Explainable Artificial Intelligence

Omar Bayoumi 1747042 (Student)

ARTICLE INFO

Keywords:
Explainable AI
Self-Interpretable Neural Network
Machine Learning
Neural Networks

ABSTRACT

Artificial intelligence has been used more and more in recent years. Every year powerful new systems are being produced that can simplify very complex problems and that can be useful in finding solutions in areas that were thought to have none. However, there remains a problem underlying neural networks, namely, how they work. We can understand the structure that a model has but it is difficult to understand the operations that take place within it. This is because most models are like black boxes in which input goes in and we trust what the black box gives as output. To solve these problems, Explainable Artificial Intelligence (XAI) was born whose task is to explain the internal mechanisms of networks. In this paper, the main XAI post-hoc methods, which are based on the analysis of the model output, will be analyzed, but self-interpretable models will also be described, which are models that due to their structure allow us to get information without the need to analyze the output.

1. Introduction

Over the past decade, artificial intelligence and machine learning systems have achieved optimal performance in fields that used to be unthinkable such as natural language processing and computer vision. Improvements in this field have been achieved through the steady increase in available information and improved hardware combined with new optimization techniques. The problematic issue with these powerful systems is their complexity, in fact, deep neural networks (DNN) have many parameters and as a result, are difficult to understand and interpret. We have systems available that allow us to understand learning performance but very often deep learning models fail to capture representations from the data that a human might consider important. Explaining why these networks decide to discard certain representations and choose others requires knowledge of the inner workings of DNNs. In addition, researchers to obtain state-of-the-art results in different domains neglect the ability to interpret AI decisions. To try to solve these problems explainable artificial intelligence (XAI, Gunning and Aha (2019)) was born, whose goal is to explain the outputs of neural networks, to understand why a DNN produced that output, for example, in the case of cancer detection (Hauser and et al. (2022)). In this paper, we will present an overview of the principal methods of XAI discussing their pros and cons. In particular, explainable artificial intelligence, its uses, and objectives will be initially described. Then it will go on to analyze what are the most important features that a model needs to detect in a given input. Next, the functioning of the neuron, the element behind neural networks, will be analyzed and different techniques for analyzing neurons will be described. After that, the internal mechanisms of neural networks will be analyzed through methods named analogies and recourse. Next, metrics for evaluating the explanations will be discussed. Finally, self-interpretable deep learning, an alternative to classical methods that rely on post-hoc analysis, will be explained and analyzed.

2. Explainable AI

The main feature of self-explainable models is to understand the why of that output given an input. A typical example of these models is the decision tree (de Ville (2013)) in which it is possible to understand the output generated by following the path that led to that response. These kinds of techniques are called Self-Explainable AI or model-intrinsic. However, many models cannot trace back why a particular response was made, these types of models are seen as Black-Box AI, where given an input you get an output by trusting the internal mechanisms of the model. Techniques called post-hoc are used to study these models. XAI aims to turn these Black-Box AI models into XAI agents. Another important use of the XAI is also for debugging. Currently, the explanations provided are used to change the hyperparameters of the model using a try-and-error mechanism. XAI models can be divided into two types, Local and Global explainers (Munoz, da Costa, Modenesi and Koshiyama (2023)). The former focuses on understanding the behavior of AI algorithms at a low hierarchical level. They consist most often of a single observation, which is why local methods provide metrics related to how each feature contributes to the final prediction output by the model. Global methods, on the other hand, seek to understand the behavior of AI algorithms at a high hierarchical level. They provide insight into how features contribute to a prediction over the whole dataset. It is possible to further distinguish the methods into model-agnostic and model-specific (Ai and Narayanan.R (2021)). Model-agnostic methods aren't limited by the characteristics of an AI algorithm and can be used to reveal the decision process of any black-box model. Model-specific methods, on the other hand, are tailored to analyze specific algorithms' characteristics and provide a more detailed understanding of how a certain algorithm improves model decision-making. However, these methods require a higher degree of skill from developers to analyze the models.

3. Most Important Feature

To understand how algorithms within models work, it is necessary to identify the most important features, this is because it allows us to understand which input feature the model focuses on to produce the current output. The most commonly used methods are LIME (Ribeiro, Singh and Guestrin (2016)) and SHAP (Lundberg and Lee (2017)). The process involves taking an input, and perturbing it, such as by removing words and collecting predictions. Then a weight is assigned to each perturbation based on the number of shared features, called proximity. This weight changes according to the algorithm, LIME rewards samples that share most of the features, and SHAP rewards both samples that share a lot of features and samples that share few features. Then the dataset is used to train a linear model to approximate the model behavior around the current input, and after that, the model weights are used as feature scores. The disadvantage of these algorithms is their lack of flexibility, for example, they are not aware of sentences and most of the importance is assigned to the endings words.

4. Understanding the neurons

Neural networks consist of neurons that learn information during the training process. Consequently, it is important to understand what they are learning. To understand this different techniques will be analyzed.

4.1. Optimization-based techniques

This set of techniques is based on the idea of generating an input that maximizes the activation of a given neuron and then trying to reconstruct the pattern. However, optimization-based techniques have problems including lack of diversity and abstract explanations.

4.2. Database-based techniques

The idea behind these kinds of techniques is to select from a dataset the images that maximize the activation of a given neuron and then try to reconstruct the pattern. One of the main problems with this type of technique is the difficulty of separating things causing behavior from things that merely correlate with the causes

4.3. Concept-based techniques

The idea behind this type of technique is similar to the previous ones but in this case, concepts are used, which represent a set of samples joined by a label. Specifically, concept annotated datasets are used, which are datasets in which there is a segmentation map. Then the concept that maximizes neuron activation is selected. An example of a concept-based technique is X-CHAR (Jeyakumar, Sarker, Garcia and Srivastava (2023)), which offers explanations in the form of human-understandable, high-level concepts while maintaining the robust performance of end-to-end deep learning models for time series data. The main problem with all these techniques is that they provide a tiny view of the model's behavior because they only look at maximal activation but there are other factors to take into account.

5. Analogies and recourse

To try to explain the internal mechanisms of the model, it is useful to try to understand what similar samples the model learned during the training process. To understand this there are different techniques, one of which is called explanation by analogies. The goal of this technique is to extract similar samples from the current input where the networks act in the same way. It is possible to use the K-NN (K-nearest neighbor) algorithm in which similar samples are searched for, as shown in Figure 1. Another alternative is to use the enhanced K-NN in which a weight is assigned to features based on feature attribution (Kenny and Keane (2019)) or the gradient (Kenny, Ford, Quinn and Keane (2021)). It is also possible to extract feature-weights from black-box models to search explanatory nearest-neighbors for test instances (Kenny, Delaney and Keane (2022)).

Figure 1: This figure shows an example of similar images to the input image

Recourse is another kind of explanation. By taking a sample we want to understand what we can change in the initial input to get a different prediction. This new sample is called counterfactual (Figure 2).

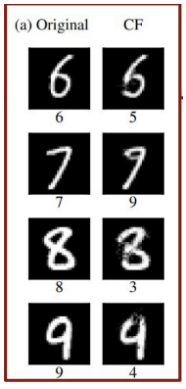

Figure 2: Example of counterfactuals

The techniques are mostly based on optimization processes called generative algorithms. These types of methods try to generate and then maximize metrics such as proximity to the input (Wachter, Mittelstadt and Russell (2018)), diversity (Mothilal, Sharma and Tan (2020)), and actionability (Poyiadzi, Sokol, Santos-Rodriguez, De Bie and Flach (2020)). Usually, one counterfactual is insufficient to obtain a good explanation, more than one must be used. These types of generative approaches, however, appear in the form of external tools to be applied to the network. It is possible, however, to have mechanisms internal to the network that

have the same operation, in this case, we talk about self-explainable neural network. Starting from the input image, images similar to it are searched and this mechanism is embedded in the last layer. An example of this type of Recourse is memory wrap (La Rosa, Capobianco and Nardi (2023)). This method works by giving input to the network a random training sample and then the network tries to find similar examples from its memory set and returns them as expressions of example and counterfactual.

6. Evaluating explanations

The methods described above are used to obtain explanations of what is happening within the network. But there is also a need to use metrics to evaluate the explanations, to see if they are valid or not. There are two types of evaluation, the first is the user-based metrics or human evaluation, which consists of taking one of the obtained explanations and letting a human being see it, and then evaluating it qualitatively and quantitatively (Vilone and Longo (2021)). Finally, there are heuristic-based methods which are mathematical entities that quantify desirable properties such as faithfulness, robustness, sensibility, speed, and optimality.

6.1. Feature attribution

Feature attribution methods seek to assign an importance value to each feature depending on its contribution to the prediction.

Monotonicity (phi Nguyen and Martínez (2020)): is defined as the Spearman's correlation between the feature's absolute performance measure of interest and corresponding expectations.

Implementation invariance (Sundararajan, Taly and Yan (2017)): states that, for any given input space, even with different architectures, if the output is the same the explanations must be similar.

Sensitivity to input perturbation (Yeh, Hsieh, Suggala, Inouye and Ravikumar (2019)): some features of the input dataset are removed, masked or altered and the explanations generated by a method for explainability from the model trained on both the original and modified inputs are compared.

