Activation Plateaus in Residual Networks: Toward the Theoretical and Empirical Understanding of Plateau Geometry

Hiroshi Nonaka

January 29, 2026

1 Introduction

Some empirical findings in ViTs, CNN-based ResNets, and toy ResNets can be found here: https://docs.google.com/presentation/d/14Nd58PsfPUG3uejcRkKIzKoFiU2W2vFZJsxjD-E8DxY/edit?usp=sharing.

In my senior thesis project, I will focus on *activation plateaus*, stable regions in an *activation space* of an LLM, in which nudging a vector to a random direction does not make a big change in the final output of the LLM. The goals of my project are to:

- Replicate activation plateaus in residual networks (ResNets).
- Provide a mathematical framework of activation plateaus.

By approaching these aims, my project could contribute to:

- Increasing the efficiency and controllability of future activation plateau research with toy ResNets.
- Identifying kinds of tasks (e.g., image classification, binary gate approximation, continuous function approximation) and training settings (e.g., classification, regression, contrastive learning with SimCLR) causing activation plateaus.
- Enabling theoretical analyses of activation plateaus.

In Section 2, I summarize past findings about activation plateaus. Section 3 introduces my views of activation plateaus, namely the **mechanistic view** discussing my speculation that activation plateaus are reproducible in simple ResNets and **mathematical view** where I try to define theoretical frameworks of activation plateaus based on mathematical analysis and dynamical systems, providing an interesting foundation for future experiments.

2 Related Work

In this section, I provide a list of discoveries about activation plateaus from prior research. Heimersheim and Mendel [2024] defines an activation plateau as **a region containing a real activation of a token, where perturbing the real activation within the region does not affect the downstream logit value much**. They discovered that when they perturb a real activation in a random direction, the change in the output logit was smaller than when they perturb it into a semantically meaningful direction. They hypothesized that plateaus enhance LLMs' inferential robustness against random noise in activations or semantic interference caused by superposition. Janiak et al. [2024] conducted a more thorough analysis and discovered that the boundaries of plateaus become sharper as the model size and training tokens increase.

Giglemiani et al. [2024] trained a sparse autoencoder [Sharkey et al., 2022] to extract disentangled features from polysemantic neurons and create new *synthetic activations* by adding some of the features with balanced weights. Then they tested the directional sensitivity and plateaus of the synthetic activations compared to real and random activations. Regarding directional sensitivity, by perturbing

each real activation *toward* a synthetic activation vs another real activation, they found that perturbing the vectors toward synthetic activations creates similar changes in the final logit calculation. Testing activation plateaus *around* synthetic vs real activations revealed that real activations enjoy more pronounced plateaus, indicating their greater robustness against noise.

Shinkle and Heimersheim [2025] interpolated two real activations (A and B) in the activation space of a layer and recorded the relative distance of each activation induced by an interpolation point to A at each downstream layer. They discovered that **the more MLP and Attention layers the model has between the interpolation layer and the recording layer, the more pronounced the plateaus and boundaries become**. They also identified that **the MLP layers, not attention layers, primarily contribute to the emergence of boundaries** in GPT2.

3 Hypotheses

Based on these research findings, I propose some hypotheses about the functioning and mechanisms of activation plateaus from different angles.

3.1 Mechanistic view

We know that the number of MLP layers determines the sharpness of plateaus in GPT2. Therefore, we can speculate that **the repeated additions of the MLP layer activations $MLP_j(x_{j,MLP})$, which are non-linearly transformed residual streams at each layer, push the residual stream $x_{j,att}$ toward a nearby real activation**. This speculation naturally gives rise to a research question: **Can we create a naturalistic plateau with a ResNet?** If this is the case, we can conduct more controlled experiments in simpler settings than past research on activation plateaus, which focuses on LLMs.

3.2 Mathematical view

Previous works have not mathematically defined activation plateaus. In this section, I define activation plateaus analytically. First, a neural network is generally continuous since it is generally differentiable. For convenience, I assume a neural network f is continuous hereafter unless otherwise specified.

3.2.1 Defining plateaus analytically

We can define a plateau as a set of points in the domain of a neural network to remove the definitional ambiguity:

Definition 3.1 (**Activation plateaus**)**.** Let $(\mathbb{R}^d, d)$ and $(\mathbb{R}^c, d)$ be metric spaces. Given a neural network $f : \mathbb{R}^d \to \mathbb{R}^c$, an activation plateau of an activation $a \in \mathbb{R}^d$, denoted $P_r(a) \subseteq \mathbb{R}^d$, is the connected component of the inverse image of an open ball $B_r(f(a)) \subseteq \mathbb{R}^c$ of radius r centered at $f(a)$ such that $a \in P_r(a)$.

Figure 1 provides an intuitive sense of an activation plateau. The distance function d of the metric spaces can be the L2-norm (the Euclidean distance). Note that KL-Divergence cannot be used since it is not symmetric. Based on this definition, the size of an activation plateau is determined in terms of how big the radius r of an open ball we consider in the codomain of the network, $\mathbb{R}^c$.

Proposition 3.2 (**Disjoint plateaus**)**.** *If $B_{r_a}(f(a))$ and $B_{r_b}(f(b))$ are disjoint, then $P_{r_a}(a)$ and $P_{r_b}(b)$ are also disjoint.*

Proof. Proof by contradiction. Suppose $P_{r_a}(a) \cap P_{r_b}(b) \neq \varnothing$. Then $\exists x \in P_{r_a}(a) \cap P_{r_b}(b)$ such that $f(x) \in B_{r_a}(f(a)) \cap B_{r_b}(f(b))$. But we know $B_{r_a}(f(a)) \cap B_{r_b}(f(b)) = \varnothing$. $\Rightarrow\!\!\leftarrow$ □

This is a fairly direct fact in topology. This statement can be extended to infinitely many activation plateaus and their open balls that are pairwise mutually exclusive.

Additionally, the following is also true.

Proposition 3.3 (**Plateau nesting**)**.** *If $r_1 < r_2$, then $P_{r_1}(a) \subseteq P_{r_2}(a)$.*

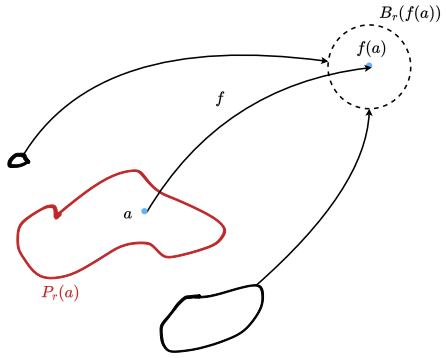

Figure 1: Diagram of an activation plateau (red). The inverse image of the open ball in the codomain $B_r(f(a))$ might be connected or disconnected. The connected component of the inverse image that includes a is the activation plateau of a, denoted $P_r(a)$.

Proof. Let $B_1 = B_{r_1}(f(a))$ and $B_2 = B_{r_2}(f(a))$. Since $B_1 \subset B_2$, $P_{r_1}(a) \subseteq f^{-1}[B_1] \subseteq f^{-1}[B_2]$. Suppose by contradiction that $P_{r_1}(a) \not\subseteq P_{r_2}(a)$. Then, $P_{r_1}(a) \cap \complement P_{r_2}(a) \neq \varnothing$. $a \in P_{r_1}(a)$ and $a \in P_{r_2}(a)$, so $P = P_{r_1}(a) \cup P_{r_2}(a)$ is also connected. Note that $\forall x \in P, f(x) \in B_2$. However, this contradicts the definition of a connected component that $P_{r_2}(a)$ is a maximal connected subset of $f^{-1}[B_2]$ since we showed that $P \supset P_{r_2}(a)$ is the maximal connected component of $f^{-1}[B_2]$ containing a. $\Rightarrow\!\!\!\!\!\!\not\Leftarrow$ □

The fact that plateaus are nested depending on the radii of open balls gives rise to an interesting visualization experiment of the **contour lines of an activation plateau**, which will be discussed in Section 4.

Based on the fact that a neural network f is continuous, **we can characterize how f "absorbs" and "pushes" activations** by decomposing the neural network into each block g_i.

Define g_i to be a continuous layer of the neural network f, such as $g_i(x) = \sigma(\ell_i))(x)$. Denote a part of the neural network from layer s to layer t as $f_{s\to t}(x) = (g_t \circ g_{t-1} \circ \cdots \circ g_s)(x)$. Note that $f = f_{0\to n-1}$. Then, we can consider the following setting:

Remark 3.4. *Since g_{n-1} is continuous, $\forall r > 0, \exists \delta_{n-1}$ such that $\forall x_{n-1} \in B_{\delta_{n-1}}(f_{0\to n-2}(a)), g(x_{n-1}) \in B_r(f(a))$. Repeat this process. Since g_i is continuous, $\forall \delta_{i+1} > 0, \exists \delta_i > 0$ such that $\forall x_i \in B_{\delta_i}(f_{0\to i-1}(a))$, $g_i(x_i) \in B_{\delta_{i+1}}(f_{0\to i}(a))$. Reaching layer 0, we eventually get such $\delta_0 > 0$ that $\forall x_0 \in B_{\delta_0}(a), g_0(x_0) \in B_{\delta_1}(f_{0\to 1}(a))$.*

Figure 2 visualizes this process.

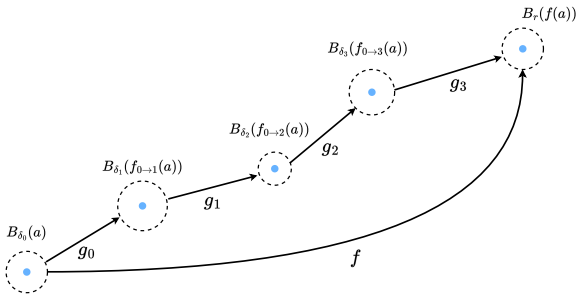

Figure 2: We consider a residual network f with $n = 4$ layers. The entire open ball maps to the open ball in the codomain by continuity. The size of the open ball is a *minimum requirement* and it is possible that elements outside the open ball map to within the open ball in the codomain.

By the continuity of each layer, we can prove the following statement:

Proposition 3.5. *Such $B_{\delta_0}(a)$ is contained in $P_r(a)$.*

Proof. Every point in $B_{\delta_0}(a)$ for sure map to an element in $B_r(f(a))$ by continuity, thus $B_{\delta_0}(a) \subseteq f^{-1}[B_r(f(a))]$. However, since $a \in B_{\delta_0}(a)$ and $B_{\delta_0}(a)$ is connected, $P_r(a)$ should be the largest connected subset containing a; $B_{\delta_0}(a) \subseteq P_r(a)$. □

From this fact, we can say the same thing for any other intermediate layers:

Corollary 3.6. *For every i for which $0 \leq i \leq n-1$, consider a partial neural network $f_{i \to n-1}$. Then, such $B_{\delta_i}(f_{0 \to i}(a)) \subseteq P_r(f_{0 \to i}(a))$.*

We note that the radius of the open ball in a domain δ_i is a minimal requirement, and it is possible that elements outside the open ball map to within the open ball in the codomain. On the other hand, **once activations in any domain map to a point in the open ball in its codomain, the activations cannot escape from this dynamics** in the sense that they eventually all map to points in $B_r(f(a))$. Therefore, it makes sense that **the more layers we have in between (e.g., n layers), the larger the plateau becomes** in the domain of the residual network $f_{0 \to n-1}$. This intuition might resonate with the findings of Shinkle and Heimersheim [2025] that the boundaries of plateaus become sharper as the number of layers in between increases. The whole argument above is purely based on the fact that f is continuous, and this even applies to a randomly initialized neural network. Therefore, **I will integrate an optimization aspect into this theory, empirically understanding how this set of trajectories evolves through training checkpoints** (to be discussed in Section 4).

3.2.2 Plateaus in ResNet as a Dynamical System

So far, we have only focused on properties of plateaus of a general neural network f. Now, we pay our attention to residual networks. Consider a self-mapping residual network $f : \mathbb{R}^D \to \mathbb{R}^D$, where interim residual blocks $F_i : \mathbb{R}^D \to \mathbb{R}^D$ are also self-maps. Let $g_i(x) = x + F_i(x)$ and $f_{s \to t}(x) = (g_t \circ \cdots \circ g_s)(x)$. **(Work in Progress)**

4 Experiments

4.1 Research Questions

To summarize, my research questions are:

- Can we create naturalistic plateaus with a toy ResNet?
- What are the datasets/tasks that create plateaus and do not create plateaus in deep toy ResNets?
- How does the geometry of activation plateaus evolve/change over time based on training epoch and tasks/datasets?
- Do plateaus emerge if you train a fully connected neural network on the same dataset?

4.2 Identifying models, tasks, and datasets that create activation plateaus

To address the questions above, I will follow the following steps to identify effective models that exhibit plateaus and perform a couple of experiments described in Sections 4.3 and 4.4.

- First, download pretrained ResNets and optionally ViTs from HuggingFace, such as microsoft/resnet-18, microsoft/resnet-50, microsoft/resnet-101, google/vit-base-patch16-224, and facebook/detr-resnet-50. Using their trained datasets, examine if each model has activation plateaus. Since the data passed from one layer to another is a feature map, not a vector, I have to come up with a clever perturbation method, such as applying local perturbations to some patches, adding values from random variables that are somewhat correlated. Flattening the feature map into a vector and discussing trajectories/contour lines might not be reasonable due to the feature dependency/locality in the feature map, but we can still measure the models' activation plateaus.
- Also, train ResNets of different sizes on other tasks, such as other datasets, symbolic regression, and function approximation. Similarly, examine if each trained model has activation plateaus. I can find such tasks from pmlb and srbench (shared by Dr. Shinkle).

- After identifying tasks/datasets that naturalistically create activation plateaus in the activation spaces, train a toy ResNet on the selected datasets. Make sure the majority of the interim layers are all $F_\ell : \mathbb{R}^D \to \mathbb{R}^D$. Then we can regard vector additions to the residual stream as actions of the model and can perform Experiment 4.4.
- Wang and Isola [2022] discovered that contrastive learning settings create representations high in alignment and uniformity. ResNets are typically trained in supervised settings, not in a self-supervised way like LLMs. Therefore, it is of interest to examine the representations of ResNets trained with SimCLR as well (weights can be found here).
- If I could not find any non-CNN-based ResNets that exhibit activation plateaus, I can still conduct the following experiments to pretrained LLM such as GPT2.

After preparing models that exhibit plateaus, we can perform the following experiments to investigate how the properties of activation plateaus vary in models of different sizes, tasks/datasets, and training epochs. To prevent a combinatorial explosion of such parameters, first identify a representative combination of the model size, training epoch size, and dataset that exhibits the "best" activation plateau, with which I compare the models of different values of a specific parameter.

4.3 Experiment 1: Contour Line of a Plateau

The idea of this experiment is inspired by Proposition 3.3. An activation plateau contains another plateau of the same point if open balls are nested in the codomain. Given a residual network, map a real activation and draw a small open ball around it. We can find a subset of the activation plateau whose points map to within the ball. If you draw a bigger open ball, the corresponding activation plateau contains the original one. Therefore, we can plot points in the domain that map to the bigger open ball but not to within the smaller one. By repeating this process, we can visualize the terrain of the activation plateau like contour lines.

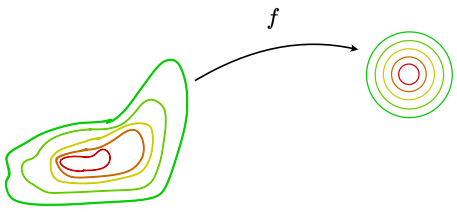

Figure 3: Contour lines of an activation plateau. The nesting property allows for this visualization of the plateau terrain.

4.4 Experiment 2: Activation Trajectories

Based on Remark 3.4 and Section 3.2.2, we can also focus on the trajectory of activations in a ResNet consisting of self-map residual blocks $F_i : \mathbb{R}^D \to \mathbb{R}^D$. As a process, simply identify three points in the domain, two of which are in the same activation plateau and one is outside the plateau. Plot a set of equidistant points so that those three points are included in the set. At each layer, record the trajectory of each point and examine if there is something that attracts nearby activations like an attractor (*semantic attractor*?). I am also interested in what kind of trajectories points inside and outside a plateau trace through.

4.5 Experiment 3: Activation Plateaus in Fully-connected Neural Networks

Another research question is, can we create plateaus in fully-connected neural networks? Again, previous research has focused on activation plateaus in LLMs. If we observe that plateaus emerge in fully-connected networks, these regions turn out to be something more universal than just error correction through residual block outputs. My speculation is yes, there should be plateaus in their activation spaces.

5 Conclusion

In this proposal, I summarized my current thoughts that activation plateaus can be reproduced in residual networks and that plateaus can be defined analytically. I proved some propositions based on the definition and designed experiments to visualize the structure of activation plateaus and the trajectory of activations. **I would appreciate any feedback, critiques, and improvements on any part of this proposal.**

References

Giorgi Giglemiani, Nora Petrova, Chatrik Singh Mangat, Jett Janiak, and Stefan Heimersheim. Evaluating synthetic activations composed of sae latents in gpt-2, 2024. URL https://arxiv.org/abs/2409.15019.

Stefan Heimersheim and Jake Mendel. Activation plateaus & sensitive directions in gpt2, 2024. URL https://www.lesswrong.com/posts/LajDyGyiyX8DNNsuF/interim-research-report-activation-plateaus-and-sensitive-1.

Jett Janiak, Jacek Karwowski, Chatrik Singh Mangat, Giorgi Giglemiani, Nora Petrova, and Stefan Heimersheim. Characterizing stable regions in the residual stream of llms, 2024. URL https://arxiv.org/abs/2409.17113.

Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders, 2022. URL https://www.lesswrong.com/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition.

Matthew Shinkle and Stefan Heimersheim. Activation plateaus: Where and how they emerge, 2025. URL https://www.lesswrong.com/posts/WMfSbt7AAcJdHzysB/activation-plateaus-where-and-how-they-emerge.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2022. URL https://arxiv.org/abs/2005.10242.