Explainability: A Comprehensive Comparison of Explainability Techniques in the Captum Library

Machine Learning for NLP II

Thomas Wyndham Bush(ID:248201), Michael Awanah Atamakira (ID: 247099)

January 2024

Abstract

The growing field of machine learning (ML) has ushered in a new era of technological advancement, with deep learning models, particularly neural networks, leading the charge in complex decision-making tasks. However, the intrinsic opaqueness of these models has raised significant concerns regarding their interpretability.

This paper presents a comprehensive comparative study of various explainability techniques implemented in the Captum library, an open-source project designed for model interpretability in PyTorch. We systematically evaluate and contrast several methods, including Layer-Wise Relevance Propagation (LRP), Gradient-Based Attribution, Integrated Gradients, and Feature Ablation applied to a Convolutional Neural Network Model trained on the CUB-200-2011 dataset (1). In addition, we run a qualitative evaluation using Gradient-Based Attribution on three different model to investigate on which sections of the image they focus the most.

We implement our code on Google Colab and make our code available on GitHub.[1]

1 Introduction

Explainable AI is a broad term for defining a series of methods that are used for obtaining explanations about the outcomes of an AI system. The field is relatively new, but it is becoming crucial with the development of more and more complex system that are difficult to understand, and see how they reach a specific outcome (2). Since the field is still new, it can be beneficial to review some of the terminology that it uses:

Interpretable Machine Learning: interpretability refers to the implicit capacity of a model to explain its reasoning. This type of machine learning techniques are called white-box. An example of interpretable machine learning is linear regression (2).

Explainability: it refers to the process by which we can obtain a justification for an output y using an external method over the model used for obtaining the output (3).

Transparency: is the property of a model that serves as a proxy for indicating how much is it understandable (3).

In our case we needed to use explainability to obtain justification for the decision of a Convolutional Neural Network classifier that has been trained to correctly classify a bird into 200 different classes.

2 Model Building

For the scope of the project we trained a CNN model to classify the type of bird given an image from the CUB-200-2011 dataset (1), which consists of: 200 different classes and a total of: $11,788$ images. In addition, the dataset have a series of ground truths annotations for each image, for our test we focused on the bounding box annotation.

[1] https://github.com/Thomasbush9/Explainable_AI

2.1 Data Processing

The dataset came with an annotation for the train/test split, after having divded the dataset we obtained a training set of: images, and a test set of: images.

Each image has been pre-processed before the training phase. The training images have been resized (size=224, 224), normalized, and augmented by using: a random horizontal flip, random rotation (8 degrees), and a random crop. For the test images we have just applied the normalisation and the resize. All the methods applied are from the *torchvision* library.

2.2 Transfer Learning and Tuning:

To correctly classify the images we have decided to use a pre-trained model: ResNet101 from Pytorch library(4).

The model has been fine tuned for our task with the addition of: a dropout layer (0.5) to limit overfitting, a linear layer followed by the ReLu activation function, and an output layer that gives as output the predicted class. The choice to add an additional layer with the ReLu activation function was due to the necessity of capturing non-linear features from the images.

The model has been trained using Stochastic Gradient Descent as optimizer, with a learning rate of: 0.008, momentum of: 0.9. The loss function used is Categorical Cross Entropy loss weighted to balance the class distribution. Finally we have added a scheduler for the regularisation of the learning rate of the optimizer with a patience of 3 and a factor of 0.1.

The training process consisted of a k-fold cross validation process, with 5 folds each of one with 25 epochs and a batch size of 32. The folding process has been done using a stratification technique from *SkLearn* library.

2.3 Results

The training process gave the following final results for the five validation folds are visibile in the table 1.

Fold	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
1	0.04	0.99	1.37	0.69
2	0.02	0.99	1.28	0.70
3	0.02	0.99	1.17	0.72
4	0.02	0.99	1.18	0.71
5	0.04	0.99	1.15	0.72

Table 1: Training Results

The model, then, has been tested on the test set, the results are visible in table 2

Test Loss	Test Accuracy
1.16	0.73

Table 2: Test Set Results

3 Explainability

3.1 Integrated Gradient Based Attribution

Integrated Gradient-based attribution computes the average gradient while the input varies along a linear pat from a baseline (usually a black image) to the original input (5).

This method calculate the gradients of the output with respect to the input, which effectively indicates how much each pixel in the input image contributes to the final decision made by the CNN. This technique highlights areas in the input image that are most significant for the model's predictions.

The right side of 1 featured a Gradient-Based Attribution map, used to understand the pixels in the input image that greatly influenced the neural network's prediction. This approach computed the gradient of the model's output with respect to the input image, which essentially measured how much each pixel's value affected the prediction score for the given class. This gradient was then used to create a visualization that highlighted the important regions for the model's decision. In the map, areas of high in-

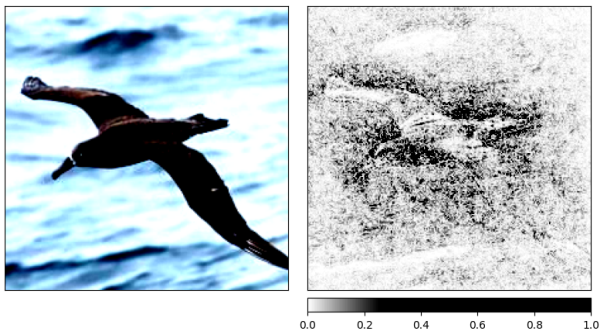

Figure 1: Gradient Based Attribution

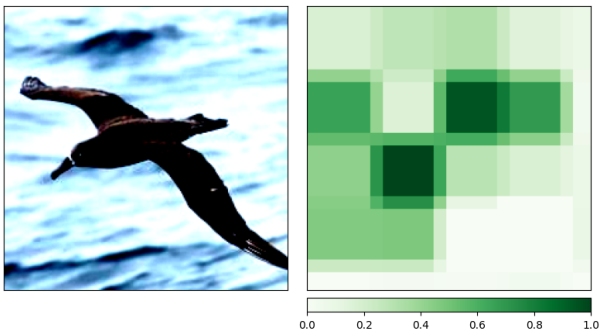

Figure 2: Occlusion-based Attribution

tensity (white or light areas) represented pixels with higher gradients and, therefore, are regions that the model considers important for making its prediction. Conversely, darker areas indicate lower gradients and had less influence on the model's output. The visualization created understanding for the sensitivity of the output to the input image on per-pixel basis.

3.2 Occlusion-based Attribution

Occlusion-based attribution is used to understand how different regions of the input data affect the network's predictions. In this method, parts of the input image are systematically occluded or masked, and the effect of each occlusion on the output is observed. By analyzing the change in the model's output one can infer the importance of the occluded region in the decision-making process.

We have applied the Occlusion function from the Captum library to run a sliding window over the image and we have computed how much each specific region contributes to the classifier's decision. In the beginning, we used a smaller size (3, 8, 8) for the window. After the first analysis, we increased the window's size to (3, 50, 50) to further see which areas of the image are crucial for the classification task.

On the left side is the original image, a bird in flight—the input image that the neural network model processed. On the right side is the occlusion-based attribution map, which is a visual representation used to interpret the decision-making process of the neural network. Parts of the input image was systematically occluded, or covered up, while the model's output was observed for changes. The map on the right shows different shades of green, where each square represents a region of the original image that has been occluded during the process. The color intensity indicates the impact of the occlusion on the model's output: darker shades signify regions where the occlusion caused a significant decrease in the model's confidence for its prediction, suggesting that these areas are signficant for the decision-making process. Lighter shades corresponded to regions where occlusion had a lesser impact on the model's output, indicating lower importance.

We con observe that the body of the bird corresponds with the area that when occluded cause the most decrease in the model confidence for the classification.

3.3 Feature Ablation

Feature ablation for inspecting CNN decisions is a technique used in machine learning to understand and evaluate the importance of input features in a CNN model's decision-making process. In this method, individual features (or groups of features) are systematically removed or "ablated" from the input data, and the impact of this removal on the model's performance is observed. By analyzing how the predictions change when certain features are absent, one can infer the significance of those features in the model's decision-making.

In the figure 3 we can observe how the cluster of

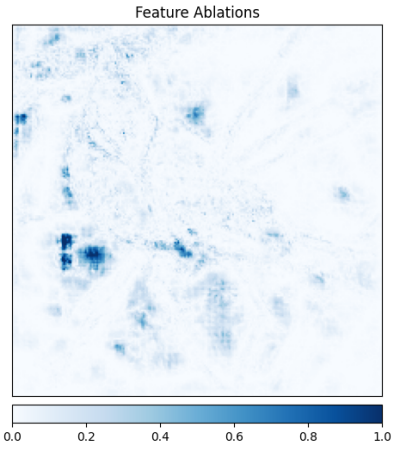

Figure 3: Feature Ablation

features containing the head of the bird play a crucial role for the model in producing its final output.

3.4 LRP-based Attribution

Layer-Wise Relevance Propagation (LRP) is an interpretability method in machine learning that explains the decision-making process of neural networks. It works by sending the prediction output back through the network's layers and giving each neuron a relevance score that shows how much it contributed to the final decision (5). This process adheres to the principle of conservation to ensure that the total relevance at the output layer is exactly redistributed back through the network to the input layer. As a result, each input feature (like a pixel in an image) is assigned a relevance score, indicating how significantly it influences the model's output. This method is particularly useful in complex models, where understanding why a model made a certain decision is as crucial as the decision itself.

In figure 3, on the left, there is the original photograph of a flying bird. This is the input fed into the neural network model for classification or detection.

On the right, we see a heatmap generated by the LRP algorithm. LRP explained the decisions of the neural network by assigning a relevance score to each pixel (or feature) of the input image, indicating its contribution to the model's output. The heatmap used a monochromatic scale with different shades of green to white, with darker shades perhaps indicating areas of higher relevance.

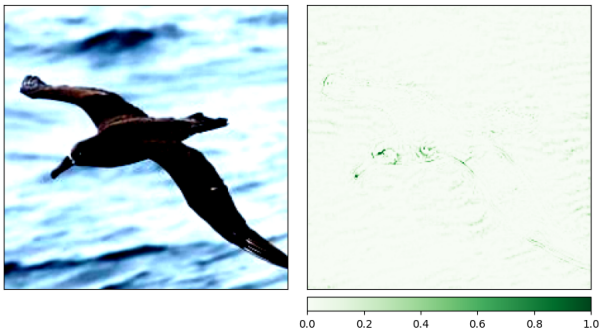

Figure 4: LRP-based Attribution

The LRP heatmap was crucial for understanding the parts of the input image that are most important for the model's decision. The LRP analysis ideally showed higher relevance scores on the pixels that make up the bird's shape(face and beak). This was helpful for us to verify that the model is focusing on the correct features for its decisions and not on background noise or irrelevant artifacts in the image.

4 Quantitative Evaluation:

To further test the predictions of our model, we decided to conduct a quantitative evaluation by testing how much attention our model focuses on the area of the image that contains the bird. Ideally, a model that correctly uses the features of the bird to classify it should focus a significant percentage of its attention on the bird, and not on the background (even though the background can give some important information about the habitat of the bird). To test whether our model does so, we decided to utilise the bounding box provided by the dataset as ground truth explanations, and calculate how much of the total attention of the model is clustered within these boxes.

4.1 Methods

We decided to conduct the analysis using the Integrated Gradient Attribution as a method for measuring the attention of the model. Our choice is due to the fact that it is not necessary to instrument the network to use this method and it is fairly efficient to apply, compared to perturbation-based methods (5). In addition, Integrated Gradients seemed to us the natural choice for a quantitative evaluation as it respects the axioms of: sensitivity and implementation invariance (6).

By computing which pixels of the image have the most effects on the final output it is possible to understand which area of the image have been used by a model to make its prediction, and by integrating this computation with the bounding box area it is possible to understand how much the area containing the bird it is used by the model in relation to its output.

We run the quantitative evaluation by selecting a random sample of images from the test set (size=600), which accounts for the 10% of the test set. To further test our model we compared its results with other two models: an untrained model, a baseline model (resnet50). Each image in the sample set has been associated with its corresponding bounding box, which has been scaled to correctly match the transformed image used as input for the model (size= 224, 224).

For each image the model made produced the class prediction, from this prediction we applied the integrated gradients attribution to the model. The attributions of each model has been summed across the three different color channels and normalised to focus on the intensity, finally, the total attribution has been obtained by the sum of the attributions. We have computed the bounding box total attribution in the same manner, but summing just the attributions within the coordinates of the bounding box. In the end the percentage of the integrated gradients attribution for that image has been calculated and stored.

At the end of the iterative process we have calculated the: mean, standard deviation and median of the percentage of integrated gradients attribution within the bounding box for each one of the three models. Then, we have compared each model's mean with the others by running an ANOVA test from the *scipy* library, to see if the average percentages of the attributions differed significantly from each others.

Figure 5: Example of overlapping between bounding box and attribution gradient

4.2 Results

From the results in Table 3 we can observe that the trained model focuses on average 60.65% of its attention within the bounding box, which is significantly more than what the other two models do. This difference has ended up being significant from the result of an ANOVA test (F-value: 208.22, p-value ¡ 0.05).

Model:	Mean:	Std:	Median:
Trained	60.65%	17.95	61.85%
Untrained	39.70%	19.63	36.03%
Baseline	44.01%	18.68	42.39%

Table 3: Results quantitative evaluation

To finalise our analysis we compared the accuracy and confidence for the three models' predictions on the sample set. The results from this test, shown in 4, shows that even if the accuracy between the three models is comparable, the confidence with which they make their prediction seems to be correlated to the percentage of attention that they cluster within the

bounding box. This suggest that the predictions made by the trained model have more explainability weight, as they can be justified by actual features of the bird (inser quote) and not by mere noise of the image.

Model:	Loss:	Accuracy:
Trained	1.25	70.00%
Untrained	24.75	70.03%
Baseline	5.39	71.00%

Table 4: Results accuracy comparison

5 Conclusion

Given the fact that the model performances are often correlate negatively with interpretability (2), it is tempting to chose a complex model over an interpretable one. Nonetheless, in many real-life scenario the reason that has brought to a decision is important nearly as much as the decision. Thus, developing techniques that allow to explain how complex models an output is extremely important.

In our study, we focused on four methods that can be used to understand how a CNN model classifies an image. By the application of these methods on our model we have gained a significant insight about the features of the image that our model considers important for the output.

To further asses whether our model focuses on the correct features of the image for making its decisions, we run a quantitative evaluation using the Integrated Gradient Based Attribution and calculate how much of the model attention was focused on the bounding box containing the bird in the image. Our results suggest that our model focuses a significant quantity of its attention on the bird. We have compared our model's results to untrained models, and we have observed how these models focus more on background part of the image. This result indicates that the training has been effective and our model has learned which parts of the image are more informative. We have also noticed that the confidence in a prediction was positive correlated with the percentage of the attention focused inside the bounding box of the bird, highlighting the quality of our model's predictions.

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011.

[2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. 58:82–115. ISSN 1566-2535. doi: 10.1016/J.INFFUS.2019.12.012. Publisher: Elsevier.

[3] Velibor Božić. *Explainable Artificial Intelligence (XAI): Enhancing Transparency and Trust in AI Systems*. doi: 10.13140/RG.2.2.23444.48007.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning pdf.

[5] Marco Ancona, Cengiz Öztireli, Enea Ceolini, and Markus Gross. TOWARDS BETTER UNDERSTANDING OF GRADIENT-BASED ATTRIBUTION METHODS FOR DEEP NEURAL NETWORKS.

[6] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep net-

works. URL https://proceedings.mlr.press/v70/sundararajan17a.html. ISSN: 2640-3498 Pages: 3319-3328.