Superposition in Neural Network Representations

- bolu ben-adeola

Mechanistic Interpretability

1. Neural networks solve an increasing number of important tasks really well.

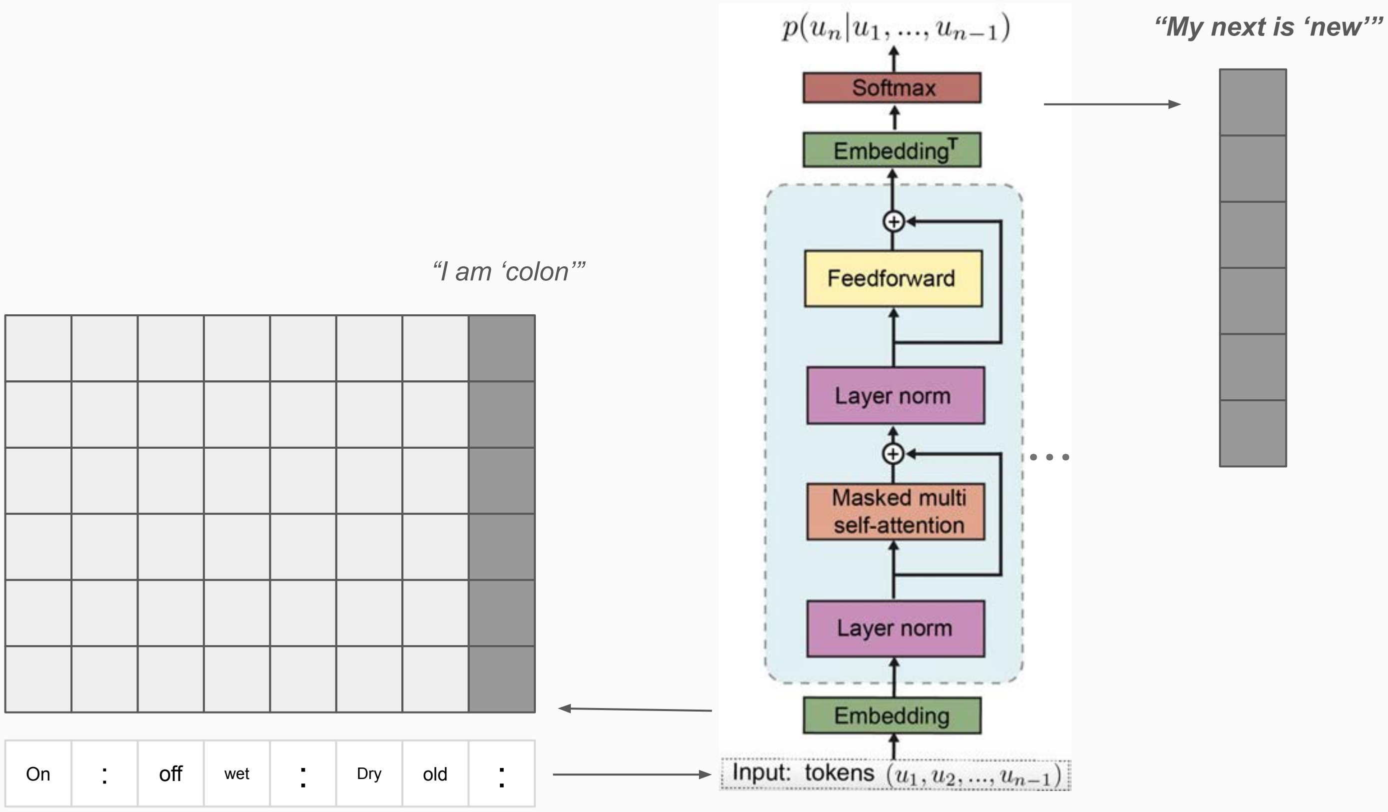
2. It would be at least interesting, and probably important to understand how.

3. Mechanistic Interpretability (Mech Interp) tackles this problem by seeking granular mechanistic explanations.

Neural Network Representations

Understanding what a model sees and how it does. I.e. what information have models found important to look for in their inputs and how is this information represented and propagated internally?

Neural Network Representations



Qualities of Representations

1. Decomposability

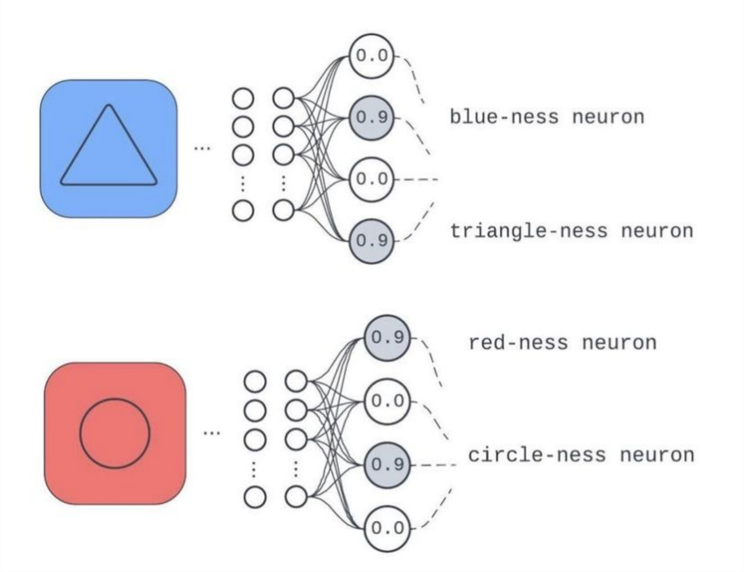
2. Linearity

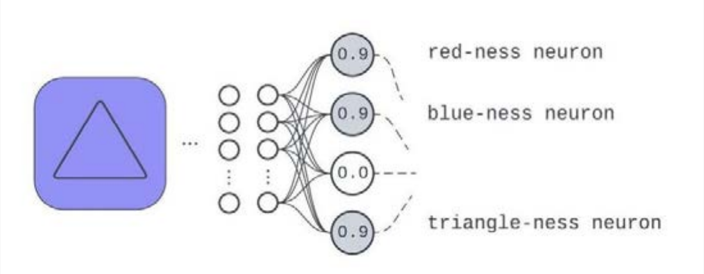
3. Composed of features

Language Model Representations are:

Linearly Decomposable Into ***Features***

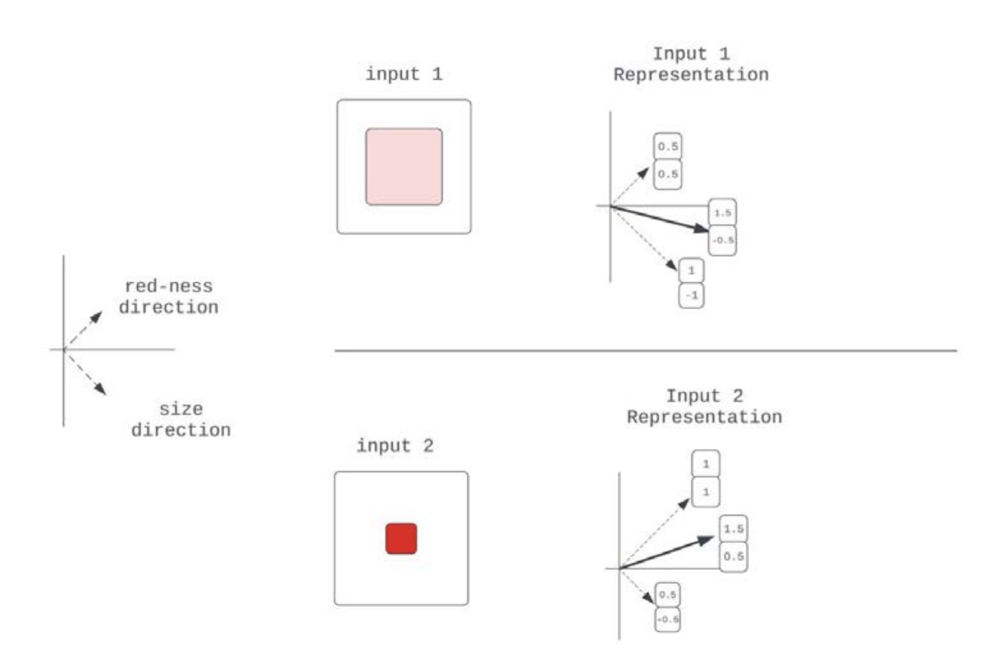
Decomposability





Linearity

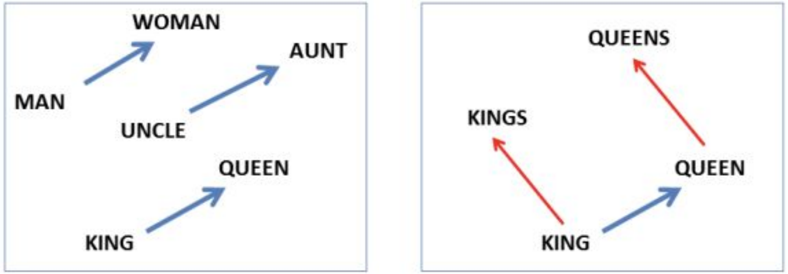
These discrete quality vectors are composed by a Sum to give the observed representation.



