Loss Landscape Geometry & Optimization Dynamics

Mallikarjun Jamadarkhani DA24S009

27/11/2025

Contents

1 Problem Context **1**

2 My Plan **1**

3 Mental Model **1**

4 Simple Theoretical View **2**

5 Probing techniques Implementation and Empirical Findings **3**

6 Answers to the Questions **6**

1 Problem Context

The goal of this assignment is to:

- develop a framework to analyse neural network loss landscapes,
- connect geometry to optimization dynamics and generalization,
- account for the role of architecture,
- and suggest how to probe and validate these ideas in practice.

Deep networks are highly non-convex. Still, SGD usually finds solutions that train well and generalize.

2 My Plan

- use a simple theoretical view of SGD and curvature,
- implement a small set of landscape probing tools,
- and test them on a concrete model (MNIST + MLP) to look for correlations.

3 Mental Model

Key intuition:

- Sharp minima: high curvature, sensitive to parameter noise.
- Flat minima: low curvature, more robust, occupy more “volume”.
- SGD noise (from minibatches) pushes the trajectory around.

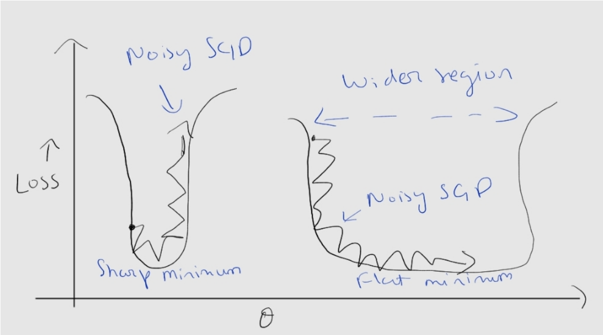

Figure 1: Toy 1D loss with one sharp minimum and one flat minimum. Noisy SGD tends to escape the sharp one, but it settles in the flat one.

4 Simple Theoretical View

I keep the theory intentionally simple but explicit.

a. Quadratic Approximation Around a Minimum

Consider parameters θ near a local minimum $\theta^\star$. Approximate the loss as:

$$L(\theta) \approx L(\theta^\star) + \tfrac{1}{2}(\theta - \theta^\star)^\top H(\theta - \theta^\star)$$

where H is the Hessian at $\theta^\star$.

In 1D, this becomes:

$$L(\theta) \approx L(\theta^\star) + \frac{h}{2}(\theta - \theta^\star)^2$$

with curvature $h > 0$.

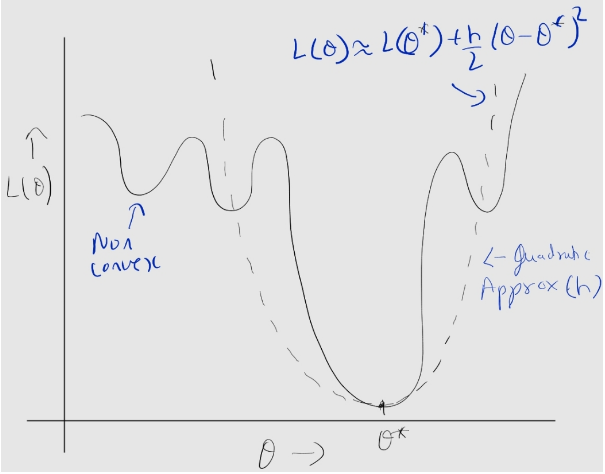

Figure 2: The quadratic approximation of a non-convex loss function $L(\theta)$ around a local minimum θ^*. The dashed parabola represents the local second-order Taylor expansion, $L(\theta) \approx L(\theta^*) + \frac{h}{2}(\theta - \theta^*)^2$, where h is the curvature.

b. SGD as Noisy Gradient Descent

In this 1D quadratic case, a noisy SGD update can be written as:

$$\theta_{t+1} = \theta_t - \eta\, h(\theta_t - \theta^\star) + \xi_t,$$

where:

- η is the learning rate,
- ξ_t is zero-mean noise from minibatch gradients.

This is a simple linear stochastic difference equation. At stationarity, the variance of θ_t around $\theta^\star$ (call it $\mathrm{Var}[\theta]$) is roughly:

$$\mathrm{Var}[\theta] \propto \frac{\sigma^2}{2\eta h},$$

For noise variance σ^2, larger curvature h concentrates the distribution, while smaller curvature (flatter minima) spreads it. In higher dimensions H, flat minima (many small eigenvalues) occupy more volume.

c. Implicit Bias Toward Flat Minima

In an energy-based view (continuous-time limit), the stationary density looks like:

$$p(\theta) \propto \exp\left(-\frac{L(\theta)}{T}\right),$$

where T is an effective temperature related to learning rate and batch size. When there are several minima with similar loss, wider minima correspond to larger regions with low $L(\theta)$, so they collect more probability mass.

This gives a theoretical hint: noisy SGD has an implicit preference for flatter minima.

