A Novel Training Method for Deep Neural Networks Leveraging Attribution Graphs

Deep neural networks have achieved remarkable success across a multitude of domains, demonstrating unparalleled capabilities in tasks ranging from image recognition and natural language processing to complex decision-making processes [1]. The increasing complexity and scale of modern deep learning models have enabled them to tackle increasingly intricate problems, pushing the boundaries of artificial intelligence [2]. While the backpropagation algorithm has served as the cornerstone for training these deep networks, its inherent limitations in terms of efficiency, interpretability, and robustness are becoming more apparent as models grow in size and are applied to more critical applications [3]. The computational cost and training time associated with backpropagation can be substantial, hindering the development and deployment of very large models [2]. Furthermore, the opacity of the learned representations and the difficulty in understanding the model's decision-making process pose significant challenges for interpretability [5]. The vulnerability of deep networks trained with backpropagation to adversarial attacks and their sensitivity to noisy data raise concerns about their reliability and robustness in real-world scenarios [6]. To address these challenges, alternative training methodologies are being explored, with the potential to offer improvements in these critical aspects of deep learning.

One promising avenue of research involves the use of attribution graphs, which provide a means to understand the inner workings of neural networks by assigning importance scores to different components of the model or the input data [5]. By visualizing these importance scores in a graph structure, researchers can gain insights into which features or neurons are most influential in the network's predictions [8]. This approach offers the potential to move beyond the global weight updates of backpropagation towards more targeted interventions within the network. This report outlines a research proposal for a novel training method that leverages attribution graphs to perform targeted neuron updates. The objective of this research is to develop a training paradigm that can enhance the efficiency, interpretability, and robustness of deep neural networks compared to traditional backpropagation by selectively updating critical neurons identified through attribution analysis.

Understanding Attribution Graphs

In the context of deep learning, "attribution" refers to the process of assigning importance scores to the inputs or internal components of a neural network with respect to a specific output [7]. Essentially, it is a method of credit assignment that highlights which dimensions of the input data or which elements within the network

had the most significant influence on the model's prediction for a given instance [5]. This can be viewed as determining the effect of an input feature on the prediction function's output, inherently addressing a causal question about the model's behavior [9]. For example, in image classification, attribution methods might highlight the pixels most responsible for the classification of an image as a cat [5].

Building upon the concept of attribution, "attribution graphs" represent a way to visualize and analyze these importance scores within the structure of a graph [8]. In a deep attribution graph, nodes can correspond to inputs, features, or even individual neurons within the network [7]. The edges in these graphs typically denote relationships or connections, weighted by the attribution scores, indicating the strength and direction of influence between the nodes [8]. For instance, in the SUMMIT system, an attribution graph visualizes highly activated neurons as vertices, with edges representing the most influential connections between them, thereby summarizing crucial neuron associations and substructures that contribute to the model's outcomes [8]. Another example is the Deep Attribution Graph for Deep Knowledge Transferability (DEPARA), where nodes represent inputs based on their vectorized attribution maps, and edges denote the relatedness between these inputs as measured by their feature similarity within a pre-trained deep neural network [7].

The significance of attribution graphs lies in their ability to provide interpretability to the often opaque decision-making processes of deep neural networks [5]. By highlighting the input dimensions that are influential to a network's prediction, attribution methods can expose the statistical regularities learned by the model [5]. If these patterns align with human intuition, it can bolster confidence in the model's predictions [5]. Conversely, if attributions reveal that the model is exploiting spurious correlations or violating common sense, they can serve as a valuable debugging tool, allowing for the identification and potential correction of issues in the dataset or model architecture [5]. In fields like drug discovery, the interpretability offered by attribution graphs can increase user trust in the model's output [5]. Furthermore, attribution graphs can reveal not only the importance of individual features but also the relationships and interactions between neurons within the network [8]. This can shed light on potential biases learned by the model and provide a deeper understanding of its internal representations [10]. The diverse applications of attribution graphs, ranging from understanding model behavior in image and text processing to analyzing graph neural networks and even facilitating knowledge transfer between models, underscore their potential as a powerful tool in deep learning research [5].

Limitations of Traditional Backpropagation

Despite its widespread success, traditional backpropagation exhibits several limitations that become increasingly pertinent in the context of modern deep learning [2]. One significant challenge is the issue of vanishing and exploding gradients, particularly in very deep networks [2]. During the backward pass, as gradients are propagated from the output layer back to the earlier layers, they can become progressively smaller (vanishing) or larger (exploding), hindering effective learning in the initial layers of the network [2]. This makes training deep networks difficult and can lead to suboptimal performance.

Another limitation of backpropagation is its inherent sequential nature [2]. The algorithm requires a complete forward pass to compute the error at the output, followed by a backward pass to calculate the gradients for all the weights in the network [12]. This sequential processing can be inefficient, especially in distributed training scenarios where parallel updates could potentially accelerate the learning process [3]. Furthermore, the "locking problem" in backpropagation means that a layer cannot be updated until the subsequent layers have completed their forward and backward computations, preventing independent and potentially more asynchronous learning across different parts of the network [3].

Backpropagation also lacks inherent interpretability [3]. While it effectively adjusts the weights of the network to minimize the error between the predicted and actual outputs, it does not directly provide insights into which input features are important for the model's decisions or how the network arrives at its conclusions [12]. The weight updates are driven by error signals, making it challenging to understand the semantic meaning or the contribution of individual neurons to the overall function of the network.

Moreover, backpropagation can be susceptible to getting stuck in local minima, especially in non-convex loss landscapes that are common in deep learning [3]. The algorithm relies on gradient descent, which can converge to a suboptimal solution if the error surface contains many local minima.

From a biological perspective, backpropagation is also considered implausible [6]. The algorithm requires bidirectional synaptic weight transport, where information needs to flow backward through the same connections used for the forward pass [13]. This is not consistent with the unidirectional flow of information observed in biological neurons [6]. Additionally, the update locking mechanism in backpropagation, where weight updates are delayed until the entire forward and backward passes are complete, is also not biologically realistic [13]. Finally, the performance of backpropagation is highly sensitive to the choice of hyperparameters, such as the learning rate, requiring careful

tuning which can be a time-consuming and computationally expensive process [14].

The multifaceted limitations of backpropagation, spanning optimization challenges, efficiency bottlenecks, lack of inherent interpretability, and biological implausibility, provide a strong motivation for exploring alternative training methods. The potential to address these limitations at a more granular, neuron-level, guided by insights from attribution analysis, could lead to significant advancements in deep learning.

Methods for Generating Attribution Graphs

Several methods exist for generating attribution graphs, each with its own strengths, weaknesses, computational cost, and interpretability [7]. These methods can be broadly categorized into gradient-based, perturbation-based, and propagation-based approaches.

(a) Gradient-based methods: These methods leverage the gradients of the output with respect to the input features to determine their importance.

- **Saliency Maps:** This approach calculates the gradient of the class score of interest with respect to the input pixels [16]. The magnitude of the gradient for each pixel is then visualized as a saliency map, highlighting the regions in the input that have the most influence on the model's prediction [17]. For example, in image classification, a saliency map might show which parts of an image (pixels) were most important for the network to classify it correctly [19]. Gradient-based methods like saliency maps are generally computationally efficient [16]. However, they can suffer from noisy gradients and sensitivity to small perturbations in the input, potentially leading to unreliable explanations [15]. Furthermore, the sign of the gradient, indicating whether increasing the pixel value would increase or decrease the class probability, is not always directly interpretable [16]. A known issue with basic saliency maps is the saturation problem, especially when using ReLU activation functions, where the gradient becomes zero for saturated neurons, potentially masking their importance [16].
- **Integrated Gradients (IG):** To address some of the limitations of basic gradient methods, Integrated Gradients computes the attribution of each input feature by integrating the gradients along a path from a defined baseline input to the current input [21]. The baseline is often chosen as a neutral or zero-value input [23]. By accumulating the gradients along this path, IG aims to provide a more comprehensive attribution that is less sensitive to local gradient saturation [21]. This method satisfies important theoretical axioms such as sensitivity and completeness [22]. Sensitivity ensures that if an input and baseline differ in one feature and result in different predictions, the differing feature gets a non-zero

attribution [24]. Completeness states that the sum of attributions should equal the difference in the model's output between the input and the baseline [22]. While IG offers a more robust attribution compared to simple gradients, its results can be sensitive to the choice of the baseline, and the computational cost is higher due to the integration process, which typically involves multiple gradient computations [21].

(b) Perturbation-based methods: These methods estimate feature importance by observing how the model's output changes when parts of the input are perturbed or removed.

- **LIME (Local Interpretable Model-agnostic Explanations):** LIME works by approximating the behavior of the complex deep learning model with a simpler, interpretable model (like a linear regression or decision tree) in the vicinity of a specific input instance [26]. It does this by generating perturbed samples around the original input, feeding them into the black-box model, and then learning a weighted local model based on how the predictions change [28]. The weights are assigned based on the proximity of the perturbed samples to the original input [29]. The coefficients of the local interpretable model then provide an explanation of the feature importance for that particular prediction [27]. LIME is model-agnostic, meaning it can be applied to any machine learning model [28]. It also provides local explanations, focusing on individual predictions [28]. However, LIME can be computationally expensive, especially for high-dimensional inputs, as it requires generating and classifying many perturbed samples [28]. The stability of the explanations can also depend on the sampling strategy and the choice of the local interpretable model [25].
- **SHAP (SHapley Additive exPlanations):** SHAP is a model-agnostic method based on game theory that aims to explain the output of any machine learning model by calculating the contribution of each feature to the prediction [30]. It considers each feature as a "player" in a game where the "payout" is the prediction, and it calculates the Shapley values, which represent the average marginal contribution of each feature across all possible feature combinations [30]. SHAP provides consistent and interpretable feature importance scores, offering both local explanations for individual predictions and global insights into the overall feature importance [30]. For deep learning models, SHAP often uses specialized explainers like DeepExplainer, which leverages the properties of neural networks to estimate Shapley values more efficiently [32]. Despite its strong theoretical foundation and desirable properties, calculating exact Shapley values can be computationally intensive, especially for complex models and large datasets, as it may involve evaluating the model on many different feature subsets

[30]. Approximations are often used to make it more tractable.

(c) Layer-wise Relevance Propagation (LRP): LRP is a propagation-based method that aims to explain the prediction of a neural network by propagating the relevance score backward through the layers of the network, starting from the output layer [33]. It uses specific local propagation rules at each layer to redistribute the relevance received by a neuron in a higher layer to the neurons in the lower layer that contributed to its activation [15]. This process continues until the input layer is reached, providing a relevance score for each input feature, indicating its contribution to the final prediction [35]. LRP can highlight the positive contributions of input features to the network's classification [36]. It also adheres to a layer-wise conservation property, where the total relevance is conserved as it is propagated backward [33]. LRP is efficient as it typically involves a single backward pass [15]. However, the choice of appropriate propagation rules for different types of layers and network architectures can be crucial and might require careful tuning [34]. The method is also inherently tied to the network's structure, unlike model-agnostic methods [34].

Table 1: Comparison of Attribution Graph Generation Methods

Method	Strengths	Weaknesses	Computational Cost	Interpretability	Relevant Snippets
Saliency Maps	Computationally efficient [16]	Noisy gradients, sensitive to perturbations, sign of gradient not always interpretable, saturation issues [15]	Low	Visual explanation of important regions [16]	[15]
Integrated Gradients	Satisfies sensitivity and completeness axioms [22], implementation invariance [24], strong theoretical justification [24]	Sensitive to baseline choice, higher computational cost than basic gradients [21]	Moderate	Provides feature importance scores relative to a baseline [24]	[15]
LIME	Model-agnostic [28], provides local explanations [28]	Computationally expensive, potential instability due to sampling, explanations can disagree with other methods [25]	High	Provides interpretable local models (e.g., linear regression coefficients) [28]	[26]

SHAP	Model-agnostic [30], consistent and interpretable feature importance scores [30], provides both local and global interpretability [30], handles complex interactions [30]	Computationally expensive, complexity of interpretation can be high, approximation methods often used [25]	High	Quantifies the contribution of each feature to the prediction [32]	[30]
LRP	Highlights positive contributions [36], layer-wise conservation of relevance [33], computationally efficient [34]	Dependent on network architecture, parameter selection for propagation rules can be challenging, potential sensitivity to rules [34]	Low	Provides relevance scores for input features, indicating their contribution to the prediction [35]	[15]

Works cited

1. Deep Learning With Spiking Neurons: Opportunities and Challenges - Frontiers, accessed March 27, 2025, https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2018.00774/full
2. Why we need a better learning algorithm than Backpropagation in Deep Learning - Medium, accessed March 27, 2025, https://medium.com/towards-data-science/why-we-need-a-better-learning-algorithm-than-backpropagation-in-deep-learning-2faa0e81f6b
3. What is Backpropagation Algorithm? Benefits & Drawbacks, accessed March 27, 2025, https://www.deepchecks.com/glossary/backpropagation-algorithm/
4. The Problem with Back-Propagation | by Anthony Repetto | TDS Archive - Medium, accessed March 27, 2025, https://medium.com/towards-data-science/the-problem-with-back-propagation-13aa84aabd71
5. proceedings.neurips.cc, accessed March 27, 2025, https://proceedings.neurips.cc/paper/2020/file/417fbbf2e9d5a28a855a11894b2e795a-Paper.pdf
6. [Discussion] What are the problems of the backpropagation algorithm? - Reddit, accessed March 27, 2025, https://www.reddit.com/r/MachineLearning/comments/70tz1n/discussion_what_are_the_problems_of_the/
7. openaccess.thecvf.com, accessed March 27, 2025, https://openaccess.thecvf.com/content_CVPR_2020/papers/Song_DEPARA_Deep_Attribution_Graph_for_Deep_Knowledge_Transferability_CVPR_2020_paper.pdf
8. SUMMIT: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations, accessed March 27, 2025, https://par.nsf.gov/servlets/purl/10183516
9. Neural Network Attributions: A Causal Perspective - Proceedings of Machine Learning Research, accessed March 27, 2025, http://proceedings.mlr.press/v97/chattopadhyay19a/chattopadhyay19a.pdf
10. Using attribution to decode binding mechanism in neural network models for chemistry | PNAS, accessed March 27, 2025, https://www.pnas.org/doi/10.1073/pnas.1820657116
11. GOAt: Explaining Graph Neural Networks via Graph Output Attribution - arXiv, accessed March 27, 2025, https://arxiv.org/html/2401.14578v1
12. Backpropagation - Engati, accessed March 27, 2025, https://www.engati.com/glossary/back-propagation
13. Learning Without Feedback: Fixed Random Learning Signals Allow for Feedforward Training of Deep Neural Networks - Frontiers, accessed March 27, 2025, https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2021.629892/full
14. Understand the Impact of Learning Rate on Neural Network Performance -

MachineLearningMastery.com, accessed March 27, 2025, https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/
15. A unified view of gradient-based attribution methods for Deep Neural Networks - Heatmapping.org, accessed March 27, 2025, http://www.interpretable-ml.org/nips2017workshop/papers/02.pdf
16. 28 Saliency Maps – Interpretable Machine Learning, accessed March 27, 2025, https://christophm.github.io/interpretable-ml-book/pixel-attribution.html
17. An Introduction to Saliency Maps in Deep Learning - MarkTechPost, accessed March 27, 2025, https://www.marktechpost.com/2022/03/07/an-introduction-to-saliency-maps-in-deep-learning/
18. Saliency map - Wikipedia, accessed March 27, 2025, https://en.wikipedia.org/wiki/Saliency_map
19. What is Saliency Map? - GeeksforGeeks, accessed March 27, 2025, https://www.geeksforgeeks.org/what-is-saliency-map/
20. Explainable machine learning #3: Saliency Maps - YouTube, accessed March 27, 2025, https://www.youtube.com/watch?v=eudf1wQmXnc
21. IG2: Integrated Gradient on Iterative Gradient Path for Feature Attribution - arXiv, accessed March 27, 2025, https://arxiv.org/html/2406.10852v1
22. Integrated Gradients — Alibi 0.9.7.dev0 documentation, accessed March 27, 2025, https://docs.seldon.io/projects/alibi/en/latest/methods/IntegratedGradients.html
23. Integrated gradients | TensorFlow Core, accessed March 27, 2025, https://www.tensorflow.org/tutorials/interpretability/integrated_gradients
24. Interpreting Deep Neural Networks using Integrated Gradients ..., accessed March 27, 2025, https://towardsdatascience.com/interpreting-deep-neural-networks-using-integrated-gradients-f9b8ecdd3c57/
25. Machine learning interpretability with feature attribution - Christian Garbin's personal blog, accessed March 27, 2025, https://cgarbin.github.io/machine-learning-interpretability-feature-attribution/
26. Understand Network Predictions Using LIME - MathWorks, accessed March 27, 2025, https://www.mathworks.com/help/deeplearning/ug/understand-network-predictions-using-lime.html
27. LIME: Local Interpretable Model-Agnostic Explanations - C3 AI, accessed March 27, 2025, https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/
28. From Opaque to Transparent: Understanding Machine Learning ..., accessed March 27, 2025, https://ishwaryasriraman.medium.com/from-opaque-to-transparent-understanding-machine-learning-models-with-lime-3f2a2d147642
29. LIME: explain Machine Learning predictions | by Giorgio Visani | TDS Archive - Medium, accessed March 27, 2025,

https://medium.com/towards-data-science/lime-explain-machine-learning-predictions-af8f18189bfe
30. How to explain neural networks using SHAP | Your Data Teacher, accessed March 27, 2025, https://www.yourdatateacher.com/2021/05/17/how-to-explain-neural-networks-using-shap/
31. Deep Learning Model Explainability with SHAP - Paperspace Blog, accessed March 27, 2025, https://blog.paperspace.com/deep-learning-model-interpretability-with-shap/
32. Can we use SHAP values to explain the performance of a Neural Network ? | ResearchGate, accessed March 27, 2025, https://www.researchgate.net/post/Can_we_use_SHAP_values_to_explain_the_performance_of_a_Neural_Network
33. 10 Layer-Wise Relevance Propagation: An Overview - Fraunhofer Heinrich-Hertz-Institut, accessed March 27, 2025, https://iphome.hhi.de/samek/pdf/MonXAI19.pdf
34. (PDF) Layer-Wise Relevance Propagation: An Overview, accessed March 27, 2025, https://www.researchgate.net/publication/335708351_Layer-Wise_Relevance_Propagation_An_Overview
35. Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers, accessed March 27, 2025, https://arxiv.org/abs/1604.00825
36. Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer's Disease Classification, accessed March 27, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC6685087/
37. Gradient-Based Attribution Methods - CGL @ ETHZ, accessed March 27, 2025, https://cgl.ethz.ch/Downloads/Publications/Papers/2019/Anc19c/Anc19c.pdf
38. Evaluating Attribution Methods in Machine Learning Interpretability, accessed March 27, 2025, https://par.nsf.gov/servlets/purl/10385266
39. Explaining Deep Learning Models for Tabular Data Using Layer-Wise Relevance Propagation - MDPI, accessed March 27, 2025, https://www.mdpi.com/2076-3417/12/1/136