Model

We start with the model from Toy Models of Superposition. Ie:

$$h = Wx$$
$$x' = \text{ReLU}(W^\top h + b)$$

with the original loss function as

$$L_1 = \sum_x \sum_i I_i (x_i - x'_i)^2$$

Quoting from the paper:

> The input vectors x are synthetic data intended to simulate the properties we believe the true underlying features of our task have. We consider each dimension x_i to be a "feature" Each one has an associated sparsity S_i and importance I_i. We let $x_i = 0$ with probability S_i, but it is otherwise uniformly distributed between $[0, 1]$. In practice, we focus on the case where all features have the same sparsity.

The model has n features and m internal dimensions ($W \in \mathbb{R}^{m \times n}$)

Behavior of the Model

The paper illustrates a number of phenomena that emerge from this model:

1. When features are very dense, the model only represents the m most important features. It does so by constructing W having its first m columns orthogonal. This allows perfect reconstruction of the top m features, but nothing else.
2. As features grow sparser, the model starts to represent more features, putting features into superposition.
3. The geometry of superposed features becomes more complex as sparsity increases (and the number of features represented grows). Features organize themselves into tegum products of simple structures, often simplices. This means that interacting features form groups, and there is no interaction between featurs in different groups.
4. "Correlated features prefer to be orthogonal, often forming in different tegum factors."

The Study

We hypothesize that when an attention head attends to a pair of tokens, that its singular vectors align with the features in the tokens that are causal for attention.

We'll model this as follows. We'll extend the above model to include a pseudo-attention head which we will train to respond to specific token pairs. **Without giving the attention head access to the feature dictionary,** we will observe whether the attention head can discover the representations of the features.

In the above model the feature dictionary is W. Assume that we decide the head should attend to a token pair (h, g) if h contains feature i (W_i) and g contains feature j (W_j). Then we ask whether the first left singular vector of the attention head aligns with W_i, and the first right singular vector aligns with W_j.

Extended Model

We now extend the model to incorporate an objective similar to what an attention does in a transformer.

We treat the pseudo-attention head as having the following objective. We assume certain pairs of features are "of interest" to the head. Define a function $s(h, g)$ where $h = Wx$ and $g = Wy$ are internal representations of x and y. Choose a particular (i, j) feature combination. Then define

$$s(h, g) = 1 \iff x_i > 0, y_j > 0, \text{ and 0 otherwise}$$

This function represents the "target logit" that the pseudo-head should output when presented with two tokens.

(Other options: $s(h, g) = x_i x_j$ for (i, j) only. Also consider different target weights : 2, 3, etc)

The head is defined by matrices $A, B \in \mathbb{R}^{r \times m}$. Given h, g, the pseudo-head calculates predicted logits:

$$\ell = h^\top A^\top B g$$

To model the goal of having the head respond to features in its input, we construct a new term for our loss function:

$$L_2 = (s(h, g) - \ell)^2$$

and our new loss fucntion becomes

$$L = L_1 + \lambda L_2.$$

Research Questions

Define $\Omega = A^\top B$.

1. Do singular vectors of Ω recover the feature representations? Short answer, at least in simple cases tried so far: yes. Longer answer: when sparsity is low, such that top m features are orthogonally represented, singular vectors can recover representations of features from the top m group.
    - Need good, thorough plots and visualizations of this.
2. What happens during training? At what point during training do SVs show alignment? Before or after W columns become orthogonal?
3. In the $m = r$ regime, with low sparsity, can all m features be recovered by Ω? What happens when the head dimension is lower than the representation dimension ($r < m$)? Which features get represented, or does something more tricky happen, like one slice works for two dimensions?
4. When one match is more important than another, is this reflected in the singular values? ie, say $s(x, y) = 2$ for some token pair, compare to one for some other token pair. [Note: check code of model to make sure this is correctly handled.]
5. As sparsity declines, features organize into a tegum product of simplices.
    1. When the head attends to features in different subspaces, do SVs span the subspaces? If there are not enough SVs, do they allocate themselves to separate subspaces?
    2. When the head attends to features in the same subspace (ie non-orthogonal features), how do SVs organize? Hypothesis: the need to pay attention to features that are non-orthogonal will shift those features to become orthogonal, and allocate distinct SVs to them. In other words, attn heads encourage orthogonal representations among the features they attend to. This sheds light on how transformers affect feature representations.
    3. Consider two features in antipodal superposition. Does attending to one of the features destroy the grouping? Does attending to both features destroy the grouping? What about three features in a simplex arrangement?
6. When a head attends to multiple features paired with a single feature, does the singular vector have components corresponding to all the features?

The latter experiments depend on getting features to self-organize in a particular way. This may be hard when also training the pseudo-head. Consider breaking the model into two pieces: first train the representations, then train the heads. This should be as easy as putting a switch on the optimize() function.