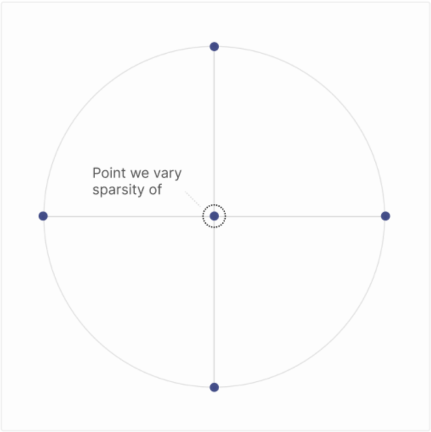
PERTURBING A SINGLE FEATURE

The simplest kind of non-uniform superposition is to vary one feature and leave the others uniform. As an experiment, let's consider an experiment where we represent $n = 5$ features in $m = 2$ dimensions. In the uniform case, with importance $I = 1$ and activation density $1 - S = 0.05$, we get a regular pentagon. But if we vary one point – in this case we'll make it more or less sparse – we see the pentagram *stretch* to account for the new value. If we make it denser, activating more frequently (yellow) the other features repel from it, giving it more space. On the other hand, if we make it sparser, activating less frequently (blue) it takes less space and other points push towards it.

If we make it sufficiently sparse, there's a phase change, and it collapses from a pentagon to a pair of digons with the sparser point at zero. The phase change corresponds to loss curves corresponding to the two different geometries crossing over. (This observation allows us to directly confirm that it is genuinely a first order phase change.)

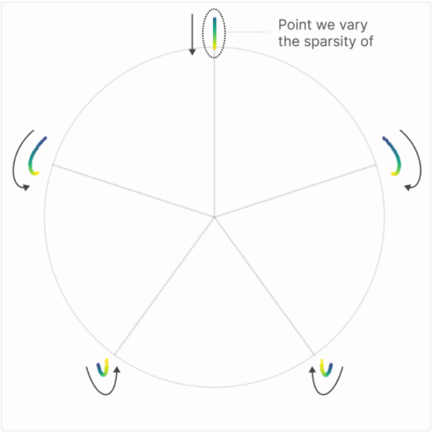
To visualize the solutions, we canonicalize them, rotating them to align with each other in a consistent manner.

Digon (Square) Solutions



When the sparsity of the varied point falls below a certain critical threshold (~2.5x less than others) the pentagon solution changes to two digons.

Pentagon Solutions



Note how vertices shift as sparsity changes

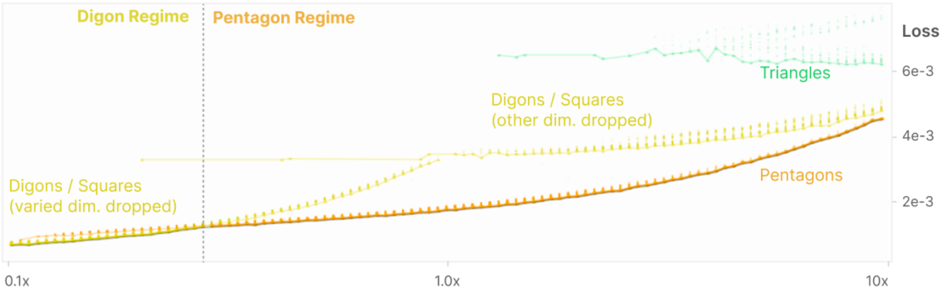
To study non-uniform sparsity, we consider models with five features, varying the sparsity of a single feature and observing how the resulting solutions change. We observe a mixture of continuous deformation and sharp phase changes.

Parameters

n = 5
m = 2
I_i = 1
1-S = 0.05 (baseline)

Relative Feature Density (1-S)

The Pentagon-Digon Phase Change Corresponds to a Loss Curve Crossover



Gradient descent has trouble moving between solutions associated with different geometries. As a result, fitting the model will often produce non-optimal solutions. By characterizing and plotting these, we can see that each geometry creates a different loss curve, and that the pentagon-digon phase change corresponds to a cross over between the curves.

These results seem to suggest that, at least in some cases, non-uniform superposition can be understood as a *deformation of uniform superposition* and *jumping between uniform superposition configurations* rather than a totally different regime. Since uniform superposition has a lot of understandable structure, but real world superposition is almost certainly non-uniform, this seems very promising!

The reason pentagonal solutions are not on the unit circle is because models reduce the effect of positive interference, setting a slight negative bias to cut off noise and setting their weights to $||W_i|| = 1/(1 - b_i)$ to compensate. Distance from the unit circle can be interpreted as primarily driven by the amount of positive interference.

A note for reimplementations: optimizing with a two-dimensional hidden space makes this easier to study, but the actual optimization process to be really challenging from gradient descent – a lot harder than even just having three dimensions. Getting clean results required fitting each model multiple times and taking the solution with the lowest loss. However, there's a silver lining to this: visualizing the sub-optimal solutions on a scatter plot as above allows us to see the loss curves for different geometries and gain greater insight into the phase change.

Correlated and Anticorrelated Features

A more complicated form of non-uniform superposition occurs when there are correlations between features. This seems essential for understanding superposition in the real world, where many features are correlated or anti-correlated.

For example, one very pragmatic question to ask is whether we should expect polysemantic neurons to group the same features together across models. If the groupings were random, you could use this to detect polysemantic neurons, by comparing across models! However, we'll see that correlational structure strongly influences which features are grouped together in superposition.

