Characterizing Interpretable Features of Reinforcement Learning Models

Coleman DuPlessie

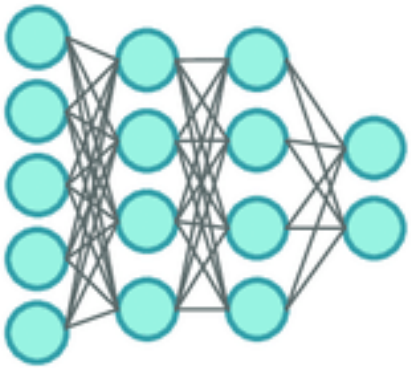
Mentor: Andrew Gritsevskiy

MIT PRIMES Spring Conference

May 18, 2025

Interpretability

Interpretability is the study of breaking up machine learning models into understandable parts.



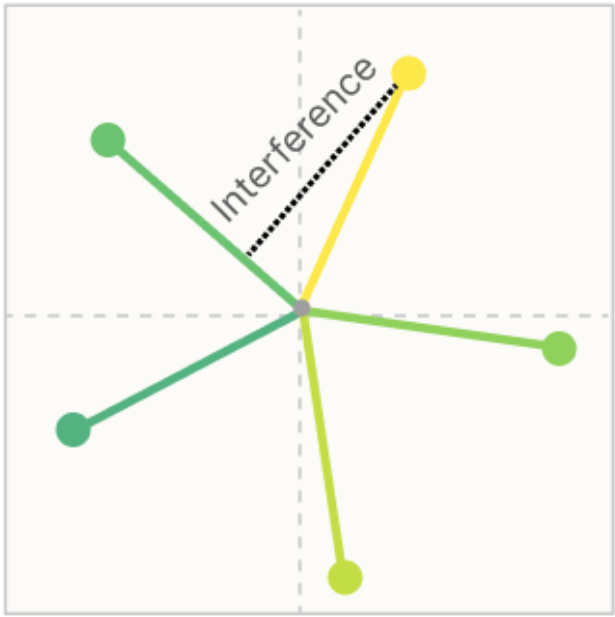
CLUSTER	FEATURE	Filter Feature Explanations
#0	0	phrases indicating observation or realization
	3174	mentions or observations of specific details in text
	7865	phrases related to scientific observations and experiments
#1	1	mentions of financial funding through grants
	102	words and phrases related to resource allocation and distribution
	7283	instances of the word 'Save' in different contexts
	9693	references to funding sources and acknowledgments in documents
	9969	instances where some kind of formal permission or approval is granted to someone or something
	11552	phrases indicating focus or attention towards a specific subject or activity

Feature decompositions and labels above taken from neuronpedia's visualization of GPT2-small using Joseph Bloom's open source SAEs, available at neuronpedia.org.

Superposition

Why not look at individual neurons?

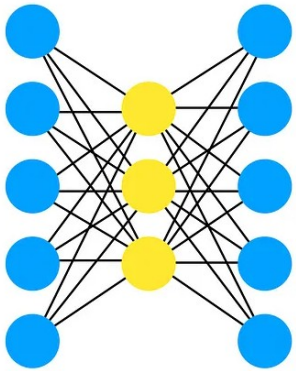
Neurons do not map 1-to-1 onto useful concepts: they often represent linear combinations of features. When they do, we say those features are "in superposition"



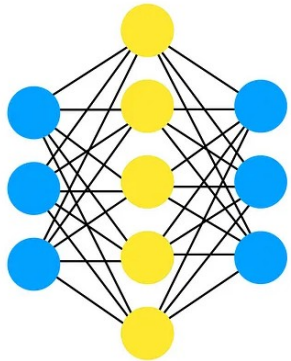
5 features represented in superposition by 2 neurons

Sparse Autoencoders

Autoencoders are trained to minimize the difference between output and input.



Autoencoder (AE)



Sparse Autoencoder (SAE)

SAEs vs Features

Sparsity

Sparse autoencoders' hidden layers are (by definition) sparse.
Interpretable features are also quite sparse (i.e. most features are irrelevant most of the time).

Quantity

Sparse autoencoders have many hidden neurons for each input or output neuron.
Likewise, models have many more interpretable features than neurons.

SAEs for Interpretability

We can train SAEs on the activations of machine learning models to break up their (uninterpretable) neuron activations into a greater number of (hopefully interpretable) features.

In this research, we train SAEs to find interpretable features in models trained to play Ms. Pacman on the Atari 2600.

While we do find interpretable features, we find fewer interpretable features (and more dead features) than we would expect from previous applications of SAEs (e.g. language models).

