Interpretable Image Recognition

Amir Akhavan Rahnama
Department of Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
arahnama@kth.se

Abstract

Machine learning models continue to have improved accuracy for image recognition tasks. While these models can be accurate, their inner working is opaque. This can introduce many challenges in adapting, comparing and improving them. In this report, a review of studies that add transparency to image recognition models are discussed.

1 Introduction

There is no consent on a definition to interpretability, or equivalently what is interpretable or not. However, in this report, we follow the definition of Miller (2017): " Interpretability is the degree to which a human can understand the cause of a decision is". In the case of a Machine Learning model, an interpretable model is a model in which its parameters and/or internal logic can be fully understood by a human. Examples of interpretable machine learning models are linear regression, logistic regression and decision trees. In interpretablity literature, the word black-box model is used for a model which is not interpretable.

Image recognition models solve a wide variety of tasks on images: object recognition, object detection, semantic segmentation to name a few. Machine learning models that perform well on image recognition tasks are considered black-boxes. This is mainly due to the fact that computer images are high-dimensional and have features that are mostly correlated in local regions of the data, therefore it is hard to learn meaningful features in them.

In recent years, many different approaches have been introduced to make image recognition models more transparent. One common approach is called post-hoc interpretability, such as LIME or SHAP. These methods provide explanations of an already trained model without knowing any information about their parameters or architecture. Post-hoc explanations are weights of an interpretable model that approximates the original black-box. The followings are some problems with these approaches that are addressed in the series of studies presented in this report:

- Explanations are based on the explanation model used
- Explanations are separate models that are trained separately and cannot really be trusted as representing the model we are trying to explain

In a direct contrast with the rationality of post-hoc interpretability, the authors of selected papers suggest the following ideas to make image recognition models interpretable:

- Models should classifying on the basis of visual feature prototypes
- Prediction of images should be done at each of the levels in a taxonomy tree of classes and not only based on dataset labels

The first and second study focused on the visual feature prototypes, while the third study focuses on taxonomies. In these selected works, transparency is enabled already during the training of a model and not after its training. Simply put, these studies propose to directly change the loss function of an image recognition model to make them interpretable. However there are adjustments that should

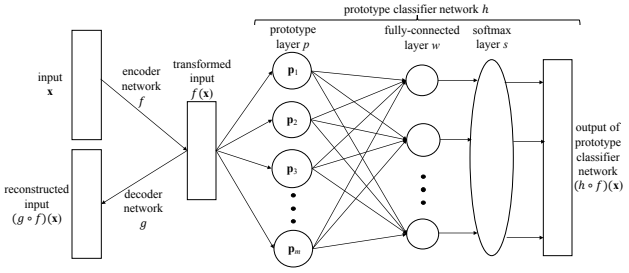

Figure 1: Proposed architecture

Figure 2: Prototype Examples with Loss functions that R2 is removed

be made to the architecture of these models as well to make the models train. the In each section, we focus on one study in three different sub-sections: In *proposed method* section, we discuss the general idea followed by *empirical* results in the next sub-section. Our conclusion is then written in its own sub-section and is related to both the method and the research paper.

2 Prototypes

2.1 Proposed approach

In this work, the authors propose a *prototype classifier* in which observations are classified based on their proximity to a prototype. This idea originally is called *prototype theory* in cognitive science.

In Figure 1, the proposed architecture of the network is shown. The most important components of the network is the prototype layer, p, which needs to be known in advanced. In other words, we need to know how many prototypes are globally feasible to learn, namely m, before we start the training phase of the model.

$$\begin{aligned} L((f, g, h), D) = E(h \circ f, D) + \lambda R(g \circ f, D) \\ +\lambda_1 R_1(p_1, ..., p_m, D) \\ +\lambda_2 R_2(p_1, ..., p_m, D) \end{aligned} \tag{1}$$

In Formula 1 you can see the loss formulation of the proposed model. In this formula, E represent the binary cross entropy loss and R represents an elastic net regularization. The R_1 term enforces each prototype to be to be close to at least one training example, whereas R_2 enforces each training example to be close to at least one prototype. As it can be seen, R_1 has a much more important role for learning diverse prototypes whereas R_2 is important to ensure meaningful prototypes.

2.2 Experiments

The authors perform multiple experiments for validating their model. The cars dataset is the one we discuss and focus on, since it includes an ablation study on the loss function of Formula 1.

