Toy Models of Superposition Replication and Findings

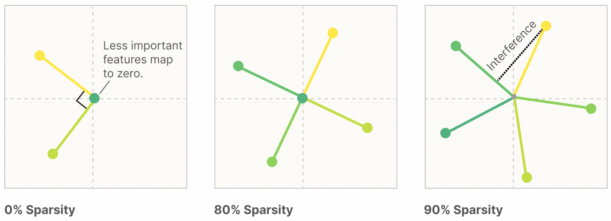
Zephaniah Roe zroe@uchicago.edu
Undergraduate Student at the University of Chicago

Abstract

Toy Models of Superposition[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. The original paper is quite extensive. As a result, this replication focuses on reproducing the most important results from the introduction and sections 2 and 3 of the original paper. It also includes some comentary on section 1.

1 Introduction

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



The basic idea is this: if you think of each feature as being represented inside of a neural network by a direction, you can graph these directions and observe them. By doing this, the authors of the original paper demonstrate that the way a model maps features as directions depends on the sparsity of it's training data. A replication of this phenomenon can be found below and the code used to generate it can be found here.

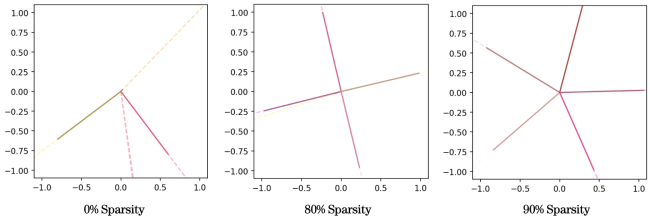


Figure 1: Graphing superposition in 2D.

The model studied in Figure 1 is designed such that each column in the weight matrix corresponds to a given input. Becasuse the weight matrix only represents 2 neurons, the columns of the matrix can be graphed in 2D. As a result, it is trivial to plot the columns as 2D vectors (each representing individual features of the input). Obersving these vectors while increasing the sparsity of the model's input reveals that the model can be trained to represent many more features than it has dimensions (despite having only 2 neurons, the model in Figure 1 can represent up to 5 features!). This is what the authors call "superposition." In future sections we will study superposition extensivly and produce the phenomenon in larger networks.

2 Background and Motivation

In this section of *Toy Models of Superposition*, the authors provide context and define terms. In this paper, I make a few additional comments about some of the key ideas from this section.

(1) Defining Features: *Toy Models of Superposition* 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 architecture may represent a feature that a larger network or a network with a more typical architecture may ignore. 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 accurately with a significantly higher probability 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 mechanistic 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 compelling yet simple definition 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 use throughout this paper.

3 Demonstrating Superposition

In the introduction, the authors of the original paper proved that models with two neurons could exhibit superposition (this result was reproduced in Figure 1). In this section, however, the authors demonstrate that superposition is also observed in models with more than two neurons.

Specifically, they begin by exploring models with 20 inputs and 5 neurons, ultimately proving that these models exhibit superposition under certain conditions. The authors demonstrate this by graphing W^TW for the weight matrix W in each model (shown in Figure 2). They represent positive numbers in the matrix as red and negative ones as blue. They also graph the length of each feature by treating each column in W as a vector. Features that are orthogonal to others in W are labeled black while features that aren't are labeled yellow (the exact details for how this is

calculated is discussed in 3.1).

