NormLime: A New Feature Importance Metric for Explaining Deep Neural Networks

Isaac Ahern,[1] **Adam Noack,** [1] **Luis Guzmán-Nateras,**[1] **Dejing Dou,**[1,2] **Boyang Li,**[2] **and Jun Huan**[2]

[1] University of Oregon
[2] Baidu Research
{iahern, anoack2, lguzmann, dou}@cs.uoregon.edu, {boyangli, huanjun}@baidu.com

Abstract

The problem of explaining deep learning models, and model predictions generally, has attracted intensive interest recently. Many successful approaches forgo global approximations in order to provide more faithful local interpretations of the model's behavior. LIME develops multiple interpretable models, each approximating a large neural network on a small region of the data manifold and SP-LIME aggregates the local models to form a global interpretation. Extending this line of research, we propose a simple yet effective method, Norm-LIME for aggregating local models into global and class-specific interpretations. A human user study strongly favored class-specific interpretations created by NormLIME to other feature importance metrics. Numerical experiments confirm that NormLIME is effective at recognizing important features.

Introduction

As the applications of deep neural networks continue to expand, the intrinsic black-box nature of neural networks creates a potential trust issue. For application domains with high cost of prediction error, such as healthcare (Phan *et al.* 2017), it is necessary that human users can verify that a model learns reasonable representation of data and the rationale for its decisions are justifiable according to societal norms (Koh and Liang 2017; Fong and Vedaldi 2018; Zhou *et al.* 2018; Lipton 2016; Langley 2019).

An interpretable model, such as a linear sparse regression, lends itself readily to model explanation. Yet due to limited capacity, these interpretable models cannot approximate the behavior of neural networks globally. A natural solution, as utilized by LIME (Ribeiro *et al.* 2016), is to develop multiple interpretable models, each approximating the large neural network locally on a small region of the data manifold. Global explanations can be obtained by extracting common explanations from multiple local approximations. However, how to best combine local approximations remains an open problem.

Extending this line of research, we propose a novel and simple feature scoring metric, *NormLIME*, which estimates

the importance of features based on local model explanations. In this paper, we empirically verify the new metric using two complementary tests. First, we examine if the Norm-LIME explanations agree with human intuition. In a user study, participants favored the proposed approach over three baselines (LIME, SmoothGrad and VarGrad) with Norm-LIME receiving 30% more votes than all the baselines combined. Second, we numerically examine if explanations created by NormLIME accurately capture characteristics of the machine learning problem at hand, using the same intuition proposed by Hooker *et al.* (2018). Empirical results indicate that NormLIME identifies features vital to the classification performance more accurately than several existing methods. In summary, we find strong empirical support for our claim that NormLIME provides accurate and human-understandable explanations for deep neural networks.

The paper makes the following contributions:

- We propose a simple yet effective extension of LIME, called NormLIME, for aggregating interpretations around local regions on the data manifold to create global and class-specific interpretations. The new technique outperforms LIME and other baselines in two complementary evaluations.

- We show how feature importance from LIME can be aggregated to create class-specific interpretations, which stands between the fine-grained interpretation at the level of data points and the global interpretation at the level of entire datasets, enabling a hierarchical understanding of machine learning models. The user study indicates that NormLIME excels at this level of interpretation.

Related Work

A machine learning model can be interpreted from the perspective of how much each input feature contributes to a given prediction. In computer vision, this type of interpretation is often referred to as *saliency maps* or *attribution maps*. A number of interpretation techniques, such as SmoothGrad (Smilkov *et al.* 2017), VarGrad (Adebayo *et al.* 2018), Integrated Gradients (Sundararajan *et al.* 2017), Guided Backpropagation (Springenberg *et al.* 2015), Guided GradCAM (Selvaraju *et al.* 2017), and Deep Taylor Expansion (Montavon *et al.* 2017), exploit gradient information, as it pro-
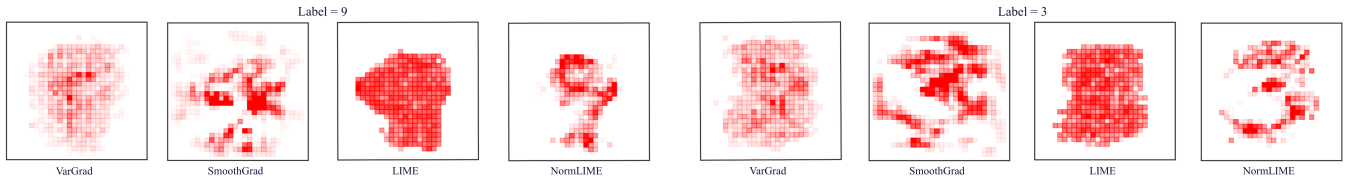
Figure 1: Explanations for the digits 9 and 3 in MNIST, generated by four different feature importance metrics: VarGrad (Adebayo *et al.* 2018), SmoothGrad (Smilkov *et al.* 2017), LIME (Ribeiro *et al.* 2016), and NormLIME (ours).