The behavior seems to be quite nuanced, with a kind of "order of preferences" for how correlated features behave in superposition. The model ideally represents correlated features orthogonally, in separate tegum factors with no interactions between them. When that fails, it prefers to arrange them so that they're as close together as possible – it prefers positive interference between correlated features over negative interference. Finally, when there isn't enough space to represent all the correlated features, it will collapse them and represent their principal component instead! Conversely, when features are anti-correlated, models prefer to have them interfere, especially with negative interference. We'll demonstrate this with a few experiments below.

SETUP FOR EXPLORING CORRELATED AND ANTICORRELATED FEATURES

Throughout this section we'll refer to "correlated feature sets" and "anticorrelated feature sets".

Correlated Feature Sets. Our correlated feature sets can be thought of as "bundles" of co-occurring features. One can imagine a highly idealized version of what might happen in an image classifier: there could be a bundle of features used to identify animals (fur, ears, eyes) and another bundle used to identify buildings (corners, windows, doors). Features from one of these bundles are likely to appear together. Mathematically, we represent this by linking the choice of whether all the features in a correlated feature set are zero or not together. Recall that we originally defined our synthetic distribution to have features be zero with probability S and otherwise uniformly distributed between [0,1]. We simply have the same sample determine whether they're zero.

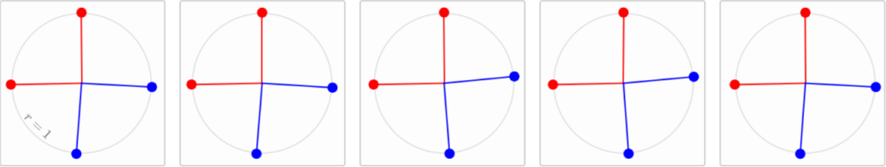
Anticorrelated Feature Sets. One could also imagine anticorrelated features which are extremely unlikely to occur together. To simulate these, we'll have anticorrelated feature sets where only one feature in the set can be active at a time. To simulate this, we'll have the feature set be entirely zero with probability S, but then only have one randomly selected feature in the set be uniformly sampled from [0,1] if it's active, with the others being zero.

ORGANIZATION OF CORRELATED AND ANTICORRELATED FEATURES

For our initial investigation, we simply train a number of small toy models with correlated and anti-correlated features and observe what happens. To make this easy to study, we limit ourselves to the $m = 2$ case where we can explicitly visualize the weights as points in 2D space. In general, such solutions can be understood as a collection of points on a unit circle. To make solutions easy to compare, we rotate and flip solutions to align with each other.

Models prefer to represent correlated features in orthogonal dimensions.

We train several models with 2 sets of 2 correlated features (n=4 total) and a m=2 hidden dimensions. We then visualize the weight column for each feature. For ease of comparison, we rotate and flip solutions to have a consistent orientation.

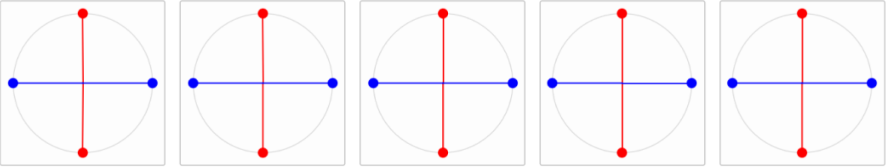


🔵🔵 and 🔴🔴 denote **correlated** feature sets.

Correlated feature sets are constructed by having them always co-occur (ie. be zero or not) at the same time.

Models prefer to represent anticorrelated features in opposite directions.

We train several models with 2 sets of 2 anticorrelated features (n=4 total) and a m=2 hidden dimensions. We then visualize the weight column for each feature. For ease of comparison, we rotate and flip solutions to have a consistent orientation.

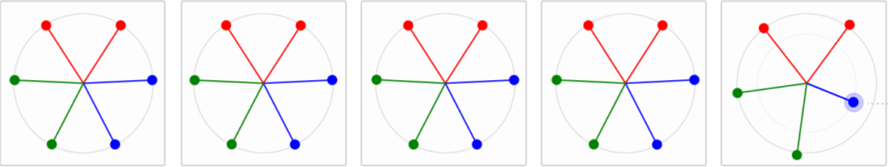


🔵🔵 and 🔴🔴 denote **anticorrelated** feature sets.

Anticorrelated feature sets are constructed by having them never co-occur (ie. be zero or not) at the same time.

Models prefer to arrange correlated features side by side if they can't be orthogonal.

We train several models with 3 sets of 2 correlated features (n=6 total) and a m=2 hidden dimensions. We then visualize the weight column for each feature. For ease of comparison, we rotate and flip solutions to have a consistent orientation. (Note that models will not embed 6 independent features as a hexagon like this.)



🔵🔵, 🔴🔴, and 🟢🟢 denote **correlated** feature sets.

Sometimes correlated feature sets "collapse". In this case it's an optimization failure, but we'll return to it shortly as an important phennomenon.

LOCAL ALMOST-ORTHOGONAL BASES

It turns out that the tendency of models to arrange correlated features to be orthogonal is actually quite a strong phenomenon. In particular, for larger models, it seems to generate a kind of "local almost-orthogonal basis" where, even though the model as a whole is in superposition, the correlated feature sets considered in isolation are (nearly) orthogonal and can be understood as having very little superposition.

To investigate this, we train a larger model with two sets of correlated features and visualize W^TW.