

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Author: Ioffe et al.

created by paper2slides

Released: 2015-02-11

Executive Summary

- **Problem:** Internal covariate shift causes slow and complicated training of deep neural networks.
- **Proposed Solution:** Batch Normalization (BN) normalizes each layer's inputs within each mini-batch.
- **Advantages:**
 - Allows higher learning rates and reduces sensitivity to initialization.
 - Acts as a regularizer, potentially eliminating the need for Dropout.
 - Achieves state-of-the-art accuracy with fewer training steps.
- **Performance:** On ImageNet, BN reduces training steps by 14x while improving accuracy.

Introduction: Background

- Deep learning has significantly advanced fields such as vision and speech.
- Training deep networks with SGD can be complex due to changing input distributions in each layer.
- This issue increases training time and complicates network optimization.
- Called **Internal Covariate Shift**, this phenomenon occurs as parameters shift during training.

Introduction: Motivation

- Normalizing inputs is known to speed up training.
- Extending this idea, we normalize activations in intermediate layers of the network.
- This can potentially stabilize and accelerate training, especially for deep networks.
- **Goal:** Reduce the internal covariate shift within the network.

Core Idea: Batch Normalization (BN)

- BN normalizes layer inputs within each mini-batch.
- Ensures mean of 0 and variance of 1 for each activation.
- BN layer can be directly inserted into existing architectures.
- For each activation x :

$$\hat{x} = \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}}$$

$$\text{BN}(x) = \gamma \hat{x} + \beta$$

- γ and β are learned parameters.

Why Batch Normalization?

- **Stabilizes Learning:** Normalizes inputs to have fixed distribution.
- **Improves Gradient Flow:** Reduces vanishing/exploding gradients.
- **Higher Learning Rates:** Enables use of higher learning rates without risking model divergence.
- **Regularization:** Acts as its own form of regularization.

Training and Inference with BN

- During training, use mini-batch statistics to normalize.
- At inference time, use moving averages of training data statistics.
- Transformation becomes deterministic during inference for consistent outputs.
- Algorithm applies the same linear transform for activations in each feature map during inference.

Batch Normalization for Convolutional Layers

- BN applied to the input of nonlinearities: $\mathbf{z} = g(\text{BN}(\mathbf{W}\mathbf{u}))$
- Normalizes over all elements in a feature map across mini-batch examples.
- Parameters γ and β are learned per feature map.
- Helps mitigate internal covariate shift particularly in deep convolutional networks.

Experimental Setup: MNIST

- Task: Digit classification on MNIST dataset.
- Model: Simple network with 3 hidden layers of 100 neurons each.
- Sigmoid activations, trained for 50000 steps with mini-batch size of 60.

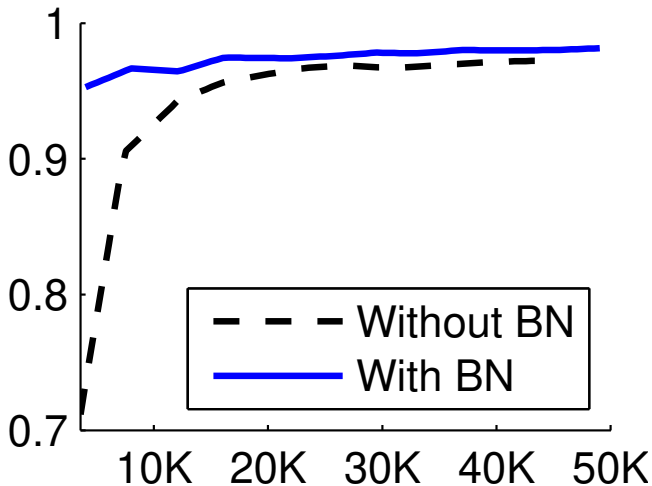


Figure: Test accuracy on MNIST with and without Batch Normalization.

Experimental Setup: ImageNet

- Task: Image classification on ImageNet dataset.
- Model: Modified Inception network with convolutional layers followed by ReLU.
- Trained using SGD with momentum, mini-batch size 32.
- Compared standard Inception to batch-normalized variants.

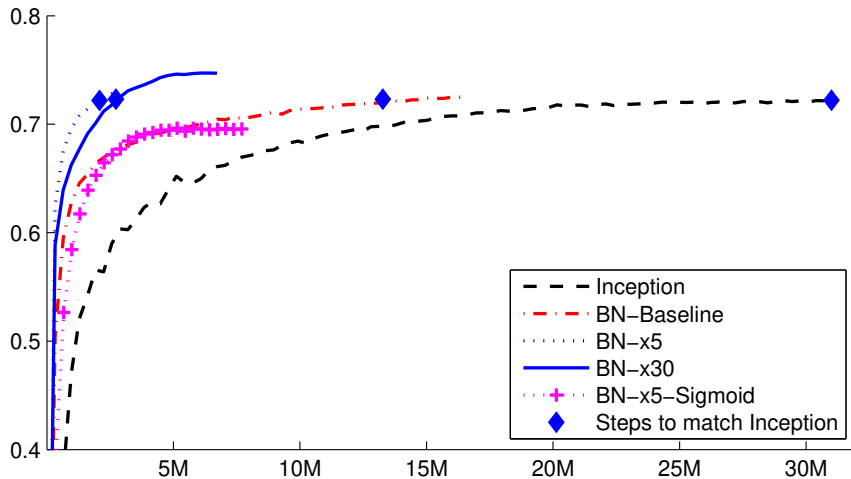


Figure: Validation accuracy for various Inception model variants.

Batch-Normalized Inception Models

- **Inception**: Original network, max accuracy 72.2% after 31M steps.
- **BN-Baseline**: BN applied, 72.2% accuracy in 13.3M steps.
- **BN-x5**: Includes modifications for faster training, 72.2% accuracy in 2.1M steps.
- **BN-x30**: Higher learning rate, 74.8% accuracy in 2.7M steps.

Power of Batch Normalization

- **Higher Learning Rates:** BN enables significantly higher learning rates, e.g., 5x-30x.
- **Removed Dropout:** Reduces need for Dropout.
- **Enabled Sigmoid Nonlinearities:** BN allows training deep networks with sigmoid.
- **Regularization:** Acts as a built-in regularizer.

Conclusion and Future Work

■ Conclusion:

- Batch Normalization accelerates deep network training.
- Achieves state-of-the-art results with fewer training steps.

■ Future Work:

- Apply BN to Recurrent Neural Networks (RNNs).
- Investigate BN's impact on domain adaptation.
- Further theoretical analysis of Batch Normalization.