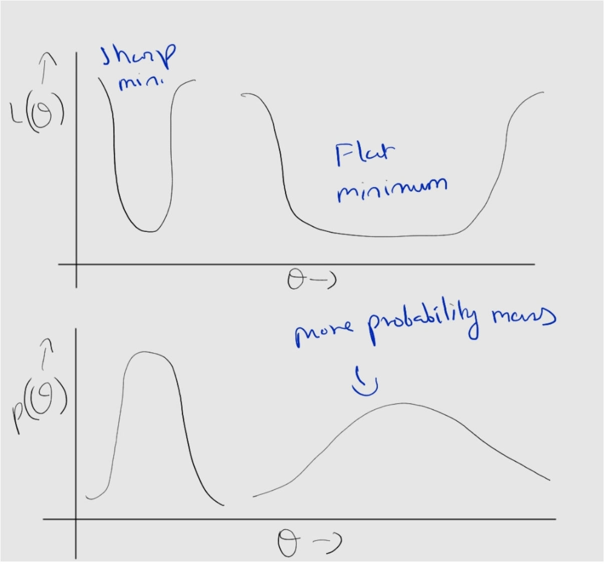

Figure 3: The stationary distribution of noisy SGD. Whether viewed through the energy-based form $p(\theta) \propto \exp(-L(\theta)/T)$, or the local quadratic approximation $\mathrm{Var}[\theta] \propto \frac{\sigma^2}{2\eta h}$, a flat minimum in $L(\theta)$ translates to a broader peak collecting more total probability mass. This explains why noisy SGD implicitly prefers more robust, flatter solutions that often generalize better.

5 Probing techniques Implementation and Empirical Findings

To keep computation lightweight, I tested the ideas on:

- MNIST classification,
- a 2-layer MLP (ReLU, 256 hidden units),
- SGD with momentum (lr = 0.05, 5 epochs).

I trained two models with different random seeds and kept the final weights for analysis.

a. Sharpness via Top Hessian Eigenvalue

Using Hessian-vector products and power iteration on a validation batch, I estimated the largest eigenvalue of the Hessian. This provides a scalar proxy for sharpness.

b. Flatness via Parameter Perturbations

I added small random perturbations to the trained weights ($\epsilon = 10^{-3}$, norm-controlled) and measured the resulting change in loss on a held-out batch.

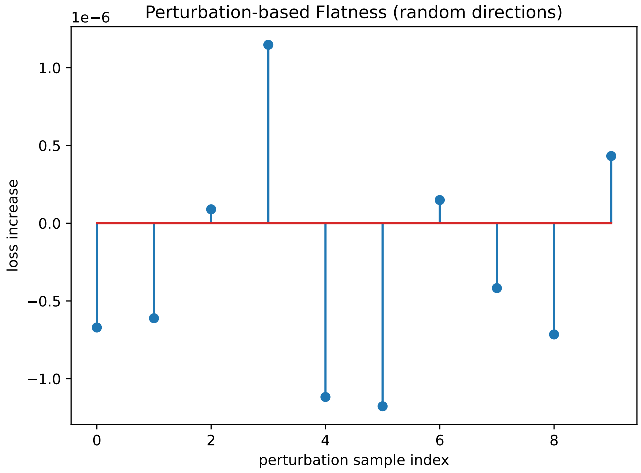

Figure 4: Perturbation-based flatness: loss increase for several random weight perturbations around a trained MLP. Values are numerically close to zero at this scale, indicating a relatively flat minimum.

c. Gradient Noise Scale (Efficient Proxy)

In addition to curvature-based measures, I estimated the *gradient noise scale* as a more efficient, first-order proxy for local geometry. For several mini-batches, I computed the per-batch gradients, estimated the mean gradient $\mathbb{E}[g]$ and the mean squared gradient norm $\mathbb{E}[||g||^2]$, and formed a scalar ratio:

$$\mathcal{S} \approx \frac{||\mathbb{E}[g]||^2}{\mathbb{E}[||g||^2] - ||\mathbb{E}[g]||^2}.$$

Intuitively, this measures how large the "signal" (mean gradient) is compared to the stochastic variation across batches. In flatter regions, gradients tend to be smaller and noisier; in sharper regions, they are often larger and more aligned.

d. Mode Connectivity

For two independently trained MLPs, I linearly interpolated between their weights and evaluated validation loss along the path to check whether the minima are isolated or connected through low-loss regions.

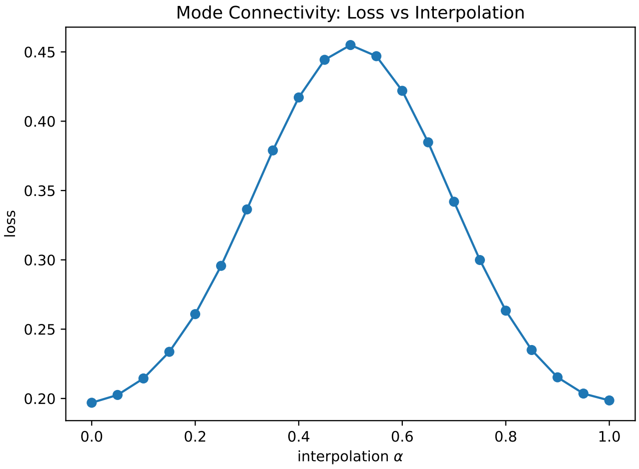

Figure 5: Mode connectivity: validation loss along the linear interpolation between two independently trained MLPs. A mild bump appears in the middle, suggesting the minima are distinct but lie in a broader low-loss region.

e. 2D Slice of the Loss Landscape

To visualize the geometry more directly, I constructed a 2D slice in parameter space:

- the α direction follows the line from MLP (seed 0) to MLP (seed 42),
- the β direction is a random vector orthogonal to this line,
- the plane is centred at the midpoint between the two models.