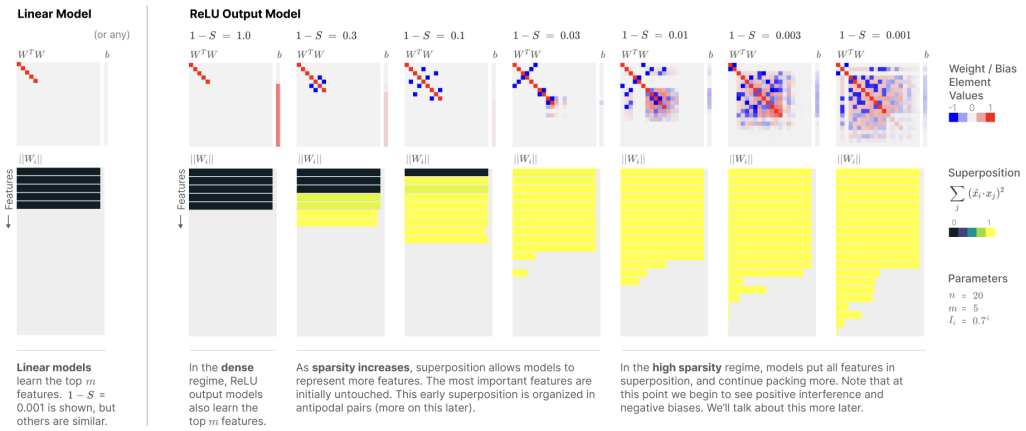


Figure 2: Graphs linear and ReLU models from *Toy Models of Superposition*[1]

In Figure 2, the authors study both a linear model and a model with a ReLU activation function. They found that both the linear and ReLU model did not exhibit superposition in the absence of sparsity. By increasing the sparsity of the input, however, the ReLU model begins to clearly exhibit superposition by ceasing to represent features orthogonally.

The first step in replicating these findings was to train the linear and ReLU models that don't perform computation in superposition. The linear model was defined by W^TWx+b and the ReLU model was defined by $\text{ReLU}(W^TWx+b)$. The objective of each model was to reconstruct the input x. Both models were trained with the Adam optimizer (learning rate $= 1*10^{-3}$) on 20,000 batches of 256 examples.

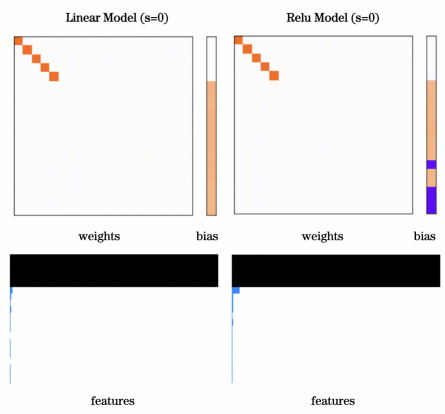


Figure 3: The generated graphics show that the model uses each of its five dimensions to graph the 5 most important features orthogonally.

Note that in Figure 3, I use orange to indicate positive numbers and purple to indicate negative ones. This is different from the red and blue in Figure 2, to distinguish my work from that of the original authors. Similarly, while the original authors use yellow to indicate features in superposition, I use blue (This is hard to see in Figure 3 but it will be more obvious going forward).

3.1 Calculating Superposition

The models in Figure 3 do not exhibit superposition. They encode the five most important features orthogonally (one feature for each neuron in the model). In this section, we will be investigating models that do not behave in this way, instead encoding features as vectors that interfere with eachother. The model explored in this section have the same architecture and objective as the models in Figure 3. The difference is that the models in this section are trained on sparse input data and, as a result, map features interally in superposition.

In order to explain this phenomenon and demonstate how models with sparse input are able to represent features in superpostion, it will be useful to dive into the math behind the concept of feature interference. The extent to which features interfere with eachother is defined 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, interference is calculated 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 width of the bars in the feature graph (shown in the bottom half of the figure). The interference equation (Equation 1) determines the color of the columns: black indicates a low value for interference while blue indicates a higher value. This means that blue bars show that a given feature is represented in superposition while black bars indicate that the feature is mapped orthogonally.

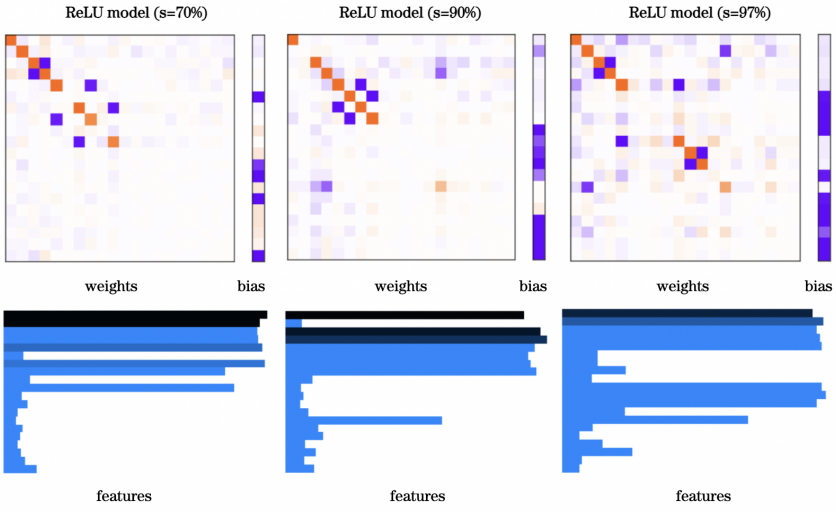


Figure 4: Superposition is observed in models trained on 70%, 90% and 97% sparse inputs (code to generate these figures can be found here). The 70% and 90% sparse models were trained on 50,000 batches of 256 examples with the "RMSProp" optimizer (learning rate = 10^{-2}). The 97% sparsity model was trained on 100,000 batches of 256 examples using the Adam opimizer (learning rate = 10^{-2}).

Unlike the models with 0% sparsity in Figure 3, the models in Figure 4 have higher levels of sparsity and, as a result, leverage superposition. The bottom half of Figure 4 shows that these models represent far many more features than the models in Figure 3, but by doing so, they are forced to represent many of their features in superposition. The models only have 5 neurons so if they "want" to represent more than 5 features, they can't represent each feature orthogonally. This tradeoff is intuatively more attractive when the model is trained on sparse inputs becase it is less likely that the model will be fed a combination of inputs that cause feature representations to conflict (because a significant percentage of the input is 0).

3.2 Models Trained on Very Sparse Data

As sparsity is increased to almost 100% the models stop representing any features orthogonally. This is displayed in Figure 5 where models are trained on 99%, 99.7% and 99.9% sparse inputs.

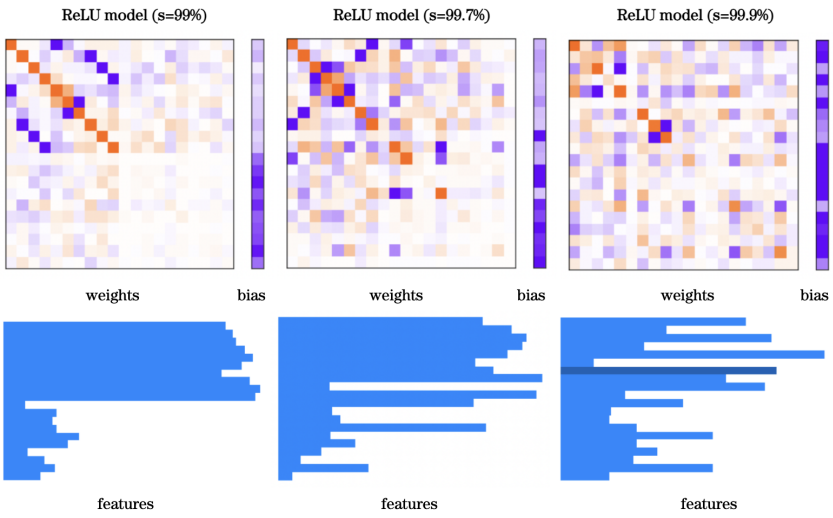


Figure 5: When models are trained on sufficiently sparse data, all feature representations are in superposition. The models in the figure were trained on 100,000 batches of 256 examples using the Adam opimizer (learning rate = 10^{-2}).

The representations of W^TW of these very sparse models (shown in the top half of Figure 5) is far less clean than previous representations we have seen. This is the same trend the original authors found when increasing sparsity of these models (Figure 2 illustrates how the original authors displayed this visually).

The feature representations, shown in the bottom have of Figure 5, are also consistent with the the findings of *Toy Models of Superposition*. Like the investigation from the original paper, these feature representations show no features mapped orthogonally (recall that features that interfere with eachother are shown in blue).

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.