Neural Networks Loss Landscape Convergence in Hessian Low-Dimensional Space

Tem Nikitin

Moscow Institute of Physics and Technology

2025

Persona: Researcher (Data Scientist)

- **Name:** Alex Researcher
- **Role:** Data Scientist / ML Researcher
- **Goals:** Understand when adding data stops improving training dynamics
- **Frustrations:** High computational cost of large datasets; unclear stopping criterion
- **Technologies:** Python, PyTorch, Hessian eigen-decompositions

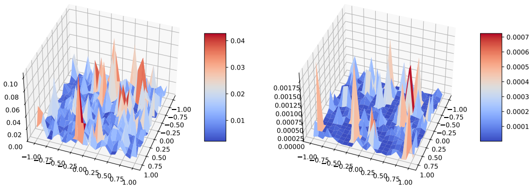
User Scenario: Background & Rising Action

Background:

- Modern neural nets achieve better accuracy with more data, but at high computational cost.
- Key question: *when* does adding more samples give negligible improvement?

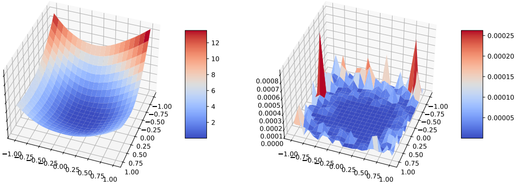
Rising Action:

- Project the high-dimensional parameter space onto top Hessian eigenvectors to capture principal curvature directions.



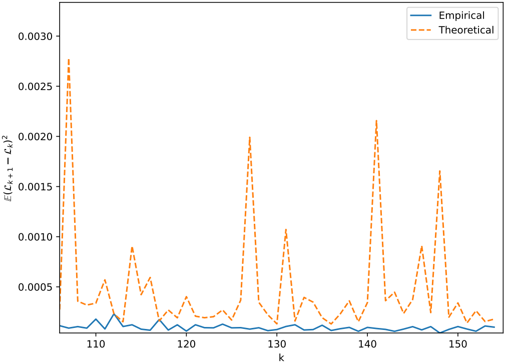
3D loss surfaces L_k and $(L_{k+1} - L_k)^2$ for small k.

User Scenario: Falling Action



3D loss surfaces L_k and $(L_{k+1} - L_k)^2$ for large k.

User Scenario: Resolution



Decay of Δ_k and $\Delta_k \cdot k^2$.

Prototype: Methodology

- Compute top d eigenvectors of Hessian $H(w^*)$ via iterative power method.
- Form low-dimensional coordinate θ such that $w = w^* + P\theta$, $P = [e_1, \ldots, e_d]$.
- Estimate $\Delta_k = E[(L_{k+1} - L_k)^2]$ via Monte Carlo in this subspace.

Prototype: Theoretical Estimate

$$\Delta_k \approx \frac{\sigma^4}{4}\left(2\sum_{i=1}^{d}(\lambda_i^{(k+1)} - \lambda_i^{(k)})^2 + \big(\sum_{i=1}^{d}(\lambda_i^{(k+1)} - \lambda_i^{(k)})\big)^2\right)$$