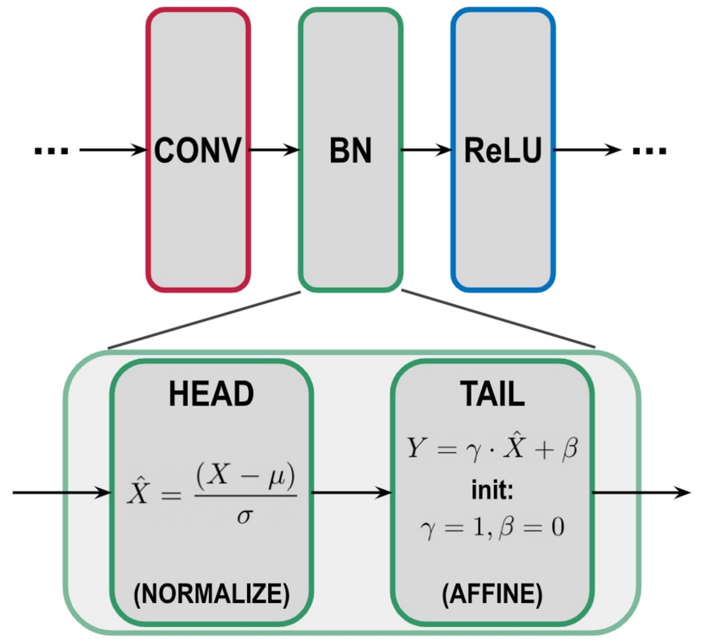
Revisiting Batch Norm Initialization

Jim Davis, **Logan Frank**
Department of Computer Science and Engineering, Ohio State University

1. Batch Normalization (BN) in Deep Neural Networks

Two per-channel operations
- **Head:** Normalizes data
- **Tail:** Learnable affine transformation

Constrains intermediate features, enabling smoother and faster optimization, and stochasticity of batch statistics can benefit generalization



2. BN: Forward Formulation

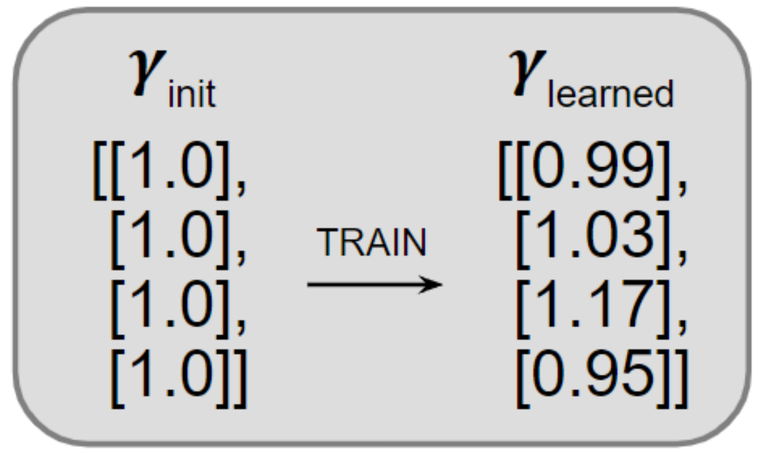
- Compute mean (μ) and variance (σ^2) across batch dimension
- Use computed statistics to normalize the data ($\mu = 0,\ \sigma^2 = 1$)
- Apply an affine transformation to the normalized data using learnable parameters: scale (γ) and shift (β)

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \;\rightarrow\; \hat{X} = \frac{X - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \;\longrightarrow\; Y = \gamma \cdot \hat{X} + \beta$$

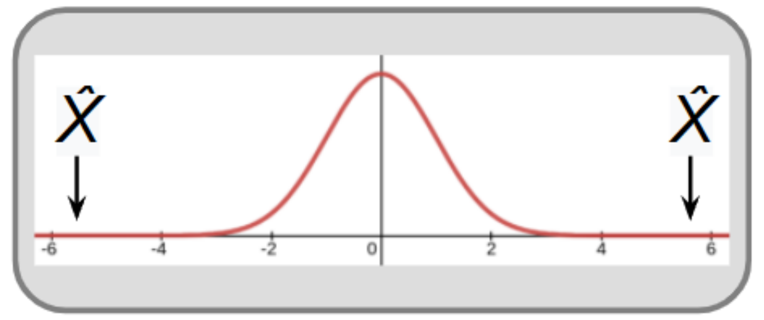
3. Observed Issues with BN

Learnable BN parameters are initialized to γ = 1 and β = 0 (identity function)

