- **Features as arbitrary functions.** One approach would be to define features as any function of the input (as in [22]). But this doesn't quite seem to fit our motivations. There's something special about these features that we're observing: they seem to in some sense be fundamental abstractions for reasoning about the data, with the same features forming reliably across models. Features also seem identifiable: cat and car are two features while cat+car and cat-car seem like mixtures of features rather than features in some important sense.
- **Features as interpretable properties.** All the features we described are strikingly understandable to humans. One could try to use this for a definition: features are the presence of human understandable "concepts" in the input. But it seems important to allow for features we might not understand. If AlphaFold discovers some important chemical structure for predicting protein folding, it very well might not be something we initially understand!
- **Neurons in Sufficiently Large Models.** A final approach is to define features as properties of the input which a sufficiently large neural network will reliably dedicate a neuron to representing.[3] For example, curve detectors appear to reliably occur across sufficiently sophisticated vision models, and so are a feature. For interpretable properties which we presently only observe in polysemantic neurons, the hope is that a sufficiently large model would dedicate a neuron to them. This definition is slightly circular, but avoids the issues with the earlier ones.

We've written this paper with the final "neurons in sufficiently large models" definition in mind. But we aren't overly attached to it, and actually think it's probably important to not prematurely attach to a definition.[4]

Features as Directions

As we've mentioned in previous sections, we generally think of *features as being represented by directions*. For example, in word embeddings, "gender" and "royalty" appear to correspond to directions, allowing arithmetic like `V("king") - V("man") + V("woman") = V("queen")` [8]. Examples of interpretable neurons are also cases of features as directions, since the amount a neuron activates corresponds to a basis direction in the representation.

Let's call a neural network representation *linear* if features correspond to directions in activation space. In a linear representation, each feature f_i has a corresponding representation direction W_i. The presence of multiple features $f_1, f_2 \ldots$ activating with values $x_{f_1}, x_{f_2} \ldots$ is represented by $x_{f_1} W_{f_1} + x_{f_2} W_{f_2}$.... To be clear, the features being represented are almost certainly nonlinear functions of the input. It's only the map from features to activation vectors which is linear. Note that whether something is a linear representation depends on what you consider to be the features.

We don't think it's a coincidence that neural networks empirically seem to have linear representations. Neural networks are built from linear functions interspersed with non-linearities. In some sense, the linear functions are the vast majority of the computation (for example, as measured in FLOPs). Linear representations are the natural format for neural networks to represent information in! Concretely, there are three major benefits: