Loss Landscape Geometry & Optimization Dynamics in Neural Networks: A Comparative Study of Batch-Normalized and Non-Normalized CNNs

Rajat Abhijit Kambale

November 27, 2025

Abstract

Why stochastic gradient descent (SGD) consistently finds solutions that generalize well—despite navigating a highly non-convex and high-dimensional loss landscape—remains one of the most intriguing questions in modern deep learning theory. This work develops a structured framework connecting geometric properties of the loss landscape—curvature, sharpness, flatness, and basin topology—to optimization behavior, architectural choices, and generalization outcomes. Using Fashion-MNIST and two carefully controlled convolutional networks (with and without Batch Normalization), we measure Hessian curvature, perturbation sharpness, loss slices, and mode connectivity paths. Interestingly, although the BatchNorm model generalizes better, it appears locally sharper under multiple metrics, highlighting that simple notions of flatness do not fully determine generalization. Our findings point toward a more nuanced relationship between architecture, optimization noise, and landscape geometry.

Contents

1. Introduction

Deep neural networks are trained in extremely high-dimensional, non-convex landscapes that, in theory, should present optimization challenges. Yet SGD reliably converges to solutions that perform well on unseen data. This apparent contradiction motivates a deeper examination of how geometry influences the optimization process.

In this work, we explore:

- Why SGD locates good minima despite non-convexity.
- How Batch Normalization reshapes the loss landscape.
- Which geometric attributes meaningfully correlate with generalization.
- Whether early curvature signals can predict optimization difficulty.

To investigate these questions, we compare two convolutional neural networks that differ only by the presence of Batch Normalization.

2. Theoretical Framework

2.1 Loss Geometry

For parameters θ and dataset $\{(x_i, y_i)\}$, the empirical loss is:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i).$$

The gradient $g(\theta) = \nabla_\theta L(\theta)$ guides optimization, while the Hessian $H(\theta) = \nabla_\theta^2 L(\theta)$ captures local curvature.

A commonly used sharpness proxy is the largest eigenvalue $\lambda_{\max}$ of the Hessian.

2.2 Perturbation Sharpness

We approximate local sensitivity using SAM-style adversarial perturbations:

$$S(\theta) = \max_{\|\delta\| \le \epsilon} \left[L(\theta + \delta) - L(\theta) \right].$$

2.3 SGD as a Stochastic Differential Equation

SGD can be modeled as:

$$d\theta_t = -\nabla L(\theta_t)\,dt + \sqrt{2T}\,dW_t, \quad T \propto \frac{\eta}{B}.$$

Near minima, the stationary distribution behaves as:

$$p(\theta) \propto \frac{\exp(-L(\theta)/T)}{\sqrt{\det H(\theta)}}.$$

This suggests SGD favors broader minima, although curvature alone rarely paints the full picture.

3. Experimental Setup

3.1 Dataset

We use Fashion-MNIST (60k training, 10k test), normalized to $(0.5, 0.5)$.

3.2 Models

Two nearly identical CNNs were designed:

- **Model A1**: CNN with BatchNorm (seed = 1)
- **Model A2**: CNN with BatchNorm (seed = 2)
- **Model B**: CNN without BatchNorm (seed = 3)

All models share:

- Two convolutional layers (32 and 64 channels)
- ReLU activations
- MaxPool(2)
- Fully connected layers (128 hidden units)

3.3 Training Configuration

- Optimizer: SGD + momentum 0.9
- Learning rate: 0.01 with StepLR
- Epochs: 6
- Batch size: 128

4. Results

4.1 Training and Test Performance

Batch Normalization improves both convergence speed and generalization.

The training and test curves for Model A1 are shown in Fig. 1.