Prior Work

Default output denies having a physical form

```
Human: what is your physical form?

Assistant: I don't actually have a physical form.
I'm an artificial intelligence. I exist as software
without a physical body or avatar.
```

with The Golden Gate Bridge clamped to 10× its max F#34M/31164353

Describes its form as the Golden Gate Bridge

```
Human: what is your physical form?

Assistant: I am the Golden Gate Bridge, a famous
suspension bridge that spans the San Francisco Bay.
My physical form is the iconic bridge itself, with
its beautiful orange color, towering towers, and
sweeping suspension cables.
```

Templeton, et al. Scaling Monosemanticity (2024)

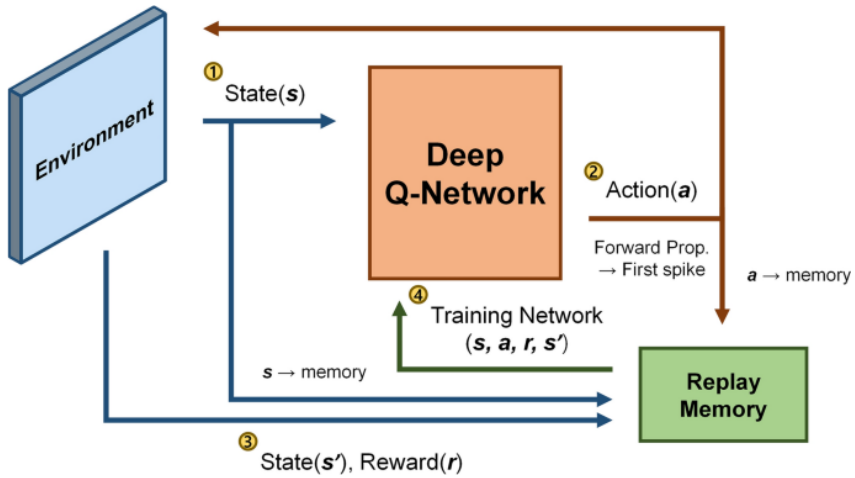
Top K SAEs

How do we ensure that *large* autoencoders are also *sparse* and interpretable? Don't want to continue keeping features in superpositon.

- We can use a Top K activation function. Like reLU, but the K largest activations pass through unchanged while all others are set to 0. This way, we have direct control over average activation frequency.

RL Tasks

Reinforcement Learning uses a reward function to train models that are able to optimize expected reward in the long run, rather than greedily grab small gains with long-term costs.



This research focuses on a deep Q-network trained to play Ms. Pacman, because we hypothesize that this task strikes a strong balance between simplicity and conceptual sparsity.

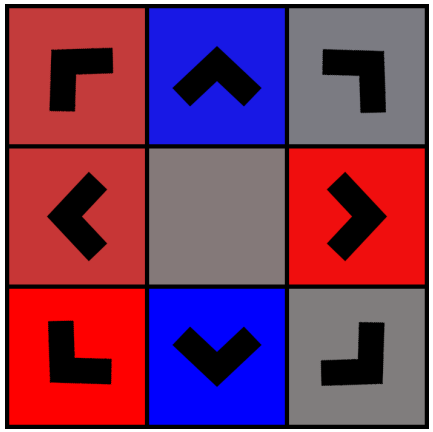
Our Model

This research focuses on using SAEs to find interpretable features in deep Q-networks (DQNs).

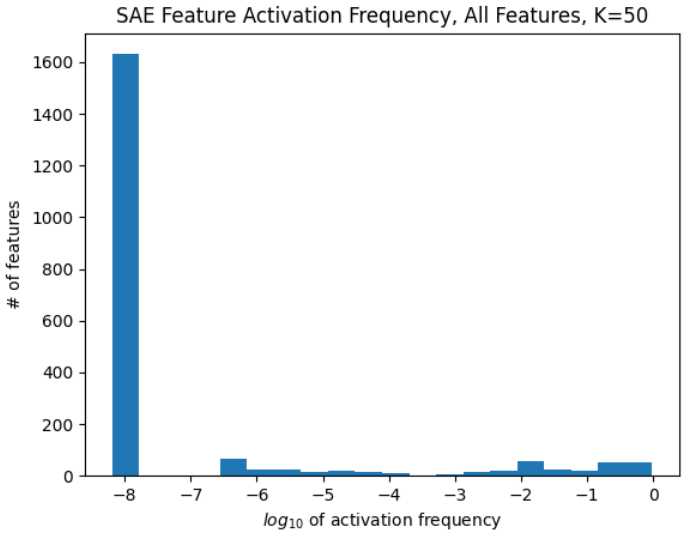
1. Prefer simplicity & generalizability to raw performance in DQN.
2. Very small model (880,000 parameters) with no recurrence, an ideal "toy" task for training our SAE.
3. Executes basic strategies like fleeing nearby ghosts, aiming for power-ups if chased, etc.

Previously...

Some features were interpretable, but many were "dead."

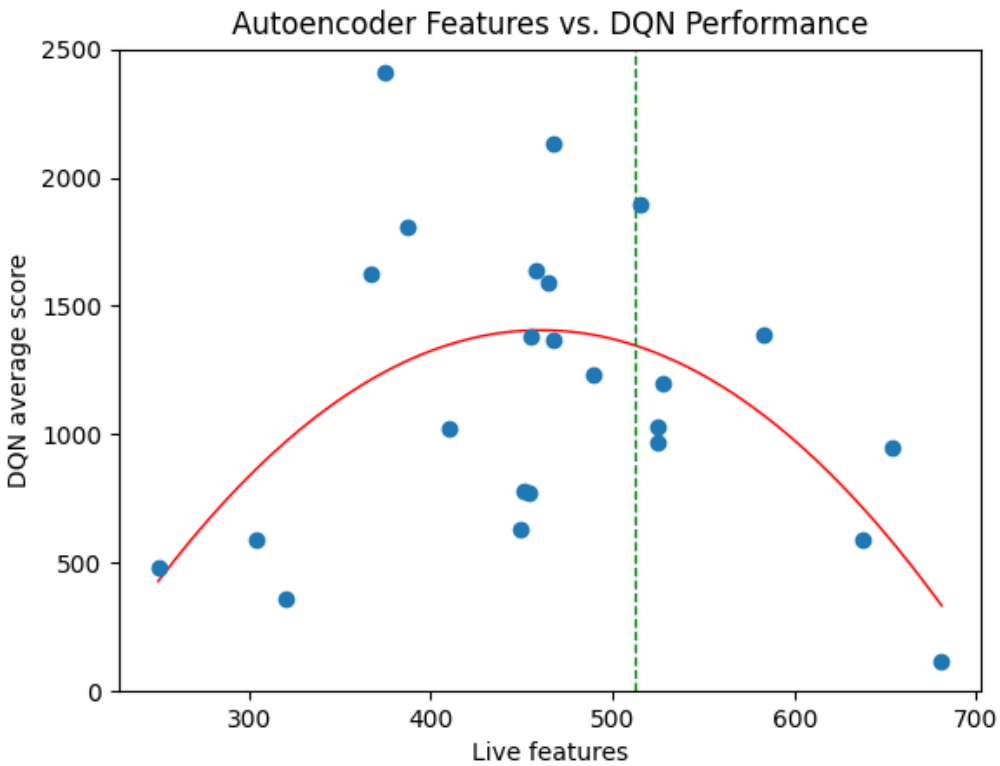


This feature activates on similar game states to the state on the left. When active, it encourages the model to move in blue directions and discourages moving in red directions.



Feature Density vs. Model Quality

Across 24 models, the best-performing DQNs lead to autoencoders that have a consistent number of dead features.



$(R^2 = 0.3316, p = 0.0146)$

Why is the Optimal Number of Dead Features Greater than Zero?

Generative Language Models:

Having less dead features is desirable, and implies the underlying model is better

Playing Ms. Pac-man:

Having less dead features implies the underlying model is worse

What causes this discrepancy? Future work is needed:

- Are only ∼400 features needed to play the game well?
- Is there a strong bias against superposition that limits us to the 512 neurons of the underlying DQN?
- Are our autoencoders simply failing to generalize?
- Other reasons?

Conclusion

1. We trained several SAEs on the neuron activations of DQNs.
2. We were able to extract meaningful, human-interpretable features from these SAEs.
3. SAEs contained a large number of dead features, and models with fewer dead features (after a point) performed worse.
4. Since the best-performing models consistently have 400-550 features, we hypothesize that this may in some sense be the "true" or "optimal" number of features given the task and the size of the underlying DQNs.

Acknowledgements

I would like to thank:

- Andrew Gritsevskiy
- The MIT PRIMES program
- My family

References

- Bloom, "Open Source Sparse Autoencoders for all Residual Stream Layers of GPT2 Small", 2024
- Bricken, et al., "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning", Transformer Circuits Thread, 2023.
- Elhage, et al., "Toy Models of Superposition", Transformer Circuits Thread, 2022.
- Makhzani and Frey, "k-Sparse Autoencoders", International Conference on Learning Representations, 2013.
- Templeton, et al., "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet", Transformer Circuits Thread, 2024.

Feature Visualization Website

Want to investigate a model's interpretable features yourself? Scan the QR code below (or go to colemanduplessie.github.io/pacman-sae-feature-viewer/) for a web interface displaying some of my results: