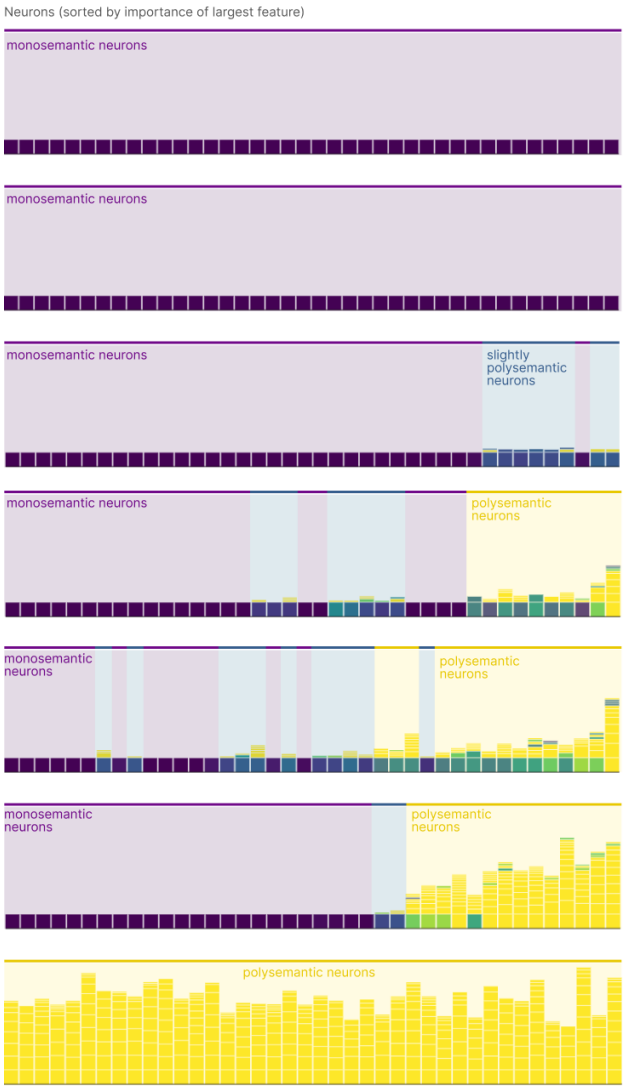
ReLU Hidden Layer Toy Model on Absolute Value Task
$n = 100;\ m = 40;\ I_i = 0.8^i$



$1 - S = 1.0$

In the dense regime, all neurons are monosemantic, dedicated to a single feature.

$1 - S = 0.3$

Neruons continue to be monosemantic to moderate sparsity levels.

$1 - S = 0.1$

Eventually, we start to see a few slightly polysemantic neurons.

$1 - S = 0.03$

As sparsity increases further, we see a small number of highly polysemantic neurons representing low importance features.

$1 - S = 0.01$

The number of polysemantic neurons grows...

$1 - S = 0.003$

And they become even more polysemantic...

$1 - S = 0.001$

Eventually, all neurons are highly polysemantic.

Much like we saw in the ReLU hidden layer models, these results demonstrate that activation functions, under the right circumstances, create a privileged basis and cause features to align with basis dimensions. In the dense regime, we end up with each neuron representing a single feature, and we can read feature values directly off of neuron activations.

However, once the features become sufficiently sparse, this model, too, uses superposition to represent more features than it has neurons. This result is notable because it demonstrates the ability of neural networks to **perform computation** even on data that is represented in superposition.[17] Remember that the model is required to use the hidden layer ReLU in order to compute an absolute value; gradient descent manages to find solutions that usefully approximate the computation even when each neuron encodes a mix of multiple features.

Focusing on the intermediate sparsity regimes, we find several additional qualitative behaviors that we find fascinatingly reminiscent of behavior that has been observed in real, full-scale neural networks: