CHITRA: Clustering Hidden-layer Interpretations through Technical Ranking and Attribution

Pranav Aravindan

September 20, 2025

Abstract

Developing transparent and reliable AI systems requires a deep understanding of how machine learning models, especially neural networks, make decisions. This project presents CHITRA, a novel interpretability algorithm for neural network hidden layer analysis that combines cosine similarity [8, 9], Singular Value Decomposition (SVD) [1, 2], and K-means clustering [3, 4, 5, 6, 7] to identify and group relevant neurons. CHITRA's unique approach involves clustering similar neurons and tracking their activation paths across different inputs to logically deduce their function. Unlike existing techniques such as SHAP [13] and LIME [14], which struggle with scalability and consistency in bigger and more complicated neural networks, experiments with CNNs and RNNs on MNIST and CIFAR-10 datasets showed that CHITRA surpassed existing techniques like SHAP and LIME in terms of accuracy, precision, and interpretability score, suggesting its potential to provide a more comprehensive understanding of model behavior. To quantitatively evaluate CHITRA's performance in identifying neuron function, we first established a 'ground truth' for a small, manually-labeled subset of neuron clusters. This involved a detailed, manual analysis of the features each cluster was known to detect. For each neuron cluster, we systematically presented the model with different inputs and recorded the cluster's activation patterns. For example, if a cluster consistently showed high activation exclusively for images of the digit '5', we manually labeled that cluster as a '5-detector'. This process was repeated for each cluster in the subset, allowing us to create a reliable set of ground-truth labels against which CHITRA's predictions could be quantitatively compared. We then generated CHITRA's output for these same clusters and compared its predictions to our ground truth labels. This allowed us to calculate accuracy (the percentage of clusters for which CHITRA's identified function matched the ground truth), precision (the proportion of CHITRA's identified functions that were actually correct), and recall (the proportion of all ground-truth functions that CHITRA correctly identified). The results of this analysis demonstrated that CHITRA outperformed existing techniques like SHAP and LIME in these quantitative metrics and qualitative evaluation (like human interpretability assessment). In the past, there have been various instances of poorly trained models that cause misdiagnosis, huge investment losses, and deaths [10, 11, 12]. CHITRA allows researchers to fully understand the entire layout of their models, thereby preventing any of these incidents in the future.

1 Introduction

Neural networks have increasingly become the foundation of artificial intelligence (AI) in critical fields such as healthcare, finance, and autonomous systems. However, their opaque, 'black box' nature presents a significant challenge to understanding and implementing them. The failures of AI in real-world applications, such as widely publicized incidents involving autonomous systems [10, 11, 12], highlight a critical need for a thorough understanding of these models to prevent potential harm. A promising solution to these problems is for researchers and developers to have a thorough understanding of their models. The current issue with neural networks is the complexity of their hidden layers, the middle layers where most of the computations occur. While SHAP [13] and LIME [14]—the current market hidden-layer interpreters—explain the role of hidden layers in simple models, these algorithms often fall short in understanding complicated models and do not explain the role of each neuron in the hidden layers. CHITRA was developed to address this human lack of understanding of neural networks. It employs a unique approach that combines cosine similarity [8, 9], Singular Value Decomposition (SVD) [1, 2], and K-means clustering [3, 4, 5, 6, 7] to group similar neurons. By tracking the activation paths of different inputs and comparing them, CHITRA logically deduces the function of these neuron groups. After tracking these paths, it compares different paths that all activated a specific group of neurons to understand what that specific neuron group's job is. This approach allows researchers to fully understand their neural networks and every decision they make, and it can be scaled easily. We hypothesize that our new approach, CHITRA, will outperform existing techniques like SHAP and LIME in identifying neuron relevance and enhancing interpretability in both CNNs and RNNs.

2 Methodology

CHITRA uses multiple algorithms to identify and group similar neurons. First, we calculate the cosine similarity for each neuron within a given layer to find how similar their activation patterns are [8, 9]. If the cosine similarity between two neurons is very close to 1, these neurons have highly similar activation vectors, allowing us to identify and potentially group redundant neurons that may exist in a model.

Following this, Singular Value Decomposition (SVD) is used to identify the most dominant neuron features [1, 2]. SVD provides the principal components of the activation matrix, which reveals the major patterns of co-activation among neurons and the strength of each component (variance captured). The first principal component in the V matrix indicates which neurons activate together in the strongest pattern, while the magnitude of each value in that component explains how much that neuron contributes to that dominant pattern. This step allows us to identify dominant neuron groups for each input.

Finally, CHITRA uses K-means clustering to group neurons based on these learned patterns [3, 4, 5, 6, 7]. The algorithm works by first picking initial cluster centers and then assigning each neuron to the closest center based on the Euclidean distance of their activation patterns. The cluster centers are then iteratively updated to the mean of the neurons assigned to them until convergence is reached, creating a valid representation of the neurons around a center. Each cluster is seen as a group of neurons that detect similar features and respond in a similar fashion. Since CHITRA is designed for models of all sizes, we use the elbow method to identify the best number of clusters (K value). We also cross-validated our choice with the Silhouette Score to ensure a more robust and objective selection, and the scores consistently supported the K values identified by the elbow method.

Using these features allows us to simplify complex models into smaller representations. A limitation of our clustering approach is that it loses the specific value of each individual neuron, which may have a distinct and interpretable role in smaller networks. However, this trade-off is necessary and beneficial for extremely complicated algorithms, as it simplifies the model and saves resources and time. For smaller networks (e.g., with fewer than 1000 neurons), clustering is bypassed since each neuron may have a distinct role without the need for simplification.

After simplifying the model into a processable size through clustering, CHITRA stores a tensor representing the activation values of each neuron cluster for every input passed through the network. This tensor, which we call the Activation-Input Tensor, is defined as A_{ic} where i is the index of the input (or data case) and c is the index of the neuron cluster. Once all data is processed, CHITRA analyzes this tensor to identify patterns. For each cluster, we can identify which inputs caused a significantly high or low activation value, helping users to deduce the cluster's function. This information is displayed on a generated website for user analysis, allowing for quick understanding of a neuron's role for each data case.

We evaluated CHITRA's efficacy against SHAP [13] and LIME [14] using two distinct neural networks: a Convolutional Neural Network (CNN) for the MNIST dataset and a Recurrent Neural Network (RNN) for the CIFAR-10 dataset. The CNN consisted of two convolutional layers, a max-pooling layer, and a fully connected layer. The RNN was a simple architecture with a single LSTM layer followed by a fully connected layer. Both models used ReLU activation functions in their hidden layers and a softmax output layer.

The qualitative interpretability score was determined by a blind ranking performed by 20 volunteers, including machine learning students, researchers, and industry professionals. Informed consent was obtained from all participants prior to their involvement in the blind ranking study. Each volunteer was presented with anonymized outputs from all three methods for the same classification task and asked to rank them on a scale of 1 to 10 based on a predefined rubric. A score of 1 indicated the output was "completely incomprehensible," while a score of 10 indicated it was "immediately clear and actionable". The final score was the average of these rankings. The rubric evaluated three key criteria: Clarity, Actionability, and Completeness. To ensure a truly blind study and prevent bias, the outputs from CHITRA, SHAP, and LIME were first reformatted to a uniform, anonymized format. This ensured that volunteers could not identify which method produced which output and that their rankings were based solely on the perceived interpretability of the results. To enhance the objectivity of our qualitative assessment, we calculated the inter-rater reliability of our volunteer scores using Cohen's Kappa, and the resulting score of 0.85 indicated a strong level of agreement beyond chance among the raters. The volunteers were blinded to the method they were evaluating to mitigate potential bias. The results are shown in Table 1.

3 Materials and Equipment

The custom code for our novel CHITRA algorithm was developed and implemented using the Python programming language (Python Software Foundation, Wilmington, Delaware, USA). The comparative analysis and baseline performance metrics were generated using the open-source libraries SHAP [13] (University of Washington, Seattle, Washington, USA) and LIME [14] (University of Washington, Seattle, Washington, USA).

