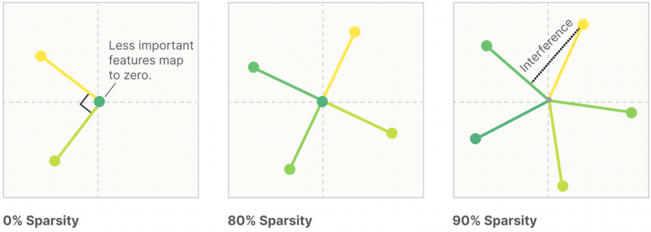
Toy Models of Superposition Replication and Findings

Abstract

Toy Models of Superpostion[1] is a groundbreaking paper published by researchers affilated with Anthropic and Harvard University in 2022. By investigating small models with under 100 neurons, the paper demonstrates that neural networks can represent more features than they have demensions. Additionally, they use these so called ”toy models” to understand the relationship between how neural networks are trained and how they represent the data internally. This paper was able to the finding from this paper and make new observations about ”toy models” and how they behave under different training circumstances.

1 Introduction

The orginal paper motivates the idea of superpostion with the following graphic:



The basic idea is this: if you think of each feature as being represented inside a nueral network as a direction, you can graph these directions and observe them. In the graphic above, the researchers studied a model with two neurons and five inputs. Recreating this model showed the same results when sparsity was varied. A replication of the original graphic can be found below and the code used to generate it can be found here.

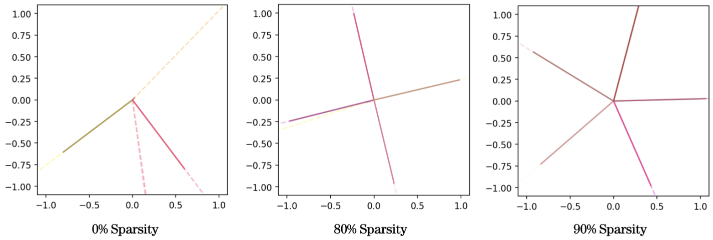


Figure 1: Graphing superposition in 2D.

Although this example only has two neurons (meaning it can be graphed in 2D), in future sections we will see how similar aproachs scale nicely to models with more neurons. As far as I understand, it is still unclear weather the ideas dicussed in the original paper can be scaled to help understand much larger models such as GTP-4 or Claude 2.

2 Background and Motivation

In this section of Toy Models of Superpostion[1], the authors provide context and define terms. In this paper, I make a few additional comments about some of the key ideas and terms the authors discuss.

(1) Defining Features: The original paper Toy Models of Superpostion defines features broadly as "properties of the input which a sufficiently large neural network will reliably dedicate a neuron to representing." The authors do however describe this definition as "slightly circular" and note that they are not "overly attached to it." I find the definition especially problematic because a network that is small or has unconventional archetecture may represent a feature that a larger network or a network with a more typical archetecture may represent. These representations are clearly still features, but are not treated as so under the original definition.

As a result, I propose an alternative definition: features are aspects of the input that a neural network represents accuratly with a higher propability than a randomly initialized network. In other words, features are parts of the input that a model determines to be important enough to represent internally.

(2) Role of linear Representations in Neural Networks: The original authors of the paper study interpretability by trying to understand the linear representations within neural networks. It is worth noting that this isn't the only way to approach interpretability research. Understanding the role of non-linearities at each level is likely also very important (and perhaps more neglected).

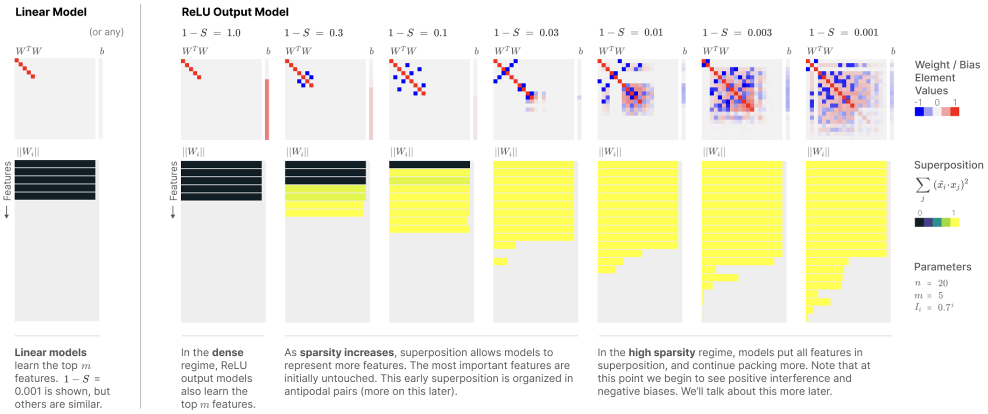
(3) Defining Superposition: The original paper has a fantastic and simple defination for Superposition: "Roughly, the idea of superposition is that neural networks 'want to represent more features than they have neurons', so they exploit a property of high-dimensional spaces to simulate a model with many more neurons." This is the definition I will used throughout this paper.

3 Demonstrating Superposition

In this section the authors of the original paper demonstrate that superposition is observable even in models with more than two neurons (like the ones we saw in the introduction). They first study models with 20 inputs and 5 neurons. They train a linear model and one with a ReLU activation function. In the ReLU model they vary sparsity and observe superposition.

The authors demonstrate this by graphing W^TW where positive numbers are red and negative ones are blue. They also graph the length of each feature (by treating each column in W as a vector). Feature that are orthogonal to others in W are labeled black. Features that interfere with others are colored yellow.

The first step in replicating these findings was to train the linear and ReLU models that don't perform computatoin in superposition. The linear model was defined by $W^TWx + b$ and the



Linear models learn the top m features. $1 - S = 0.001$ is shown, but others are similar.

In the **dense** regime, ReLU output models also learn the top m features.

As **sparsity increases**, superposition allows models to represent more features. The most important features are initially untouched. This early superposition is organized in antipodal pairs (more on this later).

In the **high sparsity** regime, models put all features in superposition, and continue packing more. Note that at this point we begin to see positive interference and negative biases. We'll talk about this more later.

Figure 2: Demostrating superposition in Toy Models of Superposition [1]

ReLU model was defined by $\text{ReLU}(W^TWx + b)$. I trained both models with Adam (learning rate $= 1 * 10^{-}3$) on 20,000 batchs of 256 examples.

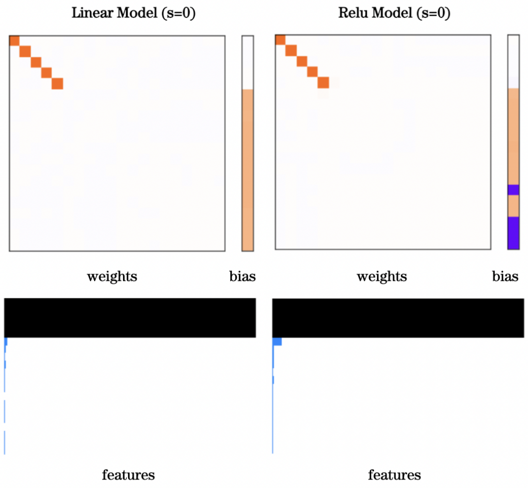


Figure 3: The generated graphics show that the model uses each of its five dimensions to graph the 5 most important features orthogonally.

There are a few small differences between my image and the original images that I think are worth noting before moving on. First, I use orange to indicate positive numbers and purple to indicate negative numbers. This is different from the red and blue used by the original authors to make it clear whose graphic is whose. Similarly, while the original authors use yellow to indicate features in superposition in Figure 2, I will use blue (hard to see in Figure 3 but will be more obvious going forward).

3.1 Calculating Superposition

The models in Figure 3 do not exibit superposition. They encode each five most important features orthogonally (one feature for each neuron in the model). Going forward however, we will be investigating models that do not behave that way, instead encoding features as vectors that interfere with eachother. In order to explain this phenomenon, and describe the graphs in Figure 4 it will be useful to dive a little bit into some math. The color of the feature bars at the bottom of Figure 4 are determined by the following equation.

$$\text{Interference} = \sum_{j \neq i} (\hat{W}_i \cdot \hat{W}_j)^2 \tag{1}$$

For a given column i in weight matrix W, it calculates interference by taking the dot product with every other column in W. Non-zero dot products indicate that the columns in W are not orthogonal. As a result, summing these dot products gives a general idea of how much the network is representing a given feature in superposition. Note that $\hat{W}_i$ is the unit vector for W_i. This is necessary because when calculating interference, we are interested in the direction of a given feature, not its length.

In Figure 4 the length of a feature (calculated by taking the length of the vector W_i) determines the lengths of the bars in the feature graph. The interference equation (Equation 1) determines the color of the columns: black indicates a low value for Equation 1 while blue indicates a higher value. Blue bars show that a given feature is represented in superposition while black bars indicate that the feature is mapped orthogonally.

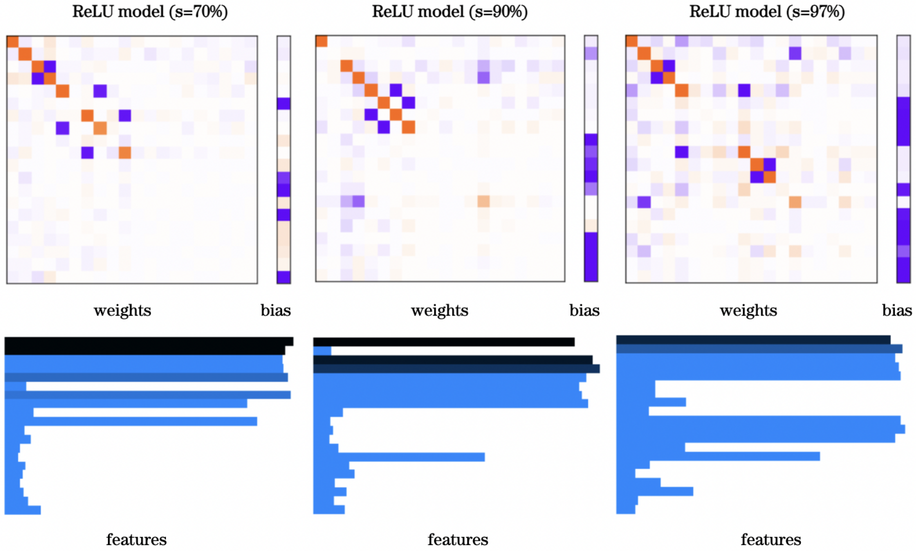


Figure 4: By varying sparsity, superposition observed in ReLU models. This figure shows three models with increasing sparsity levels. As sparsity is increased, more features are mapped in superposition.

References

[1] 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, and Christopher Olah. Toy models of superposition, 2022.