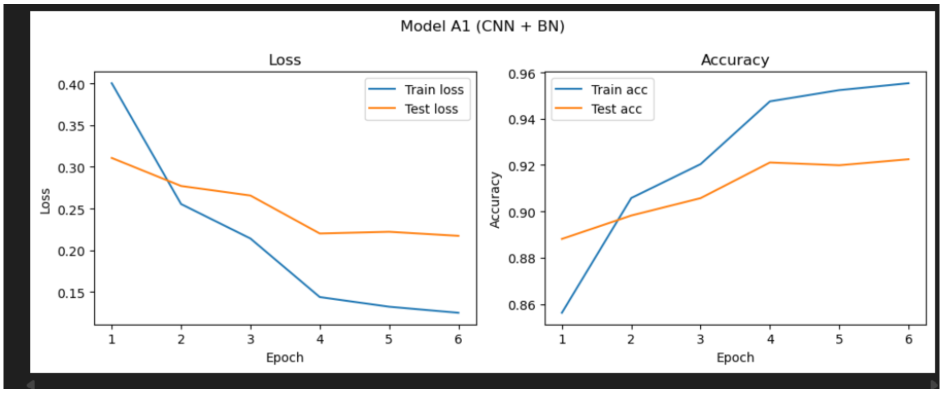

Figure 1: Training and test curves for Model A1 (CNN + BatchNorm).

The corresponding curves for Model B are shown in Fig. 2.

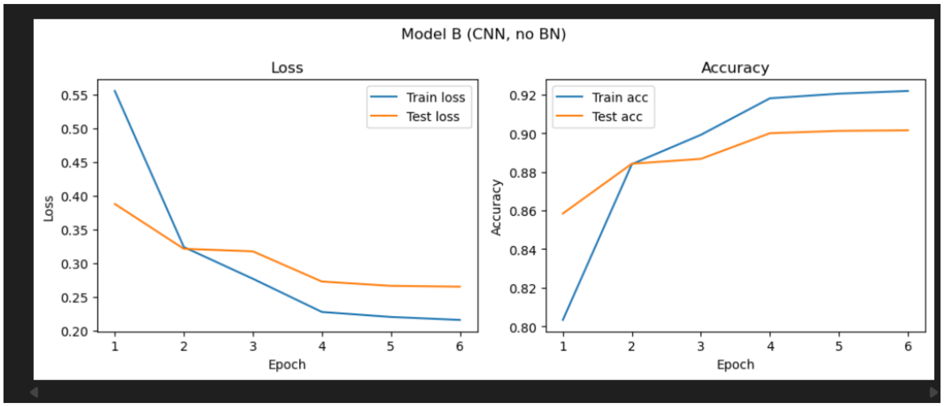

Figure 2: Training and test curves for Model B (No BatchNorm).

4.2 Curvature: Top Hessian Eigenvalue

$$\lambda_{\max}(A1) \approx 57.18, \qquad \lambda_{\max}(B) \approx 37.51.$$

4.3 Perturbation Sharpness

$$S(A1) \approx 0.1837, \qquad S(B) \approx 0.0886.$$

4.4 1D Loss Slice

Fig. 3 shows the loss along a random direction around the minima.

