Superposition in a Privileged Basis

So far, we've explored superposition in a model *without a privileged basis*. We can rotate the hidden activations arbitrarily and, as long as we rotate all the weights, have the exact same model behavior. That is, for any ReLU output model with weights W, we could take an arbitrary orthogonal matrix O and consider the model $W' = OW$. Since $(OW)^T(OW) = W^TW$, the result would be an identical model!

Models without a privileged basis are elegant, and can be an interesting analogue for certain neural network representations which don't have a privileged basis – word embeddings, or the transformer residual stream. But we'd also (and perhaps primarily) like to understand neural network representations where there are neurons which do impose a privileged basis, such as transformer MLP layers or conv net neurons.

Our goal in this section is to explore the simplest toy model which gives us a privileged basis. There are at least two ways we could do this: we could add an activation function or apply L1 regularization to the hidden layer. We'll focus on adding an activation function, since the representation we are most interested in understanding is hidden layers with neurons, such as the transformer MLP layer.

This gives us the following "ReLU hidden layer" model:

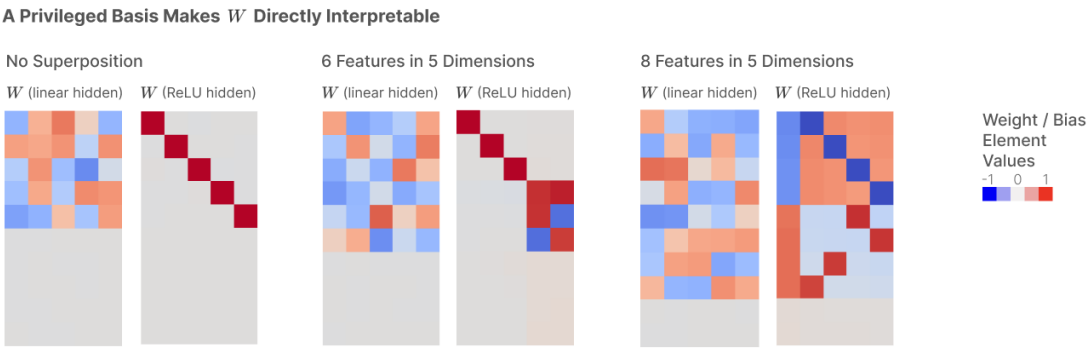
$$h = \text{ReLU}(Wx)$$

$$x' = \text{ReLU}(W^Th + b)$$

We'll train this model on the same data as before.

Adding a ReLU to the hidden layer radically changes the model from an interpretability perspective. The key thing is that while W in our previous model was challenging to interpret (recall that we visualized W^TW rather than W), W in the ReLU hidden layer model can be directly interpreted, since it connects features to basis-aligned neurons.

We'll discuss this in much more detail shortly, but here's a comparison of weights resulting from a linear hidden layer model and a ReLU hidden layer model:



Recall that we think of basis elements in the input as "features," and basis elements in the middle layer as "neurons". Thus W is a map from features to neurons.

What we see in the above plot is that *the features are aligning with neurons in a structured way*! Many of the neurons are simply dedicated to representing a feature! (This is the critical property that justifies why neuron-focused interpretability approaches – such as much of the work in the original Circuits thread – can be effective in some circumstances.)

Let's explore this in more detail.