Convergence of Loss Function Surface in Transformers

Egor Petrov

Moscow Institute of Physics and Technology
Course: My first scientific paper
Consultant: Nikita Kiselev, BSc
Expert: Andrey Grabovoy, PhD

2025

Loss Function Landscape Convergence for transformers

Training a neural network involves searching for the minimum point of the loss function, which defines the surface in the space of model parameters.

Goal

Investigation of the Loss Function's Landscape for Transformer's architecture to find the minimal dataset size

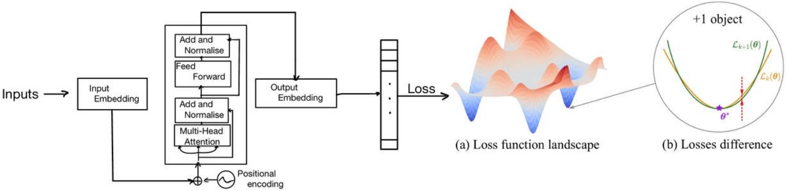
Problem

Determine the minimal dataset size k^* for loss function surface convergence in transformers within a predefined error threshold.

Solution

1) Hessian-based approach to find the critical sufficient dataset size for transformer architectures.
2) Empirical studies on the task of image classification using ViT's
3) Reduce of computational resources with the minimal data size

Loss function Landscape Convergence for a Transformer Block



(a) Loss function landscape (b) Losses difference

1. In the neighborhood of a local minimum, the loss function can be approximated by a quadratic form
2. When incrementally adding samples to the training set, we observe convergence in the optimization landscape

$$\mathcal{L}_k(\boldsymbol{w}) \approx \mathcal{L}_k(\boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}^*)^\top \mathbf{H}^{(k)}(\boldsymbol{w}^*)(\boldsymbol{w} - \boldsymbol{w}^*)$$

The described convergence will provide estimates on the minimum data size for efficient training.

Problem Statement

Objective

Determine the minimal dataset size k^* for loss function surface convergence in transformers within a predefined error threshold.

Challenges

- Analyze Hessian $\mathbf{H}_k(\mathbf{w})$ to quantify landscape evolution with k.
- Derive bounds for $\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})$.
- Validate empirically for transformers (e.g., ViTs).

Motivation

Efficient training in data-scarce domains (e.g., medical imaging) with limited resources

Solution

Approach

Decompose Hessian: $\mathbf{H}_k = \mathbf{H}_o + \mathbf{H}_f$.
Use Taylor approximation at $\mathbf{w}^*$:

$$\mathcal{L}_k(\mathbf{w}) \approx \mathcal{L}_k(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^\top \mathbf{H}_k(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*).$$

Bound

$$|\mathcal{L}_{k+1} - \mathcal{L}_k| \leq \frac{1}{k+1}\left|l_{k+1} - \mathcal{L}_k\right| + \frac{\|\mathbf{w} - \mathbf{w}^*\|_2^2}{2(k+1)}\left\|\mathbf{H}_{k+1} - \frac{1}{k}\sum \mathbf{H}_i\right\|_2.$$

Outcome

Estimate k^* for minimal dataset size, reducing computational costs.

Vectorization and Norms in Matrix Calculus

Vectorization: For a matrix $\mathbf{F} \in \mathbb{R}^{m \times n}$, $\text{vec}_r(\mathbf{F})$ transforms $\mathbf{F}$ into a column vector by stacking its rows.

Example: If $\mathbf{F} = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix}$, then $\text{vec}_r(\mathbf{F}) = \begin{pmatrix} f_{11} \\ f_{12} \\ f_{21} \\ f_{22} \end{pmatrix}$.

Derivatives with Vectorization:
First derivative: $\dfrac{\partial \mathbf{F}}{\partial \mathbf{W_i}} := \dfrac{\partial \text{vec}_r \mathbf{F}}{\partial \text{vec}_r \mathbf{W_i}}$, where $\text{vec}_r \left(\dfrac{\partial \mathbf{F}}{\partial \mathbf{W_i}} \right)$ is its vectorized form.

Second derivative: $\dfrac{\partial^2 \mathbf{F}}{\partial \mathbf{W_i} \partial \mathbf{W_j}} := \dfrac{\partial \text{vec}_r \dfrac{\partial \mathbf{F}}{\partial \mathbf{W_i}}}{\partial \text{vec}_r \mathbf{W_j}}$.

Norm Equivalence: The second norm of the vectorized matrix equals the Frobenius norm of the original matrix:

$$\|\text{vec}_r(\mathbf{A})\|_2 = \|\mathbf{A}\|_F$$

where $\|\cdot\|_2$ is the Euclidean norm and $\|\cdot\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$.

Hessian Norm Bound for Self-Attention

Theorem 1

For the Self-Attention layer, the Hessian norm is bounded by

$$\|\mathbf{H}_i(\mathbf{w}^*)\|_2 \leq M$$

where M is a constant depending on the model and data:

$$M = \max\Big(\frac{2X_{\max}^2}{Ld_V}, \frac{2W_{\max}^4 X_{\max}^6}{Ld_V d_K} + \frac{2R_{\max} W_{\max}^3 X_{\max}^5}{Ld_V d_K}, \frac{2X_{\max}^4 W_{\max}^2}{Ld_V\sqrt{d_K}} + \frac{2R_{\max} S_{\max} X_{\max}^3 W_{\max}}{Ld_V\sqrt{d_K}}, \frac{2W_{\max}^4 X_{\max}^6}{Ld_V d_K} + \frac{2R_{\max} W_{\max}^3 X_{\max}^5}{Ld_V d_K} + \frac{2R_{\max} W_{\max} X_{\max}^3 S_{\max}}{Ld_V\sqrt{d_K}}\Big)$$

$X_{\max}$, $W_{\max}$: maximum singular values of input data and weight matrices, L, d_V, d_K: sequence length, value, key dimensions

Convergence of Loss Function Difference

Theorem 2

For a single self-attention block, the difference in loss functions satisfies

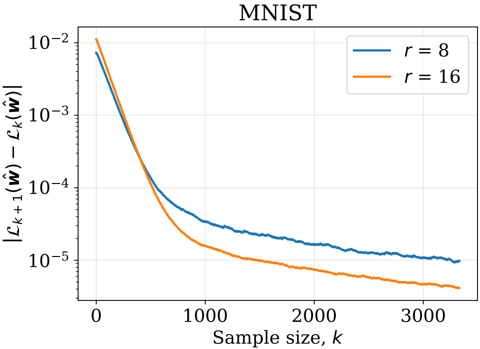
$$|\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})| \leq \frac{2L}{k+1} + \frac{M\|\mathbf{w} - \mathbf{w}^*\|_2^2}{k+1}$$

where L is the bound on the loss function, and M is the bound on the Hessian norm from Theorem 1.

This shows that as the dataset size k increases, the loss function surface stabilizes.
Practically, this helps estimate the minimal dataset size k^* for efficient training.

Transformer Experiment

Fine-tuned ViT on small image datasets (LoRA, unfreezing layers). Monitored accuracy and $|\mathcal{L}_{k+1} - \mathcal{L}_k|$.



Conclusion

Summary

- Hessian-based Loss function differennce convergence analysis.
- Theoretical convergence validated via ViT experiments.
- Practical for resource-efficient training.

Future Work

- Extend to multi-layer transformers and new structures, e.g. LayerNorm.
- Apply to specific tasks (e.g., medical imaging).