vides a first-order approximation of the input's influence on the output (Simonyan *et al.* 2013; Ancona *et al.* 2018). Seo *et al.* (2018) analyze the theoretical properties of Smooth-Grad and VarGrad. When the gradient is not easy to compute, Baehrens *et al.* (2010) places Parzen windows around data points to approximate a Bayes classifier, from which gradients can be derived. DeepLift (Shrikumar *et al.* 2017) provides a gradient-free method for saliency maps. Though gradient-based techniques can interpret individual decisions, aggregating individual interpretations for a global understanding of the model remains a challenge.

Local interpretations are beneficial when the user is interested in understanding a particular model decision. They become less useful when the user wants a high-level overview of the model's behavior. This necessitates the creation of global interpretations. LIME (Ribeiro *et al.* 2016) first builds multiple sparse linear models that approximate a complex model around small regions on the data manifold. The weights of the linear models can then be aggregated to construct a global explanation using Submodular Pick LIME (SP-LIME). Ribeiro *et al.* (2018) introduce anchor rules to capture interaction between features. Tan *et al.* (2018) approximate the complex model using a sum of interpretable functions, such as trees or splines, which capture the influence of individual features and their high-order interactions. The proposed NormLIME technique fits into this "neighborhood-based" paradigm. We directly change the normalization for aggregating weights rather than the function forms (such as rules or splines).

Ibrahim *et al.* (2019) rank feature importance and cluster data points based on their ranking correlation. The interpretations for cluster medoids are used in the place of a single global interpretation. Instead of identifying clusters, in this paper, we generate interpretations for each class in a given dataset. The class-level interpretation provides an intermediate representation so that users can grasp behaviors of machine learning models at different levels of granularity.

Proper evaluation of saliency maps can be challenging. Adebayo *et al.* (2018) show that, although some techniques produce reasonable saliency maps, such maps may not faithfully reflect the behavior of the underlying model. Thus, visual inspection by itself is not a reliable evaluation criterion. Kindermans *et al.* (2017) adopt linear classifiers as a sanity check. Feng and Boyd-Graber (2019) propose cooperative games, where humans can see the interpretation of their AI teammate, as a benchmark. Hooker *et al.* (2018) propose ablative benchmarks for the evaluation of feature importance maps. When features that are considered important are re-moved from the input, the model should experience large drops in performance. The opposite should happen when features deemed unimportant are removed. In this paper, we evaluate the proposed technique using the Keep-and-Retrain (KAR) criterion from Hooker *et al.*.

Methodology

As a method for explaining large neural networks, LIME (Ribeiro *et al.* 2016) first builds shallow models where each approximates the complex model around a locality on the data manifold. After that, the local explanations are aggregated using SP-LIME. In this work, we extend the general paradigm of LIME with a new method, which we call Norm-LIME, for aggregating the local explanations.

Building Local Explanations

LIME constructs local interpretable models to approximate the behavior of a large, complex model within a locality on the data distribution. This process can be analogized with understanding how a hypersurface $f(\boldsymbol{x})$ changes around $\boldsymbol{x}_0$ by examining the tangent hyperplane $\nabla f(\boldsymbol{x}_0)$.

