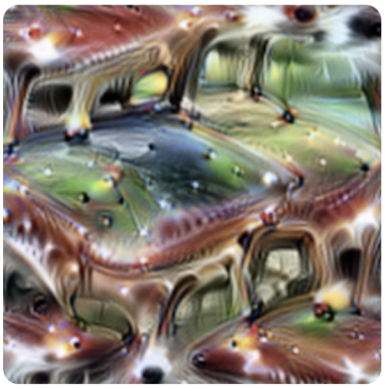
Superposition refers to the concept that a neural network can sometimes use the same set of parameters to encode the recognition of multiple different disparate features. Since these features might not be related to each other this concept of superposition makes the task of interpretability of neural networks considerably harder. This concept is closely related to the concept of polysemantic neurons.


Related

- What are polysemantic neurons?
- What is feature visualization?
- How much can we learn about AI with interpretability tools?
- What is interpretability and what approaches are there?

Sources

- https://distill.pub/2020/circuits/zoom-in/
- https://distill.pub/2017/feature-visualization/
- https://www.transformer-circuits.pub/2022/mech-interp-essay/index.html

---

Scratchpad

It would be very convenient if the individual neurons of artificial neural networks corresponded to cleanly interpretable features of the input. For example, in an “ideal” ImageNet classifier, each neuron would fire only in the presence of a specific visual feature, such as the color red, a left-facing curve, or a dog snout. Empirically, in models we have studied, some of the neurons do cleanly map to features. But it isn't always the case that features correspond so cleanly to neurons, especially in large language models where it actually seems rare for neurons to correspond to clean features. This brings up many questions. Why is it that neurons sometimes align with features and sometimes don't? Why do some models and tasks have many of these clean neurons, while they're vanishingly rare in others?

0% sparsity : Each feature is given its own independent orthogonal direction

Not only can models store additional features in superposition by tolerating some interference, but we'll show that, at least in certain limited cases, models can perform computation while in superposition. (In particular, we'll show that models can put simple circuits computing the absolute value function in superposition.) This leads us to hypothesize that the neural networks we observe in practice are in some sense noisily simulating larger, highly sparse networks. In other words, it's possible that models we train can be thought of as doing "the same thing as" an imagined much-larger model, representing the exact same features but with no interference.

That is, we show a case where interpreting neural networks as having sparse structure in superposition isn't just a useful post-hoc interpretation, but actually the "ground truth" of a model.
This explains why neurons are sometimes "monosemantic" responding to a single feature, and sometimes "polysemantic"

Linear Algebra version
One might think of this as two separate properties, which we'll explore in more detail shortly:

- **Decomposability:** Network representations can be described in terms of independently understandable features.
- **Linearity:** Features are represented by direction.

If we hope to reverse engineer neural networks, we *need* a property like decomposability. Decomposability is what allows us to reason about the model without fitting the whole thing in our heads! But it's not enough for things to be decomposable: we need to be able to access the decomposition somehow. In order to do this, we need to *identify* the individual features within a representation. In a linear representation, this corresponds to determining which directions in activation space correspond to which independent features of the input.

Why dont we always geed monosemanticity?

- **Privileged Basis:** Only some representations have a *privileged basis* which encourages features to align with basis directions (i.e. to correspond to neurons).
- **Superposition:** Linear representations can represent more features than dimensions, using a strategy we call *superposition*. This can be seen as neural networks *simulating larger networks*. This pushes features *away* from corresponding to neurons.

major results motivating our thinking:

- **Word Embeddings** - A famous result by *Mikolov et al.* 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*).
- **Latent Spaces** - Similar "vector arithmetic" and interpretable direction results have also been found for generative adversarial networks (e.g.).
- **Interpretable Neurons** - There is a significant body of results finding neurons which appear to be interpretable (*in RNNs* ; *in CNNs* ; *in GANs*), activating in response to some understandable property. This work has faced some skepticism . 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.* , but also).

- **Universality** - Many analogous neurons responding to the same properties can be found across networks .
- **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 .

