Investigation of Interpretability of Different Local Explainers for Deep Neural Network

Chengyang Wang, Fangzhou Li, Xiawei Wang

November 29, 2020

Abstract

Local Interpretable Model-agnostic Explanations (LIME) is a popular approach which uses a linear model, an explainer, to generate local explanations for individual predictions made by complex models, such as deep neural networks. While linear models are easily interpretable, it has two limitations: 1) It is not good at classification problems, and 2) it fails to capture the importance of feature interactions. In this paper, we investigate the levels of LIME's explainability while using different local explainers: 1) the logistic regression model and 2) the decision tree model. We apply these variations of LIME to an image classification model to compare the plausibility and usefulness of the generated explanations. Our experiment shows that the LIME with the decision tree explainer generates better explanations for image classifiers. The LIME with the decision tree explainer outperforms the original LIME and the LIME with the logistic explainer respectively by 7.1% and 31.6% in human expert evaluation. This paper highlights the potential of decision tree models for generating local explanations and serves as a stepping stone for the application of different interpretable models to complex models.

1 Introduction

Deep learning has been increasingly showing its potential in various tasks, such as natural language generation and object detection [1]. However, the opacity of deep learning prevents people from understanding why the models behave in a certain way, which makes it difficult to deploy deep learning models in real-world policy making. In the last decade, the field of Interpretable Machine Learning (IML) developed at a significantly fast pace

[2]. Many model-agnostic [3, 4] as well as model-specific methods [5, 6] have helped scientists to acquire insights from supposed blackboxes. The model-agnostic approaches of IML are especially popular due to their versatility. LIME is one of the most popular model-agnostic approaches to interpret complex models [7]. It generates the explanation for a model's prediction by perturbing samples around that prediction and fitting the perturbed samples to a linear model, an explainer that is intrinsically interpretable by humans. Although linear models are easily understandable, they have two limitations due to its nature of linearity. First, it performs poorly on tasks that require sparse data, such as classification problems. Second, it is not capable of capturing the importance of feature interaction. These limitations of linear models prevent them from being great explainers for image classification models which involve not only the sparse model outputs but also the relation between adjacent pixels.

In this paper, we compare the levels of interpretability of LIME with different model explainers: the linear regression model, the logistic regression model, and the decision tree. We test these variations of LIME on a well-trained image classification model and have human experts manually evaluate the resulting explanations. With three human experts, each evaluating 150 explanations in total, the LIME with the tree explainer outperforms the original LIME with the linear explainer and our LIME with the logistic explainer respectively by 7.1% and 31.6% in terms of our evaluation metric. See our Results for details. We conclude that the LIME with tree explainer performs better than the original LIME with the linear explainer in explaining predictions made by image classifiers. The LIME with the logistic explainer does not show extraordinary performance, but we discuss its potential later in this paper and plan to further explore its possibility in the future investigation.

2 Methods

2.1 LIME

LIME is a technique that uses the linear regression model to locally explain a complex model's individual predictions. [7] LIME perturbs and resamples the neighbor of a prediction to generate a lower-dimensional dataset for training the linear explainer. It is slightly different while LIME is dealing with different data types, but we focus on image data in our paper. Given a prediction of an image data, we start from applying the quick-shift method [8] to get the image segmentations for the input, where we refer each segment as a superpixel. LIME then repeatedly masks random subsets of superpixels to generate perturbed images which are fed to the image classifier to create training labels for the linear explainer. The inputs of the linear explainer are then binary vectors where each element indicates the presence of the corresponding superpixel.

LIME only focuses on one prediction at a time, which makes the outcome of LIME local explanations. A local explanation heavily relies on its explainer so that we are interested in the qualities of explanations generated by different interpretable models.

2.2 Interpretable Model

2.2.1 Linear Regression

Linear regression is one of the simplest interpretable models,

$$f(x) = w_0 + w_1 x_1 + \cdots + w_p x_p$$

whose feature coefficients are intrinsically indicating feature importances. To get a stable solution, we add the L2 Ridge penalty $\alpha = 1$ to our linear explainer.

2.2.2 Logistic Regression

A logistic regression model g $g$is essentially a composition of a sigmoid function and a linear regression model:

$$g(x) = \sigma(f(x)) = \frac{1}{1 + e^{-f(x)}}$$

where the output of g is the probability of x classified to a class.

The interpretation of logistic regression models vary from linear regression models. Consider a binary classification problem, the odds is then calculated as:

$$\text{odds} = \frac{\mathbb{P}(y=1)}{1 - \mathbb{P}(y=0)} = e^{f(x)} = e^{w_0 + w_1 x_1 + \cdots + w_p x_p}$$

If we increase one of the feature weights by 1 and observe the ratio of the two odds:

$$\frac{\text{odds'}}{\text{odds}} = \frac{e^{w_0 + w_1 x_1 + \cdots + w_j (x_j + 1) + \cdots + w_p x_p}}{e^{w_0 + w_1 x_1 + \cdots + w_j x_j + \cdots + w_p x_p}} = e^{w_j}$$

where w_j is the weight of the increased feature. Thus, the interpretation of a logistic regression model is: Increasing a feature x_j by one unit multiplies the log odds ratio by the value of that feature's weight [9].

In our paper, we want to focus on superpixels that are positively influencing the prediction according to the logistic regression explainer, so we collect the feature weights with positive values.

2.2.3 Decision Tree Regression

Different from both linear and logistic regression models, a decision tree model is capable of capturing the importance of feature interaction, which makes it a potential explainer for image classification tasks. Recently, Shi et al. [10] experimented with the performance of LIME with the decision tree explainer on image classifier predictions. The interpretation of a decision tree model is achieved by its feature importance. Here we use Gini importance to calculate the feature importance of x_p:

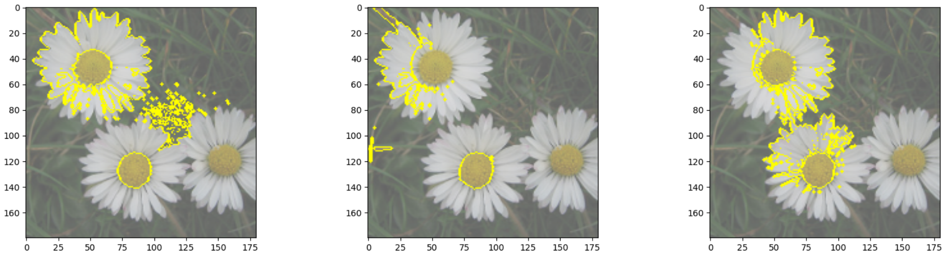
$$I(x_p) = \sum_{s \in S_p} (N_s \cdot G_s - N_{s.left} \cdot G_{s.left} - N_{s.right} \cdot G_{s.right})$$

where S_p is a set of all nodes that use the feature x_p as the split points, and N_s, G_s are the number of samples, the Gini impurity of the node s respectively. The feature importance values are then normalized so that they add to one.

3 Experiments

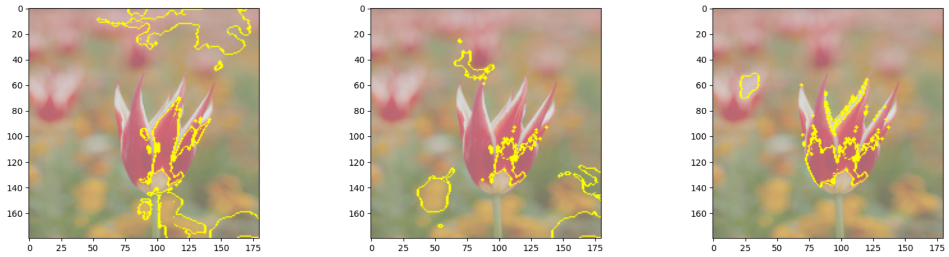
We implemented image classifiers based on convolutional neural nets as our complex models. We used the three local explainers to identify features from images that cause the classifier to make such predictions.

Fig.1 and Fig.2 are examples of identified superpixels from 3 local explainers. They show that LIME_tree outperforms LIME_line and LIME_log in identifying more relevant features. Fig.1 shows that LIME_tree identifies more pedals and cores of daisy flowers. Fig.2 shows that LIME_tree is the most accurate in extracting tulip petals, even those in the background. Meanwhile LIME_line and LIME_log incorrectly extract background information as important features.



(a) Daisy_linear (b) Daisy_logistic (c) Daisy_tree

Figure 1: Local Explainers on Daisy



(a) Tulip_linear (b) Tulip_logistic (c) Tulip_tree

Figure 2: Local Explainers on Tulip

3.1 Setup

We first choose a convolutional neural network architecture, CNN_good, which consists of 3 consecutive (convolutional layers + Maxpooling layer) and two dense layers. We train it by Google TF-Flower Dataset [11] which contains 3670 images and 5 classes and CNN_good achieves 85% training accuracy and 78% validation accuracy. We test the variations of LIME on it. For each class, we randomly pick 10 images that are not in the training dataset and apply three LIME variations, LIME_line, LIME_log, and LIME_tree, to these 50 images. We then have three human experts evaluate the generated 150 explanations by scoring them between 1 to 3 with 3 standing for the highest quality of an explanation. We evaluate the performances of the LIME variations by averaging all the scores given by three human evaluators for each local explainer.

3.2 Results

The result of human evaluation is shown in Table 1. LIME_tree outperforms both LIME_line and LIME_log by 7.3% and 31.7%. Fig.1 and Fig.2 are examples of three explanations generated by different explainers on the same images.

We also noticed that among 5 classes, Daisy has the best score, Rose and Tulip the worst 2 classes. We further investigated the images. It turns out that most daisy images are close-up of daisy. However, rose pictures often involve different backgrounds, such as a wedding ceremony, and tulip pictures often involve a tulip field.

	LIME_linear	LIME_logistic	LIME_tree	Class Average
Daisy	2.40	1.93	2.7	2.34
Dandelion	2.26	1.93	2.47	2.22
Rose	2.23	1.50	2.30	2.01
Sunflower	2.30	2.07	2.20	2.19
Tulip	2.07	1.83	2.40	2.10
Exlainer Average	2.25	1.832	2.414	

Table 1: Average Score of Local Interpretability by Expaliner and Class

4 Discussion

4.1 Decision Tree as an Explainer

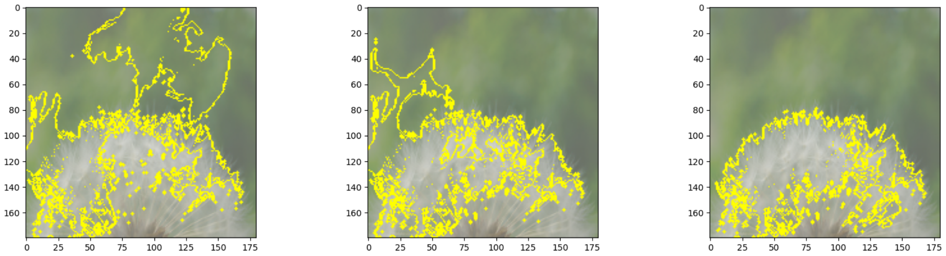
Our result shows that LIME with a decision tree explainer achieves the overall best result compared to the other two. We believe that the decision tree model is more expressive due to its capacity of considering feature interaction. While decision trees usually suffer from unstable outcomes, LIME_tree's feature importance is not affected too much because of its binary splits, i.e. the presence of a superpixel. This mitigates the randomness of growing a decision tree, resulting in a relatively stable feature importance calculation.

4.2 Logistic Regression as an Explainer

The logistic regression model does not perform well in the experiment even though we expect that the logistic regression model should have behaved better than the linear model regression model in sparse data distribution. Our speculation is that the sparsity of data decreases explanations' faithfulness towards the original predictions. For example, let's have a binary classification model which generates a prediction saying a sample has a 51% probability of being a dandelion. While the linear regression model generates an explanation that is faithful to that 51% probability, the logistic regression model treats all probabilities that are higher than the threshold as 1. This means that the logistic explainer fails to convey an accurate explanation for predictions with low confidence. Also, the logistic regression model occasionally returns only superpixels with no positive influences on predictions. We suspect that a few features are assigned with explosive weights while training the explainer, resulting in negative weights for other features.

4.3 Images Noises

The variations of LIME respond differently to images with noises. Some images consist of random background or scattering objects, like a garden of roses or sprouts of a dandelion. It heavily influences the quality of image segmentation, hence the outcome of explanation. We observe that the tree explainer is somewhat more resilient to the image noise than the linear and logistic explainers are Fig.3. We will leave this investigation to our future study.



(a) Dandelion_linear (b) Dandelion_logistic (c) Dandelion_tree

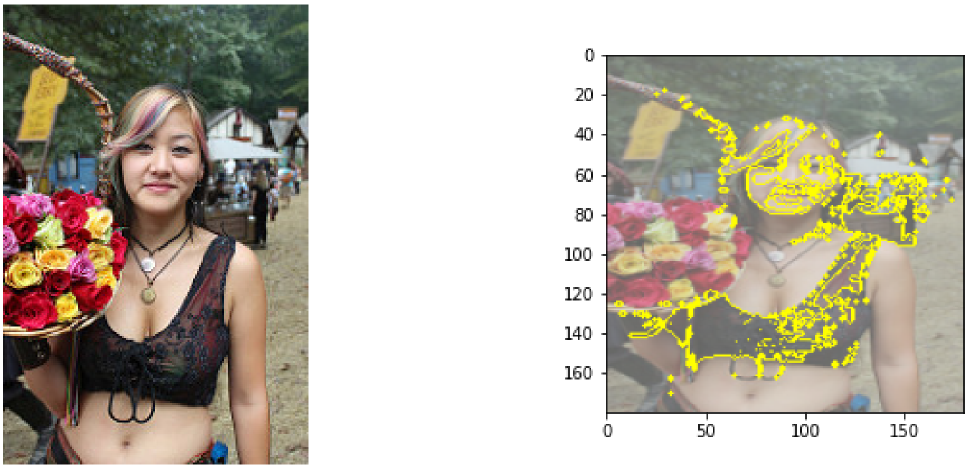
Figure 3: Local Explainers on Dandelion

4.4 Detection of Bad Classifier

We are also interested in whether the explanations of the LIME variations can help people to identify the ill-training of their models. We purposely created an ill-trained overfitted model with training and validation accuracy of 91% and 55% respectively. Fig.4 is the explanation for a correctly classified prediction, generated by the LIME with tree explainer. As we can see, the observer can easily detect the model has learned a wrong representation for this image classification task.

5 Conclusion

In this paper, we are interested in studying the quality of explanations generated by LIME with different explainers: the linear regression, the logistic regression, and the decision tree models. We succeed to present the comparison among explanations generated by different explainers. More specifically, we show that the LIME with decision tree explainer is able to generate more comprehensive and detailed explanations for predictions of image classifiers.



(a) Original picture

(b) Explanation by Decision_tree

Figure 4: Local explainer of a bad calssifier

We will further investigate the reason for the relative resilient performance of the decision tree explainer. Also, to overcome the poor feature extraction of logistic regression, the explosive weights can be avoided by adding more sophisticated regularization. We believe that our work highlights the potential of decision tree models for generating local explanations, as well as the sufficiency of their expressiveness for the model interpretability. We hope our work serves as a stepping stone for the future study in the explainability of complex models via simple interpretable methods.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. *arXiv preprint arXiv:2010.09337*, 2020.

[3] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, 2019.

[4] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[5] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2018.

[6] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences, 2019.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[8] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *In European Conference on Computer Vision, volume IV*, pages 705–718, 2008.

[9] Christoph Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[10] Sheng Shi, Xinfeng Zhang, and Wei Fan. Explaining the predictions of any image classifier via decision trees, 2020.

[11] The TensorFlow Team. Flowers, jan 2019.

Author contributions

Fangzhou Li implemented the variation of LIME with the logistic regression explainer and generated the figures. Chengyang Wang implemented the decision tree local explainer and generated figures. Xiawei Wang implemented image preprocessing. All authors contributed equally to the documentation.