- **Linear representations are the natural outputs of obvious algorithms a layer might implement.** If one sets up a neuron to pattern match a particular weight template, it will fire more as a stimulus matches the template better and less as it matches it less well.
- **Linear representations make features "linearly accessible."** A typical neural network layer is a linear function followed by a non-linearity. If a feature in the previous layer is represented linearly, a neuron in the next layer can "select it" and have it consistently excite or inhibit that neuron. If a feature were represented non-linearly, the model would not be able to do this in a single step.
- **Statistical Efficiency.** Representing features as different directions may allow *non-local generalization* in models with linear transformations (such as the weights of neural nets), increasing their statistical efficiency relative to models which can only locally generalize. This view is especially advocated in some of Bengio's writing (e.g. [5]). A more accessible argument can be found in this blog post.

It is possible to construct non-linear representations, and retrieve information from them, if you use multiple layers (although even these examples can be seen as linear representations with more exotic features). We provide an example in the appendix. However, our intuition is that non-linear representations are generally inefficient for neural networks.

One might think that a linear representation can only store as many features as it has dimensions, but it turns out this isn't the case! We'll see that the phenomenon we call *superposition* will allow models to store more features – potentially many more features – in linear representations.

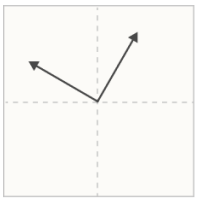
For discussion on how this view of features squares with a conception of features as being multidimensional manifolds, see the appendix “What about Multidimensional Features?”.

Privileged vs Non-privileged Bases

Even if features are encoded as directions, a natural question to ask is which directions? In some cases, it seems useful to consider the basis directions, but in others it doesn't. Why is this?

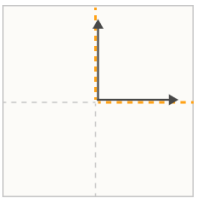
When researchers study word embeddings, it doesn't make sense to analyze basis directions. There would be no reason to expect a basis dimension to be different from any other possible direction. One way to see this is to imagine applying some random linear transformation M to the word embedding, and apply M^{-1} to the following weights. This would produce an identical model where the basis dimensions are totally different. This is what we mean by a *non-privileged basis*. Of course, it's possible to study activations without a privileged basis, you just need to identify interesting directions to study somehow, such as creating a gender direction in a word embedding by taking the difference vector between "man" and "woman".

But many neural network layers are not like this. Often, something about the architecture makes the basis directions special, such as applying an activation function. This "breaks the symmetry", making those directions special, and potentially encouraging features to align with the basis dimensions. We call this a privileged basis, and call the basis directions "neurons." Often, these neurons correspond to interpretable features.



In a **non-privileged basis**, features can be embedded in any direction. There is no reason to expect basis dimensions to be special.

Examples: word embeddings, transformer residual stream



In a **privileged basis**, there is an incentive for features to align with basis dimensions. This doesn't necessarily mean they will.

Examples: conv net neurons, transformer MLPs