Linearity

Linguistic Regularities in Continuous Space Word Representations

Tomas Mikolov*, Wen-tau Yih, Geoffrey Zweig
Microsoft Research
Redmond, WA 98052



$$cars_r - car_r + apple_r \approx apples_r$$

https://aclanthology.org/N13-1090.pdf

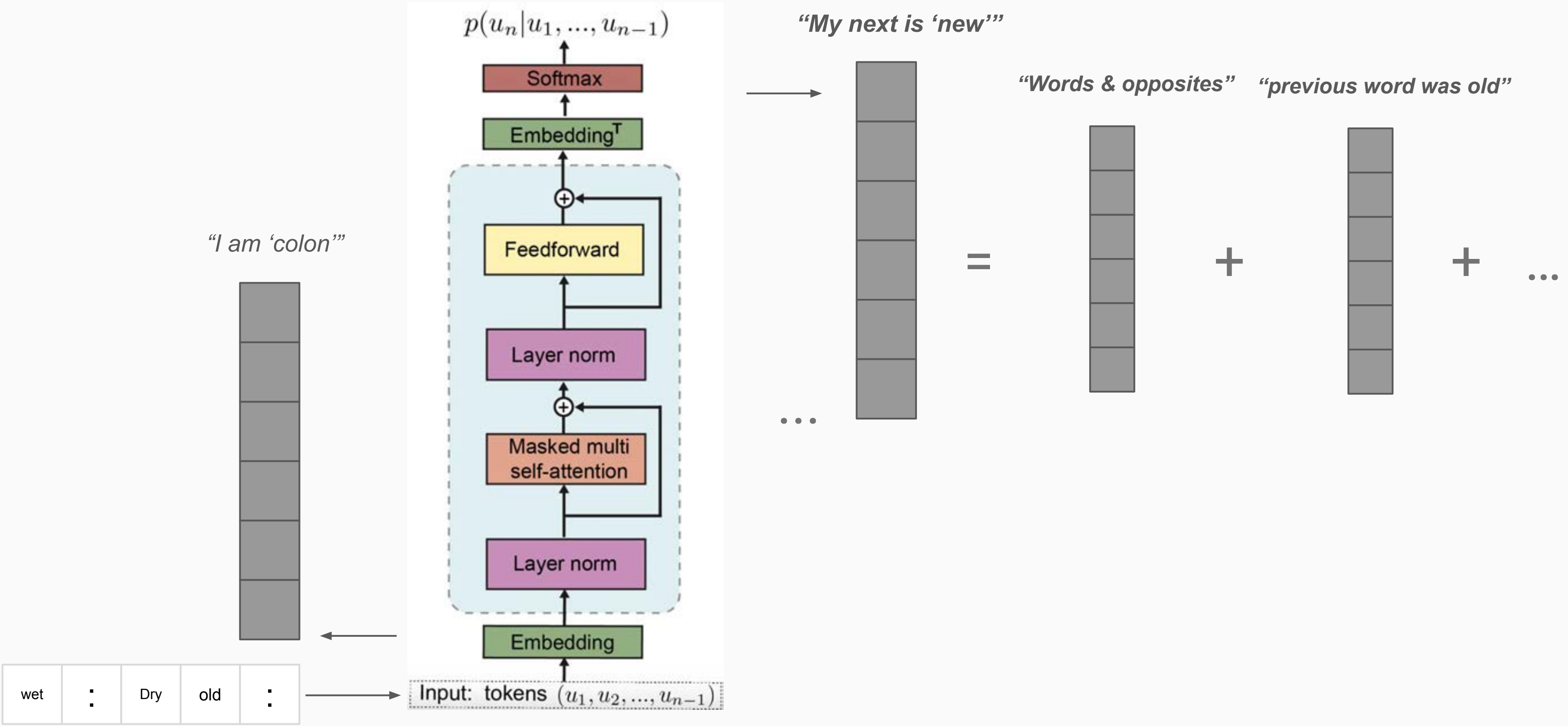
Linearity

What could non-Linear composition look like?

```python
def compress_values(x1, x2, precision=1):
    z = 10 ** precision
    compressed_val = (floor(z * x1) + x2) / z
    return round(compressed_val, precision * 2)
```

$$\underset{\text{purpleness-ness}}{\begin{bmatrix} 0.32 \\ 0.74 \end{bmatrix}} = \underset{\text{red-ness}}{\begin{bmatrix} 3 \\ 7 \end{bmatrix}} \oplus \underset{\text{blue-ness}}{\begin{bmatrix} 2 \\ 4 \end{bmatrix}}$$

Linear Representations



Linear Composition as a Compression Scheme

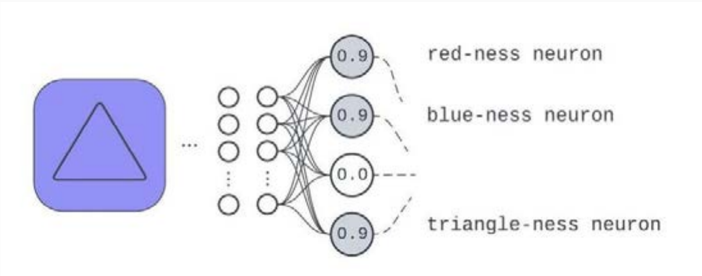
Linearity is great because it helps us narrow down to one compression algorithm in a very large function space.

This understanding aids diagnostics (and maybe even steering) in AI safety contexts.

Effectively, mind control

Demands of Linearity

But Linearity also has pretty stringent demands: As a compression scheme, it requires as many vector dimensions as the number of discrete qualities you want to encode.



$$
\underset{\text{Input Representation}}{\begin{bmatrix} 0.9 \\ 0.9 \\ 0 \\ 0.9 \end{bmatrix}} = \underset{\text{red-ness}}{\begin{bmatrix} 0.9 \\ 0 \\ 0 \\ 0 \end{bmatrix}} + \underset{\text{blue-ness}}{\begin{bmatrix} 0 \\ 0.9 \\ 0 \\ 0 \end{bmatrix}} + \underset{\text{square-ness}}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}} + \underset{\text{triangle-ness}}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.9 \end{bmatrix}}
$$

Language Model Representations are:

Linearly Decomposable Into ***Features***

The Linear Representation Puzzle

Linearity

We have some evidence that LLMs represent inputs with linear combinations (of features.)

Neuron - Feature Requirements

Lossless Linear combinations requires as many dimensions (neurons) as features.

Experience with LLMs

Common experience suggests LLMs have more features than they have neurons.

(GPT2-Small has on the order of 100k+ Neurons, and probably encodes more features than that.)

How?

Towards Monosemanticity: Decomposing Language Models With Dictionary Learning

Using a sparse autoencoder, we extract a large number of interpretable features from a one-layer transformer.

Browse A/1 Features →

Browse All Features →

AUTHORS

Trenton Bricken*, Adly Templeton*, Joshua Batson*, Brian Chen*, Adam Jermyn*, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, Chris Olah

AFFILIATIONS

Anthropic

PUBLISHED

Oct 4, 2023

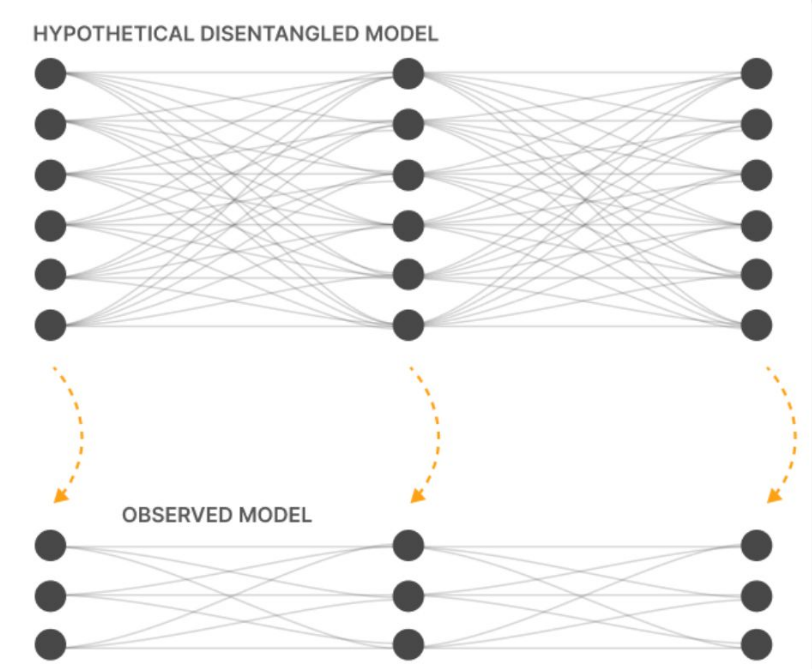
* Core Contributor; Correspondence to colah@anthropic.com; Author contributions statement below.

https://transformer-circuits.pub/2023/monosemantic-features

The Superposition Hypothesis

The superposition hypothesis suggests that Neural Networks represent more features than they have neurons to by exploiting feature sparsity and relative feature importance.

Effectively it says networks trade off lossless compression for increased feature representation to achieve good performance on training tasks.

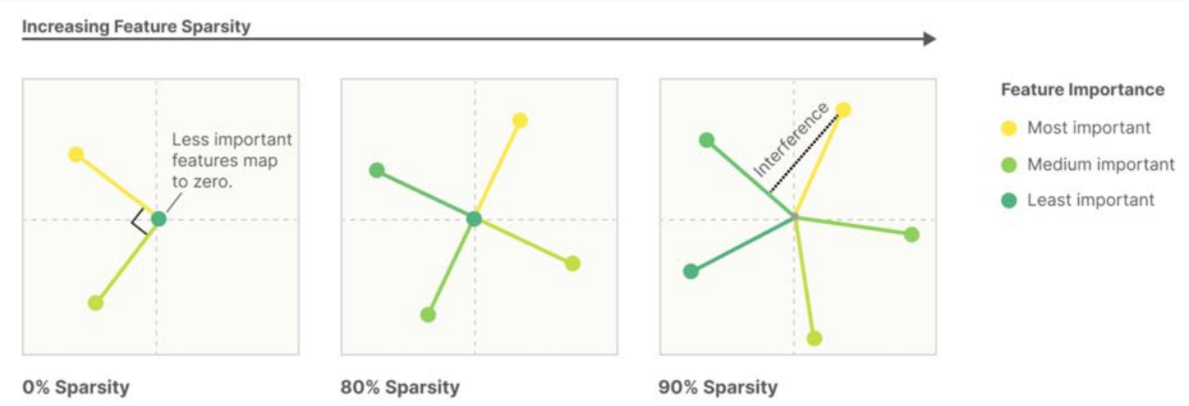


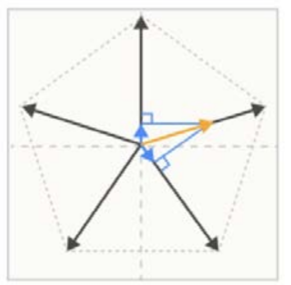
transformer-circuits.pub/2022/toy_model/

Sparsity

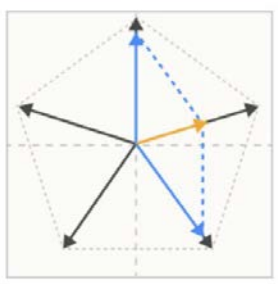
A key reason why this works is sparsity. Although language and other representation tasks have a very large number of helpful features that would be worth representing, they don't all show up in any given input at the same time.

This means as sparsity increases, the interference costs of having more features than neurons drops off.





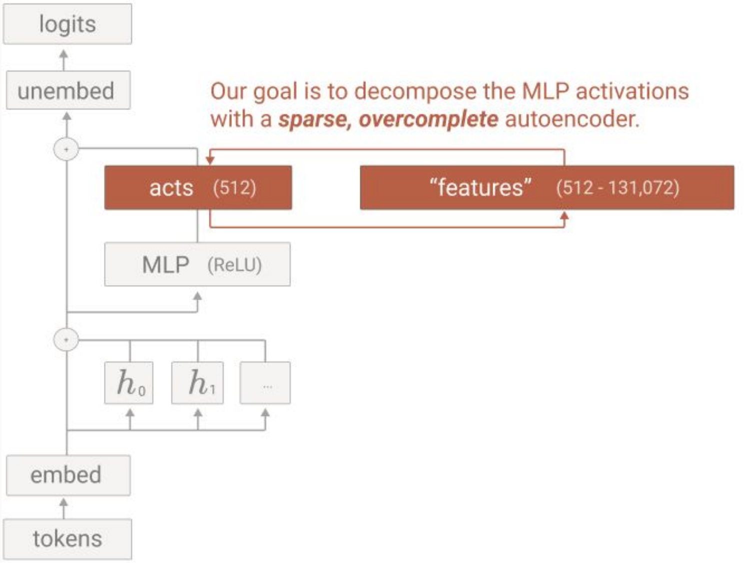
One sparsely activated Vector with little interference



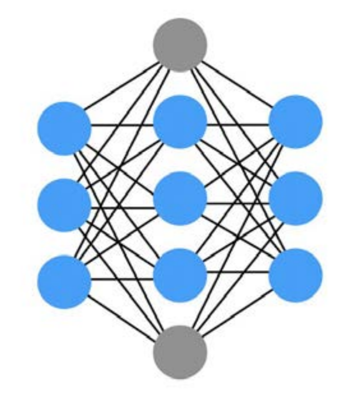
Two activated Vectors being misinterpreted.

transformer-circuits.pub/2022/toy_model/

Recovering Features in Superposition



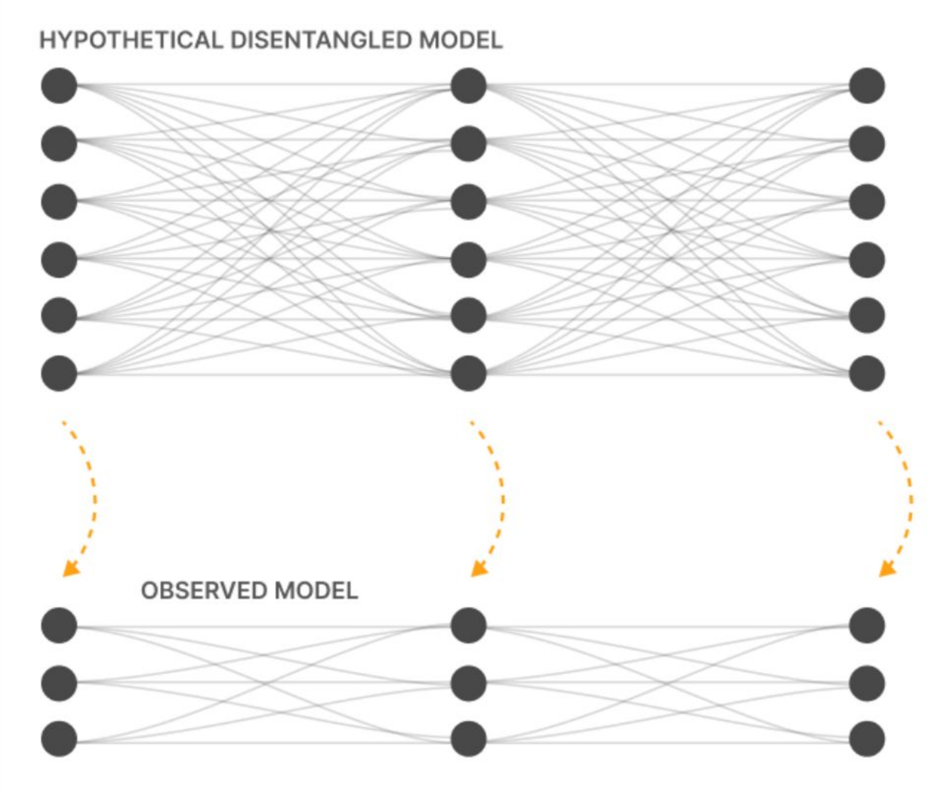
One-layer transformer



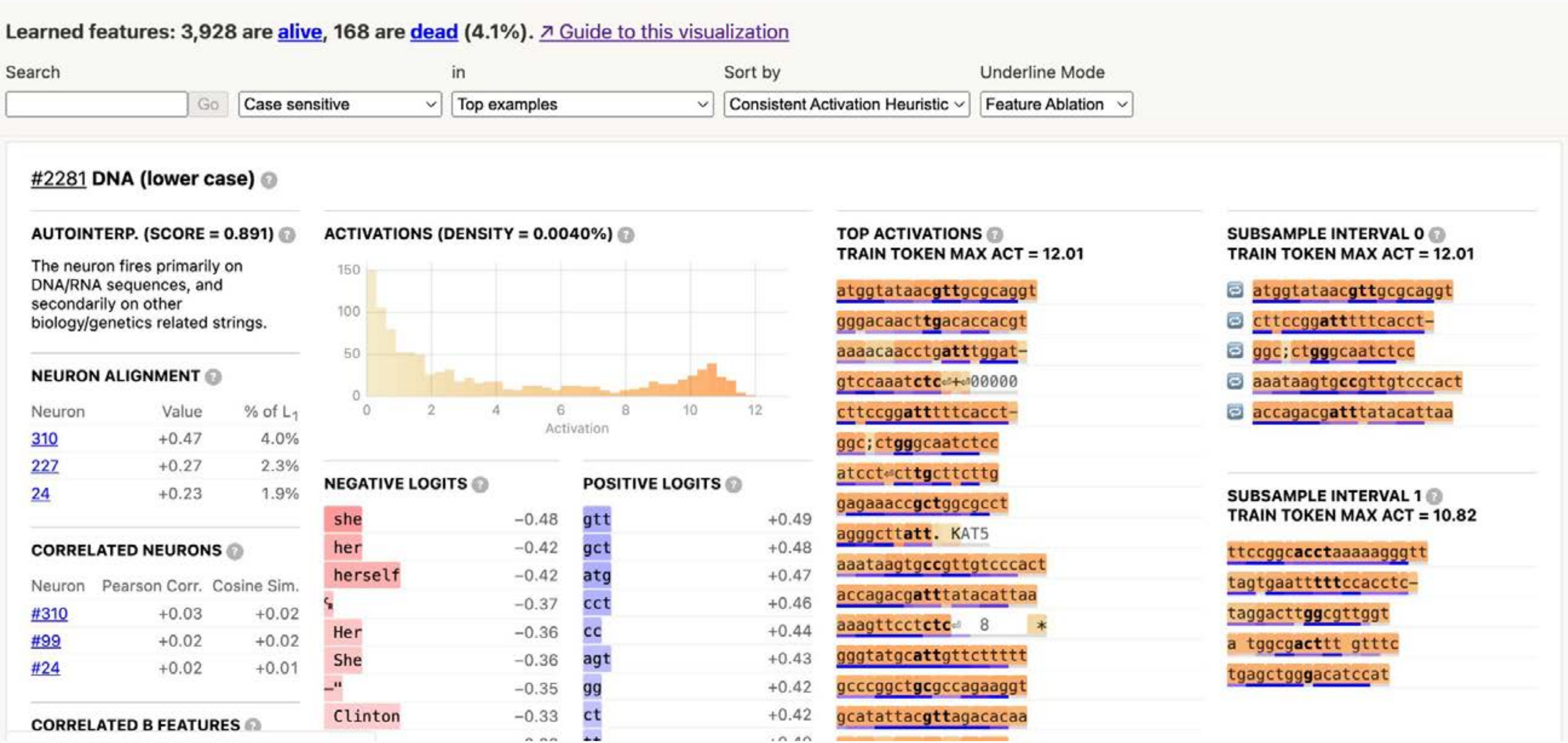
Sparse Overcomplete Autoencoder

https://transformer-circuits.pub/2023/monosemantic-features

Recovering the Disentangled Model



Feature Exploration



https://transformer-circuits.pub/2023/monosemantic-features

Thanks