In Figure 2, we can see some of the decoded prototypes learned by the model. As it can be seen, they are both fine-grained and easy-to-understand prototypes that depict different rotations of the car and textures, representing different prototypical instances.

One might pose the question on whether any model can learn these prototypes. As it is known, different layers in a neural network can learn abstract features such as edge detection. The authors perform an ablation study in this regard. As you can see in Figure 3 we can see that prototypes that were learned by removing R_2 from the loss function do not own the same quality in depth and

Figure 3: Prototype Examples with Loss functions that R2 is removed

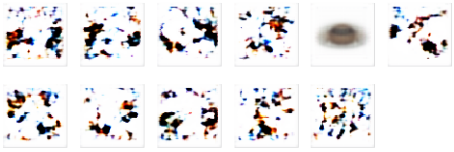

Figure 4: Prototype Examples with Loss functions that R1 is removed

sharpness of the original prototypes. In addition, the same results are depicted in Figure 4, when R_1 is removed from the loss equation. As it can be seen, the presence of R_1 has more importance in learning meaningful prototypes as emphasized earlier.

2.3 Discussion

The main contribution of the proposed model is the formulation of the loss function. Traditionally, in image recognition community, the assumption is that transparency is hard to achieve, however this study proves otherwise and becomes a first step into that direction. The experiments are inclusive and the ablation studies can help to understand each components in the loss formulation.

One of the shortcomings of the paper is the lack of discussion on training the model. In addition, there are no discussion on how to find the optimal number of prototypes for each dataset, namely m, as it is somewhat impossible to find that through looking at a dataset and its underlying classes.

3 ProtoPNet

In [2], the authors show that they can extend the ideas of [1] into developing a model in which a measure of similarity between the image and a similar prototype is found. After that the image is predicted using a weighted combination between the parts of the image and the learned prototypes. The authors find the work on class attention maps as a related work, however class attention maps do not provide any similarity measure to any known prototype. Therefore their idea is an extended version of class attention maps. Their work is also a case-based classifier similar to [1].

In Figure 5 you can see the architecture of ProtoPNet[1]. After passing the image through the convolutional layers, a set of m prototype nodes are added (as done in [1]). The fully connected layer learns the weight of each image with regards to each prototype.

3.1 Experiments

In this study, there are multiple cases that are studied. However the most important case is around bird species identification. Using the proposed model in Figure and the loss formulation that will be mentioned in details in Section 4.1, the model can achieve a good predictive ability and detect similar features of the image at hand and the learned prototypes as shown in Figure 6.

3.2 Discussion

The paper is very much detailed and well-written. The experiments are inclusive and experiments on large-scale datasets are included. The loss formulation is formulated brilliantly and clearly defines the transparent components of the model. It seems that most of the work of [3] was done and achieved in this paper already, even though the work in [2] was published after [3].

[1]Due to the fact that ProtoPNet architecture is extended in [3], the details of ProtoPNet model is written and explained in details in section 4 to avoid duplicated work

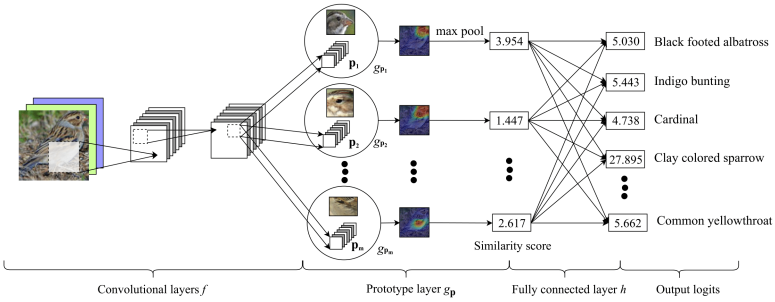

Figure 5: Training classes

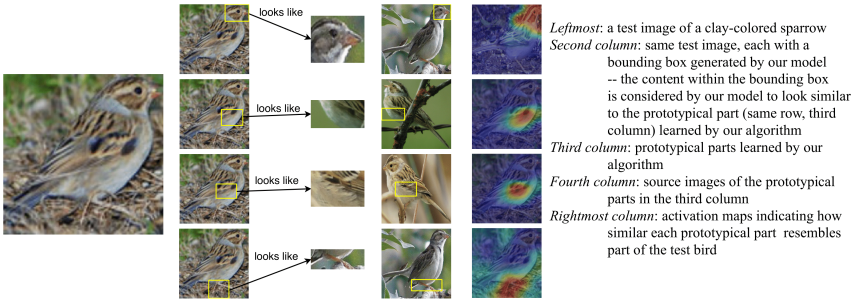

Figure 6: Training classes

4 Hierarchical Prototypes

4.1 Proposed Approach

In the next two studies, authors shift their approach toward learning hierarchical classes and prototypes. Some of the rationale for the importance of hierarchical prototypes in computer vision is the human vision itself. Humans organize the world into "inductively rich" categories that relate to each other. Humans also explain their visual judgment by pointing to prototypes with regards to a class, e.g. a certain animal is a tiger because it is a large cat with black stripes. Lastly, humans can explain each taxonomy in the same class or across different classes with different layers of abstractions

There are a number of ways in which this work is different than similar attempts in learning hierarchical structures in data, such as graphical Bayesian models. In this work, there are different neural networks that learn different sub-classification tasks. In addition, the main strength of this study is that it learns and uses prototypes from the latent space of a neural network and not at the pixel space. On the other hand, other similar approaches are using hierarchical class labeling without learning hierarchical structures in the latent space of a neural network.

In hierarchical image recognition, the task is to predict an images class at each level of taxonomy tree where each y_i has k elements and $y_i^{(k)}$ is the image's label at the kth level at the tree. In this regard, $y^{(0)}$ is the root and $y^{(1)}$ is the first node after the root and so on. Each branch can a different depth. Figure 7 is an example of a hierarchical class that this study uses. We learn a function f that

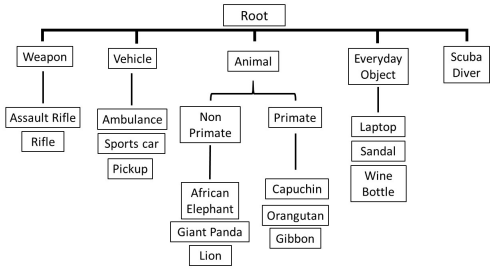

Figure 7: Training classes

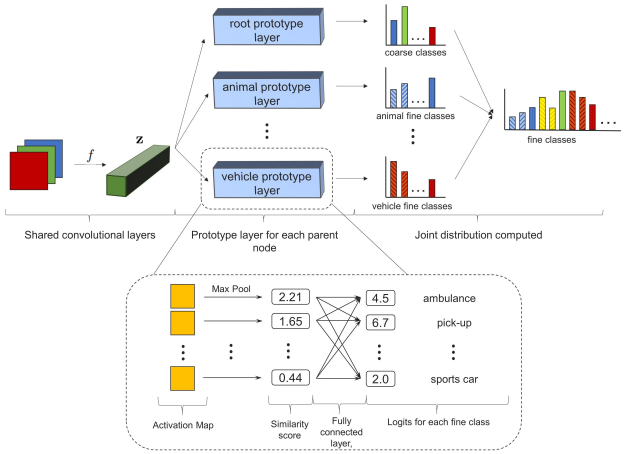

Figure 8: Network Architecture

approximates $P(Y|X)$ over paths in the tree in full labels {animal, dog} with impossible paths as being 0, like {animal, car}. The task is to learn each of the factors of the probability as in Formula 2.

$$P(Y|X) = P(Y^{(1)}|X) \times ... \times P(Y^{(K)}|Y^{(K-1)}, X) \tag{2}$$

After that, each distribution represents the multinomial distribution over children classes $Y^{(K)}$. The proposed architecture in [3] can be seen in Figure 8. This model is an extension of [2]. In this study, the encoder part of VGG-16 maps the images to the input space, namely $\tilde{z}$. For each parent node, there is a prototype layer that operates directly on $\tilde{z}$ and produces a similarity score. During training, m prototypes are learned by mapping each instance and selecting for which prototype the activation map is in its maximum. Before training, authors set pre-determined number of prototypes evenly to child class of $P^{c^{(k)}}$. In following formulas, the formulation of the loss function is visible. The interpretations of clustering and separation are equivalent of R_1 and R_2 respectively as in section 2.1:

$$\sum_{c^{(k)} \in C} [\sum_{c^{(k)}} \text{Cross Entropy}(h^{c^{(k)}} \circ g_{P^{c^{(k)}}} \circ f(x_i), y_i)$$

$$+\lambda_1 \text{Clust}(P^{c^{(k)}}, X, Y) + \lambda_2 \text{Sep}(P^{c^{(k)}}, X, Y) + \lambda_3 \text{Reg}(h^{c^{(k)}})$$

$$\text{Clust}(P^{c^{(k)}}, X, Y) = \sum_{i: y_i^{(k)}} c^{(k)} \min_{j: P_j \in P_{c_i^{(k+1)}}} \min_{\tilde{z} \in \text{patches}(f(x_i))} ||\tilde{z} - p_j||_2^2$$

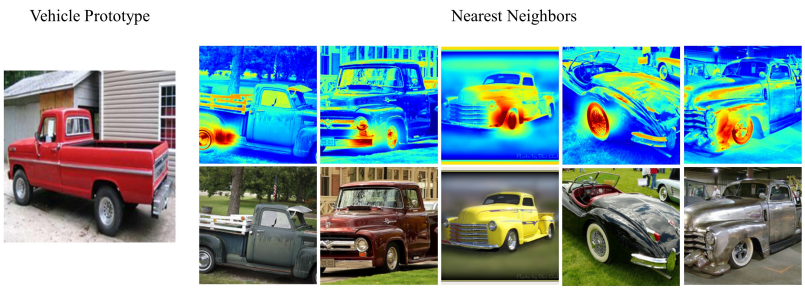

Figure 9: Similar images to vehicle prototypes

$$\text{Sep}(P^{c^{(k)}}, X, Y) = -\sum_{i:y_i^{(k)}} c^{(k)} \min_{j:P_j \notin P_{c_i^{(k+1)}}} \min_{\tilde{z} \in \text{patches}(f(x_i))} ||\tilde{z} - p_j||_2^2$$

In these formulas, cross entropy E handles the accuracy of predictions over joint distribution of fine-grained data. Clustering cost encourages the model to map at least one path vector of each image close to a prototype corresponding to its class. Separation costs discourages the mapping of patches to different classes. Regularization terms is both an L1 regularization term (to nullify the weights of the mapping between an image with different classes) while L2 regularization handles the weights for within-class weights.

The training of the model is not end-to-end and it can prove to be challenging as each networks needs to map their own coarse-layer into similar prototype and prototypes are then learned in the latent space of the model.

4.2 Experiments

There are many inclusive studies in [3], however we have selected the ones that are most significant to this short report. The main experiment in [3] resolves around predicting on a subset of hierarchical classes in ImageNet as shown in Figure 7. In Figure 9, some of the prototypes learned for the class vehicle using nearest neighbour is shown.

In Figure 10, the set of vehicle prototypes that have the highest similarity to the image of the forklift are shown. In addition to that, one can see highlights of which part of the image resembles which part of the image that is explained. This is somewhat an extension of [2] where same similarity features can be extended for hierarchical classes.

4.3 Discussions

This work overall seems to be a more extended version of the model in [2]. However the experiments are not inclusive as expected. The subset of ImageNet is rather small and limited to draw conclusions about. In addition, since the training is not end to end, it is hard replicate their work as not so much details are revealed for the training of the model. Overall, it is an interesting extension of [2], however it seems like the idea needs more experimental results to strengthen their arguments.

Reference

[1] Li, O., Liu, H., Chen, C., & Rudin, C. (2018) Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. *In Thirty-Second AAAI Conference on Artificial Intelligence*.

[2] Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). This looks like that: deep learning for interpretable image recognition. *In Advances in Neural Information Processing Systems*

[3] Hase, P., Chen, C., Li, O., Rudin, C. (2019). Interpretable Image Recognition with Hierarchical Prototypes. *In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (Vol. 7, No. 1, pp. 32-40).

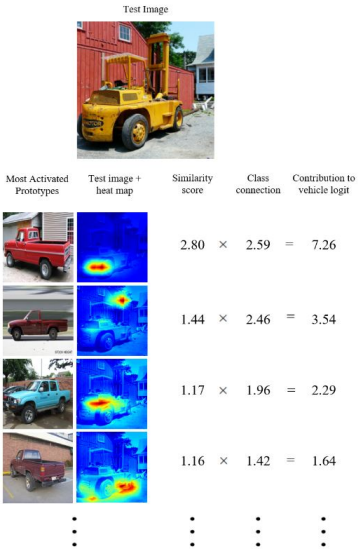

Figure 10: Case Study