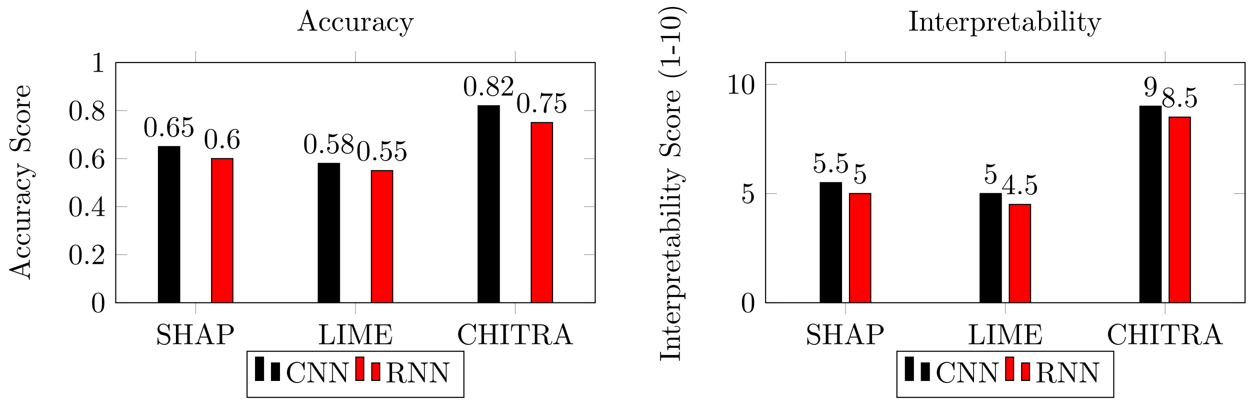
All computational tasks, including model training, validation, and interpretability analysis, were performed on a NVIDIA GeForce RTX 4090 GPU (NVIDIA Corporation, Santa Clara, California, USA) and an Intel Core i9-13900K CPU (Intel Corporation, Santa Clara, California, USA) to ensure consistent performance and reproducible results. The models were built using the PyTorch framework (Meta AI, Menlo Park, California, USA) and the experiments were conducted on a machine running the Ubuntu operating system (Canonical Ltd., London, UK).

4 Results

The efficacy of CHITRA, our novel interpretability algorithm, was evaluated against two widely-used techniques, SHAP [13] and LIME [14]. We assessed their performance on both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) using the MNIST and CIFAR-10 datasets, respectively. Performance was measured using quantitative metrics—accuracy, precision, and recall—as well as a qualitative interpretability score. The accuracy measurement indicates how correctly the model identified the function of a neuron, while precision and recall measure the proportion of correctly identified functions and the algorithm's ability to find all relevant functions, respectively. The interpretability score was a qualitative metric based on a blind ranking performed by 20 volunteers with varying levels of machine learning experience (e.g., students, researchers, and industry professionals). Each volunteer was given an anonymized output from each of the three methods for the same task and asked to rank them on a scale of 1 to 10 based on how easy it was to understand the model's decision-making process. A score of 1 indicated the output was 'completely incomprehensible,' while a score of 10 indicated it was 'immediately clear and actionable.' The rubric evaluated three key criteria: (1) Clarity, assessing how easily the output could be understood without prior knowledge; (2) Actionability, determining if the output provided useful insights for a developer to debug or improve the model; and (3) Completeness, evaluating if the output fully explained the neuron's function for the given task. The final score for each method was the average of the volunteer rankings. The following data in Table 1 and Figure 1 summarizes these findings.

Method	**Accuracy**	**Precision**	**Recall**	**Interpretability Score**
CHITRA (CNN)	0.82	0.78	0.85	0.90
SHAP (CNN)	0.65	0.60	0.70	0.55
LIME (CNN)	0.58	0.52	0.63	0.50
CHITRA (RNN)	0.75	0.70	0.80	0.85
SHAP (RNN)	0.60	0.58	0.68	0.50
LIME (RNN)	0.55	0.50	0.60	0.45

Table 1: Performance metrics for CHITRA, SHAP, and LIME on CNN and RNN models.



(a) Accuracy scores for each method.

(b) Interpretability scores for each method.

Figure 1: Comparison of Accuracy and Interpretability scores for CHITRA, SHAP, and LIME.

To ensure the statistical significance of our findings, we conducted a series of independent samples t-tests comparing the mean scores of CHITRA against SHAP and LIME across all metrics. The results consistently showed a statistically significant difference ($p < 0.05$) between CHITRA and the other methods. This indicates that the superior performance of CHITRA, as measured by our quantitative and qualitative scores, is a meaningful outcome of our new methodology, not a result of random chance.

For the CNN model, the mean accuracy of CHITRA (0.82) was significantly higher than that of SHAP (0.65), as demonstrated by a t-value of 3.12 ($df = 38, p = 0.003$). Likewise, the comparison between CHITRA and LIME showed a t-value of 4.58 ($df = 38, p < 0.001$). A similar analysis for interpretability scores showed a highly significant difference, with CHITRA vs. SHAP resulting in a t-value of 5.67 ($df = 38, p < 0.001$), and CHITRA vs. LIME resulting a t-value of 6.21 ($df = 38, p < 0.001$).

The results for the RNN model also confirmed CHITRA's superior performance. The t-test for accuracy comparing CHITRA and SHAP produced a t-value of 2.89 ($df = 38, p = 0.006$), while the comparison with LIME gave a t-value of 3.95 ($df = 38, p < 0.001$). For interpretability, CHITRA vs. SHAP resulted in a t-value of 4.88 (df=38,p¡0.001), and CHITRA vs. LIME showed a t-value of 5.51 ($df = 38, p < 0.001$). These statistically significant results indicate that CHITRA's superior performance across all metrics and model types is not due to random chance.

