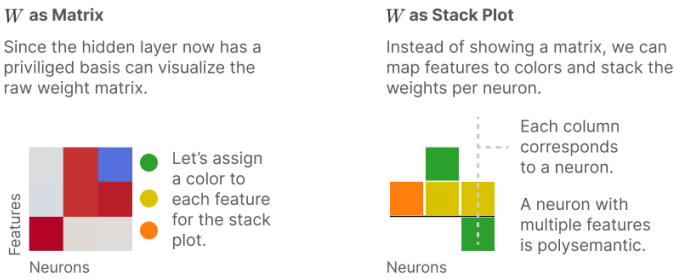
VISUALIZING SUPERPOSITION IN TERMS OF NEURONS

Having a privileged basis opens up new possibilities for visualizing our models. As we saw above, we can simply inspect W. We can also make a per-neuron stacked bar plot where, for every neuron, we visualize its weights as a stack of rectangles on top of each other:

- Each column in the stack plot visualizes one column of W.
- Each rectangle represents one weight entry, with height corresponding to the absolute value.
- The color of each rectangle corresponds to the feature it acts on (i.e. which row of W it's in).
- Negative values go below the x-axis.
- The order of the rectangles is not significant.

This stack plot visualization can be nice as models get bigger. It also makes polysemantic neurons obvious: they simply correspond to having more than one weight.



We'll now visualize a ReLU hidden layer toy model with $n = 10;\ m = 5; I^i = 0.75^i$ and varying feature sparsity levels. We chose a very small model (only 5 neurons) both for ease of visualization, and to circumvent some issues with this toy model we'll discuss below.

However, we found that these small models were harder to optimize. For each model shown, we trained 1000 models and visualized the one with the lowest loss. Although the typical solutions are often similar to the minimal loss solutions shown, selecting the minimal loss solutions reveals even more structure in how features align with neurons. It also reveals that there are ranges of sparsity values where the optimal solution for all models trained on data with that sparsity have the same weight configurations.

The solutions are visualized below, both visualizing the raw W and a neuron stacked bar plot. We color features in the stacked bar plot based on whether they're in superposition, and color neurons as being monosemantic or polysemantic depending on whether they store more than one feature. Neuron order was chosen by hand (since it's arbitrary).