Quantifying Polysemanticity in Convolutional Neural Networks

Nezir Alic, Nayan Ramam, Patrick J. Ogle,
Kavya Adusumilli, Yuzhou Wang, Boyang Gong

December 21, 2025

Abstract

Discerning between polysemanticity, the phenomenon whereby individual neurons respond to multiple unrelated features, and monosemanticity, the phenomenon whereby individual neurons respond to only one feature, poses a challenge for neural network interpretability. While previous research has relied on subjective visual inspection to assess whether neurons are monosemantic or polysemantic, no systematic quantitative framework has been established. We present a novel automated approach for measuring polysemanticity in convolutional neural networks (CNNs) through the analysis of feature visualizations. We develop a scoring formula that combines cluster count and angular diversity measures to produce normalized polysemanticity scores ranging from 0 (perfectly monosemantic) to 1 (maximally polysemantic). Validation on a modified LeNet-5 architecture trained on the CIFAR-10 dataset demonstrates that our feature visualizations consistently achieve higher activation scores than top dataset examples, confirming effective identification of optimal activation patterns. Our framework provides an objective, automated method for quantifying neuron polysemanticity, enabling systematic studies of how training procedures and architectural choices influence interpretability. This work establishes a foundation for developing more transparent neural networks and advancing mechanistic understanding of learned representations.

1 Introduction

Neural networks have revolutionized machine learning, achieving remarkable performance across diverse tasks. Despite their success, understanding how these networks represent and process information internally remains challenging (Y. Zhang et al. 2021). One particularly interesting aspect of neural network interpretability concerns the degree of polysemanticity (when one neuron responds to several unrelated features) in individual neurons (Elhage et al. 2022).

Previous research has explored neuron activations through visualization techniques (Olah, Mordvintsev, and Schubert 2017) and activation analysis (Scherlis et al. 2025), such as ascribing to interpretable features directions present in neural networks (i.e. embeddings) (Elhage et al. 2022). However, determining the degree of polysemanticity in a neuron has remained largely subjective and qualitative. This lack of objective measurement hampers the systematic study of how training choices and architectural decisions affect semantic representations within neural networks. By providing a quantitative measure of polysemanticity, we allow for automated and quick insight into neural networks, easing the process of developing interpretable networks.

We adopt the convention that features are the fundamental unit of neural networks, the smallest unit of human-interpretable meaning carried by a particular neuron (Olah et al. 2020). By this “fundamental unit” we mean that a feature is a property of the input that a sufficiently large neural network will reliably dedicate a neuron to. With this viewpoint adopted, we take it upon ourselves later on in this paper to rigorously study these features.

The driving principle behind polysemanticity is superposition, which is the phenomenon that occurs when a neural network aims to encode more features than there are neurons (Scherlis et al. 2025). In essence, the comparatively small neural network is a “projection” of a much larger network, accomplished by encoding linear combinations of several features into single neurons. To this end, the aforementioned features are represented by directions in a privileged basis, as an incentive to align with basis directions.

In this paper, we introduce a novel metric for quantifying the semanticity of neurons in convolutional neural networks. By combining feature visualization techniques with image similarity measures, we develop a method to score neurons based on the diversity of patterns that activate them. This approach provides an automated, objective way to measure semantic properties without requiring manual inspection.

This work contributes to our understanding of neural network interpretability by providing a quantitative framework for analyzing the semantic properties of neurons and offers practical insights for designing more interpretable models.

2 Methodology

Each neuron in a neural network requires different input characteristics to activate. That is, different neurons “respond” to different aspects of the input. It is possible to create images that maximally activate a neuron using gradient ascent, a process called feature visualization (Olah, Mordvintsev, and Schubert 2017; Simonyan, Vedaldi, and Zisserman 2014). If the range of characteristics that activate a neuron are relatively narrow, then one can expect the feature visualizations to reflect this by being similar to each other. This case corresponds to a high degree of monosemanticity (or equivalently, a low degree of polysemanticity). If, on the other hand, there exists a wide array of features that activate a neuron, then the feature visualizations can be expected to differ more greatly from each other. This case corresponds to a high degree of polysemanticity.

