# Candy Clouds: Sharp Minima Can Generalize for Deep Nets

Candice Yao [1] Carina Sun [1] Dipesh Tharu Mahato [1] Tanishq Sardana [1]

[1]New York University, Center for Data Science

## Introduction

Deep neural networks often generalize well, and one explanation attributes this to the flatness of minima found during training. This paper examines whether commonly used definitions of flatness—based on curvature or local loss variation—are valid in the context of deep, overparameterized networks, and investigates how architectural properties like reparametrization and symmetry affect these definitions.

## Definition of Flatness/Sharpness

These definitions attempt to quantify how "flat" or "sharp" a minimum is in the loss landscape.

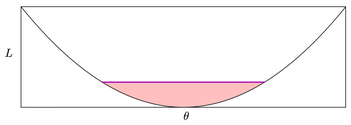



Figure 1. An illustration of $\varepsilon$-flatness and $\varepsilon$-sharpness.

- **Definition 1:** A minimum is $\varepsilon$-flat if there is a large connected region around it where the loss remains within $\varepsilon$ of the minimum. The volume of this region, $\mathcal{C}(L, \theta, \varepsilon)$, defines how flat the minimum is—larger volume means flatter.
- **Definition 2:** $\varepsilon$-sharpness measures how much the loss can increase within a small Euclidean ball of radius $\varepsilon$ around a minimum $\theta$:

$$\varepsilon\text{-sharpness} = \frac{\max_{\theta' \in B_2(\varepsilon,\theta)} (L(\theta') - L(\theta))}{1 + L(\theta)}$$

Using a second-order Taylor approximation around the minimum, this can be estimated as:

$$\varepsilon\text{-sharpness} \approx \frac{\|\nabla^2 L(\theta)\|_2 \cdot \varepsilon^2}{2(1 + L(\theta))}$$

## Properties of Deep Rectified Networks

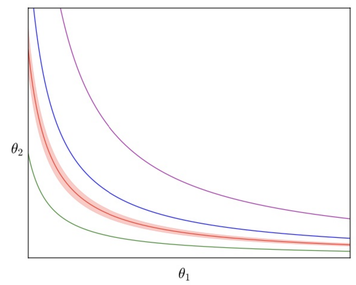



Figure 2. An illustration of the effects of non-negative homogeneity. Level curves of the loss function $L$ in 2-d parameter space. Each curve represents parameters $(\theta_1, \theta_2)$ that yield the same prediction function $f_\theta$, illustrating observational equivalence due to non-negative homogeneity.

- **Definition 3: Deep Rectified Network Output**
  $y = \phi_{\text{rect}}\left(\phi_{\text{rect}}\left(\cdots \phi_{\text{rect}}(x \cdot \theta_1) \cdots\right) \cdot \theta_{K-1}\right) \cdot \theta_K$
  *Where:*
  - $x$: input vector
  - $\phi_{\text{rect}}$: elementwise ReLU, $\phi(z) = \max(z, 0)$
- **Definition 4: Non-negative Homogeneity**
  $\phi(\alpha z) = \alpha\phi(z), \quad \forall \alpha > 0$
- **Theorem 1:**

  ReLU is non-negative homogeneous.
- **Implication for Deep Networks:**
  $\phi_{\text{rect}}(x \cdot (\alpha\theta_1)) \cdot \theta_2 = \phi_{\text{rect}}(x \cdot \theta_1) \cdot (\alpha\theta_2)$
- **Definition 5: $\alpha$-scale Transformation**
  $T_\alpha(\theta_1, \theta_2) = (\alpha\theta_1, \alpha^{-1}\theta_2)$

  This keeps the output of the network unchanged while enabling manipulation of sharpness.

## Flatness ≠ Better Generalization

This paper challenges a prevailing assumption: that flatter minima found by stochastic gradient methods imply better generalization in deep networks. Instead, the authors show:

- **Volume-based flatness measures can be misleading.** Due to the non-identifiability and symmetries in ReLU networks, every minimum can be transformed into an observationally equivalent one with infinite $\varepsilon$-flatness.
- **Hessian-based sharpness is not invariant.** Through $\alpha$-scale transformations, the curvature (e.g., spectral norm of the Hessian) can be arbitrarily manipulated without affecting function behavior.
- **Reparametrizations change the geometry.** The perceived flatness or sharpness of a loss landscape depends heavily on the choice of parameter or input space.

*Conclusion:* Generalization cannot be inferred from flatness alone. Metrics that don't account for reparametrization and network symmetries fail to capture true model behavior.

## Deep Rectified Networks and flat minima

## Volume ε- flatness

For rectified neural networks, the flatness of a minimum can be manipulated without changing the function it represents.

**Theorem:** Let $\theta = (\theta_1, \theta_2)$ be a minimum of a one-hidden-layer rectified neural network, where $\theta_1 \neq 0$ and $\theta_2 \neq 0$. Then for any $\varepsilon > 0$, the $\varepsilon$-flatness volume $\mathcal{C}(L, \theta, \varepsilon)$ is infinite.

**Implication:** Due to the non-identifiability and $\alpha$-scale transformations $T_\alpha(\theta_1, \theta_2) = (\alpha\theta_1, \alpha^{-1}\theta_2)$, we can construct infinitely many distinct parameter configurations that represent the same function and remain within an $\varepsilon$-loss neighborhood. Thus, all minima are infinitely flat under this definition, making volume $\varepsilon$-flatness ineffective for measuring generalization.

## Hessian Flatness

Hessian matrix represents the second-order curvature of the loss landscape around a minimum. **Key Metrics**: Spectral Norm and Trace

**Key Challenges: Parameter Symmetries in ReLU Networks**

**Non-negative homogeneity property**: For ReLU networks, the transformation

$$\phi_{\text{ReLU}}(x \cdot \alpha\theta_1) \cdot \alpha^{-1}\theta_2 = \phi_{\text{ReLU}}(x \cdot \theta_1) \cdot \theta_2$$

preserves the network's output function, demonstrating *observational equivalence*.

**Observational Equivalence**: Infinitely many parameter sets yield the same function but different Hessians.

**Scaling the Hessian via $\alpha$-Transformation**

The Hessian under $\alpha$-scale transformation

$$T_\alpha(\theta_1, \theta_2) = (\alpha\theta_1, \alpha^{-1}\theta_2)$$

transforms as: $(\nabla^2 L)(\alpha\theta_1, \alpha^{-1}\theta_2) = \begin{bmatrix} \alpha^{-1}\mathbb{I}_{n_1} & 0 \\ 0 & \alpha\mathbb{I}_{n_2} \end{bmatrix} (\nabla^2 L)(\theta_1, \theta_2) \begin{bmatrix} \alpha^{-1}\mathbb{I}_{n_1} & 0 \\ 0 & \alpha\mathbb{I}_{n_2} \end{bmatrix}$.

For any critical point $\theta$, the spectral norm of the Hessian can be made **arbitrarily large** (sharp) or **small** (flat) by choosing $\alpha \to 0$ or $\alpha \to \infty$.

**Implications for Flatness Matrics**

**Spectral Norm**: $\lambda_{\max}(\nabla^2 L) \propto \alpha^{-2}$ or $\alpha^2$ (arbitrary scaling!).

**Trace**: $\text{Tr}(\nabla^2 L) \propto \alpha^{-2}\,\text{Tr}(\nabla^2_{\theta_1} L) + \alpha^2\,\text{Tr}(\nabla^2_{\theta_2} L)$.

## $\varepsilon$-Sharpness.

While the formal definition of $\varepsilon$-sharpness appears earlier, we now explore how it can be artificially increased without changing the network's predictions.

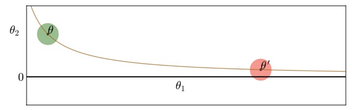


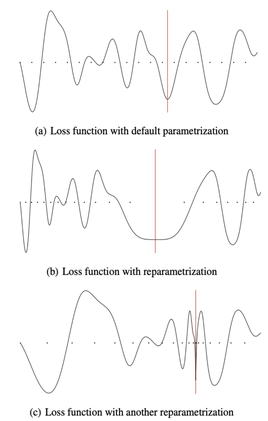

Figure 3. Observationally equivalent parameters with differing sharpness.[1]

Using the transformation:

$$T_\alpha(\theta_1, \theta_2) = \left(\frac{\varepsilon\theta_1}{\|\theta_1\|_2}, \alpha^{-1}\theta_2\right)$$

We can create a new parameter $\theta' \in B_2$ that is observationally equivalent to $\theta$ but has higher sharpness.

## Allowing Reparametrizations




**Model Reparametrization:**
Letting $\eta = g^{-1}(\theta)$ and deriving from our previous derivation on Hessians, then:

$$L_\eta(\eta) = L(g(\eta))$$

We derive the expression for $\nabla^2 L_\eta(\eta)$, noting that at critical points, $\nabla L(g(\eta)) = 0$:

$$\nabla^2 L_\eta(\eta) = \nabla g(\eta)^T \cdot \nabla^2 L(g(\eta)) \cdot \nabla g(\eta)$$

At critical points, this transformation reshapes the curvature arbitrarily.

**Example – Weight Normalization:**
While the output $w$ remains invariant, the flatness in $v$-space can be arbitrarily adjusted via simple rescaling.

$$\text{Let } w = s \cdot \frac{v}{\|v\|_2}.$$

While the output $w$ remains unchanged under rescaling of $v$, the curvature of the loss surface in $v$-space — and thus flatness — can be arbitrarily adjusted. This demonstrates that flatness is not intrinsic to the function but depends on the parameter representation.

**Input Representation:**

Even if the model is fixed, reparametrizing the input space (e.g., via whitening or feature standardization) alters the geometry of the function.

$$\frac{\partial f_\xi}{\partial u^\top}(\xi(u)) = \frac{\partial f}{\partial x^\top}(\xi(u)) \cdot \frac{\partial \xi}{\partial u^\top}(u)$$

This can distort gradients and affect measures of sensitivity or robustness — weakening any generalization claims based solely on gradient magnitude or Lipschitz continuity.

## References

[1] Vijay Badrinarayanan, Bamdev Mishra, and Roberto Cipolla. Understanding symmetries in deep networks. *arXiv preprint arXiv:1511.01029*, 2015.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[3] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[4] Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. pages 2422–2430, 2015.