We observed that the final learned parameter values tend to remain close to their initialization



Furthermore, we observed that the BN normalization head can yield overly large values ($\pm 6\sigma$) for the proceeding layer, which can be undesirable for training



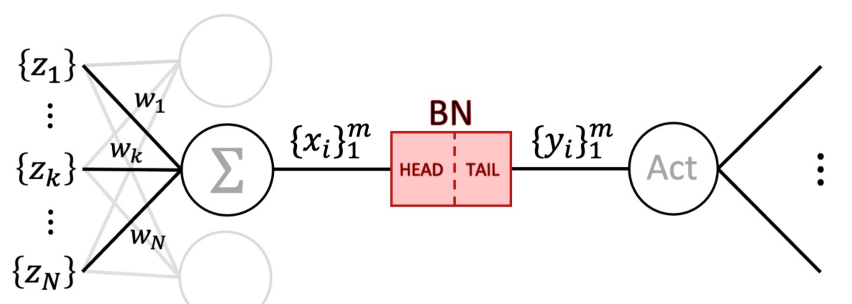
4. Proposed Adjustments to BN Scale Parameter γ

Initialize γ to a value in $(0, 1]$
- Directly addresses overly large values after normalization by immediately scaling down the data (with *no additional parameters*)
- Enables BN shift parameter β to have a broader reach on scaled data before the proceeding activation function (in many cases ReLU)

Reduce the learning rate α on γ
- Divide learning rate on γ by constant c ($\alpha_\gamma = \alpha / c$)
- Allows for fine-grained search near initialization value
- Leave β with original learning rate, enabling it to have a broader and now more stable search of the normalized and scaled data

5. BN: Gradients and Insights

Using a fully-connected layer as illustration (below, left), we use the gradients given in the original BN paper to derive the gradient of the loss with respect to the input $\partial\mathcal{L}/\partial x_i$ (below, right)



$$\frac{\partial\mathcal{L}}{\partial x_i} = \frac{\gamma}{\sqrt{\sigma_B^2 + \epsilon}} \cdot \left[\frac{\partial\mathcal{L}}{\partial y_i} - \frac{1}{m}\sum_{i=1}^{m}\frac{\partial\mathcal{L}}{\partial y_i}(1 - \hat{x}_i) \right]$$

$$= \frac{\gamma}{\sqrt{\sigma_B^2 + \epsilon}} \cdot G_i$$

Insights
- No effects introduced by $\gamma < 1$ for the first backward pass
- *Negligible* effects for remainder of training
- More insights shown in the paper

$$\sigma_B^2 = \sigma_{act}^2 \cdot \gamma_{prev}^2 \sum_{k=1}^{N} w_k^2 \longrightarrow \frac{\gamma_{curr}}{\gamma_{prev} \cdot \sqrt{\sigma_{act}^2 \sum w_k^2 + \epsilon}} = \frac{1}{\sqrt{\sigma_{act}^2 \sum w_k^2 + \epsilon}}$$

6. Training Details

BN scale initialization: $\gamma \in \{\, 0.01,\ 0.05,\ 0.1,\ 0.25,\ 0.5,\ 0.75,\ 1.0 \,\}$
- Examine subset of values after initial CIFAR-10 experiments

BN scale learning rate reduction factor: c = 100

7. Statistical Significance

Different RNG seeds can cause variations in final score (accuracy)

For ***each*** experiment, we conduct 15 runs with different seeds, aggregate the results of each run (to report a mean and standard deviation), and compare to a baseline (or related approach) using a one-sided paired t-test (using a p-value of 0.05)