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. We'd like to use the term "feature" to encompass all these properties. Rather than offer a single definition we're confident about, we consider three potential working definitions:

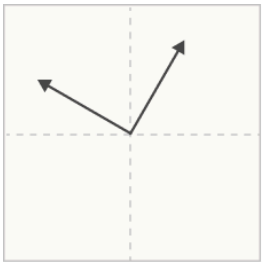
- **Features as arbitrary functions.** One approach would be to define features as any function of the input (as in). But this doesn't quite seem to fit our motivations. There's something special about these features that we're observing: they seem to in some sense be fundamental abstractions for reasoning 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.This definition is trickier than it seems. Specifically, something is a feature if there a large enough model size such that it gets a dedicated neuron. This create a kind "epsilon-delta" like definition. Our present understanding – as we'll see in later sections – is that arbitrarily large models can still have a large fraction of their features be in superposition. However, for any given feature, assuming the feature importance curve isn't flat, it should eventually be given a dedicated neuron. This definition can be helpful in saying that something a feature – curve detectors are a feature because you find them in across a range of models larger than some minimal size – but unhelpful for the much more common case of features we only hypothesize about or observe in superposition. 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.

Let's call a neural network representation linear if features correspond to directions in activation space. We don't think it's a coincidence that neural networks empirically seem to have linear representations.

Privileged vs Non-privileged Bases

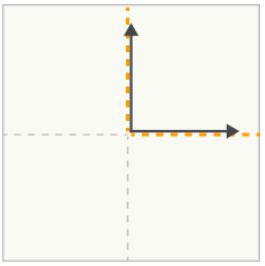
Even if features are encoded as directions, a natural question to ask is which directions? In some cases, it seems useful to consider the basis directions, but in others it doesn't. Why is this?

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

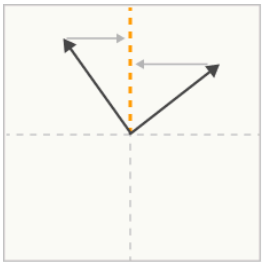


In a **privileged basis**, there is an incentive for features to align with basis dimensions. This doesn't necessarily mean they will.

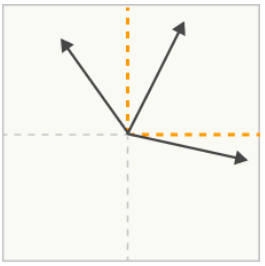
Examples: conv net neurons, transformer MLPs

The Superposition Hypothesis

Even when there is a privileged basis, it's often the case that neurons are "polysemantic", responding to several unrelated features. One explanation for this is the superposition hypothesis. Roughly, the idea of superposition is that neural networks "want to represent more features than they have neurons", so they exploit a property of high-dimensional spaces to simulate a model with many more neurons.

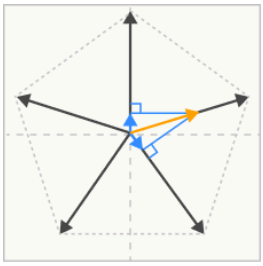


Polysemanticity is what we'd expect to observe if features were not aligned with a neuron, despite incentives to align with the privileged basis.

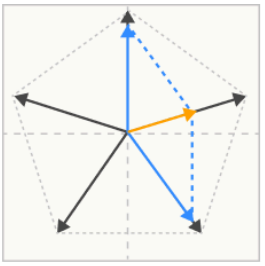


In the **superposition hypothesis**, features can't align with the basis because the model embeds more features than there are neurons. Polysemanticity is inevitable if this happens.

Concretely, in the superposition hypothesis, features are represented as almost-orthogonal directions in the vector space of neuron outputs. Since the features are only almost-orthogonal, one feature activating looks like other features slightly activating. Tolerating this "noise" or "interference" comes at a cost. But for neural networks with highly sparse features, this cost may be outweighed by the benefit of being able to represent more features!



Even if only **one sparse feature** is active, using linear dot product projection on the superposition leads to **interference** which the model must tolerate or filter.



If the features aren't as sparse as a superposition is expecting, **multiple present features** can additively interfere such that there are multiple possible nonlinear reconstructions of an **activation vector**.

Demonstrating Superposition

If one takes the superposition hypothesis seriously, a natural first question is whether neural networks can actually noisily represent more features than they have neurons. If they can't, the superposition hypothesis may be comfortably dismissed.

———————————————————— EOF ————————————————————