

$$L \sim \sum_i I_i (1 - \|W_i\|^2)^2 + \sum_{i \neq j} I_j (W_j \cdot W_i)^2$$

**Feature benefit** is the value a model attains from representing a feature. In a real neural network, this would be analogous to the potential of a feature to improve predictions if represented accurately.

**Interference** between  $x_i$  and  $x_j$  occurs when two features are embedded non-orthogonally and, as a result, affect each other's predictions. This prevents superposition in linear models.

The Saxe results reveal that there are fundamentally two competing forces which control learning dynamics in the considered model. Firstly, the model can attain a better loss by representing more features (we've labeled this "feature benefit"). But it also gets a worse loss if it represents more than it can fit orthogonally due to "interference" between features.<sup>11</sup> In fact, this makes it never worthwhile for the linear model to represent more features than it has dimensions.<sup>12</sup>

Can we achieve a similar kind of understanding for the ReLU output model? Concretely, we'd like to understand  $L = \int_x \|I(x - \text{ReLU}(W^T W x + b))\|^2 d\mathbf{p}(x)$  where  $x$  is distributed such that  $x_i = 0$  with probability  $S$ .

The integral over  $x$  decomposes into a term for each sparsity pattern according to the binomial expansion of  $((1-S) + S)^n$ . We can group terms of the sparsity together, rewriting the loss as  $L = (1-S)^n L_n + \dots + (1-S) S^{n-1} L_1 + S^n L_0$ , with each  $L_k$  corresponding to the loss when the input is a  $k$ -sparse vector. Note that as  $S \rightarrow 1$ ,  $L_1$  and  $L_0$  dominate. The  $L_0$  term, corresponding to the loss on a zero vector, is just a penalty on positive biases,  $\sum_i \text{ReLU}(b_i)^2$ . So the interesting term is  $L_1$ , the loss on 1-sparse vectors:

$$L_1 = \sum_i \int_{0 \leq x_i \leq 1} I_i (x_i - \text{ReLU}(\|W_i\|^2 x_i + b_i))^2 + \sum_{i \neq j} \int_{0 \leq x_i \leq 1} I_j \text{ReLU}(W_j \cdot W_i x_i + b_j)^2$$

If we focus on the case  $x_i = 1$ , we get something which looks even more analogous to the linear case:

$$= \sum_i I_i (1 - \text{ReLU}(\|W_i\|^2 + b_i))^2 + \sum_{i \neq j} I_j \text{ReLU}(W_j \cdot W_i + b_j)^2$$

**Feature benefit** is similar to before. Note that ReLU never makes things worse, and that the bias can help when the model doesn't represent a feature by taking on the expected value.

**Interference** is similar to before but ReLU means that negative interference, or interference where a negative bias pushes it below zero, is "free" in the 1-sparse case.

This new equation is vaguely similar to the famous Thomson problem in chemistry. In particular, if we assume uniform importance and that there are a fixed number of features with  $\|W_i\| = 1$  and the rest have  $\|W_i\| = 0$ , and that  $b_i = 0$ , then the feature benefit term is constant and the interference term becomes a generalized Thomson problem – we're just packing points on the surface of the sphere with a slightly unusual energy function. (We'll see this can be a productive analogy when we resume our empirical investigation in the following sections!)

Another interesting property is that ReLU makes negative interference free in the 1-sparse case. This explains why the solutions we've seen prefer to only have negative interference when possible. Further, using a negative bias can convert small positive interferences into essentially being negative interferences.

What about the terms corresponding to less sparse vectors? We leave explicitly writing these out to the reader, but the main idea is that there are multiple compounding interferences, and the "active features" can experience interference. In a [later section](#), we'll see that features often organize themselves into sparse interference graphs such that only a small number of features interfere with another feature – it's interesting to note that this reduces the probability of compounding interference and makes the 1-sparse loss term more important relative to others.

## Superposition as a Phase Change

The results in the previous section seem to suggest that there are three outcomes for a feature when we train a model: (1) the feature may simply not be learned; (2) the feature may be learned, and represented in superposition; or (3) the model may represent a feature with a dedicated dimension. The transitions between these three outcomes seem sharp. Possibly, there's some kind of phase change.<sup>13</sup>

One way to understand this better is to explore if there's something like a "phase diagram" from physics, which could help us understand when a feature is expected to be in one of these regimes. Although we can see hints of this in [our previous experiment](#), it's hard to really isolate what's going on because many features are changing at once and there may be interaction effects. As a result, we set up the following experiment to better isolate the effects.

