The goal of this section will be to motivate these ideas and unpack them in detail.

It's worth noting that many of the ideas in this section have close connections to ideas in other lines of interpretability research (especially disentanglement), neuroscience (distributed representations, population codes, etc), compressed sensing, and many other lines of work. This section will focus on articulating our perspective on the problem. We'll discuss these other lines of work in detail in Related Work.

Empirical Phenomena

When we talk about "features" and how they're represented, this is ultimately theory building around several observed empirical phenomena. Before describing how we conceptualize those results, we'll simply describe some of the major results motivating our thinking:

- **Word Embeddings** - A famous result by *Mikolov et al.* [8] found that word embeddings appear to have directions which correspond to semantic properties, allowing for embedding arithmetic vectors such as `V("king") - V("man") + V("woman") = V("queen")` (*but see* [9]).
- **Latent Spaces** - Similar "vector arithmetic" and interpretable direction results have also been found for generative adversarial networks (e.g. [10]).
- **Interpretable Neurons** - There is a significant body of results finding neurons which appear to be interpretable (*in RNNs* [11, 12]; *in CNNs* [13, 14]; *in GANs* [15]), activating in response to some understandable property. This work has faced some skepticism [16, 17]. In response, several papers have aimed to give extremely detailed accounts of a few specific neurons, in the hope of dispositively establishing examples of neurons which truly detect some understandable property (notably Cammarata *et al.* [6], but also [18, 19]).
- **Universality** - Many analogous neurons responding to the same properties can be found across networks [20, 1, 18].
- **Polysemantic Neurons** - At the same time, there are also many neurons which appear to not respond to an interpretable property of the input, and in particular, many *polysemantic neurons* which appear to respond to unrelated mixtures of inputs [21].

As a result, we tend to think of neural network representations as being composed of *features* which are *represented as directions*. We'll unpack this idea in the following sections.

What are Features?

Our use of the term "feature" is motivated by the interpretable properties of the input we observe neurons (or word embedding directions) responding to. There's a rich variety of such observed properties![2] We'd like to use the term "feature" to encompass all these properties.

But even with that motivation, it turns out to be quite challenging to create a satisfactory definition of a feature. Rather than offer a single definition we're confident about, we consider three potential working definitions:

- **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:

- **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.

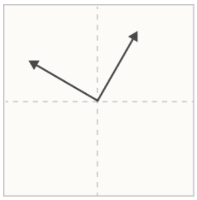
For discussion on how this view of features squares with a conception of features as being multidimensional manifolds, see the appendix “What about Multidimensional Features?”.

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?

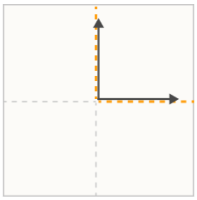
When researchers study word embeddings, it doesn't make sense to analyze basis directions. There would be no reason to expect a basis dimension to be different from any other possible direction. One way to see this is to imagine applying some random linear transformation M to the word embedding, and 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