch sizes

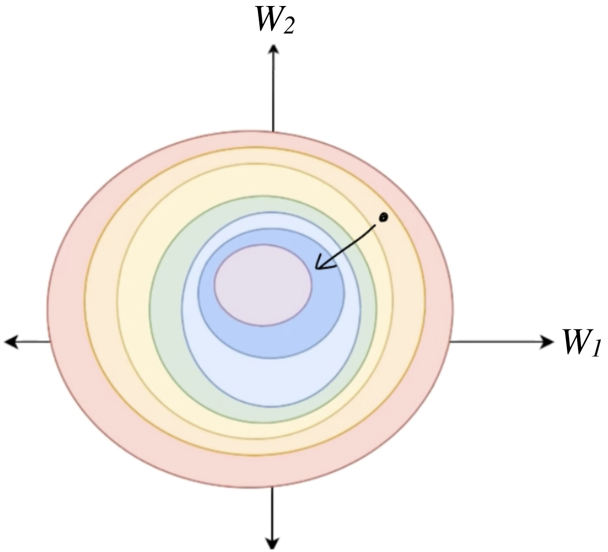


Benefits of Batch Normalization

❑ Accelerates training speed

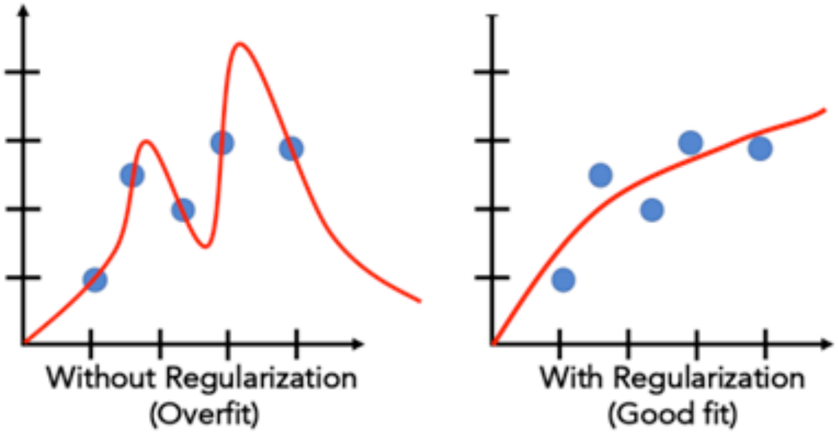
- Makes loss surface significantly smoother
- Smoothness induces predictive gradients, allowing for a larger learning rate
- Reduces sensitivity to initialization





Benefits of Batch Normalization

- **Regularizes the model**
  - Non-deterministic outputs can little yield different results for the same sample
  - Slight randomness helps prevent overfitting and improves generalization



INDEX

Experiments

- MNIST dataset with MLP

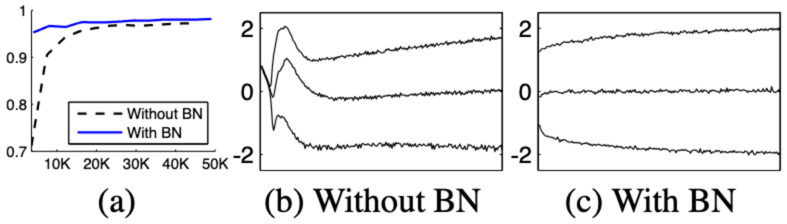


Figure 1: (a) *The test accuracy of the MNIST network trained with and without Batch Normalization, vs. the number of training steps. Batch Normalization helps the network train faster and achieve higher accuracy.* (b, c) *The evolution of input distributions to a typical sigmoid, over the course of training, shown as* $\{15, 50, 85\}$*th percentiles. Batch Normalization makes the distribution more stable and reduces the internal covariate shift.*

Experiments

❑ ImageNet classification with CNN

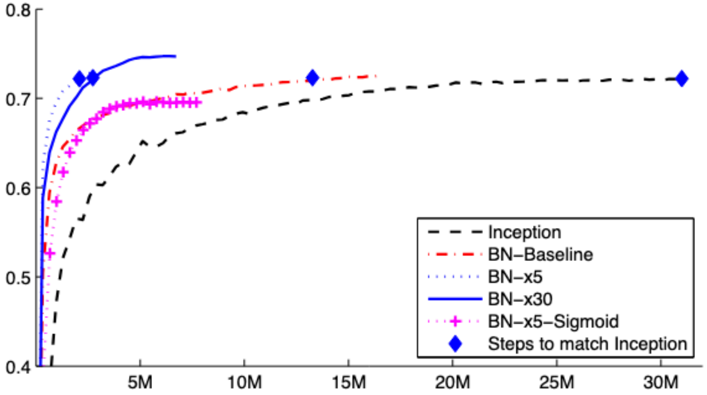


Figure 2: *Single crop validation accuracy of Inception and its batch-normalized variants, vs. the number of training steps.*

Model	Steps to 72.2%	Max accuracy
Inception	$31.0 \cdot 10^6$	72.2%
BN-Baseline	$13.3 \cdot 10^6$	72.7%
BN-x5	$2.1 \cdot 10^6$	73.0%
BN-x30	$2.7 \cdot 10^6$	74.8%
BN-x5-Sigmoid		69.8%

Figure 3: *For Inception and the batch-normalized variants, the number of training steps required to reach the maximum accuracy of Inception (72.2%), and the maximum accuracy achieved by the network.*

Conclusion

- Batch normalization reduces internal covariant shift, accelerating training
- Smoother loss surface allows larger learning rate reduces sensitivity to initialization
- Also batch normalization regularizes the model

Thank you!