Formally, for a given model $f : \mathcal{X} \mapsto \mathcal{Y}$, we may learn an interpretable model g, which is local to the region around a particular input $\boldsymbol{x}_0 \in \mathcal{X}$. To do this, we first sample from our dataset according to a Gaussian probability distribution $\pi_{\boldsymbol{x}_0}$ centered around $\boldsymbol{x}_0$. Repeatedly drawing $\boldsymbol{x}'$ from $\pi_{\boldsymbol{x}_0}$ and applying $f(\cdot)$ yield a new dataset $\mathcal{X}' = \{(\boldsymbol{x}', f(\boldsymbol{x}'))\}$. We then learn a sparse linear regression $g(\boldsymbol{x}', \boldsymbol{x}_0) = \boldsymbol{w}_{\boldsymbol{x}_0}^\top \boldsymbol{x}'$ using the local dataset $\mathcal{X}'$ by optimizing the following loss function with $\Omega(\cdot)$ as a measure of complexity.

$$\underset{\boldsymbol{w}_{\boldsymbol{x}_0}}{\arg\min} \, \mathcal{L}(f, g, \pi_{\boldsymbol{x}_0}) + \Omega(\boldsymbol{w}_{\boldsymbol{x}_0}) \tag{1}$$

where $\mathcal{L}(f, g, \pi_{\boldsymbol{x}_0})$ is the squared loss weighted by $\pi_{\boldsymbol{x}_0}$

$$\mathcal{L}(f, g, \pi_{\boldsymbol{x}_0}) = \underset{\boldsymbol{x}' \sim \pi_{\boldsymbol{x}_0}}{\mathbb{E}} \, (f(\boldsymbol{x}') - g(\boldsymbol{x}', \boldsymbol{x}_0))^2 \tag{2}$$

For $\Omega(\boldsymbol{w}_{\boldsymbol{x}_0})$, we impose an upper limit K for the number of non-zero components in $w_{\boldsymbol{x}_0}$, so that $\Omega(\boldsymbol{w}_{\boldsymbol{x}_0}) = \infty \cdot \mathbb{1}(\|\boldsymbol{w}_{\boldsymbol{x}_0}\|_0 > K)$. The optimization is intractable, but we approximate it by first selecting K features with LASSO regression and performing regression on only the top K features.

This procedure yields $g(\boldsymbol{x}', \boldsymbol{x}_0)$, which approximates the complex model $f(\boldsymbol{x})$ around $\boldsymbol{x}_0$. The components of the weight vector $\boldsymbol{w}_{\boldsymbol{x}_0}$ indicate the relative influence of the individual features of $\boldsymbol{x}$ in the sample $\mathcal{X}'$ and serve as the local explanation of $f(\boldsymbol{x})$. Figure 2 illustrates such a local explanation.
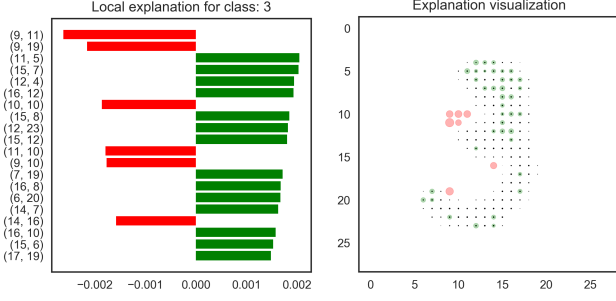
Figure 2: A LIME-based Local Explanation. On the right, green and red indicate pixels whose presence/absence offers support for the class label "3". On the left, we show the 20 pixels with the largest weights.

NormLIME

After a number of local explanations have been constructed, we aim to create a global explanation. NormLIME is a method for aggregating and normalizing multiple local explanations and estimating the global relative importance of all features utilized by the model. NormLIME gives a more holistic explanation of a model than the local approximations of LIME.

We let c_i denote the ith feature, or the ith component in the feature vector $\boldsymbol{x}$. Since the local explanation weights are sparse, not all local explanations utilize c_i. We denote the set of local explanations that do utilize c_i as $E(c_i)$, which is a set of weight vectors $\boldsymbol{w}_{x_j}$ computed at different locales $\boldsymbol{x}_j$. In other words, for all $\boldsymbol{w} \in E(c_i)$, the corresponding weight component $w_i \neq 0$.

The NormLIME "importance score" of the feature c_i, denoted by $\mathcal{S}(c_i)$, is defined as the weighted average of the absolute values of the corresponding weight $\boldsymbol{w}_i, \forall \boldsymbol{w} \in E(c_i)$.

$$\mathcal{S}(c_i) := \frac{1}{|E(c_i)|} \sum_{\boldsymbol{w}_{x_j} \in E(c_i)} \gamma(\boldsymbol{w}_{x_j}, i) \left| w_{x_j, i} \right|, \quad (3)$$

where the weights γ are computed as follows.

$$\gamma(\boldsymbol{w}_{x_j}, i) := \frac{|w_{x_j, i}|}{\sum_k |w_{x_j, k}|} = \frac{|w_{x_j, i}|}{\|\boldsymbol{w}_{x_j}\|_1}. \quad (4)$$

Here, $\gamma(\boldsymbol{w}_{x_j}, i)$ represents the relative importance of the feature c_i in the local model built around the data point x_j. If a feature c_i is not utilized in any local models, we set its importance $\mathcal{S}(c_i)$ to 0.

We now introduce a slightly different perspective of NormLIME, which helps us understand the difference between this approach and the aggregation and feature importance approach used in LIME. Consider the global feature weight matrix M, whose rows are the local explanation w computed at different locales. Let $\boldsymbol{\omega}_i$ be the ith column of the matrix, which contains the weights for the same feature in different local explanations. Let $\boldsymbol{v}$ be the vector representing the L1 norms of the rows of M. We can express the NormLIME global feature importance function as

$$\mathcal{S}(c_i) = \frac{1}{\|\boldsymbol{\omega}_i\|_0} \boldsymbol{\omega}_i^\top \text{diag}(\boldsymbol{v}^{-1}) \boldsymbol{\omega}_i, \quad (5)$$

where $\text{diag}(\boldsymbol{v}^{-1})$ denotes a matrix with $\boldsymbol{v}^{-1}$ on the diagonal and zero everywhere else. $\|\cdot\|_0$ is the L0 norm, or the number of non-zero elements.

In comparison, the submodular pick method (SP-LIME) by Ribeiro *et al.* (2016) employs the L2 norm

$$I^{SP}(c_i) = \|\boldsymbol{\omega}_i\|_2 \quad (6)$$

to measure the importance of c_i. Contrasting Eq. (5) and (6), it is apparent that the difference between the two methods lies in in the normalization of the column weights.

Class-specific Interpretations

As discussed above, NormLIME estimates the overall relative importance assigned to feature c_i by the model. For binary classification problems, this is equivalent to a representation of the importance the model assigns to the feature c_i in distinguishing between the two classes, i.e., recognizing a class label. In multi-class problems, however, this semantic meaning is lost as the salience computation above does not distinguish between classes, and class-relevant information becomes muddled together.

It is straightforward to recover the salience information associated with individual class labels, by partitioning $E(c_i)$ based on the class label of the initial point x_j that the local approximation is built around. The partition $E_y(c_i)$ contains the local explanation w_{x_j} if and only if $f(\cdot)$ assigns the label y to x_j. More formally,

$$E_y(c_i) = \{w_{x_j} \in E(c_i) \mid f(x_j) = y\}. \quad (7)$$

It is easy to see that if $E_y(c_i) \neq \varnothing, \forall y$, and $f(x)$ is a single-label classification problem, then the family of sets $E_y(c_i)$ forms a partition of $E(c_i)$:

$$E(c_i) = \bigcup_y E_y(c_i), \quad E_{y_j}(c_i) \cap E_{y_k}(c_i) = \varnothing \, \forall y_j \neq y_k. \quad (8)$$

Computing salience of c_i for a given label is performed via

$$\mathcal{S}_y(c_i) := \frac{1}{|E_y(c_i)|} \sum_{\boldsymbol{w}_{x_j} \in E_y(c_i)} \gamma(\boldsymbol{w}_{x_j}, i) \left| w_{x_j, i} \right|. \quad (9)$$

Compared to the global interpretation, the class-specific salience $\mathcal{S}_y(c_i)$ yields higher resolution information about how the complex model differentiates between classes. We use $\mathcal{S}_y(c_i)$ as prediction-independent class-level explanations in the human evaluation, described in the next section.

Human Evaluation

In order to put the interpretations generated by NormLIME to test, we administered a human user study across Amazon Mechanical Turk users. In the study, we compare class-specific explanations generated by other standard feature importance functions and the proposed salience method on the MNIST dataset.

Saliency Map Baselines

To avoid showing too many options to the human participants, which may cause cognitive overload, we selected a

For each set of images, please select the image that best highlights the pixels that are most important in distinguishing this particular number from the other nine numbers.

For example, with the image of the number four, the red pixels inside of the blue oval constituting the corner of the arm of the four might be considered more important than other pixels in the image because these pixels are NOT activated in images of most representations of the other nine numbers but ARE activated often in images of other fours. Therefore, a model that produces an image that highlights these circled pixels when generating the explanatory image might be considered better at explaining the number four than others that don't.
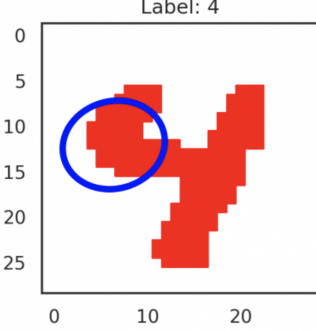
Pay close attention to the number that each model is trying to represent. This number is highlighted in yellow in the description above each set of four images. This number is always the same as the "Label" above each image in each set.

Figure 3: Instructions given to survey takers.

From the following set of four images, **please select the image that best highlights the pixels that distinguish the number 9 from the other nine possible digits.**

From the following set of four images, **please select the image that best highlights the pixels that distinguish the number 3 from the other nine possible digits.**

Figure 4: Top: example questions from the user study Bottom: example attention checks from the user study. The correct choice is option 2. Options 1 and 4 are duplicates and cannot distinguish well between labels "3" and "8".

few baseline techniques that we consider to be the most promising for saliency maps (Seo *et al.* 2018). We select SmoothGrad and VarGrad because they aim to reduce noise in the gradient, which should facilitate the aggregation of individual decisions to form a class-level interpretation. The aggregation of these individual interpretations are performed by taking the mean importance scores over interpretations from a sample of datapoints corresponding to each label. The details are discussed below.

SmoothGrad (Smilkov *et al.* 2017) This technique generates a more "interpretable" salience map by averaging out local noise typically present in gradient interpretations. We add random noise $\eta \sim \mathcal{N}(0, \sigma^2)$ to the input x. Here we follow the default implementation using the "SmoothGrad Squared" formula, which is an expectation over η:
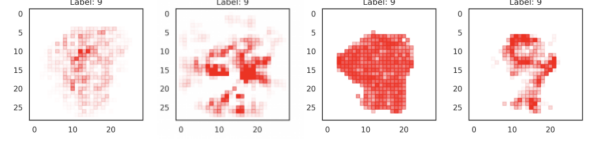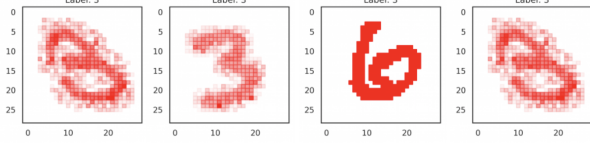
$$I^{SGSQ}(x) = \mathbb{E}\left[|\frac{\partial f(x+\eta)}{\partial c_k}|^2\right]$$

as noted in (Hooker *et al.* 2018). In practice, we approximate the expectation with the average of 100 samples of η drawn from $\mathcal{N}(0, \sigma^2)$ where σ is 0.3. The class-level interpretation is computed as the average of the saliency maps for 10 images randomly sampled from the target class.

VarGrad (Adebayo *et al.* 2018) Similar to SmoothGrad, VarGrad uses local sampling of random noise $\eta \sim \mathcal{N}(0, \sigma^2)$ to reduce the noise of the standard gradient map interpretation. VarGrad perturbs an input x randomly via η, and then computes the component-wise variance of the gradient over the sample

$$I^{VG}(x) = \text{Var}(|\frac{\partial f(x+\eta)}{\partial c_k}|).$$

Similar to SmoothGrad, we compute the variance using 100 samples of η from a normal distribution with zero mean and standard deviation of 0.3. We use the average saliency map over 10 randomly sample images in the desired class as the class-level interpretation.

LIME (Ribeiro *et al.* 2016) We compute importance as $I^{SP}(x)$ as in Eq. (6), but conditioned on the label y to capture feature that were positively correlated with the specific label.

For both LIME and NormLIME, we only show the input features that are positively correlated with the prediction. That is, when $I^{SP}(c_i)$ or $\mathcal{S}(c_i)$ is positive. The purpose is to simplify the instructions given to human participants and avoid confusion, since most participants may not have sufficient background in machine learning.

Experimental Design

The design of the study is as follows: We administered a questionnaire featuring 25 questions, each containing four label-specific explanations for the same digit. We were able to restrict participants through Mechanical Turk to users who had verified graduate degrees. Survey takers were instructed to evaluate the different explanations based on how well they captured the important characteristics of each digit in the model's representation. Figure 3 shows the instructions. To account for response ordering bias, the order of the methods presented for each question was randomized. In order to catch participants who cheat by making random choices, we included 5 attention checks with a single acceptable answer that is relatively obvious. We only include responses that pass at least 4 of the 5 attention checks and disregard the rest.

We conducted the experiment on MNIST with 28×28 single-channel images. We trained a 5-layer convolutional network that achieves 99.04% test accuracy. This model consisted of two blocks of convolution plus max-pooling operations, followed by three fully connected layers with

dropout in-between, and a final softmax operation. The number of hidden units for the three layers was 128, 128, and 10, respectively. Class-specific explanations were generated for each of the digits from 0 to 9.

It is important to note that none of the explanations generated for the study represented a particular prediction on a particular image, but instead represented how well the importance functions captured the important features for a label (digit) in the dataset.

Results and Discussion

After filtering responses that failed the attention check, we ended up with 83 completed surveys. From their responses, the number of votes for each method were: 939 for Norm-LIME, 438 for LIME, 151 for VarGrad, and 132 for Smooth-Grad. We analyzed the data by examining each user's response as a single sample of the relative proportions of the various explanation methods for that user, and performed a standard one-way ANOVA test against the hypothesis that the explanations were preferred uniformly. We obtained a statistically significant result, with an F statistic of 338 and a p-value of 5.22×10^{-100}, allowing us to reject the null hypothesis. We conclude that a significant difference exist between how users perceived the explanations.

A subsequent Tukey HSD post hoc test confirms that the differences between NormLIME and *all* other methods are highly statistically significant ($p < 0.001$). It also shows that the difference between LIME and the gradient-based interpretations is statistically significant ($p < 0.001$). We conclude that overall, the NormLIME explanations were preferred over all other baseline techniques, including LIME and that NormLIME and LIME were preferred over the gradient-based methods.

Observing Figure 1, the interpretations of SmoothGrad and VarGrad do not appear to resemble anything semantically meaningful. This may be attributed to the fact that these methods are not designed with class-level interpretation in mind. LIME captures the shape of the digits to some extent but cannot differentiate the most important pixels. In contrast, the normalization factor in NormLIME helps to illuminate the differences among the important pixels, resulting in easy-to-read interpretations. This suggests that proper normalization is important for class-level interpretations.

Numerical Evaluation with KAR

Visual inspection alone may not be sufficient for evaluating saliency maps (Adebayo *et al.* 2018). In this section, we further evaluate NormLIME using a technique akin to Keep And Retrain (KAR) proposed by Hooker *et al.* (2018). The underlying intuition of KAR is that features with low importance are less relevant to the problem under consideration. This gives rise to a principled "hypothesis of least damage": removal of the least important features as ranked by a feature importance method should impact the model's performance minimally. Thus, we can compare two measures of feature importance by comparing the predictive performance after removing the same number of features as ranked by each method as the least important.

Specifically, we first train the same convolutional network as in the human evaluation with all input features and use one of the importance scoring method to rank the features. We remove a number of least important features and retrain the model. Retraining is necessary as we want to measure the importance of the removed features to prediction rather than how much one trained model relies on the removed features. After that, we measure the performance drop caused by feature removal; a smaller performance drop indicates more accurate feature ranking from the interpretation method.

We perform KAR evaluation on two set of features. The first set is the raw pixels from the images. The second set of features are the output from the second convolutional layers of the network. The baselines and results are discussed below.

Baselines

We evaluated NormLIME against various baseline interpretation techniqes on the MNIST dataset (LeCun 1998). For NormLIME and LIME, we use the absolute value of $I(c_i)$ as the feature importance. In addition to the existing baselines used in the human evaluation, we introduce the following baselines.

SHAP (Lundberg and Lee 2017). The Shapley value measures the importance of a feature by enumerating all possible combinations of features and compute the average of performance drop when the feature is removed. The value is well suited to situations with heavy interactions between the features. While theoretically appealing, the computation is intractable. A number of techniques (Chen *et al.* 2019; Lundberg and Lee 2017; Strumbelj and Kononenko 2010) have been proposed to approximate the Shapley value. Here we use SHapley Additive exPlanations (SHAP), an approximation based on sampled least-squares.

Random. This baseline randomly assigns feature importance. This serves as a "sanity check" baseline. Notable, in the experiments of Hooker *et al.* (2018), some commonly used saliency maps perform worse than random.

Results and Discussion

Figure 6 shows the error gained after a number of least important features are removed averaged over 5 independent runs. We use removal thresholds from 10% to 90%. When 50% of the features or less are removed, NormLIME performs better or similarly with the best baselines, though it picks up more error when more features are removed. The best prediction accuracy among all is achieved by Norm-LIME at 50% feature reduction with 0% error gain. This is matched by SHAP also at 50% feature reduction and Var-Grad at 20% feature reduction. All other baselines observe about at least 0.25% error gain at 50% feature reduction,. SHAP and LIME perform better than other methods, including NormLIME, when 60% or more features are removed. The gradient-based methods, are outperformed by Norm-LIME and LIME.

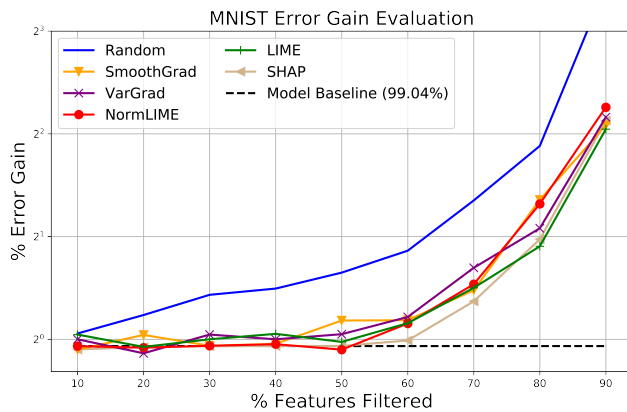Figure 6 shows the same measure on the convolutional features. On these features, NormLIME outperforms the

Figure 5: Error gain when the least important features are removed from the input. The horizontal axis indicate the percentage of of features removed. The vertical axis shows absolute error gain in log scale.
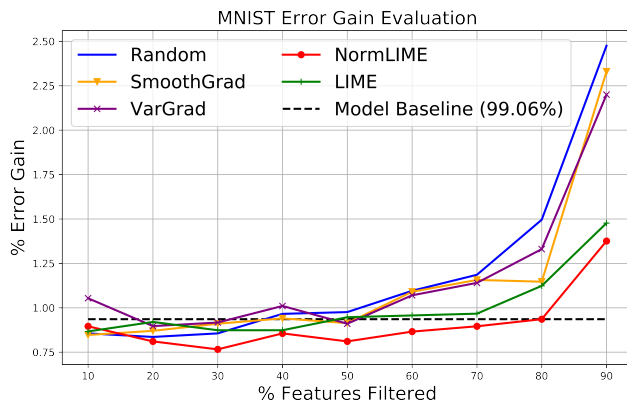

Figure 6: Error gain when the least important features are removed from the output of the convolutional layers. The horizontal axis indicate the percentage of of features removed. The vertical axis shows absolute error gain.

other methods by larger margins, compared to the input features. NormLIME achieves better results than the original model (at 99.06%) when 70% or less features are removed, underscoring the effectiveness of dimensionality reduction. The best performance is achieved at 30% removal with a classification accuracy of 99.31%. The second best is achieved by LIME at 99.1% improvement when filtering 40% of features, comparable with NormLIME performance at the same level.