6.2. Explanation by Examples

Explanation by examples methods provide insight into the model using representative examples, or high-level concepts (Chen, Li, Tao, Barnett, Su and Rudin (2019)). Given a prediction an explanation by examples method explains the prediction by providing examples of the same prediction (Guidotti, Monreale, Matwin and Pedreschi (2020))

6.3. Counterfactuals

In the counterfactual evaluation process, we try to understand how the output of the model might change if we change some characteristics of the input. The work of Ge and et al. (2021) show an example of a metric for counterfactuals that consists of using validity, to measure degrees of prediction switches, and proximity to measure degrees of perturbations. Validity is defined as the ratio of the counterfactuals that have the desired outputs over the total number of data points, while proximity is defined as the average edit distance between original inputs and the corresponding counterfactual.

6.4. Dataset

This type of evaluation process consists of creating synthetic data with label explanation. In this way, it can be understood whether the model is learning the correct information. However, one of the disadvantages of benchmark datasets is that they don't allow us to understand whether an explanation is correct or not, we can only evaluate their usefulness.

7. Self-interpretable Deep learning

Most of the methods seen so far are based on post-hoc analysis but these very often are not sufficient to adequately describe a model, this is especially the case when dealing with complex data or architectures. In this case, it is necessary to have an alternative. One possible choice is to use variational autoencoders (VAE) since the latent space encodes the extracted features used to obtain the final prediction. Also, by imposing constraints on the latent space it is possible to make the neural network self-explainable. To interpret the latent space, Higgins and et al. (2016) propose β-VAE, a modified VAE through the introduction of an adjustable hyperparameters β that balances latent channel capacity and independence constraints with reconstruction accuracy. Another study of Tan, Gao, Khan and Guan (2022) shows that we can use the fact that filters in a convolutional neural network are activated by certain patterns and in this way it is possible to organize an architecture such that similar filters are clustered together. This is because some neurons can be activated by particular geometries or patterns.

Alvarez-Melis and Jaakkola (2018) propose an alternative technique that is Self-Explainable Neural Networks (SENN) which are a particular class of neural networks that are interpretable by design. The architecture is shown in Figure 3.

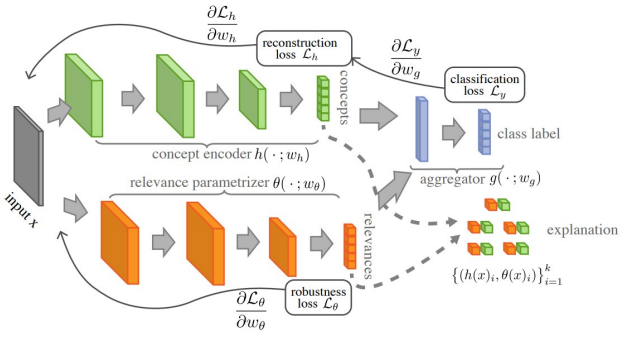

Figure 3: SENN architecture

The idea behind these models is to have a network composed of two different substructures, the first is a concept encoder that encodes concepts, and the second is a

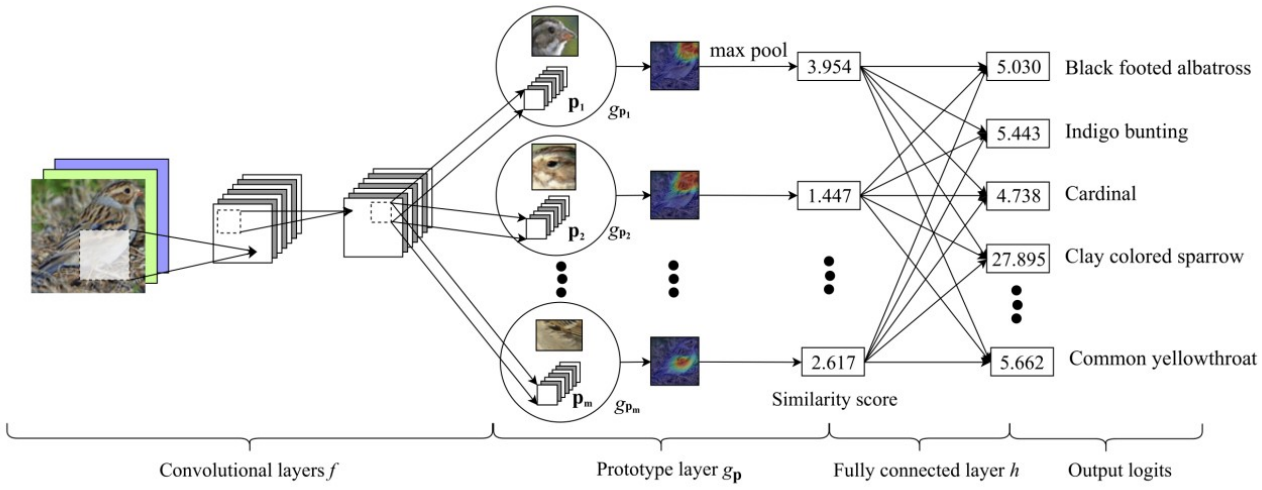

Figure 4: ProtoPNet architecture

relevance calculator estimator that allows to calculate the relevance of an input concerning these concepts. These two pieces of information are merged through an aggregator and then passed to a linear layer. Usually, the majority of self-interpretable models work on the linear layers because are highly interpretable.

One of the most important architectures in the field of self-interpretable deep learning is ProtoPNet (Chen et al. (2019)), which modifies the architecture of SENNs by removing the decoder and replacing it with a prototype layer, trained using image patch similarities. The architecture is shown in Figure 4.

The idea is to learn N prototypes for each class. Then the final prediction is made using the L2 similarities between the latent representation of an image and the learned prototypes. One of the problems with this method is that the prototypes often learn very similar things because of their large number.

Another approach is concept whitening (Chen, Bei and Rudin (2020)) which is based on organizing latent space using already labeled concepts. This method consists of using a batch whitening layer that is mean-centered, normalized, and correlated. It is then rotated with a trainable rotation matrix to align the representation with the concept. This layer is then applied to other layers of the neural network substituting the classical batch normalization layer.

XAI is used in various fields such as natural language processing, reinforcement learning, and graph neural networks. The goal of reinforcement learning is to train an agent to do actions through a reward maximization process. The XAI in this area is to understand the important parts of a state and to understand why the agent performs certain actions. An example is PW-Net (Kenny, Tucker and Shah (2023)) whose architecture is very similar to ProtoPNet but instead of patches, it uses states. In addition, weights and states are defined manually.

Graph neural networks (GNNs) are special types of neural networks that work with graph data types and are used to perform node classification, graph classification, and other tasks. The XAI again has the task of understanding the inner workings of these networks and Ragno, La Rosa and Capobianco (2022) propose an example of application. The idea is to consider images as particular graph types and replace 2D convolution layers with graph ones. In particular, ProtoPNet prototypes are adapted so that they can represent node embeddings and are capable of identifying relevant class-aware motifs. The model then uses the prototypes to generate the prediction and, subsequently, they can be used to extract an explanation about the behavior of the model.

When using high-dimensional graphs, pooling can be used to compress representations and identify substructures. Pooling can be done through either node clustering or node filtering. MemPool (Khasahmadi, Hassani, Moradi, Lee and Morris (2020)) pools nodes by clustering them with a memory that keeps track of the cluster centroids. The idea is to constrain the clustered space to identify class-aware relevant substructures. Next, nodes are assigned to K+1 clusters, one for each class and one for unimportant nodes.

8. Conclusion

A better understanding of how neural networks work would allow us to make the most of them and thus enable us to improve them more and more. This is the goal set by XAI, and in this paper, we have analyzed the main methods both with regard to post-hoc analysis such as LIME and SHAP that are effective because of their low complexity but we have also shown alternate methods known as self-interpretable neural networks and analyzed the advantages and disadvantages of each method.

References

Ai, Q., Narayanan.R, L., 2021. Model-agnostic vs. model-intrinsic interpretability for explainable product search, in: Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management, ACM. URL: http://dx.doi.org/10.1145/3459637.3482276, doi:10.1145/3459637.3482276.

Alvarez-Melis, D., Jaakkola, T.S., 2018. Towards robust interpretability with self-explaining neural networks. arXiv:1806.07538.

Chen, C., Li, O., Tao, C., Barnett, A.J., Su, J., Rudin, C., 2019. This looks like that: Deep learning for interpretable image recognition. arXiv:1806.10574.

Chen, Z., Bei, Y., Rudin, C., 2020. Concept whitening for interpretable image recognition. Nature Machine Intelligence 2, 772–782. URL: http://dx.doi.org/10.1038/s42256-020-00265-z, doi:10.1038/s42256-020-00265-z.

Ge, Y., et al., 2021. Counterfactual evaluation for explainable ai. arXiv:2109.01962.

Guidotti, R., Monreale, A., Matwin, S., Pedreschi, D., 2020. Black box explanation by learning image exemplars in the latent feature space. arXiv:2002.03746.

Gunning, D., Aha, D.W., 2019. Darpa's explainable artificial intelligence program. AI Magazine 40, 44–58. URL: https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v40i2.2850, doi:https://doi.org/10.1609/aimag.v40i2.2850, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1609/aimag.v40i2.2850.

Hauser, K., et al., 2022. Explainable artificial intelligence in skin cancer recognition: A systematic review. European Journal of Cancer 167, 54–69. URL: https://www.sciencedirect.com/science/article/pii/S095980492200123X, doi:https://doi.org/10.1016/j.ejca.2022.02.025.

Higgins, I., et al., 2016. beta-vae: Learning basic visual concepts with a constrained variational framework, in: International Conference on Learning Representations. URL: https://api.semanticscholar.org/CorpusID:46798026.

Jeyakumar, J.V., Sarker, A., Garcia, L.A., Srivastava, M., 2023. X-char: A concept-based explainable complex human activity recognition model. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 7. URL: https://doi.org/10.1145/3580804, doi:10.1145/3580804.

Kenny, E., Keane, M., 2019. Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ann-cbr twins for xai .

Kenny, E.M., Delaney, E.D., Keane, M.T., 2022. Advancing nearest neighbor explanation-by-example with critical classification regions. URL: https://openreview.net/forum?id=sBT5nxwt18Q.

Kenny, E.M., Ford, C., Quinn, M., Keane, M.T., 2021. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. Artificial Intelligence 294, 103459. URL: https://www.sciencedirect.com/science/article/pii/S0004370221000102, doi:https://doi.org/10.1016/j.artint.2021.103459.

Kenny, E.M., Tucker, M., Shah, J., 2023. Towards interpretable deep reinforcement learning with human-friendly prototypes, in: The Eleventh International Conference on Learning Representations. URL: https://openreview.net/forum?id=hWwY_Jq0xsN.

Khasahmadi, A.H., Hassani, K., Moradi, P., Lee, L., Morris, Q., 2020. Memory-based graph networks. arXiv:2002.09518.

La Rosa, B., Capobianco, R., Nardi, D., 2023. A self-interpretable module for deep image classification on small data. Applied Intelligence 53, 9115–9147. URL: https://doi.org/10.1007/s10489-022-03886-6, doi:10.1007/s10489-022-03886-6.

Lundberg, S., Lee, S.I., 2017. A unified approach to interpreting model predictions. arXiv:1705.07874.

Mothilal, R.K., Sharma, A., Tan, C., 2020. Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, ACM. URL: http://dx.doi.org/10.1145/3351095.3372850, doi:10.1145/3351095.3372850.

Munoz, C., da Costa, K., Modenesi, B., Koshiyama, A., 2023. Local and global explainability metrics for machine learning predictions. arXiv:2302.12094.

phi Nguyen, A., Martínez, M.R., 2020. On quantitative aspects of model interpretability. arXiv:2007.07584.

Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P., 2020. Face: Feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, Association for Computing Machinery, New York, NY, USA. p. 344–350. URL: https://doi.org/10.1145/3375627.3375850, doi:10.1145/3375627.3375850.

Ragno, A., La Rosa, B., Capobianco, R., 2022. Prototype-based interpretable graph neural networks. IEEE Transactions on Artificial Intelligence , 1–11doi:10.1109/TAI.2022.3222618.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "why should i trust you?": Explaining the predictions of any classifier. arXiv:1602.04938.

Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks, in: Precup, D., Teh, Y.W. (Eds.), Proceedings of the 34th International Conference on Machine Learning, PMLR. pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html.

Tan, R., Gao, L., Khan, N., Guan, L., 2022. Interpretable artificial intelligence through locality guided neural networks. Neural Networks 155, 58–73. URL: https://www.sciencedirect.com/science/article/pii/S0893608022003094, doi:https://doi.org/10.1016/j.neunet.2022.08.009.

de Ville, B., 2013. Decision trees. WIREs Computational Statistics 5, 448–455. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1278, doi:https://doi.org/10.1002/wics.1278, arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1278.

Vilone, G., Longo, L., 2021. Notions of explainability and evaluation approaches for explainable artificial intelligence. Information Fusion 76, 89–106. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001093, doi:https://doi.org/10.1016/j.inffus.2021.05.009.

Wachter, S., Mittelstadt, B., Russell, C., 2018. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. arXiv:1711.00399.

Yeh, C.K., Hsieh, C.Y., Suggala, A., Inouye, D.I., Ravikumar, P.K., 2019. On the (in)fidelity and sensitivity of explanations, in: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/a7471fdc77b3435276507cc8f2dc2569-Paper.pdf.