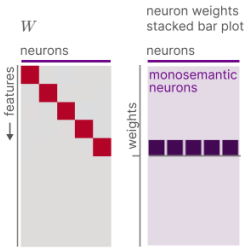
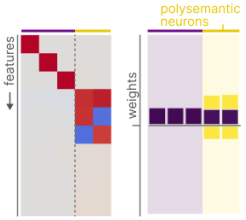


All Monosemantic Neurons (approximately 0.5 ≤ 1-S)

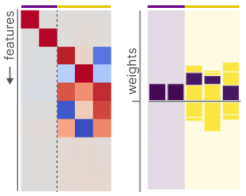
In this regime, every feature gets a dedicated neuron. That is, the model consists entirely of monosemantic neurons.



Three Features in Two Neurons (approximately 0.2 ≤ 1-S ≤ 0.5)

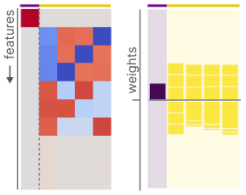
In this regime, the three most important features still get dedicated neurons, but the next three are represented in a kind of binary code by two neuron's activations.

These two neurons form a binary code for three features. Note that the (1,1) feature is orthogonal to the others, while the other two are an antipodal pair.



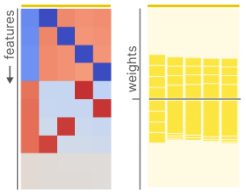
Five Features in Three Neurons (1-S = 0.15)

At this sparsity level (which seems quite narrow), we still see two monosemantic neurons, and three polysemantic neurons. The three polysemantic neurons imlement a code that doesn't have any very simple explanation.



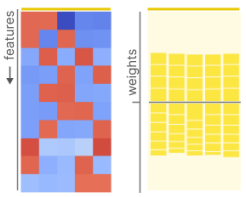
Six Features in Four Neurons (1-S = 0.12)

In this regime, we have one monosemantic neuron. The four polysemantic neurons implement an interesting code. One neuron seems to distinguish between important and unimportant features. Important features are then encoded as sets of two of the other three neurons, while unimportant features are represented by one neuorn.



Eight Features in Five Neurons (0.05 ≤ 1-S ≤ 0.08)

In this regime, all neurons are polysemantic. The code is a kind of extension of the one described previously: one distinguishes imporant and unimportant features. Then either sets of three of the other neurons, or a single other neuron, are used to distinguish the specific feature.

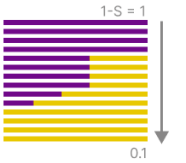


Features are Pairs of Neurons (1-S ≤ 0.04)

In this regime, each feature simply correspond to a *pair of neurons*.

Presumably if there were more features, increasing sparsity would eventually produce a dense binary code. But with only 10 features, this is the densest code that forms.

As feature sparsity increases, we see neurons shift from being monosemantic to being polysemantic.



Individual features are colored based on whether they're in superposition