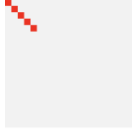


$W^T W$ 

It tends to be easier to visualize  $W^T W$  than  $W$ . Here we see that  $W^T W$  is an **identity matrix** for the most important features and **0** for less important ones.

 $b$ 

We can also look at the bias,  $b$ . The bias is **zero** for features learned to pass through, and the **expected value** (a positive number) for others.

Weight / Bias Element Values

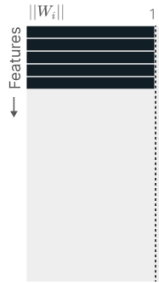


But the thing we really care about is this hypothesized phenomenon of superposition – does the model represent "extra features" by storing them non-orthogonally? Is there a way to get at it more explicitly? Well, one question is just how many features the model learns to represent. For any feature, whether or not it is represented is determined by  $\|W_i\|$ , the norm of its embedding vector.

We'd also like to understand whether a given feature shares its dimension with other features. For this, we calculate  $\sum_{j \neq i} (\hat{W}_i \cdot W_j)^2$ , projecting all other features onto the direction vector of  $W_i$ . It will be 0 if the feature is orthogonal to other features (dark blue below). On the other hand, values  $\geq 1$  mean that there is some group of other features which can activate  $W_i$  as strongly as feature  $i$  itself!

We can visualize the model we looked at previously this way:

Features



We want to understand which features the model chooses to represent in its hidden representation, and whether they're orthogonal to each other.

To do this, we visualize the norm of each feature's direction vector,  $\|W_i\|$ . This will be  $\sim 1$  if a feature is fully represented, and zero if it is not. For each feature, we also use color to visualize whether it is orthogonal to other features (i.e. in superposition).

This model simply dedicates one dimension to each of the most important features, representing them orthogonally.

Superposition

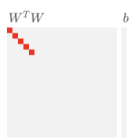
$$\sum_j (\hat{x}_i \cdot x_j)^2$$



Now that we have a way to visualize models, we can start to actually do experiments. We'll start by considering models with only a few features ( $n = 20$ ;  $m = 5$ ;  $I_i = 0.7^i$ ). This will make it easy to visually see what happens. We consider a linear model, and several ReLU-output models trained on data with different feature sparsity levels:

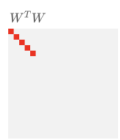
### Linear Model

(or any)

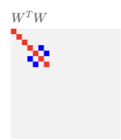


**Linear models** learn the top  $m$  features.  $1 - S = 0.001$  is shown, but others are similar.

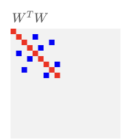
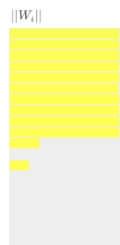
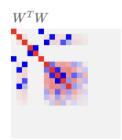
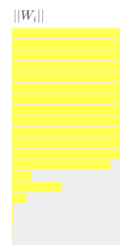
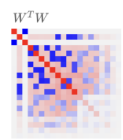
### ReLU Output Model

 $1 - S = 1.0$ 

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

 $1 - S = 0.3$ 

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).

 $1 - S = 0.1$  $1 - S = 0.03$  $1 - S = 0.01$  $1 - S = 0.003$  $1 - S = 0.001$ 

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.

Weight / Bias Element Values



Superposition

$$\sum_j (\hat{x}_i \cdot x_j)^2$$



Parameters

$n = 20$   
 $m = 5$   
 $I_i = 0.7^i$