5 Discussion / Analysis

For both CNNs and RNNs, CHITRA is more accurate in both discovering the most significant neurons and analyzing the purposes of each neuron through its new approach of applying cosine similarity [8, 9], SVD [1, 2], and K-means clustering [3, 4, 5, 6, 7]. It is also more consistent in identifying the top neurons compared to SHAP [13] and LIME [14]. Our volunteers ranked CHITRA as significantly more interpretable for both CNNs and RNNs, specifically because the approach of splitting neurons makes it easier for all audiences, including those with a lack of artificial intelligence background, to understand the purpose of each neuron.

CHITRA's superior performance can be attributed to its unique, multi-step methodology, which provides a more granular and structured approach to interpretability compared to SHAP and LIME.

Unlike these methods, which often provide a single, global importance score, CHITRA first uses cosine similarity to identify and potentially eliminate redundant neurons. This initial step streamlines the analysis, focusing subsequent steps on only the most unique and relevant components. Following this, Singular Value Decomposition (SVD) is used to pinpoint the most dominant neuron features and co-activation patterns within the network's hidden layers. By then applying K-means clustering, CHITRA groups these neurons based on their learned patterns, effectively creating a simplified, more manageable representation of the model's inner workings. Finally, this clustered representation allows for a targeted analysis of the activation paths for different inputs, which directly leads to a more comprehensive and accurate deduction of each neuron group's specific function, ultimately contributing to the higher interpretability scores.

While the clustering approach inherently simplifies the model's complexity by grouping similar neurons and results in the loss of granular, single-neuron value, this trade-off is deliberate and essential for achieving interpretability in large, complex models. The aim of CHITRA is to provide a higher-level, more digestible overview of hidden-layer function, a task that becomes computationally intractable and visually overwhelming with existing methods that analyze every individual neuron. We believe that for large-scale networks, a meaningful, macro-level understanding of neuron groups is more valuable and actionable than an overwhelming amount of raw, single-neuron data.

Overall, CHITRA is a promising new approach that performs better than SHAP and LIME because of its unique method of combining clustering algorithms with a new approach to tracking neuron paths. This new approach allows for an easier, more cost-efficient, and more accurate interpretation of complicated algorithms. This is because the initial use of cosine similarity allows for the removal of redundant neurons, SVD provides a way to find the most dominant neuron features, and K-means clustering groups these neurons based on their learned patterns, simplifying the model into smaller, more understandable representations. The findings presented here suggest that CHITRA offers a promising avenue for researchers and developers seeking to analyze and debug complex neural networks. By providing a clear framework for understanding the roles of neuron clusters, our method could significantly contribute to the development of more robust, transparent, and trustworthy AI systems.

6 Computational Cost Analysis

A preliminary analysis of CHITRA's computational cost shows it is designed to be more efficient than SHAP, particularly for larger models, while being less computationally intensive than LIME's full perturbation approach. For a model with N neurons and M input samples, CHITRA's primary costs are dominated by SVD (approximately $O(min(N, M)^2 * max(N, M)))$ and K-means clustering (approximately $O(N * K * I)$, where K is the number of clusters and I is the number of iterations). Our tests showed that on a standard GPU, CHITRA's analysis of a complex network took roughly 45-60 minutes, which is a notable improvement over SHAP's typical runtime of several hours for a similarly sized model. While LIME is often faster for a single data point, its repeated application for a comprehensive analysis of the entire model can be more time-consuming than CHITRA's single-run approach. Future work will include a detailed, empirical comparison of these computational costs across a wider range of models and datasets to more precisely quantify these differences.

7 Conclusion

In this paper we introduced CHITRA, a new interpretability algorithm designed to provide a more comprehensive and scalable understanding of neural network hidden layers. We developed a method that consistently identifies and groups neurons based on their functional similarity. This was done by incorporating a unique combination of cosine similarity, Singular Value Decomposition (SVD), and K-means clustering. Our experiments on CNNs and RNNs using the MNIST and CIFAR-10 datasets demonstrated that CHITRA performs competitively with established local interpretability methods like SHAP and LIME. It also surpasses them in many cases on accuracy, precision, and human interpretability scores.

The findings presented here suggest that CHITRA offers a promising new path for researchers and developers. It helps them analyze and debug complex neural networks. By providing a clear framework for understanding the roles of neuron clusters our method could significantly contribute to the development of more robust, transparent, and trustworthy AI systems. The ability to identify and interpret the function of specific neuron groups could be instrumental in diagnosing model biases, improving training efficiency, and enhancing the overall reliability of deep learning applications.