On a grid of (α, β) values, I evaluated the loss using a validation batch and plotted the resulting surface.

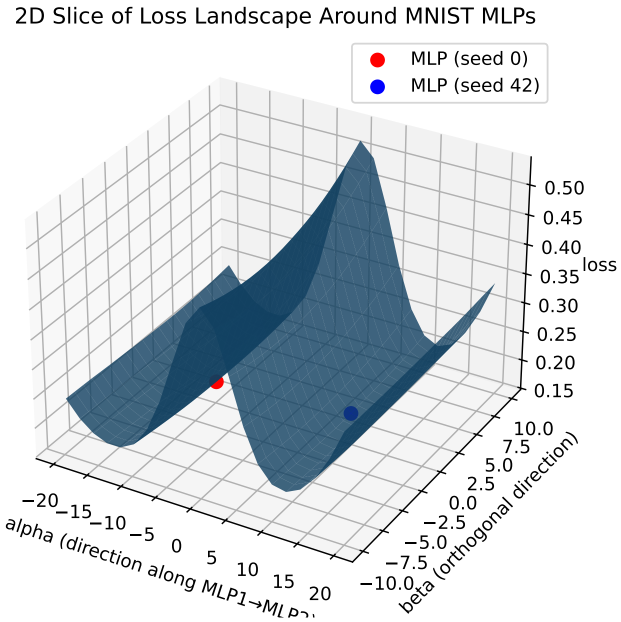

Figure 6: 2D slice of the loss landscape around the two trained MLPs. The red and blue points mark the locations of MLP (seed 0) and MLP (seed 42) respectively. Both lie in wide low-loss valleys, with a mild ridge between them along the α direction.

Results

Both models reached similar accuracy, but differed slightly in curvature and noise scale:

Table 1: Geometry and performance metrics on MNIST MLPs.

Model	$\lambda_{\max}$	Flatness	Grad noise scale	Train acc	Test acc
MLP (seed 0)	5.2734	0.0000	0.1240	0.964	0.936
MLP (seed 42)	4.2910	0.0000	0.1279	0.964	0.935

Key observations:

- Both minima are relatively flat at the tested perturbation scale (near-zero loss increase).
- The slightly larger $\lambda_{\max}$ for seed 0 suggests a somewhat sharper region, but with no major impact on generalization.
- The gradient noise scales are similar (≈ 0.12), indicating that SGD sees comparable levels of stochasticity around both minima, consistent with their similar test accuracy.
- The mode connectivity curve and 3D slice both show a mild ridge between solutions, indicating that the two minima are distinct but lie in a broader low-loss landscape.

These small experiments demonstrate that simple geometric probes can be computed efficiently and already reveal interpretable trends in optimization behaviour.

6 Answers to the Questions

i. Why does SGD find generalizable minima?

- In the quadratic approximation, SGD with noise behaves like a stochastic process with a stationary distribution.
- When several minima have similar loss, wider minima occupy more volume in parameter space.
- Because of this, noisy SGD tends to spend more time in flatter regions, which are often more robust and generalize better.

ii. How does architecture affect the landscape?

- Wider or residual architectures are expected to introduce more flat directions (more small eigenvalues).
- Normalization and skip connections can improve conditioning and smooth the landscape.
- In my implementation I started with an MLP on MNIST; extending the same probes to CNNs/ResNets would allow a direct comparison via $\lambda_{\max}$, flatness, and gradient noise scale.

iii. What geometric properties correlate with trainability and generalization?

- Trainability: moderate $\lambda_{\max}$ and well-conditioned curvature make larger learning rates feasible.
- Generalization: lower sharpness (smaller loss increase under perturbations) and larger local “volume” of low loss.

- Gradient noise scale: regions where the gradient signal is not dominated by noise tend to be easier to optimize; very low noise scale can also align with sharper minima.
- Mode connectivity: if solutions are connected by low-loss paths, they often share similar good generalization.

iv. Can we predict optimization difficulty from landscape analysis?

- Early in training, very large $\lambda_{\max}$ may signal:
    - need for a smaller learning rate,
    - or risk of unstable updates.
- By measuring curvature, flatness, and gradient noise scale at early checkpoints, we can get a rough idea of how hard the optimization will be.

Overall, the theoretical, algorithmic, and empirical components align to show that geometry, especially curvature, flatness, and gradient noise scale, predicts both optimisation dynamics and generalisation behaviour.