8. Results

- Significant improvements across multiple initial values of γ and learning rates for CIFAR-10 (T1.a), as well as CIFAR-100, CUB-200, and Stanford Cars (T2.b)
- Even greater gains with deeper network architectures (T2.a)
- Outperforms other existing related approaches which require additional parameters and computations (T2.b)

γ	Learning Rate (α) 0.1	0.01	0.001
0.01	85.50 ± 0.39	**87.11** ± 0.23	**80.37** ± 0.58
0.05	**90.19** ± 0.32	**88.84** ± 0.32	**76.98** ± 0.71
0.10	**90.80** ± 0.20	**87.31** ± 0.37	**74.48** ± 0.55
0.25	**90.32** ± 0.24	**85.33** ± 0.43	**73.83** ± 0.64
0.50	**90.17** ± 0.19	84.60 ± 0.35	**72.80** ± 0.68
0.75	**90.19** ± 0.18	84.43 ± 0.30	**72.01** ± 0.58
1.00	**89.81** ± 0.46	84.48 ± 0.33	71.15 ± 0.56
BASE	89.44 ± 0.45	84.64 ± 0.25	71.32 ± 0.60

Table 1.a

Network	γ	Learning Rate (α) 0.1	0.01
ResNet-50	0.05	**91.23** ± 0.20	**89.60** ± 0.19
	0.10	**91.28** ± 0.26	**87.67** ± 0.20
	0.50	**89.49** ± 0.27	84.74 ± 0.37
	BASE	86.94 ± 1.23	85.04 ± 0.32
ResNet-101	0.05	**91.58** ± 0.22	**90.02** ± 0.22
	0.10	**91.26** ± 0.18	**88.35** ± 0.28
	0.50	**89.89** ± 0.74	**85.23** ± 0.50
	BASE	88.28 ± 1.39	84.74 ± 0.56
ResNet-152	0.05	**91.20** ± 0.16	**90.00** ± 0.17
	0.10	**90.89** ± 0.41	**88.31** ± 0.33
	0.50	**90.17** ± 0.23	**85.23** ± 0.62
	BASE	88.73 ± 0.62	84.15 ± 0.79

Table 2.a

Dataset	Method	Learning Rate (α) 0.1	0.01
CIFAR10	RBN	**90.17** ± 0.22	84.72 ± 0.29
	RBN⁻	**90.11** ± 0.24	84.50 ± 0.36
	IEBN	**90.18** ± 0.26	**85.34** ± 0.39
	IEBN⁻	**90.15** ± 0.24	**85.29** ± 0.35
	Ours	**90.80** ± 0.20	**88.84** ± 0.32
	BASE	89.44 ± 0.45	84.64 ± 0.25
CIFAR100	RBN	**66.95** ± 0.57	**58.95** ± 0.42
	RBN⁻	**66.82** ± 0.55	**58.90** ± 0.61
	IEBN	**66.94** ± 0.39	**60.61** ± 0.40
	IEBN⁻	**66.95** ± 0.32	**60.89** ± 0.41
	Ours	**68.80** ± 0.49	**64.01** ± 0.54
	BASE	66.01 ± 0.95	58.48 ± 0.53
CUB-200	RBN	48.68 ± 1.56	**44.68** ± 0.59
	RBN⁻	47.14 ± 2.72	**43.02** ± 1.22
	IEBN	**54.12** ± 0.60	**44.92** ± 0.74
	IEBN⁻	**53.81** ± 0.76	**44.09** ± 0.65
	Ours	**58.52** ± 0.69	**45.31** ± 0.59
	BASE	46.26 ± 1.59	41.61 ± 1.03
ST-Cars	RBN	**68.17** ± 1.84	51.87 ± 1.34
	RBN⁻	**67.84** ± 2.96	52.30 ± 1.73
	IEBN	**73.60** ± 0.92	51.06 ± 0.87
	IEBN⁻	**74.04** ± 1.55	51.08 ± 0.78
	Ours	**78.29** ± 0.44	51.18 ± 2.16
	BASE	64.73 ± 2.87	51.86 ± 1.80

Table 2.b

9. QR Codes:

Paper

GitHub

Supported by the U.S. Air Force Research Laboratory, Sensors Directorate