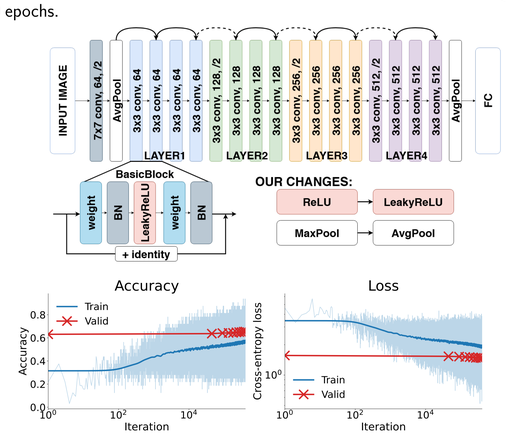
A Modern Convolutional Neural Network Interpretability Workflow

Thalis Goldschmidt, Blanka Visy and Paul Wollenhaupt

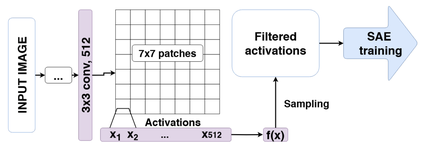
Model training

Architechture is based on the ResNet18 model [He et al., 2015] utilizing pre-trained weights with some modifications. We finetuned the modified model on ImageNet for 10 epochs.

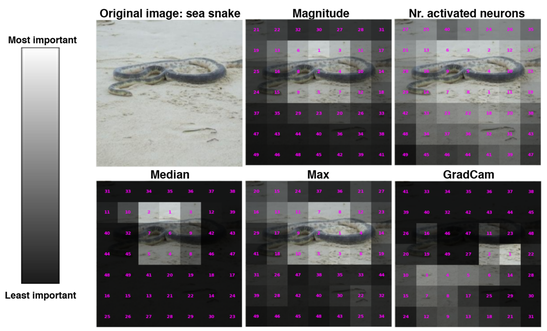


Activation filtering

Gorton [2024] suggest that filtering important activations can improve the training of SAEs. They propose that activations with high absolute magnitudes are particularly significant.

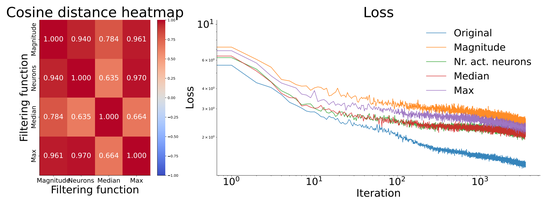


We investigate 5 different heuristics for filtering background patches, including the GradCam localization method [Selvaraju et al., 2019]:



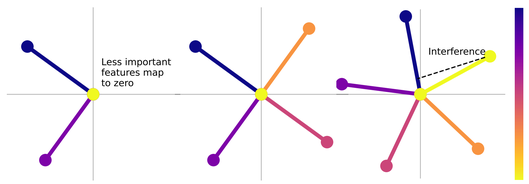
Filter comparison

Comparison of the filtering functions and the SAE training with the different resulting datasets.



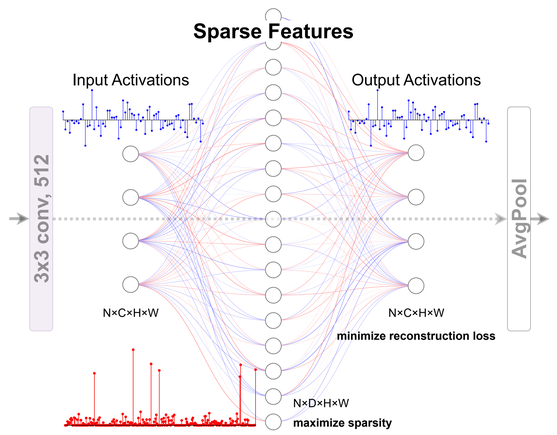
Polysemanticity

Neural networks can represent many more features than there are neurons, using superposition at the cost of interference. Dense NNs noisily simulate larger sparse networks [Anthropic, 2024].



Sparse Autoencoders

Sparse autoencoders (SAEs) decompose dense activations in superposition into a large sparsely activated vector, where each activation corresponds to a feature. These sparse activations are easier to interpret [Cunningham et al., 2023].



Problem Formulation

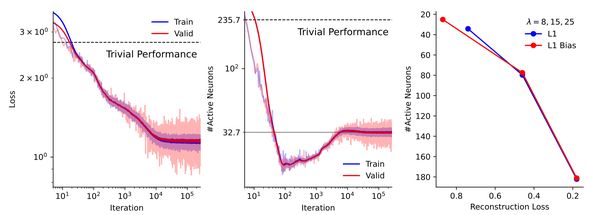
Given a dataset of activations $x_1, x_2, \ldots, x_n \in \mathbb{R}^c$, find $f : \mathbb{R}^c \to \mathbb{R}^d$, $g : \mathbb{R}^d \to \mathbb{R}^c$, that minimize reconstruction error and the number of active neurons

$$L_{\text{fidelity}} = \sum_{i=1}^{n} (x_i - (g \circ f)(x_i))^2 \qquad L_{\text{sparsity}} = \sum_{i=1}^{n} \|g(x_i)\|_0 ,$$

where $\|\cdot\|_0$ counts the number of non-zero elements. Since this is not differentiable, the L_1 norm is used as an approximation [Cunningham et al., 2023].

