Mechanistic Interpretability

On the Internals of LLMs

Why Analyze LLMs?

- They are ubiquitous
  - We live in a post-GPT4 world
- Their training doesn’t lend itself towards trust:
  - Unsupervised pretraining
  - Supervised finetuning
  - RLHF
- An understandable fear of hallucinations and malicious outputs.

Existing methods of Analysis

- Mostly input-output maps
- CAMs and their variants, attention visualizations, etc.
- Some internals-based analysis

“Mechanistic” Interpretability?

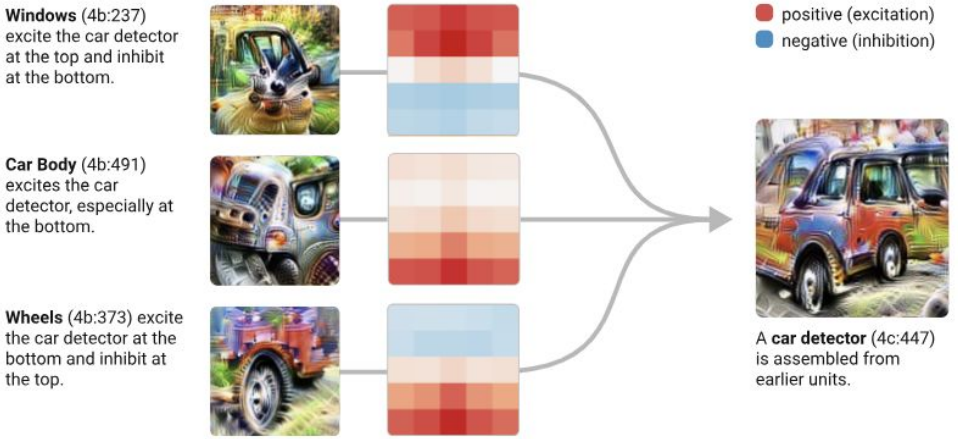
- “Mechanistic interpretability seeks to reverse-engineer neural networks”
- What does this mean?

Analysis of Computation

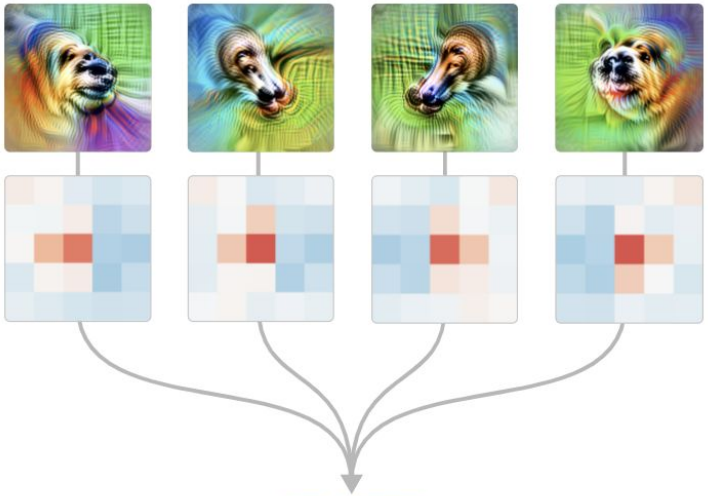
MI focuses on what computations in the model *mean*.

Hence the term ‘mechanistic’.

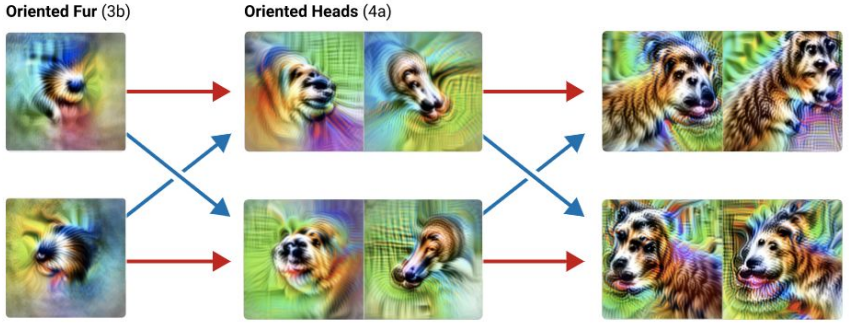
Lots of initial focus on CNNs



Olah, et al., "Zoom In: An Introduction to Circuits", Distill, 2020.