As an initial experiment, we consider models with 2 features but only 1 hidden layer dimension. We still consider the ReLU output model,  $\text{ReLU}(W^T W x - b)$ . The first feature has an importance of 1.0. On one axis, we vary the importance of the 2nd "extra" feature from 0.1 to 10. On the other axis, we vary the sparsity of all features from 1.0 to 0.01. We then plot whether the 2nd "extra" feature is not learned, learned in superposition, or learned and represented orthogonally. To reduce noise, we train ten models for each point and average over the results, discarding the model with the highest loss.

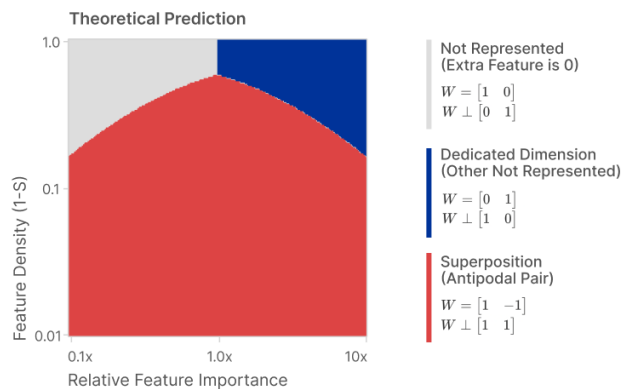
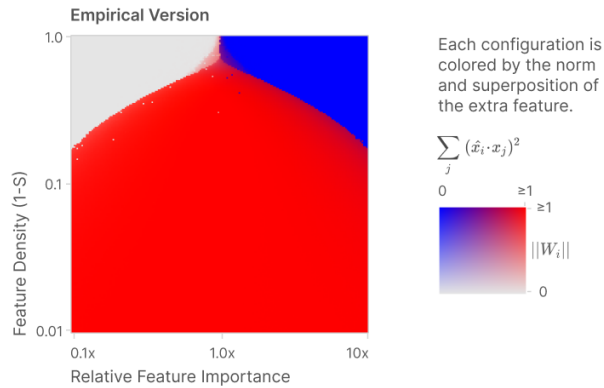
We can compare this to a theoretical "toy model of the toy model" where we can get closed form solutions for the loss of different weight configurations as a function of importance and sparsity. There are three natural ways to store 2 features in 1 dimension:  $W = [1, 0]$  (ignore  $[0, 1]$ , throwing away the extra feature),  $W = [0, 1]$  (ignore  $[1, 0]$ , throwing away the first feature to give the extra feature a dedicated dimension), and  $W = [1, -1]$  (store the features in superposition, losing the ability to represent  $[1, 1]$ , the combination of both features at the same time). We call this last solution "antipodal" because the two basis vectors  $[1, 0]$  and  $[0, 1]$  are mapped in opposite directions. It turns out we can analytically determine the loss for these solutions (details can be found in [this notebook](#)).

## Sparsity-Relative Importance Phase Diagram (n=2, m=1)

What happens to an "extra feature" if the model can't give each feature a dimension? There are three possibilities, depending on feature sparsity and the extra feature's importance relative to other features:

- Extra Feature is Not Represented
- Extra Feature Gets Dedicated Dimension
- Extra Feature is Stored In Superposition

We can both study this empirically and build a theoretical model:



As expected, sparsity is necessary for superposition to occur, but we can see that it interacts in an interesting way with relative feature importance. But most interestingly, there appears to be a real phase change, observed in both the empirical and theoretical diagrams! The optimal weight configuration discontinuously changes in magnitude and superposition. (In the theoretical model, we can analytically confirm that there's a first-order phase change: there's crossover between the functions, causing a discontinuity in the derivative of the optimal loss.)

We can ask this same question of embedding three features in two dimensions. This problem still has a single "extra feature" (now the third one) we can study, asking what happens as we vary its importance relative to the other two and change sparsity.

For the theoretical model, we now consider four natural solutions. We can describe solutions by asking "what feature direction did  $W$  ignore?" For example,  $W$  might just not represent the extra feature – we'll write this  $W \perp [0, 0, 1]$ . Or  $W$  might ignore one of the other features,  $W \perp [1, 0, 0]$ . But the interesting thing is that there are two ways to use superposition to make antipodal pairs. We can put the "extra feature" in an antipodal pair with one of the others ( $W \perp [0, 1, 1]$ ) or put the other two features in superposition and give the extra feature a dedicated dimension ( $W \perp [1, 1, 0]$ ). Details on the closed form losses for these solutions can be found in [this notebook](#). We do not consider a last solution of putting all the features in joint superposition,  $W \perp [1, 1, 1]$ .