When 80% of features are removed, NormLIME demonstrates zero error gain, whereas the second best method, LIME, gains 0.3% absolute error. When 90% of features are removed, NormLIME shows 0.45% error gain while LIME observes .6% error gain and all others receive at least 1.25%.

Overall, gradient ensemble methods, SmoothGrad and VarGrad, perform better than Random, but they compare unfavorable against "additive local model approximation"

schemes like SHAP, LIME, and NormLIME.

The advantage of NormLIME is more pronounced when pruning the convolutional features than the input features. Further, we can achieve better performance by removing convolutional features but not the input features. This suggests that there is more redundancy in the convolutional features and NormLIME is able to exploit that phenomenon.

Conclusions

Proper interpretation of deep neural networks is crucial for state-of-the-art AI technologies to gain the public's trust. In this paper, we propose a new metric for feature importance, named NormLIME, that helps human users understand a black-box machine learning model.

We extend the LIME / SP-LIME technique (Ribeiro *et al.* 2016), which generates local explanations of a large neural network and aggregates them to form a global explanation. NormLIME adds proper normalization to the computation of global weights for features. In addition, we propose label-based salience, which provides finer-grained interpretation in a multi-class setting compared to LIME which in contrast focuses on selecting an optimal selection of individual predictions to explain a model.

Experimental results demonstrate that the NormLIME explanations agree with human intuition of features that separate different digits in MNIST. The human evaluation study shows that explanations generated by the salience metric are strongly favored over comparable ones generated by LIME, SmoothGrad, and VarGrad with strong statistical significance. Further, using the Keep-And-Retrain evaluation method, we show that explanations formed by the Norm-LIME metric are faithful to the problem at hand, as it identifies input features and convolutional features whose removal is not only harmless, but may improve the prediction accuracy. By improving the interpretability of machine learning models, the proposed salience metric lays the groundwork for further adoption of AI technologies for the benefits of all of society.

Acknowledgments

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems 31*. 2018.

Marco Ancona, Enea Ceolini, Cengiz Oztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, 2010.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*, 2019.

Shi Feng and Jordan Boyd-Graber. What can AI do for me? evaluating machine learning interpretations in cooperative play. In *24th International Conference on Intelligent User Interfaces*, 2019.

Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2018.

Sara Hooker, Dumitru Erhan, Pieter jan Kindermans, and Been Kim. Evaluating feature importance estimates. *arXiv*, 2018.

Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. Global explanations of neural networks: Mapping the landscape of predictions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pages 279–287, New York, NY, USA, 2019. ACM.

Pieter-Jan Kindermans, Kristof T. Schütt, Klaus-Robert Müller Maximilian Alber, Dumitru Erhan, Been Kim, and Sven Dahne. Learning how to explain neural networks: Patternnet and patternattribution. 2017.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894, 2017.

Pat Langley. Explainable, normative, and justified agency. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

Yann LeCun. MNIST dataset, 1998. data retrieved from Yann Lecun's online archive, http://yann.lecun.com/exdb/mnist/.

Zachary C. Lipton. The mythos of model interpretability. In *2016 ICML Workshop on Human Interpretability in Machine Learning*, 2016.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. 2017.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recogn.*, 65(C):211–222, May 2017.

Nhathai Phan, Dejing Dou, Hao Wang, David Kil, and Brigitte Piniewski. Ontology-based deep learning for human behavior prediction with explanations in health social networks. *Information Sciences*, 384:298 – 313, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

Junghoon Seo, Jeongyeol Choe, Jamyoung Koo, Seunghyeon Jeon, Beomsu Kim, and Taegyun Jeon. Noise-adding methods of saliency map as series of higher order partial derivative. In *ICML Workshop*, 2018.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3145–3153, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. In *ICML Workshop*, 2017.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations*, 2015.

Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, pages 1–18, March 2010.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 2017.

Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. Learning global additive explanations for neural nets using model distillation. *ArXiv Preprint. ArXiv 1801.08640*, 2018.

Bolei Zhou, Yiyou Sun, David Bau, , and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint. ArXiv: 1806.02891*, 2018.