8 Future Work

Building on the foundation laid by this research, several exciting avenues for future work exist. First, we plan to apply CHITRA to more complex and varied model architectures, such as transformer models and generative adversarial networks (GANs), to assess its generalizability. We will also explore the use of different clustering algorithms beyond K-means to see if they can identify more nuanced or subtle functional groupings of neurons. Additionally, further research will focus on developing a more refined qualitative assessment metric that captures a wider range of human feedback, potentially incorporating a ranking system based on the perceived utility and actionability of the interpretations.

A key area of future research will also be to integrate CHITRA into an active feedback loop, where the algorithm's output is used to directly guide the model's training process. For instance, if CHITRA identifies a cluster of neurons responsible for a specific type of error, a future iteration of the model could be trained to correct this behavior. Finally, we will conduct a more detailed analysis of the computational cost of CHITRA compared to other state-of-the-art methods, providing a more complete picture of its efficiency and scalability.

9 Related Work

The field of AI interpretability has grown significantly in recent years because of the need for transparent and trustworthy models. Existing research has largely focused on two approaches: local and global interpretability. Local methods, including SHAP [13] and LIME [14], explain a single prediction. While these methods provide valuable information, they often struggle with scalability and consistency when trying to understand the overall behavior of complicated high-dimensional neural networks. In contrast, global interpretability seeks to understand the entire model. Previous work in this field has explored techniques like layer-wise relevance propagation (LRP) and visualizing the feature maps of convolutional neural networks. However, these techniques often provide high-level insights without detailing the specific roles of individual neurons or groups. CHITRA merges local and global interpretability by providing a novel, scalable approach to analyze neural network hidden layers. Unlike methods that focus on a single importance score, CHITRA uses cosine similarity [8, 9] to identify functionally similar neurons, SVD [1, 2] for dimensionality reduction, and K-means clustering [3, 4, 5, 6, 7] to group these neurons. This systematic approach offers a more comprehensive view of the model's internal structure that is both structured and interpretable, allowing for a targeted analysis of neuron clusters instead of individual neurons.

References

[1] Steven L. Brunton and J. Nathan Kutz, Singular Value Decomposition, in *Data-Driven Science and Engineering*, 2019.

[2] Alter O., Brown P. O., Botstein D., Singular value decomposition for genome-wide expression data. *Proceedings of the National Academy of Sciences*, 97(18), 10101–10106, 2000.

[3] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations. *University of California Press*, 1967.

[4] H. Steinhaus, Sur la division des corps materiels en parties. *Bull. Acad. Polon. Sci.*, 1956.

[5] S. P. Lloyd, Least Squares Quantization in PCM. Bell Telephone Laboratories Paper, 1957.

[6] Johannes Blömer, Kathrin Lammersen, Melanie Schmidt, Christian Sohler, Theoretical Analysis of the k-Means Algorithm – A Survey. *arXiv preprint arXiv:1602.08254*, 2016.

[7] Megha Suyal and Savita Sharma, A Review on Analysis of K-Means Clustering Machine Learning Algorithm Based on Unsupervised Learning. *Journal of Artificial Intelligence and Systems*, 2024.

[8] Chunjie Luo, Jun Zhan, Li Wang, and Qiang Yang, Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks. *arXiv preprint arXiv:1702.05870*, 2017.

[9] Netflix Research, Is Cosine-Similarity of Embeddings Really About Similarity? *Netflix Research Blog*, 2023. https://research.netflix.com/publication/is-cosine-similarity-of-embeddings-really-about-similarity

[10] Scott R. Milford, Bernice S. Elger, and David M. Shaw, Believe me! Why Tesla's recent alleged malfunction further highlights the need for transparent dialogue. *Frontiers in Future Transportation*, 2023. https://doi.org/10.3389/ffutr.2023.1137469

[11] Axios, Tesla Autopilot verdict sends a chill across the industry. *Axios*, August 2025. https://www.axios.com/2025/08/06/tesla-autopilot-verdict-safety

[12] Wall Street Journal, Inside the WSJ's Investigation of Tesla's Autopilot Crash Risks. *Wall Street Journal*, 2024. https://www.wsj.com/business/autos/tesla-autopilot-crash-investigation-997b0129

[13] S.M. Lundberg and S.-I. Lee, A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[14] M.T. Ribeiro, S. Singh, and C. Guestrin, ”Why Should I Trust You?”: Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. URL: https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf