Transformer Circuits Thread

Toy Models of Superposition

AUTHORS

Nelson Elhage*, Tristan Hume*, Catherine Olsson*, Nicholas Schiefer*, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg*, Christopher Olah‡

AFFILIATIONS

Anthropic, Harvard

PUBLISHED

Sept 14, 2022

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

Abstract: Neural networks often pack many unrelated concepts into a single neuron – a puzzling phenomenon known as 'polysemanticity' which makes interpretability much more challenging. This paper provides a toy model where polysemanticity can be fully understood, arising as a result of models storing additional sparse features in "superposition." We demonstrate the existence of a phase change, a surprising connection to the geometry of uniform polytopes, and evidence of a link to adversarial examples. We also discuss potential implications for mechanistic interpretability.

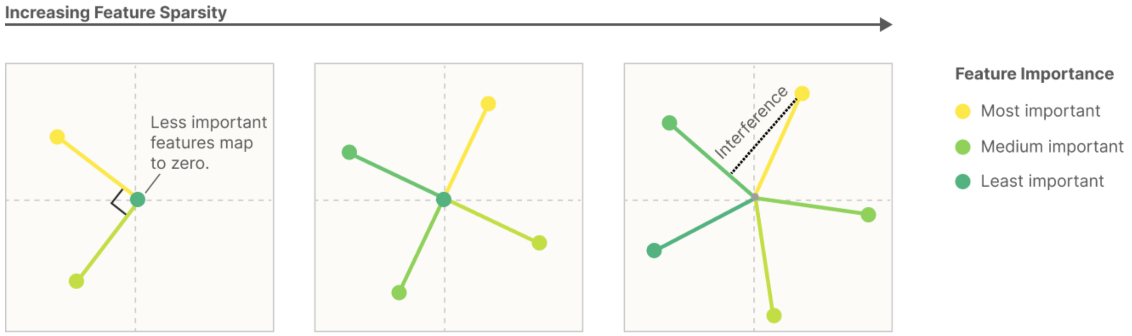
We recommend reading this paper as an HTML article.

It would be very convenient if the individual neurons of artificial neural networks corresponded to cleanly interpretable features of the input. For example, in an “ideal” ImageNet classifier, each neuron would fire only in the presence of a specific visual feature, such as the color red, a left-facing curve, or a dog snout. Empirically, in models we have studied, some of the neurons do cleanly map to features. But it isn't always the case that features correspond so cleanly to neurons, especially in large language models where it actually seems rare for neurons to correspond to clean features. This brings up many questions. Why is it that neurons sometimes align with features and sometimes don't? Why do some models and tasks have many of these clean neurons, while they're vanishingly rare in others?

In this paper, we use toy models — small ReLU networks trained on synthetic data with sparse input features — to investigate how and when models represent more features than they have dimensions. We call this phenomenon **superposition**. When features are sparse, superposition allows compression beyond what a linear model would do, at the cost of "interference" that requires nonlinear filtering.

Consider a toy model where we train an embedding of five features of varying importance[1] in two dimensions, add a ReLU afterwards for filtering, and vary the sparsity of the features. With dense features, the model learns to represent an orthogonal basis of the most important two features (similar to what Principal Component Analysis might give us), and the other three features are not represented. But if we make the features sparse, this changes:

As Sparsity Increases, Models Use "Superposition" To Represent More Features Than Dimensions



0% Sparsity

The two most important features are given **dedicated orthogonal dimensions**, while other features are **not embedded**.

80% Sparsity

The four most important features are represented as **antipodal pairs**. The least important features are **not embedded**.

90% Sparsity

All five features are embedded **as a pentagon,** but there is now "positive interference."

This figure and a few others can be reproduced using the toy model framework Colab notebook in our Github repo

Not only can models store additional features in superposition by tolerating some interference, but we'll show that, at least in certain limited cases, *models can perform computation while in superposition*. (In particular, we'll show that models can put simple circuits computing the absolute value function in superposition.) This leads us to hypothesize that *the neural networks we observe in practice are in some sense noisily simulating larger, highly sparse networks*. In other words, it's possible that models we train can be thought of as doing "the same thing as" an imagined much-larger model, representing the exact same features but with no interference.

Feature superposition isn't a novel idea. A number of previous interpretability papers have speculated about it [1, 2], and it's very closely related to the long-studied topic of compressed sensing in mathematics [3], as well as the ideas of distributed, dense, and population codes in neuroscience [4] and deep learning [5]. What, then, is the contribution of this paper?

For interpretability researchers, our main contribution is providing a direct demonstration that superposition occurs in artificial neural networks given a relatively natural setup, suggesting this may also occur in practice. We offer a theory of when and why this occurs, revealing a phase diagram for superposition. We also discover that, at least in our toy model, superposition exhibits complex geometric structure.

But our results may also be of broader interest. We find preliminary evidence that superposition may be linked to adversarial examples and grokking, and might also suggest a theory for the performance of mixture of experts models. More broadly, the toy model we investigate has unexpectedly rich structure, exhibiting phase changes, a geometric structure based on uniform polytopes, "energy level"-like jumps during training, and a phenomenon which is qualitatively similar to the fractional quantum Hall effect in physics. We originally investigated the subject to gain understanding of cleanly-interpretable neurons in larger models, but we've found these toy models to be surprisingly interesting in their own right.

KEY RESULTS FROM OUR TOY MODELS

In our toy models, we are able to demonstrate that:

- **Superposition is a real, observed phenomenon**.
- **Both monosemantic and polysemantic neurons can form.**
- **At least some kinds of computation can be performed in superposition.**
- **Whether features are stored in superposition is governed by a phase change.**
- **Superposition organizes features into geometric structures** such as digons, triangles, pentagons, and tetrahedrons.

Our toy models are simple ReLU networks, so it seems fair to say that neural networks exhibit these properties in at least some regimes, but it's very unclear what to generalize to real networks.

Definitions and Motivation: Features, Directions, and Superposition

In our work, we often think of neural networks as having *features of the input* represented as *directions in activation space*. This isn't a trivial claim. It isn't obvious what kind of structure we should expect neural network representations to have. When we say something like "word embeddings have a gender direction" or "vision models have curve detector neurons", one is implicitly making strong claims about the structure of network representations.

Despite this, we believe this kind of "linear representation hypothesis" is supported both by significant empirical findings and theoretical arguments. One might think of this as two separate properties, which we'll explore in more detail shortly:

- **Decomposability:** Network representations can be described in terms of independently understandable features.
- **Linearity:** Features are represented by direction.

If we hope to reverse engineer neural networks, we *need* a property like decomposability. Decomposability is what allows us to reason about the model without fitting the whole thing in our heads! But it's not enough for things to be decomposable: we need to be able to access the decomposition somehow. In order to do this, we need to *identify* the individual features within a representation. In a linear representation, this corresponds to determining which directions in activation space correspond to which independent features of the input.

Sometimes, identifying feature directions is very easy because features seem to correspond to neurons. For example, many neurons in the early layers of InceptionV1 clearly correspond to features (e.g. curve detector neurons [6]). Why is it that we sometimes get this extremely helpful property, but in other cases don't? We hypothesize that there are really two countervailing forces driving this:

- **Privileged Basis:** Only some representations have a *privileged basis* which encourages features to align with basis directions (i.e. to correspond to neurons).
- **Superposition:** Linear representations can represent more features than dimensions, using a strategy we call *superposition*. This can be seen as neural networks *simulating larger networks*. This pushes features *away* from corresponding to neurons.

Superposition has been hypothesized in previous work [1, 2]. However, we're not aware of feature superposition having been unambiguously demonstrated to occur in neural networks before ([7] demonstrates a closely related phenomenon of model superposition). The goal of this paper is to change that, demonstrating superposition and exploring how it interacts with privileged bases. If superposition occurs in networks, it deeply influences what approaches to interpretability research make sense, so unambiguous demonstration seems important.