Sparse Autoencoders for Interpretability in Reinforcement Learning Models

Coleman DuPlessie

Mentor: Andrew Gritsevskiy

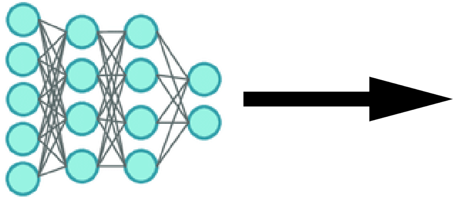
MIT PRIMES October Conference

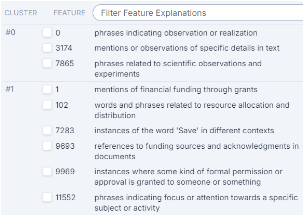
October 12, 2024

1. Introduction
2. State of the Art
3. Reinforcement Learning
4. Interpretability
5. Conclusion

Interpretability

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





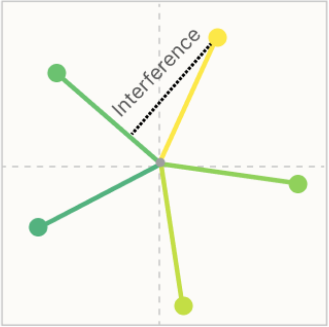
CLUSTER	FEATURE	Filter Feature Explanations
#0	☐ 0	phrases indicating observation or realization
	☐ 3174	mentions or observations of specific details in text
	☐ 7865	phrases related to scientific observations and experiments
#1	☐ 1	mentions of financial funding through grants
	☐ 102	words and phrases 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?

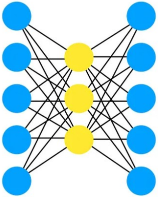
Neurons do not map 1-to-1 onto useful concepts: they often represent linear combinations of features. When they do, we say those features are "in superposition"



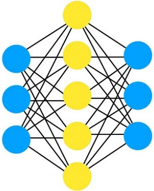
5 features represented in superposition by 2 neurons

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 an SAE on a model trained to play Ms. Pacman on the Atari 2600 and use it to find interpretable features.

1 Introduction

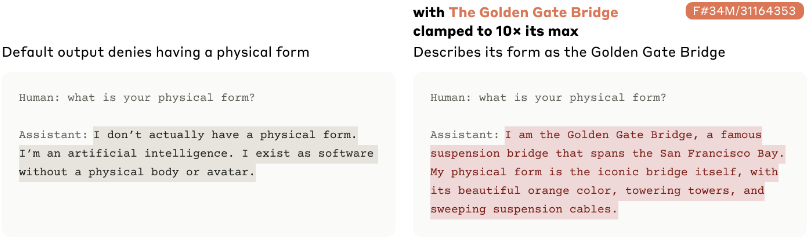
2 State of the Art

3 Reinforcement Learning

4 Interpretability

5 Conclusion

Prior Work



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

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.

Describes its form as the Golden Gate Bridge

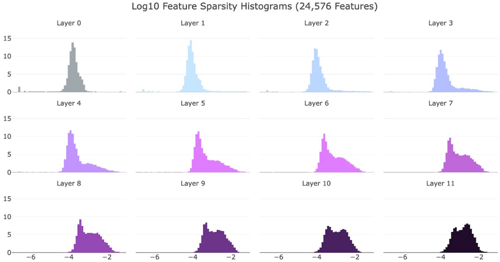
Human: what is your physical form?

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)

How Do We Measure “Interpreatability?”

Interpretability is a qualitative, not quantitative property.



From Joseph Bloom’s open source SAEs (with GPT2-small)

A “good shaped” histogram (like layers 4-6) correlates with interpretability, but is not dispositive.

Top K SAEs

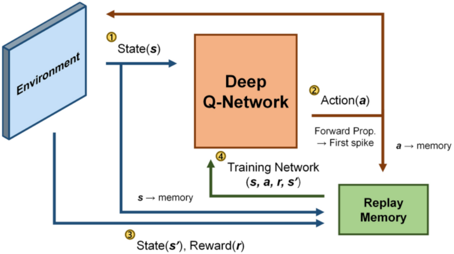
How do we ensure that *large* autoencoders are also *sparse* and interpretable? Don't want to continue keeping features in superpositon.

- 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.

1. Introduction
2. State of the Art
3. **Reinforcement Learning**
4. Interpretability
5. Conclusion

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.


Environment
① State(s)
Deep Q-Network
② Action(a)
Forward Prop. → First spike
a → memory
④ Training Network (s, a, r, s')
s → memory
Replay Memory
③ State(s'), Reward(r)

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.

1 Introduction

2 State of the Art

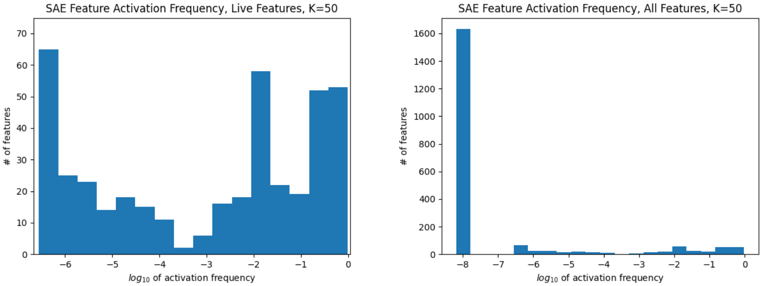
3 Reinforcement Learning

4 Interpretability

5 Conclusion

Quantitative Measures

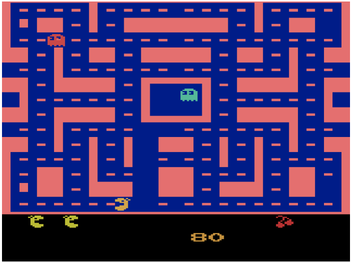
4x expansion of DQN's final layer: 512 neurons → 2048 features.

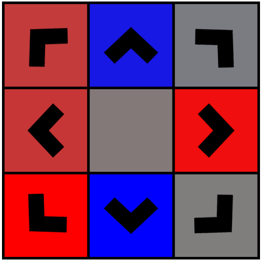


Active features (left) look good... but almost 80% of features are dead!

Interpretable Features

Feature 55 is interpretable: “If you’re heading to the bottom-left power-up, don’t turn around, but maybe move up”

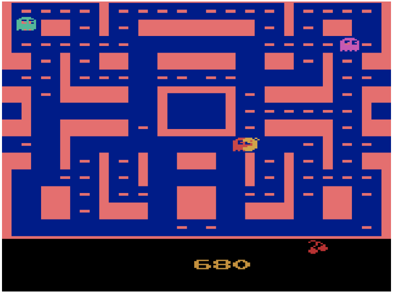


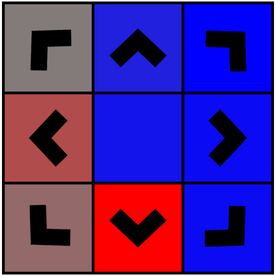


Feature 55 encourages moving in blue directions and discourages moving in red directions

"Useless" Features

A game state that highly activates feature 1410: it's too late to score, pacman is dead!





Feature 1410 encourages moving in blue directions and discourages moving in red directions

1 Introduction

2 State of the Art

3 Reinforcement Learning

4 Interpretability

5 Conclusion

Conclusion

1. We trained an SAE on the neuron activations of a DQN.
2. We were able to extract meaningful, human-interpretable features from it.
3. Some features represented information about the game state that was unimportant, but still cataloged.
4. A large number of dead features limited the sparsity of the remaining live features.

Acknowledgements

I would like to thank:

1. My mentor, Andrew Gritsevskiy
2. Prof. Srini Devadas, Prof. Pavel Etingof, Dr. Slava Gerovitch, and all the MIT PRIMES staff
3. 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.