Toy Models of Superposition

Tristan Thomas

November 2025

Paper Introduction/Key Questions

In the paper[1] studied as part of this project, the authors build toy models of superposition.
Key questions:

- What is superposition?
- What is meant by toy model in this case?
- Is there extension from the "toy model" to describe effects in larger models?

[1] Nelson Elhage et al. *Toy Models of Superposition*. 2022. arXiv: 2209.10652 [cs.LG]. URL: https://arxiv.org/abs/2209.10652.

What is superposition?

- Superposition is a phenomenon in which neural network models are able to represent more features than they have dimension.
- Take digit classification as an example.
  - Ideally, neurons are monosemantic
  - E.g. One neuron detects if there is a closed loop in the number, one detects straight edges, etc.
- Is this really what happens? Typically, no.

The fuel for superposition-feature sparsity

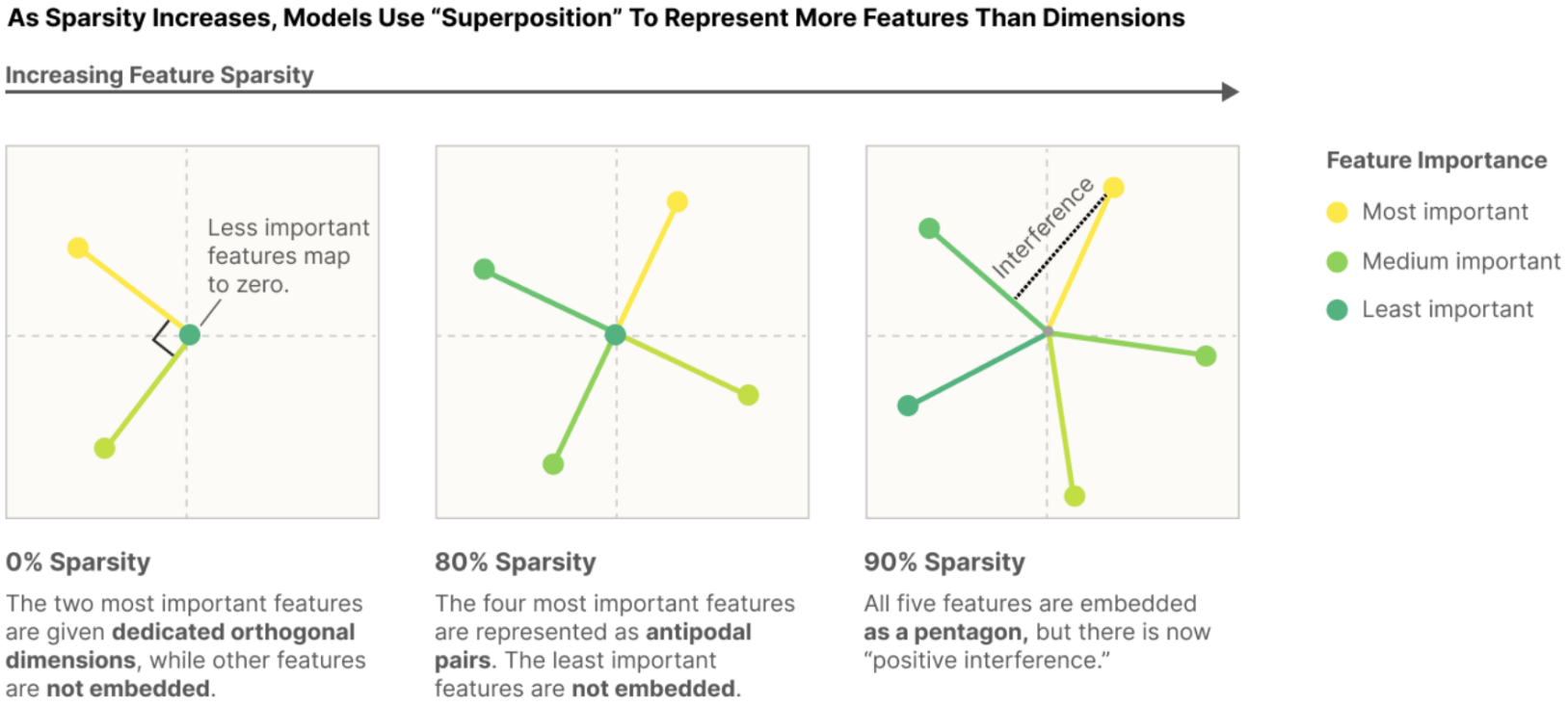


Figure: Example with training embedding 5 features in 2D space.

Related Work

Mikolov et. al[2] Investigated the semantic properties of word representations. i.e. King + Male - Queen = Female.
There is a lot of work on polysemantic neurons. These neurons may respond to very unrelated words or concepts in language models (Clemson football and South African safari tours).[3]

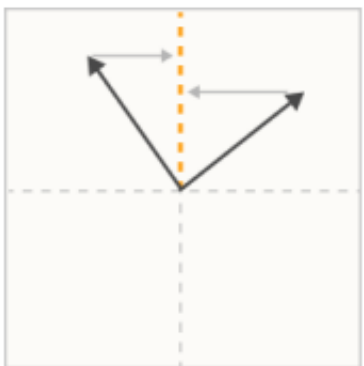
---

[2] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL]. URL: https://arxiv.org/abs/1301.3781.

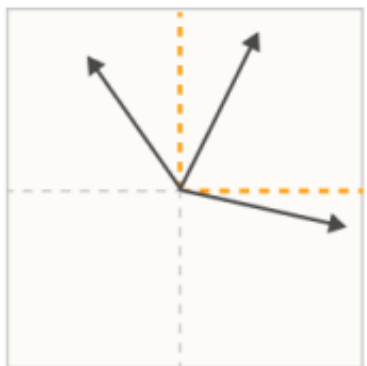
[3] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature Visualization". In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

The Superposition Hypothesis

- Non-privileged basis: Features are embedded in any direction
- Privileged basis: Features are incentivized to align with basis dimensions
- Superposition says the network want to represent more features than there are neurons. This is mathematically justified by the Johnson-Lindenstrauss Theorem.

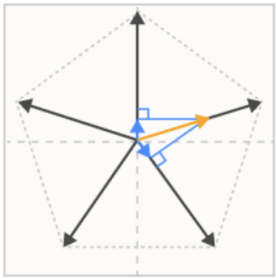


Polysemanticity is what we'd expect to observe if features were not aligned with a neuron, despite incentives to align with the privileged basis.

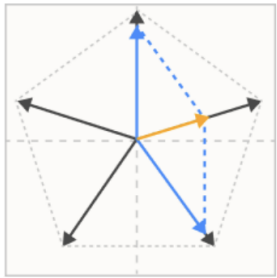


In the **superposition hypothesis**, features can't align with the basis because the model embeds more features than there are neurons. Polysemanticity is inevitable if this happens.

Issues



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**.

- Setup: To understand the left figure, we define a concrete example with 3 features (Universities) packed into 2 neurons.
- Feature Directions: We assign a vector in 2D space for each university:
  - Clemson: $v_1 = (\frac{\sqrt{3}}{2}, \frac{1}{2})$
  - Duke: $v_2 = (-\frac{\sqrt{3}}{2}, \frac{1}{2})$
  - NC State: $v_3 = (0, -1)$
- Weight Matrix (W): Stacking these as columns gives us our encoding matrix:

$$W = \begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & -1 \end{pmatrix}$$

Quantifying Interference

- In low dimensional space, we can't have all three vectors be even close to mutually orthogonal.
- We project the Duke vector onto the Clemson vector:

$$\text{proj}_C(D) = \underbrace{\left(\begin{pmatrix} -\frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix}\right)}_{\text{Dot Product}} \underbrace{\begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix}}_{\text{Direction}}$$

$$= \left(-\frac{3}{4} + \frac{1}{4}\right) \text{Clemson} = -0.5\text{Clemson}$$

- Result: It is obvious from this derivation that activating Duke will have an effect on the apparent activation of Clemson.

Ambiguity

- Activate Clemson and Duke both at **0.9**.
- $\begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & -1 \end{pmatrix} \begin{pmatrix} 0.9 \\ 0.9 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.9 \end{pmatrix}$
- From previous results, if you de-embed this vector you will get half the Clemson feature value (because the Duke projects on and cancels half).
- NC State Result: How does this look to the NC State feature $\begin{pmatrix} 0 \\ -1 \end{pmatrix}$?

$$\text{Activation} = \begin{pmatrix} 0 \\ 0.9 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -0.9$$

- Conclusion: The presence of Clemson and Duke creates an ambiguous signal of -0.9 on the NC State channel.

The Full Dembedding Math

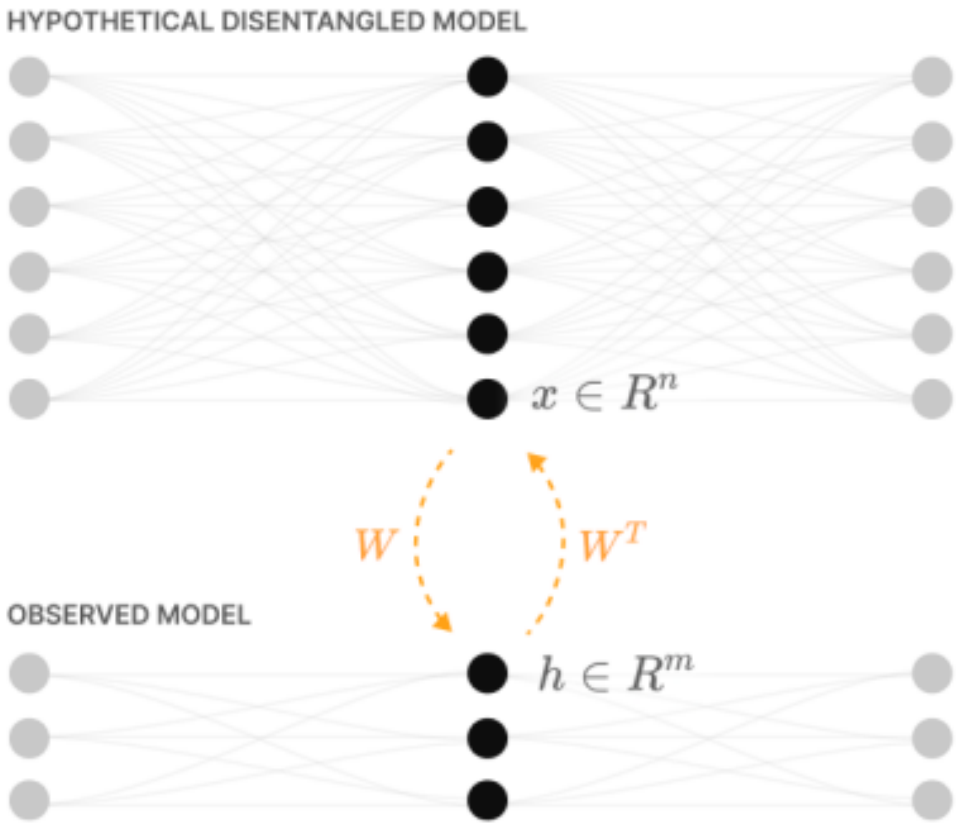
- Recovery ($x' = W^T h$): We get the activation of each feature by multiplying our activation vector by the transpose of the feature matrix.

$$\begin{bmatrix} x'_{\text{Clemson}} \\ x'_{\text{Duke}} \\ x'_{\text{NC State}} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \\ 0 & -1 \end{bmatrix}}_{W^T \text{ (Feature Rows)}} \underbrace{\begin{bmatrix} 0 \\ 0.9 \end{bmatrix}}_{\text{Activation } h} = \begin{bmatrix} 0.45 \\ 0.45 \\ -0.9 \end{bmatrix}$$

- Results:
  - Clemson & Duke: Recovered at 0.45
  - NC State: Introduced a nonzero value (-0.9).
- Interpretation: We now have no idea what features were really present at the input. A problem due to not having sparsity.

Testing the Superposition Hypothesis

- Can we project a vector from high dimensional space to low dimensional space and then recover it?
- Data is based on each x_i having a sparsity S_i and important I_i. $x_i = 0$ with probability S_i, otherwise uniformly sampled from $[0, 1]$.



Our first experiments will test the extent to which the idealized activations of an imagined larger model can be **stored** and **recovered** from a lower-dimensional space.

Two Models to Consider

- Linear model
  - $h = Wx$
  - $x' = W^T h + b$
  - $x' = W^T Wx + b$
- ReLU Output Model
  - $h = Wx$
  - $x' = \text{ReLU}(W^T h + b)$
  - $x' = \text{ReLU}(W^T Wx + b)$
- Note that we add new things to the models from the previous illustrative example. We have a bias that can help put features at their expected value (if we know they should be sparse). We also add activation.

Results Visualization Example

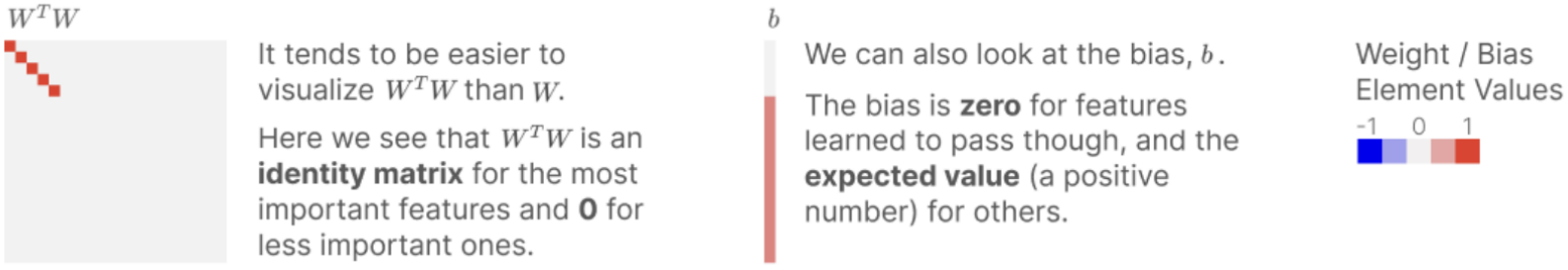


Figure: If a feature is represented, it should have a norm close to 1. The biases will be zero if we have a representation of the feature, otherwise it can be used to help us get a closer representation for features not well represented by embedding.

Results

- Will show them directly in paper for better viewing experience
- Conclusion:
  - The linear model only give priority to the 5 most important features
  - As we look at ReLU models, they are still only embedding important features as sparsity being so high will damage feature recovery
  - As sparsity is increased, more features are stored as interference and ambiguous effects are mitigated

Citations

Figures were borrowed from [1]