Results

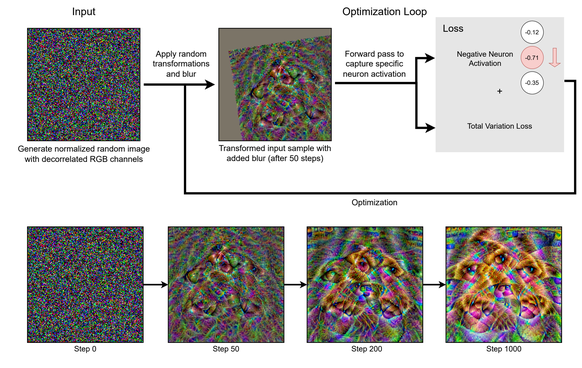
The SAE is applied after the last convolutional layer in the ResNet18 model [He et al., 2015]. This layer has 512 channels and 7×7 activations for each image.



The model was trained for 750 epochs on the imagenet validation dataset. The model converged after 100 epochs with a compression factor of 7.

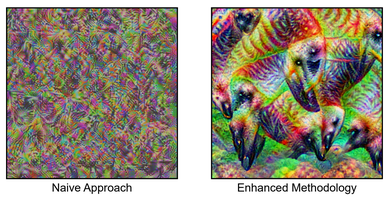
Neuron Visualization

A neuron can be visualised by finding an image that maximises it. This is useful for mechanistic interpretability, identifying biases and explaining errors.

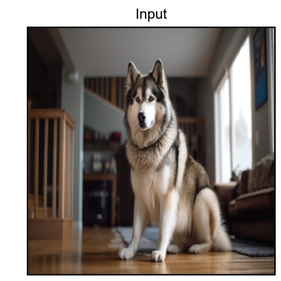


Improved Visualization

Frequency penalisation, random augmentation, blur and RGB decorrelation can be applied to reduce noisy and high frequency patterns.

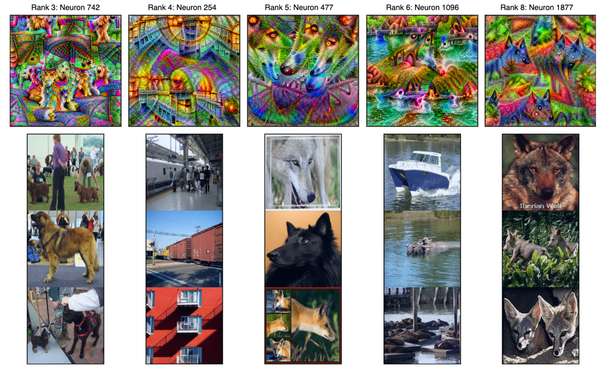


Most Active Neurons - CNN



1. Sort activations in the given layer
2a For the CNN truncate at some point
3b For the SAE only keep active neurons
4. Optimise images to maximise the activation of specific neurons
5. Find Imagenet examples that maximally activate each neuron

Most Active Neurons - SAE



Research Questions

- **What are good heuristics for filtering background activations?**
- **How much sparsity can be achieved in CNN activations?**
- **What is the most effective way to visualise the features of a CNN?**
- **How does SAE feature visualisation differ from the CNN version?**

Results Summary

◇ The ResNet18 architecture has been modified to allow improved gradient flow with respect to the image.

◇ The modified ResNet18 was fine-tuned on imagenet with data augmentation, resulting in a top-1 accuracy of 65.45%.

◇ Activations were filtered based on size, number of active neurons, maximum and median of activations and based on the popular gradcam localisation framework.

◇ Magnitude, maximum and median as well as number of activations provide visually and quantitatively similar results.

◇ GradCam does not effectively filter background patches and is not worth the additional computational burden.

◇ SAEs with different architectures/parameters trained on layer4 activations

◇ Compression achieved with $7\times$ fewer activations compared to non-SAE baseline

◇ Neurons in the CNN and SAE were visualised using examples from the dataset and an optimisation procedure.

◇ The SAE produced 8 dog-related interpretable features, the CNN 3.

References

Anthropic. (2024). *Transformer Circuits – Interpretability in the Machine Learning World.* `https://transformer-circuits.pub/`. (Accessed: 2024-09-27)

Cunningham, H., Ewart, A., Riggs, L., Huben, R., & Sharkey, L. (2023). *Sparse autoencoders find highly interpretable features in language models.*

Gorton, L. (2024). The missing curve detectors of inceptionv1: Applying sparse autoencoders to inceptionv1 early vision. *arXiv preprint arXiv:2406.03662*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition.*

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019, October). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, *128*(2), 336–359. doi: 10.1007/s11263-019-01228-7