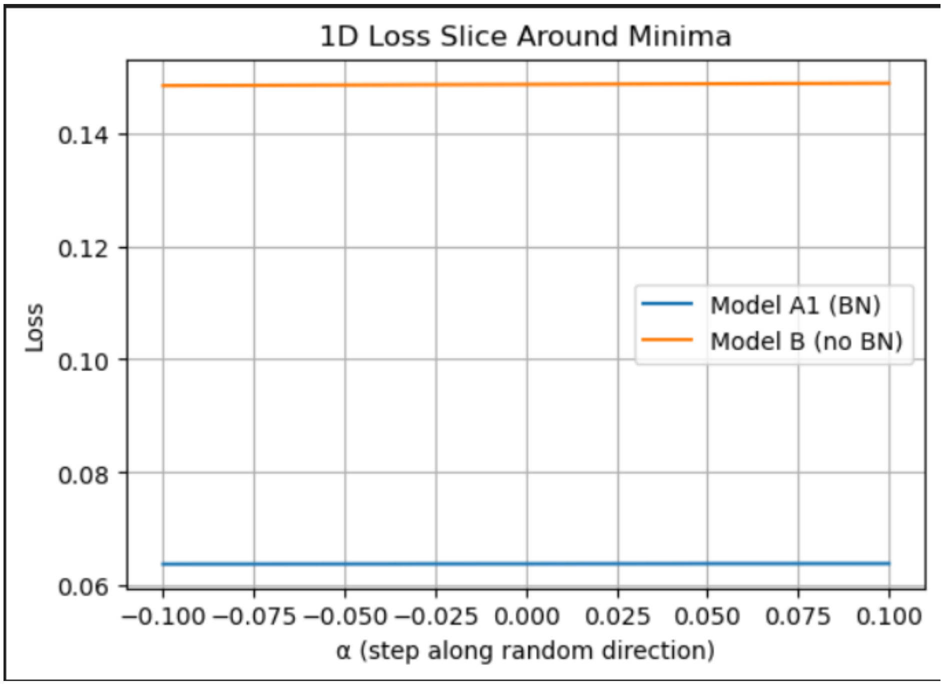

Figure 3: 1D loss slice for A1 (BN) and B (No BN).

4.5 Mode Connectivity (A1 A2)

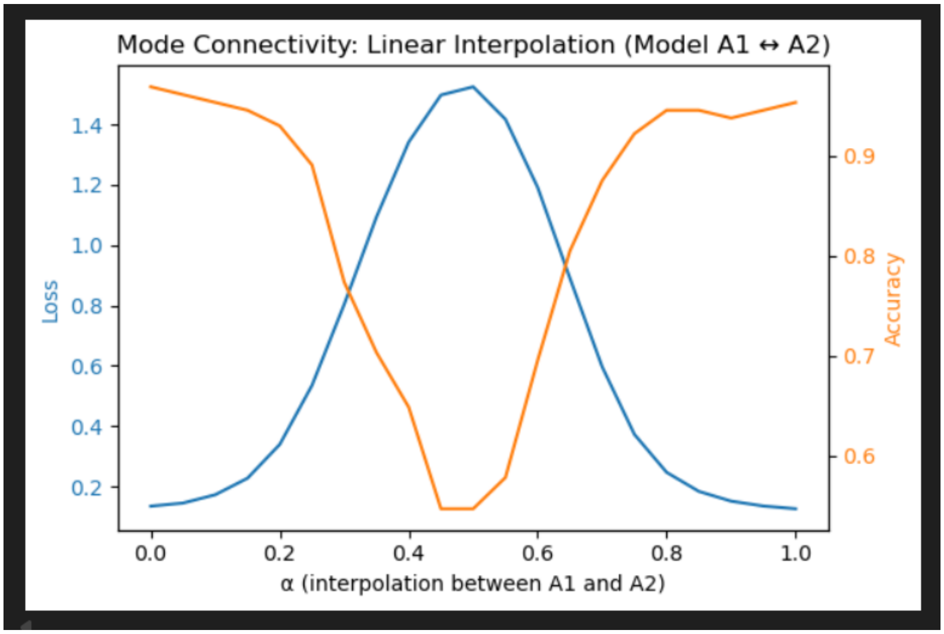

Figure 4: Linear interpolation between A1 and A2 reveals a loss barrier.

5. Discussion

5.1 BN Generalizes Better but Appears Sharper

BatchNorm improves accuracy yet produces higher curvature and greater perturbation sharpness. This indicates:

- local flatness is not the sole factor governing generalization,
- architectural effects can dominate naive sharpness measures,
- global basin shape matters more than local curvature.

5.2 Topology of Minima

The interpolation barrier between A1 and A2 suggests they reside in distinct basins—consistent with observations that linear mode connectivity rarely holds without curved paths.

5.3 Predicting Optimization Difficulty

Lower curvature in Model B does not translate to better optimization or generalization, showing early Hessian metrics alone are unreliable predictors.

6. Conclusion

This work presents a structured framework for analyzing neural network loss geometry and applies it to two closely matched CNN architectures. While Batch Normalization improves optimization stability and generalization, it also produces sharper local curvature—challenging the traditional notion that flatter minima always generalize better.

Overall, our findings highlight that generalization arises from a combination of local curvature, global basin geometry, architectural design, and the stochastic nature of SGD. These results emphasize the importance of evaluating multiple geometric perspectives rather than relying on any single metric when analyzing or designing neural network models.