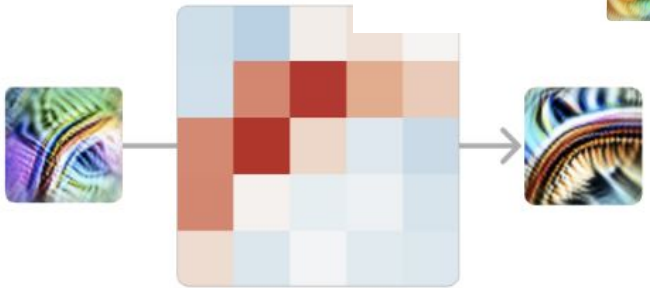


This pose invariant dog detector (4b:418) is **excited** by dogs oriented either way.



Union over left and right cases.

Union over left and right cases.



The raw weights between the early curve detector and late curve detector in the same orientation are a curve of **positive weights** surrounded by small **negative** or zero weights.

Olah, et al., "Zoom In: An Introduction to Circuits", Distill, 2020.

The Frontier Model Forum

ANTHROP\C Google Microsoft OpenAI

Frontier Model Forum: Advancing Safe AI Development

The Frontier Model Forum will draw on the technical and operational expertise of its member companies to benefit the entire AI ecosystem, advancing AI safety research and supporting efforts to develop AI applications to meet society's most-pressing needs.

Anthropic

A Mathematical Framework for Transformer Circuits

AUTHORS

Nelson Elhage*† Neel Nanda*, Catherine Olsson*, Tom Henighan† Nicholas Joseph†
Ben Mann† Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma,
Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones,
Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark,
Jared Kaplan, Sam McCandlish, Chris Olah‡

AFFILIATION

Anthropic

PUBLISHED

Dec 22, 2021

* Core Research Contributor; † Core Infrastructure Contributor; ‡ Correspondence to colah@anthropic.com;
Author contributions statement below.

Introduced a formalism to the analysis of Transformers

Features

Defining a "feature" in a satisfying way is surprisingly hard - Chris Olah

Very generally, features are tensors that occur during the computations performed by a model.

These could be individual neuron activations, or some function of a bunch of them.

Features are interesting because there tends to be human-understandable meaning associated with them surprisingly often.

Concepts to Cover

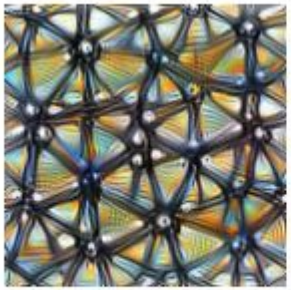
- Features
- Circuits
- The Activation Space
- Privileged Bases
- A reframing of the Transformer

Features

Olah, et al., "Feature Visualization", Distill, 2017.

Neuron

$layer_n[x,y,z]$



Channel

$layer_n[:,:,z]$

Layer/DeepDream

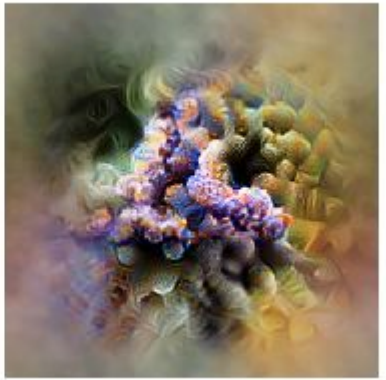
$layer_n[:,:,:]^2$

Class Logits

`pre_softmax[k]`

Class Probability

`softmax[k]`

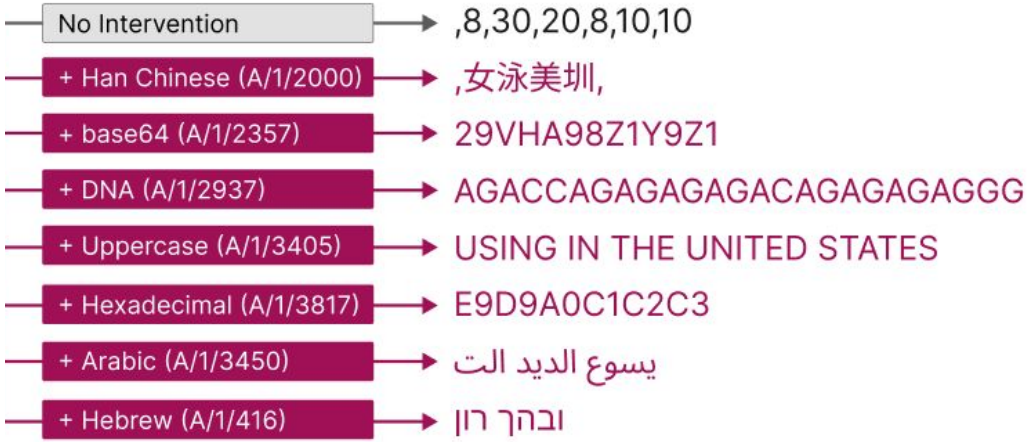


Features

What is the “meaning” associated with a feature?

Generally, it’s the concept in the input that causes the greatest activation of the feature.

Language Features?



Greater variety.

Broader range of concepts in language modeling.

Low level: the script of the input.

High level: Semantics contained in text.

Bricken, et al., *"Towards Monosemanticity: Decomposing Language Models With Dictionary Learning", Transformer Circuits Thread, 2023.*

Circuits

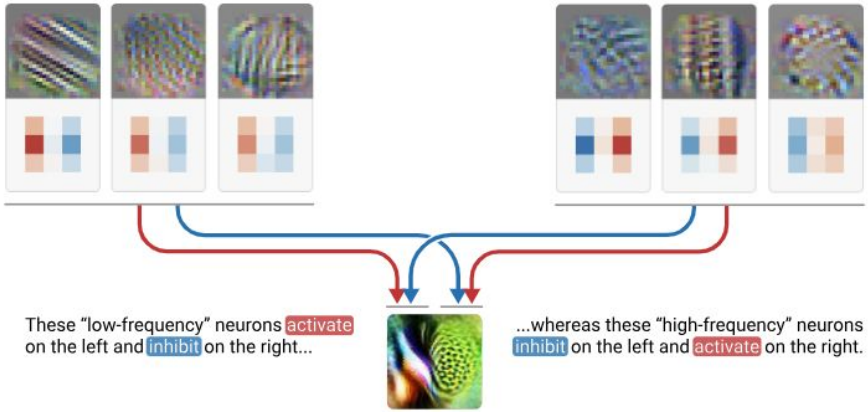
How are these features connected?

Is there meaning associated with these connections? Are higher-level features intuitively created from lower-level ones?

Circuits chart out these connections.

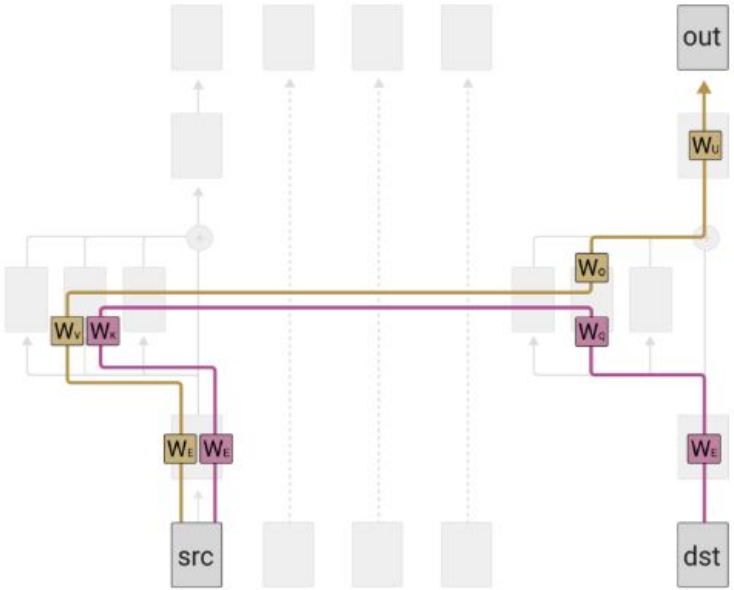
Each circuit is a smaller network extracted from the larger architecture.

Does not have to correspond to architectural computations.



Schubert, et al., "High-Low Frequency Detectors", Distill, 2021.

GPT Circuits



The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

GPT circuits are understood as compositions of two "atomic" circuits, the "query-key" and "output-value" circuits.

Elhage, et al., "A Mathematical Framework for Transformer Circuits", Transformer Circuits Thread, 2021.

Activation Space

The vector space in which all activations exist.

If the dimensionality of a layer's output is n, then the activation space is a subspace of R^n (most analysis assumes it to just be R^n).

Features, when thought of as mathematical constructs, exist in activation space.

Privileged Basis

Neurons can be thought of as bases for the activation space.

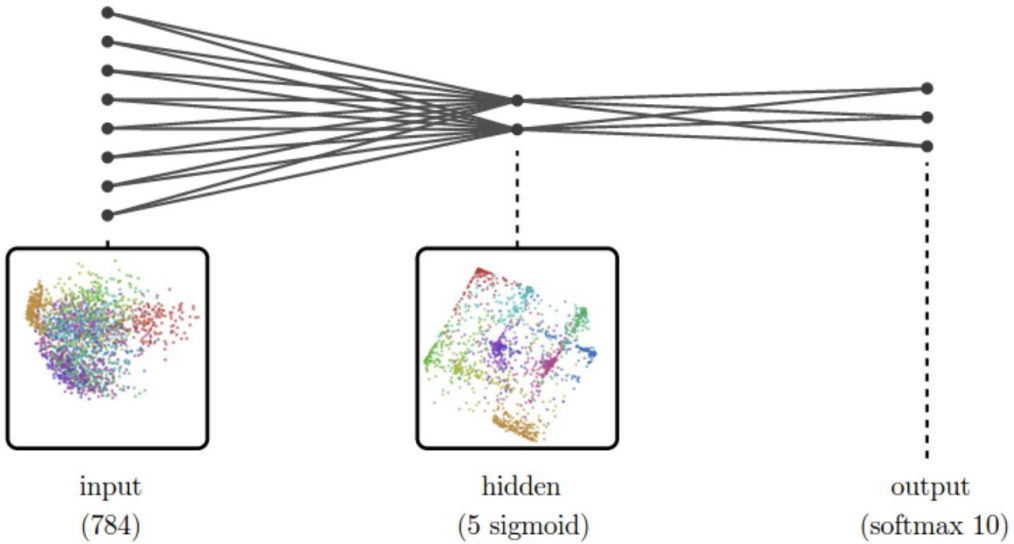
When meaningful, interesting features align with neurons, we call them 'privileged' bases.

Peculiarities within the architecture, dataset, and training process are what cause neurons to become privileged bases.

Privileged Bases

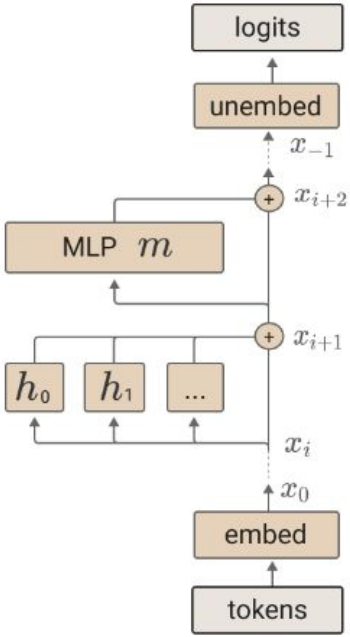
Features appear to align along a 5-dimensional hypercube.

This is because sigmoids tend to cause activations to be near 0 or 1, which in higher dimensions are the vertices of a hypercube.



https://colah.github.io/posts/2015-01-Visualizing-Representations/

The Transformer Reframed



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

An MLP layer, m, is run and added to the residual stream.

$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

Each attention head, h, is run and added to the residual stream.

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

One residual block

Token embedding.

$$x_0 = W_E t$$

Elhage, et al., "A Mathematical Framework for Transformer Circuits", Transformer Circuits Thread, 2021.

The Residual Stream

Similar to Highway Networks, this framing centres the residual connections in the model's visualization.

As wide as the model itself.

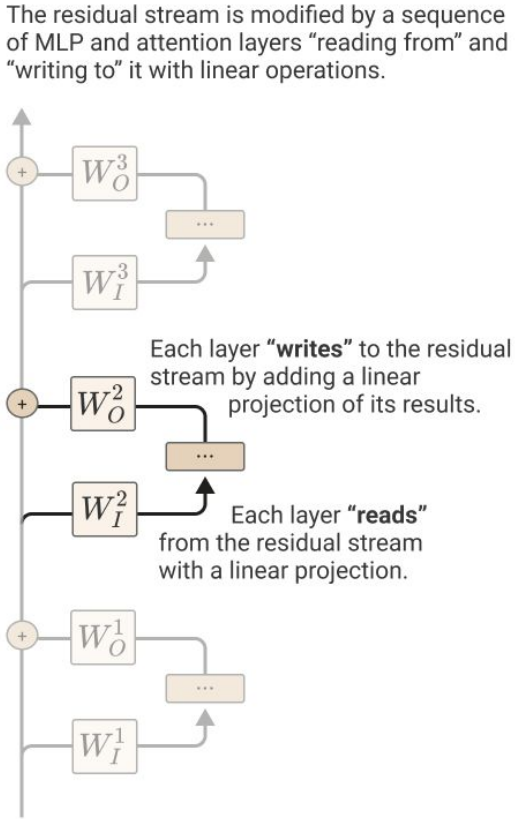
At any point during the forward pass, the residual stream is simply the sum of the activations of all prior (Attention+MLP) layers along with the initial embedding.

Elhage, et al., "A Mathematical Framework for Transformer Circuits", Transformer Circuits Thread, 2021.

The Residual Stream

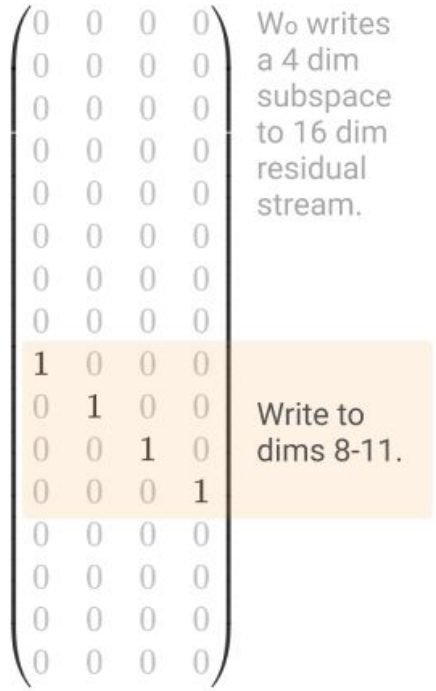
Attention heads use their W_v and W_o matrices to read and write from the residual stream.

These matrices help understand which portions of the residual stream individual attention heads modify, as well as which portions they use to perform this modification

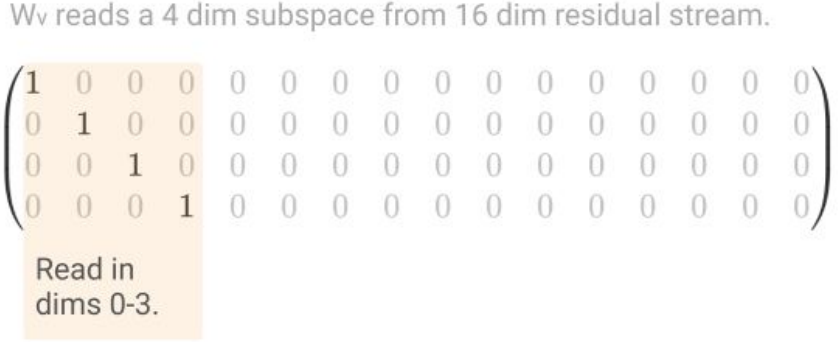


Elhage, et al., "A Mathematical Framework for Transformer Circuits", Transformer Circuits Thread, 2021.

The Residual Stream

$W_O^1 =$ 

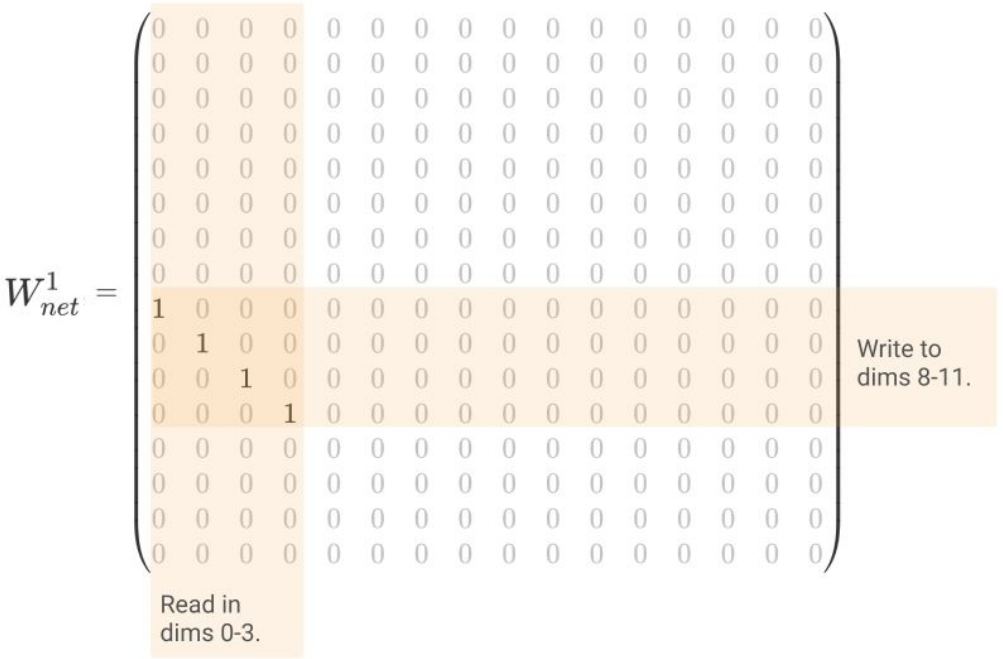
This framing allows us to understand how attention heads in different layers interact.

$W_V^1 =$ 

https://transformer-circuits.pub/2021/exercises/index.html

The Residual Stream

The net effect is what is termed the “output-value” circuit.

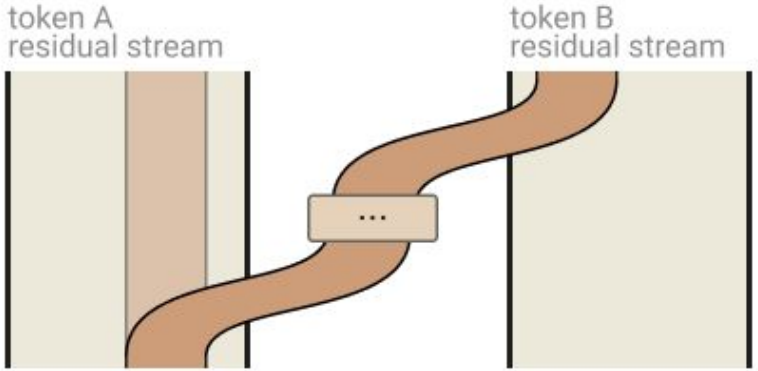


https://transformer-circuits.pub/2021/exercises/index.html

The Residual Stream

Attention heads move information across residual streams.

MLPs move information across dimensions within a residual stream.



Attention heads copy information from the residual stream of one token to the residual stream of another. They typically write to a different subspace than they read from.

The Residual Stream

The “query-key” circuit’s job is to determine which residual stream is read from and which is written to.

$$A = \text{softmax}(x^T W_Q^T W_K x).$$

Together, the circuits work, for a given token, to select which dimensions within its representation get updated, and which other tokens’ representations they use for this.

This sort of analysis allows us to see if different substreams within the residual stream have clear roles in the computation.

Pseudo-linearized Attention

A few simplifying assumptions are made to understand the attention mechanism in isolation:

- The model has no MLP layers
- It has no LayerNorm

MLP Layers tend to elude interpretation because of *superposition*

$$h(x) = \underbrace{(\mathrm{Id} \otimes W_O)}_{\text{Project result vectors out for each token } (h(x)_i = W_O r_i)} \cdot \underbrace{(A \otimes \mathrm{Id})}_{\text{Mix value vectors } \textit{across} \text{ tokens to compute result vectors } (r_i = \sum_j A_{i,j} v_j)} \cdot \underbrace{(\mathrm{Id} \otimes W_V)}_{\text{Compute value vector for each token } (v_i = W_V x_i)} \cdot x$$

$$h(x) = \underbrace{(A \otimes W_O W_V)}_{A \text{ mixes across tokens while } W_O W_V \text{ acts on each vector independently.}} \cdot x$$

As a result of this simplification, we can mathematically quantify the interaction between two arbitrary heads

$$(A^{h_2} \otimes W_{OV}^{h_2}) \cdot (A^{h_1} \otimes W_{OV}^{h_1}) = (A^{h_2} A^{h_1}) \otimes (W_{OV}^{h_2} W_{OV}^{h_1})$$

The Toy Model Approach

When trying to understand individual computations within *extremely* large models, the architecture only adds to the complexity.

The Toy Model Approach addresses this:

- Probe the larger model for interesting behaviour
- Implement the simplest possible architecture that would be required to capture this behaviour while using the fundamental mechanisms of the larger architecture
- Analyze the behaviour within this “toy” model

The toy model abstracts away all architectural complexities, leaving us with only those pieces needed to study the behaviour of interest.

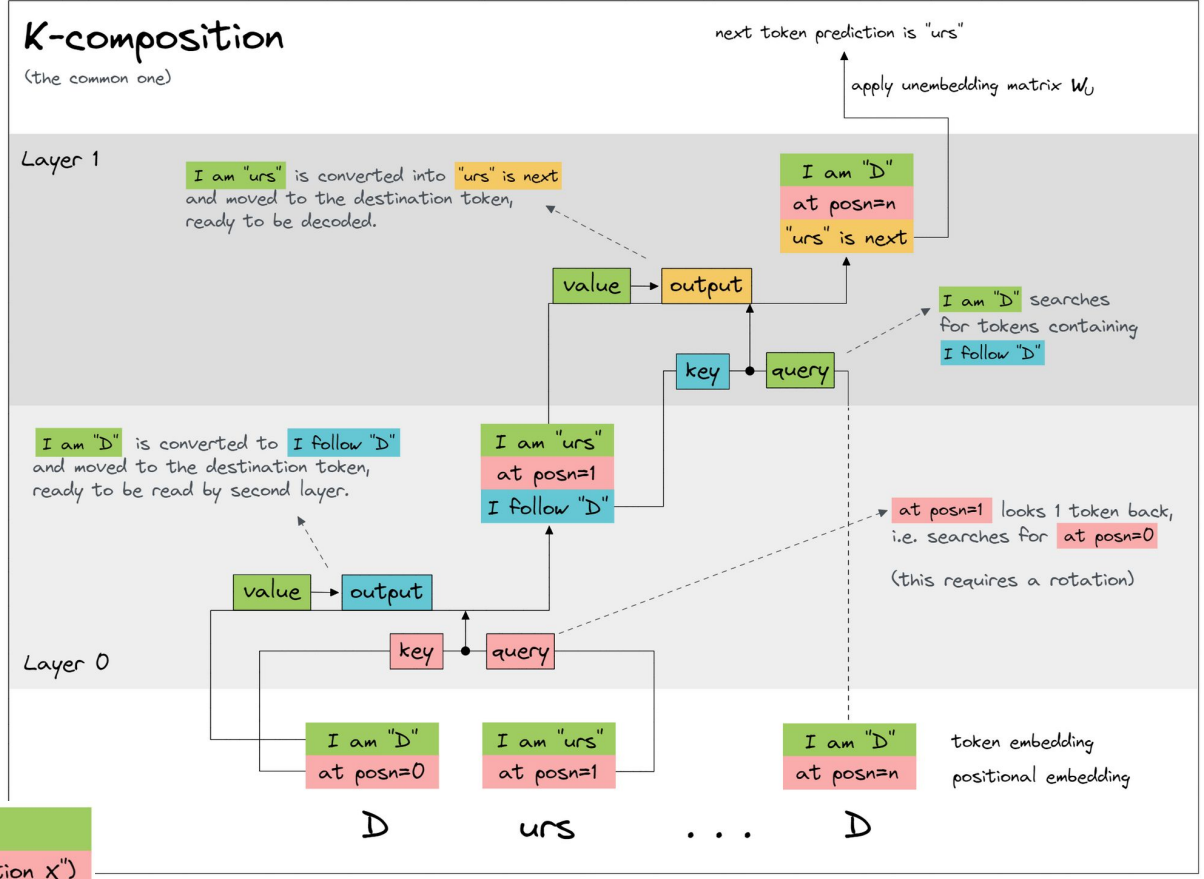
Induction Heads

Pairs of attention heads that have the role of recreating patterns observed in prior input, i.e. `[a][b] … [a] → [b]` .

These heads arise in models with more than 2 attention layers. Their analysis is motivated on the following two ideas:

- The method by which this pattern-recreation is performed validates the notion of substreams within the residual stream; it is also helpful in understanding the information-movement purpose behind attention heads.
- There is evidence that induction heads might play an important role in the in-context learning ability of transformers of any size.

An example of how the 2-layer toy transformer learns the word “Dursley” in context.



token encoding subspace (i.e. "this token is X")

positional encoding subspace (i.e. "this token is at position X")

decoding subspace (i.e. "the next token will be X")

prev token subspace (i.e. "the previous token was X")

https://www.lesswrong.com/posts/TvrfY4c9eaGLeyDkE/induction-heads-illustrated

Superposition

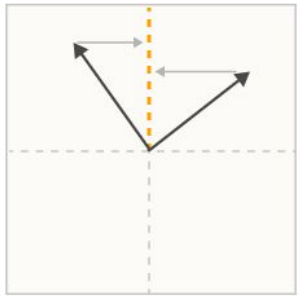
The phenomenon observed when neurons try to embed more features than they have dimensions

Olah, et al., "Feature Visualization", Distill, 2017.

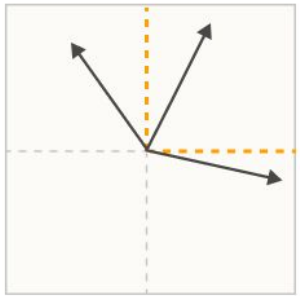
Superposition

Superposition seems to arise as a result of two factors:

- For an n-dimensional vector space, you can find exp(n) “almost orthogonal” vectors below some threshold cosine similarity. (cf. Johnson-Lindenstrauss)
- When projecting from high-dimensional to low-dimensional space, if the projection is almost orthonormal, then the initial vector can be reconstructed through L1 Optimization (cf. Candes and Tao)



Polysemanticity is what we’d expect to observe if features were not aligned with a neuron, despite incentives to align with the privileged basis.

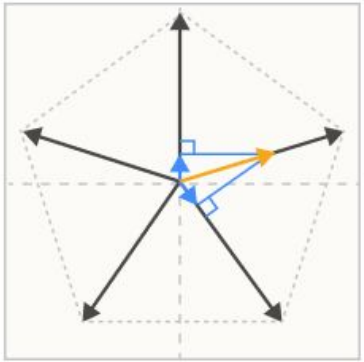


In the **superposition hypothesis**, features can’t align with the basis because the model embeds more features than there are neurons. Polysemanticity is inevitable if this happens.

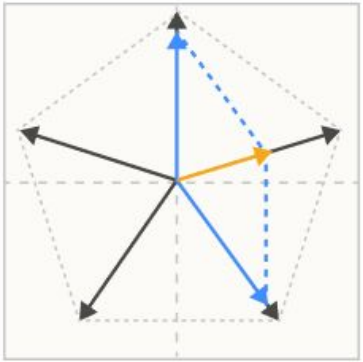
Elhage, et al., "Toy Models of Superposition", Transformer Circuits Thread, 2022.

Superposition

“Sparse” projections essentially mean that projections of high-dimensional vectors do not lie along multiple “almost orthogonal” vectors.



Even if only **one sparse feature** is active, using linear dot product projection on the superposition leads to **interference** which the model must tolerate or filter.



If the features aren’t as sparse as a superposition is expecting, **multiple present features** can additively interfere such that there are multiple possible nonlinear reconstructions of an **activation vector**.

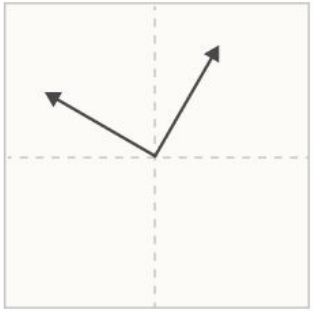
Elhage, et al., "Toy Models of Superposition", Transformer Circuits Thread, 2022.

Superposition

Superposition is a prominent roadblock for the standard set of MI methods, since you cannot find a “clean” set of features.

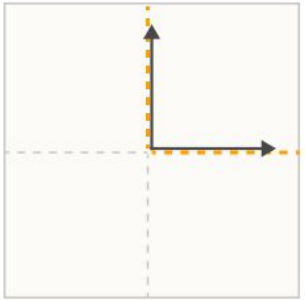
Features tend to interfere with each other, defying analysis.

This is the source of the problem with MLPs. While they are biased towards having privileged bases, superposition pushes representations away from being aligned.



In a **non-privileged basis**, features can be embedded in any direction. There is no reason to expect basis dimensions to be special.

Examples: word embeddings, transformer residual stream



In a **privileged basis**, there is an incentive for features to align with basis dimensions. This doesn’t necessarily mean they will.

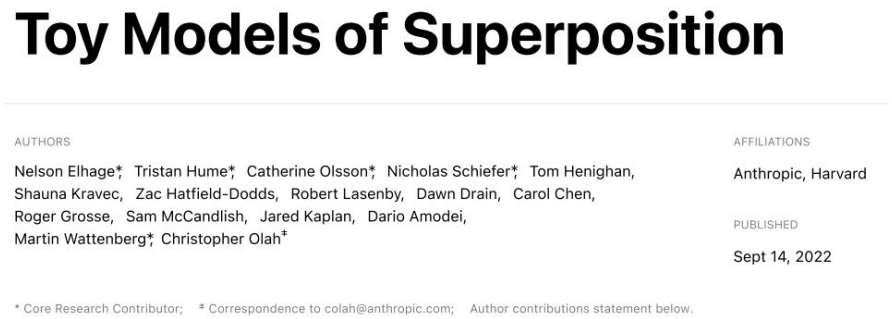
Examples: conv net neurons, transformer MLPs

Elhage, et al., "Toy Models of Superposition", Transformer Circuits Thread, 2022.

MI Research on Superposition

To account for the fact that superposition is a significant roadblock in the MI method, researchers have applied the Toy Model approach to develop a deep understanding of this phenomenon.

This culminated in a 62-page report that developed a theory of superposition in neural networks.



Covers:

- Superposition as the result of a mapping from high-dimensional NNs to low-dimensional ones
- Topological analysis of spaces under superposition.
- Computation across superposed subspaces.
- The effects of superposition on learning dynamics.