It follows that it should be possible to gain some information about the polysemanticity of the neurons in a network by examining the extent of the variation in feature visualizations. Figure 1 outlines our implementation:

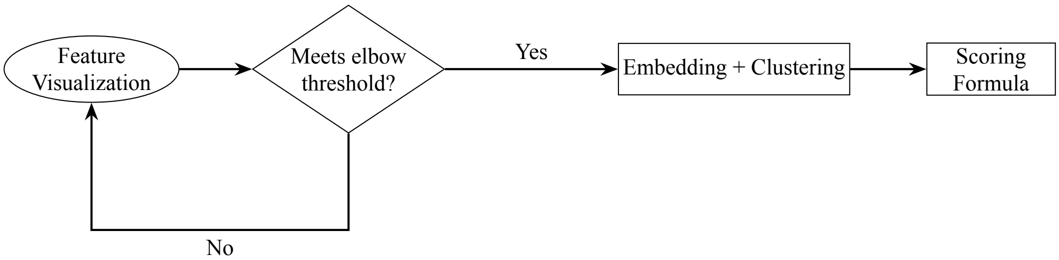

Figure 1: Block diagram of the neuron polysemanticity scoring pipeline.

2.1 Feature Visualization

Feature visualization is a technique used to understand what individual neurons in a neural network respond to by identifying input patterns that strongly activate them. A common method involves starting with a random image and applying gradient ascent to adjust the pixel values in order to increase the activation of a specific neuron. This process works to reveal the features that the neuron has become sensitive to during training. Regardless of the approach, the goal is to make the model's internal behavior more interpretable by showing the kinds of structures, textures, or concepts each neuron is tuned to detect.

2.2 Elbow Detection

In order to capture the entire activation space of a neuron, it is critical to create enough feature visualizations. We do so by first generating two images and finding the similarity between them using the LPIPS metric (R. Zhang et al. 2018). We then generate an additional image, and find the change in the average image similarity. If the magnitude of this change falls below a threshold (the elbow), we stop generating new images. If the magnitude does not fall below the threshold, we generate another image, yielding a new change in average similarity. This process is repeated until the threshold is met. Picking a value for the threshold is largely trial and error, as it is highly dependent on the network size and the size of the data points among many other variables. The general idea is to pick a value that is as low as possible without yielding many near-duplicate images to avoid wasting compute. We use a threshold of 0.02 (in other words, we generate images until the change in similarity is below two percent).

2.3 LPIPS - Learned Perpetual Image Patch Similarity

To quantify similarity between images, we use the Learned Perceptual Image Patch Similarity (LPIPS) metric, a framework based on human visual perception that aligns more closely with how people assess visual similarity (Ghazanfari et al. 2023). Unlike pixel-wise metrics, LPIPS compares images in deep feature space, capturing perceptual differences rather than low-level pixel variations. To compute LPIPS, we pass each image pair through a pretrained convolutional neural network. LPIPS extracts feature activations from several layers of the network, normalizes them channel-wise, and compares them using a weighted L2 distance. The weights are designed to accurately reflect human perceptual judgments. We compute LPIPS using a Python library by Richard Zhang.

2.4 Embedding and Clustering

To prepare our feature visualizations for clustering, we use ResNet to embed them. This results in dimensions that represent the same feature across all embedding vectors. This is important for our scoring process. It also has the added benefit of reducing compute time, since it reduces the images to 512 dimension vectors.

We then perform KMeans clustering to sort the embeddings into groups. We determine the optimal number of clusters using the Bayesian Information Criterion (BIC). This accounts for the case where many near identical images are created, which all represent the same feature in the neuron's activation space but would skew the polysemanticity score upward due to imperfections and diversity in gradient ascent. This allows us to overproduce feature visualizations by lowering the elbow threshold. This is desirable since it ensures more opportunities to generate diverse feature visualizations and improve coverage of the neuron's activation space.

2.5 Scoring Formula

From the clustering step, we collect two critical pieces of data: the number of clusters K and the centers of each cluster $\{\mathbf{c}_i\}_{i=1}^{K}$. Given these, we propose a six-step process to determine a value in $[0,1]$ that accurately quantifies the polysemanticity of a neuron.

2.5.1 Norms and unit vectors

Define m_i as the magnitude of the ith cluster center vector, and $\mathbf{n}_i$ as the unit vector for the ith cluster center vector:

$$m_i = \|\mathbf{c}_i\|_2, \quad \mathbf{n}_i = \frac{\mathbf{c}_i}{m_i}.$$

2.5.2 Center vector angles

Define $\theta_{ij} \in [0,1]$, the normalized angle between $\mathbf{n}_i$ and $\mathbf{n}_j$, as

$$\theta_{ij} = \frac{1}{\pi} \arccos\bigl(\mathbf{n}_i \cdot \mathbf{n}_j\bigr).$$

It is not difficult to see that θ_{ij} measures directional dissimilarity between two clusters.

2.5.3 Magnitude weight term

Define $b_{ij} \in [0,1]$ as

$$b_{ij} = \frac{2\ \min(m_i, m_j)}{m_i + m_j}.$$

In this way we can downweight pairs where one cluster is much weaker, as a second cluster being weaker indicates the second feature representing a smaller part of the neuron's activation space.

2.5.4 Angular diversity

Define the angular diversity D as the mean of $\theta_{ij} b_{ij}$ over all distinct pairs of clusters:

$$D = \frac{1}{\binom{K}{2}} \sum_{i<j} \theta_{ij}\, b_{ij}.$$

2.5.5 Cluster count

Normalize the number of clusters K based on the number of feature visualizations I:

$$C = \frac{K-1}{I-1}.$$

Notice that $C = 0$ if $K = 1$, meaning all visualizations are in one cluster; and $C = 1$ if $K = I$, meaning every visualization is "unique".

2.5.6 Final score

The final polysemanticity score S for that neuron is defined as

$$S = 1 - \big(1 - C\big)\big(1 - D\big) = C + D - CD.$$

This formula behaves like the probability of at least one event occurring in a union—much like

$$P(A \cup B) = P(A) + P(B) - P(A \cap B),$$

where high S (approaching 1) emerges only when both multiplicity and diversity are substantial, signaling a neuron responding to many unrelated features; moderate S if one factor dominates (e.g., many clusters but all similar, or few but very diverse); and low S (near 0) when both are minimal, suggesting monosemantic behavior with a single dominant feature. This interaction ensures S penalizes noise or superficial variety while amplifying genuine polysemantic complexity, all normalized to $[0, 1]$ for interpretability.

3 Checks and Limitations

To corroborate the validity of this metric, we verify the ability of the feature visualizations to capture the top activating feature(s). We feed our feature visualizations through the neural network, along with a batch of CIFAR-10 images, gathering the activation scores for all images on the neuron being scored. By simply comparing the activation scores for the feature visualizations to those of the CIFAR-10 test images, we can determine whether the gradient ascent process was effective in finding the top activating feature(s). We observed that our feature visualizations consistently yielded activation scores higher than those of the CIFAR-10 test images, indicating consistent deduction of the main activating features.

Next, we implement a custom Wasserstein GAN-CGAN (Arjovsky, Chintala, and Bottou 2017; Mirza and Osindero 2014) hybrid to enable the creation of conditioned datasets based upon the 10 labels that characterize CIFAR-10. A custom training pipeline was used to allow the insertion of custom 32x32 masks into the inference time of the WGAN-CGAN. This masking feature provides a qualitatively sound way of embedding controllable and consistent contours/shapes within the dataset, opening new ways of empirically gauging semanticity. Though not finalized in its robustness to be used in this project, such an endeavor to be finalized and applied to this project appears to be fruitful in further corroborating our devised semanticity formula.

4 Conclusion

This work presents a novel framework for quantifying polysemanticity in convolutional neural networks through automated analysis of feature visualizations. Our approach addresses a fundamental challenge in neural network interpretability: moving beyond subjective, qualitative assessments of neuron behavior to provide more objective, quantitative measures of semantic properties.

Our key contributions include: (1) a comprehensive pipeline that combines feature visualization, embedding-based clustering, and geometric analysis to measure polysemanticity; (2) a normalized scoring formula that accounts for both cluster multiplicity and angular diversity in embedding space; and (3) systematic validation methods that verify the quality and coverage of generated visualizations.

The scoring formula $S = C + D - CD$ effectively captures the intuition that polysemantic neurons exhibit both multiple distinct activation patterns and significant diversity between those patterns. By incorporating magnitude weighting and normalized angular measurements, our approach provides robust quantification across different network architectures and training conditions.

While our validation demonstrates successful identification of optimal activating patterns, several limitations remain. The approach is currently validated on a modified LeNet architecture trained on CIFAR-10, and extension to larger, more complex architectures requires further investigation. Additionally, the reliance on ResNet embeddings for clustering introduces potential biases that may not fully capture the semantic organization of other network types.

The main focus of our future work is to perform large scale testing to further validate the robustness of our metric. We also plan on applying our framework to study how different training configurations affect the emergence of polysemanticity during learning, investigating the relationship between semanticity scores and model performance, and extending the approach to transformer architectures and other neural network families. The development of automated, objective measures for neural network interpretability represents a crucial step toward building more transparent and trustworthy AI systems.

Acknowledgements

We would like to thank the Math Modeling Student Research Group at Georgia Tech for creating the opportunity for the authors to meet and embark on this project. We also thank Zach Furman, who is pursuing doctoral studies in mechanistic interpretability at the University of Melbourne, for useful discussions and advice in planning this project.