

Regularizing Black-box Models for Improved Interpretability

Gregory Plumb Maruan Al-Shedivat Eric Xing Ameet Talwalkar

Abstract

Most work on interpretability in machine learning has focused on designing either inherently interpretable models, that typically trade-off interpretability for accuracy, or post-hoc explanation systems, that lack guarantees about their explanation quality. We propose an alternative to these approaches by directly regularizing a black-box model for interpretability at training time. Our approach explicitly connects three key aspects of interpretable machine learning: the model’s innate explainability, the explanation system used at test time, and the metrics that measure explanation quality. Our regularization results in substantial (up to orders of magnitude) improvement in terms of explanation fidelity and stability metrics across a range of datasets, models, and black-box explanation systems. Remarkably, our regularizers also slightly improve predictive accuracy on average across the nine datasets we consider. Further, we show that the benefits of our novel regularizers on explanation quality provably generalize to unseen test points.

1. Introduction

Complex learning-based systems are increasingly shaping our daily lives, and, in order to monitor and understand these systems, we require clear explanations of model behavior. While model interpretability has many definitions and is often largely application specific (Lipton, 2016), local explanations are a popular and powerful tool (Ribeiro et al., 2016). Recent work on local interpretability in machine learning ranges from proposals of new models that are interpretable *by-design* (e.g., Wang & Rudin, 2015; Caruana et al., 2015) to model-agnostic, *post-hoc* algorithms for interpreting complex, black-box predictors such as ensembles and deep neural networks (e.g., Lei et al., 2016; Ribeiro et al., 2016; Lundberg & Lee, 2017; Selvaraju et al., 2017; Kim et al., 2018). Despite the variety of technical approaches, the underlying goal of all of these works is to

develop an interpretable predictive system that produces two outputs: a prediction and its underlying explanation.

Both interpretability by-design and post-hoc explanation strategies have limitations. On the one hand, the by-design approaches are restricted to working with model families that provide inherent explainability, potentially at the cost of accuracy. On the other hand, by performing two disjointed steps, there is no guarantee that post-hoc explainers applied to an arbitrary model will produce explanations of suitable quality. Moreover, recent approaches that claim to overcome this apparent tradeoff between prediction quality and explanation quality are in fact by-design proposals that impose certain constraints on the underlying model families they consider (Al-Shedivat et al., 2017; Plumb et al., 2018; Melis & Jaakkola, 2018). In this work, we propose a novel alternative strategy called *Explanation-based Optimization* (ExpO) that aims to address both of these shortcomings by adding an *interpretability regularizer* to the loss function of an arbitrary predictive model.

To motivate ExpO, consider a situation where Bob’s loan application is denied by a machine learning system (a toy illustration of the model is given in Figure 1). In this setting, a good local explanation can help Bob understand how to improve his application in order to get the loan. Unfortunately, as we see from Figure 1, a standard model—a multi-layer perceptron trained via SGD—is difficult to explain ‘faithfully’ because it has many kinks and abrupt changes. Indeed, we can quantitatively measure the faithfulness or quality of its local explanations using the standard *fidelity* and *stability* explanation metrics (see Sec. 2 for formal definitions of these metrics). In order to make the learned model more amenable to local explanation, ExpO augments its objective function with fidelity or stability-based regularizers, effectively controlling the degree of local smoothing.

The specific contributions of our work are threefold:

Interpretability Regularizers. We introduce two explanation regularizers associated with the fidelity and stability explanation metrics. The first, ExpO-Fidelity, is designed for semantic features and explainers that directly make predictions, such as Ribeiro et al. (2016); Lundberg & Lee (2017); Plumb et al. (2018). The second, ExpO-Stability, is tailored for non-semantic features (e.g., images) and explainers like Saliency Maps (Simonyan et al., 2013) which

Machine Learning Department, Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Gregory Plumb <gdplumb@andrew.cmu.edu>.

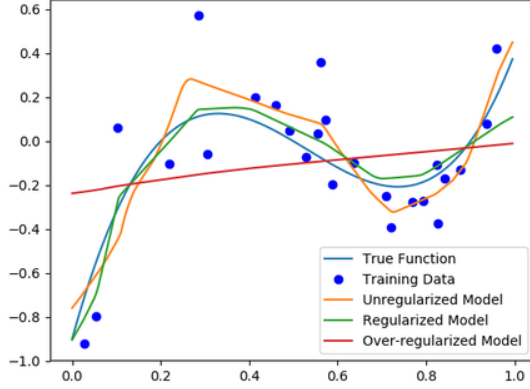


Figure 1. The effects of our regularizer on a model predicting Bob’s credit rating. The abrupt kinks in the unregularized model make the local linear approximations both less faithful to the model and less stable to small perturbations. The regularized model is much smoother and therefore easier to explain. Also observe that, when over-regularized, we can force the learned function to be essentially linear.

identify inputs that are influential on a prediction. Both regularizers are differentiable, can be used to augment the objective function of an arbitrary model, and are amenable to gradient-based optimization. In Sec. 3.1, we discuss how they differ from the classical approaches for local approximation and function smoothing.

Empirical Results. We evaluate a number of models trained with and without the proposed regularizers on a variety of regression and classification tasks with semantic and image features. We show experimentally that our regularizers in fact slightly improve predictive performance on average across our nine datasets (seven UCI regression tasks, a medical classification task, and MNIST). Moreover, from an interpretability perspective, our results demonstrate significant improvement in terms of explanation quality as measured by the fidelity and stability explanation metrics. In particular, our regularization technique improved explanation fidelity by at least 30% on the UCI datasets and by an order of magnitude on the medical classification task; stability on MNIST was improved by orders of magnitude.

Generalizable Explanation Quality. Finally, we analyze the properties of the explanation quality metrics and show that the benefits of our regularization generalize to unseen points. Specifically, we derive a bound on the gap between the fidelity of explanations on training and held out points and connect it with the local variance of the learned model.

2. Background and Related Work

In this section, we introduce our notation and provide the necessary background on local explanations and the fidelity and stability metrics commonly used to evaluate them.

Consider a supervised learning problem, where our goal is to estimate a model, $f : \mathcal{X} \mapsto \mathcal{Y} \mid f \in \mathcal{F}$, that maps input feature vectors, $x \in \mathcal{X}$, to targets, $y \in \mathcal{Y}$, and is trained using data, $\{x_i, y_i\}_{i=1}^N$. If the class of functions used for modeling the data is complex, we can understand the behavior of f in some neighborhood, $N_x \in \mathcal{P}[\mathcal{X}]$ (the space of probability distributions over \mathcal{X}), by generating a local *explanation*. We denote algorithms that produce explanations (i.e., *explainers*) as $e : \mathcal{X} \times \mathcal{F} \mapsto \mathcal{E}$, where \mathcal{E} is the output space of the explainer and defines the set of possible explanations. The choice of \mathcal{E} generally depends on whether or not \mathcal{X} consists of *semantic features*, and will be defined more precisely in the following subsections.

2.1. Semantic Features

These are features that people can reason about (e.g., a person’s income, the concentration of a chemical, etc.) and understand what it means when their values change. Consequently, local explanations try to predict how the model’s output would change if the input was perturbed (Ribeiro et al., 2016; Lundberg & Lee, 2017; Plumb et al., 2018). Hence, we can define the output space of the explainer as $\mathcal{E}_s := \{g \in \mathcal{G} \mid g : \mathcal{X} \mapsto \mathcal{Y}\}$, where \mathcal{G} is a class of interpretable (usually linear) functions.

FIDELITY-METRIC

When the explainer’s output space is \mathcal{E}_s , the explanation is defined as a function $g : \mathcal{X} \mapsto \mathcal{Y}$, and it is natural to evaluate how accurately g models f across the neighborhood N_x (Ribeiro et al., 2016). Thus, we define the fidelity of the explanation as:

$$L(f, e, N_x) := \mathbb{E}_{x' \sim N_x} [(g(x') - f(x'))^2], \quad (1)$$

which we call the *neighborhood-fidelity metric*.¹ This metric is sometimes evaluated with N_x as a point mass on x and we call this version of the metric the *point-fidelity metric*. While Plumb et al. (2018) argued that point-fidelity can be misleading because it does not measure how $e(x, f)$ generalizes across N_x , we report it in our experiments along with neighborhood-fidelity for completeness.

BLACK-BOX EXPLANATION SYSTEMS

Various explainers have been proposed to generate local explanations of the form $g : \mathcal{X} \mapsto \mathcal{Y}$, usually assuming that

¹For classification problems, the squared error can be computed directly on the logits or output probability vectors.

g is linear. In particular, LIME (Ribeiro et al., 2016), one of the most popular black-box explanation system,² solves the following optimization problem:

$$e(x, f) := \arg \min_{e \in \mathcal{E}_s} \mathcal{L}(f, e, N_x) + \Omega(e), \quad (2)$$

where $\Omega(e)$ stands for an additive regularizer that encourages certain desirable properties of the explanations (e.g., sparsity). Clearly, LIME’s formulation is closely related to the fidelity metric and subsequently the ExpO-Fidelity regularizer. Consequently, we would expect our regularizer to improve the quality of its explanations substantially. Our experimental results in Section 4.1 in fact corroborate this hypothesis, demonstrating a dramatic improvement in the quality of LIME’s local explanations while slightly increasing the predictive accuracy of the underlying model.

MAPLE (Plumb et al., 2018) is another black-box explanation tool that we consider in this work. This explanation system, derived from a by-design interpretable model based on tree ensembles, differs substantially from LIME in that its *neighborhood function is learned from the data* rather than given as a parameter, N_x . In our experiments, we evaluate the quality of MAPLE-generated local explanations for models regularized via ExpO-Fidelity, but do not embed MAPLE’s learned neighborhood function into ExpO-Fidelity. We view this experiment as a good test case to see how optimizing the fidelity metric for one neighborhood generalizes to another one (see Sec. 3 for a more detailed discussion of this point). Remarkably, we show empirically in Section 4.1 that regularizing for LIME neighborhoods in fact improves MAPLE’s explanation quality.

2.2. Non-Semantic Features

Non-semantic features lack an inherent interpretation. Images are the canonical example because in general it is not clear what it means to perturb pixel values or whether the resulting perturbation is even a ‘real’ image.³ When \mathcal{X} consists of non-semantic inputs, we cannot interpret the difference between x and x' , so it does not make sense to explain the difference between $f(x)$ and $f(x')$. As a result, fidelity is not an appropriate explanation metric. Instead, in this context, local explanations try to identify which parts of the input were particularly influential on a prediction (Sundararajan et al., 2017) and consequently we consider explanations of the form $\mathcal{E}_{ns} := \mathbb{R}^d$, where d is the number of features in \mathcal{X} .

²SHAP (Lundberg & Lee, 2017) is a popular variation of LIME that proposes a theoretically-motivated neighborhood sampling function, but requires explanations to be linear models that act on binary features. This requirement is too limiting in our case, hence SHAP is not used in our study.

³ In certain cases, such as scientific imaging (de Oliveira et al., 2016), each pixel value may have a precise meaning because of the way the images are processed.

STABILITY METRIC AND SALIENCY MAPS

When the explainer’s output space is \mathcal{E}_{ns} , the explanation is a vector in \mathbb{R}^d , and cannot be directly compared to the underlying model itself, as in the case of the fidelity metric. Instead, the focus in this setting is on the degree to which the explanation changes between points in a local neighborhood (Melis & Jaakkola, 2018). Consequently, we define the stability of $e(x, f)$ with respect to f across N_x as:

$$\mathcal{S}(e, f, N_x) := \mathbb{E}_{x' \sim N_x} [d(e(x, f), e(x', f))] \quad (3)$$

where $d(\cdot, \cdot)$ is a suitable distance measure between d -dimensional vectors.

Various explainers (Adebayo et al., 2018; Melis & Jaakkola, 2018) have been proposed to generate local explanations in \mathcal{E}_{ns} , with Saliency Maps (Simonyan et al., 2013) being one popular approach that we consider in this work. Saliency Maps assign importance to the pixels of an image based on the magnitude of the gradient of the predicted class with respect to the image.

Note that the stability metric can also be considered in the context of semantic features in addition to the fidelity metric. In our experiments with semantic features, we report results for both metrics. In either case, more stable explanations tend to be more trustworthy (Melis & Jaakkola, 2018).

2.3. Self-explaining Neural Networks

We conclude this section by discussing a highly relevant related explanation framework called self-explaining neural networks (SENN) (Melis & Jaakkola, 2018). At a high level, SENN can also be viewed as performing explanation optimization, as it proposes a regularizer that indirectly optimizes the stability metric. However, SENN is interpretable by-design, and as discussed in Section 1, this implies that 1) SENN works with a restricted model family that in practice adversely impacts predictive accuracy (e.g., increases the error on MNIST from less than 1% to 3%), and 2) the SENN regularizer cannot be applied to an arbitrary model.

Moreover, ExpO and SENN differ significantly from a technical perspective. SENN produces local explanations by splitting the network into two components: $h(x)$ which learns a feature embedding of x and $\theta(x)$ which learns the coefficients of the linear explanation that act on $h(x)$. It indirectly optimizes the model for explanation stability using $\|\nabla_x f(x) - \theta(x)^T J_x h(x)\|$ as a regularizer. When we have semantic inputs (i.e., $h(x) = x$), we can see that this encourages $\theta(x)$ to be close to the first order Taylor approximation of $f(x)$ at x . In Sec. 3.1, we demonstrate how Taylor approximations are very different from and more difficult to use than the neighborhood-based local explanations used by ExpO.

3. Explanation Optimization

Running black-box explainers on arbitrary models does not guarantee the quality of the produced explanations. To address this, we define a regularizer that can be added to the loss function and used to train an arbitrary model f . The local explanations generated for such model are expected to be of higher quality, as measured by fidelity or stability metrics. Specifically, we propose to learn a model from data, $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, regularized for explainability by solving the following optimization problem:

$$\hat{f} := \arg \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f, x_i, y_i) + \gamma \mathcal{R}(f, N_x^{\text{reg}}), \quad (4)$$

where $\mathcal{L}(f, x_i, y_i)$ is a standard predictive loss (*e.g.*, squared error for regression or cross-entropy for classification), $\mathcal{R}(f, N_x^{\text{reg}})$ is a regularizer that encourages explainability of f in the neighborhood of x_i , and $\gamma > 0$ controls regularization strength. Note that while our regularizer is data-dependent, it also decomposes into a sum over the training points, which makes the whole objective amenable to stochastic optimization.

We define $\mathcal{R}(f, N_x^{\text{reg}})$ based on either the neighborhood-fidelity, Eq. (1), or the neighborhood-stability, Eq. (3). Note that to compute these metrics, we need to run an explainer algorithm, e , that would produce local explanations. However, the process of fitting the explainer might be non-differentiable or just too computationally expensive to optimize directly. Therefore, we approximate the explanation with a local linear regression fit across some neighborhood, N_x^{reg} , for the fidelity metric. The stability metric is approximated by the variance of the model f across the selected neighborhood, N_x^{reg} . Finally, we approximate the expectations in Eq. (1) and Eq. (3) by sampling points from N_x^{reg} .

Defining a good regularization neighborhood, N_x^{reg} , requires to take the following into consideration. On the one hand, we would like N_x^{reg} to be similar to N_x as defined in Eq. 1 or Eq. 3, so that the neighborhoods used for regularization and for evaluation match. On the other hand, we also would like N_x^{reg} to be consistent with the ‘local neighborhood’ defined by the explainer algorithm e internally, which may differ from N_x . For LIME, this is not a problem since the internal definition of the ‘local neighborhood’ is a hyperparameter that we can set. However for MAPLE, ‘local neighborhood’ is learned from the data, and hence the regularization and explanation neighborhoods may differ. Ultimately, we left resolving this tension to future work.

Algorithm 1 summarizes our proposed approach to learning with ExpO regularization. The key idea is to keep estimating fidelity or stability of the model in the neighborhood of each point in the mini-batch using either Algorithm 2 or 3, and using the obtained value (as a function the model parameters)

Algorithm 1 Learning with ExpO regularization.

input \mathcal{D} – dataset, γ – regularization coefficient,
 N_x^{reg} – neighborhood sampling function,
 T – number of optimization epochs.
 1: Initialize model f with parameters θ_0 .
 2: **for** t in $0 \dots T$ **do**
 3: **for** each of mini-batch \mathcal{B} in \mathcal{D} **do**
 4: **for** each point x_i in \mathcal{B} **do**
 5: Regularizer: $r(\theta, x_i) = \mathcal{R}(f_\theta, N_{x_i}^{\text{reg}})$
 (compute using Algorithm 2 or 3).
 6: **end for**
 7: Construct loss:

$$L(\theta) := \frac{1}{|\mathcal{B}|} \sum_{x_i \in \mathcal{B}} \mathcal{L}(\theta, x_i) + \gamma r(\theta, x_i).$$

 8: Update model: $\theta_{t+1} \leftarrow \text{update}(\theta_t, \nabla L(\theta_t))$.
 9: **end for**
 10: **end for**
output Learned model, f_{θ_T} .

Algorithm 2 Neighborhood-fidelity regularizer.

input f_θ – model, x – point of interest, m – number of
 samples, N_x^{reg} – neighborhood sampler.
 1: Sample points: $x'_1, \dots, x'_m \sim N_x^{\text{reg}}$.
 2: Compute predictions:

$$\hat{y}_j(\theta) = f_\theta(x'_j), \quad j = 1, \dots, m.$$

 3: Produce a local linear explanation:

$$\beta_x(\theta) = \arg \min_{\beta} \sum_{j=1}^m (\hat{y}_j(\theta) - \beta^\top x'_j)^2$$

output Fidelity, $\frac{1}{m} \sum_{j=1}^m (\hat{y}_f(\theta) - \beta_x(\theta)^\top x'_j)^2$.

Algorithm 3 Neighborhood-stability regularizer.

input f_θ – model, x – point of interest, m – number of
 samples, N_x^{reg} – neighborhood sampler.
 1: Sample points: $x'_1, \dots, x'_m \sim N_x^{\text{reg}}$.
 2: Compute predictions:

$$\hat{y}_j(\theta) = f_\theta(x'_j), \quad j = 1, \dots, m.$$

output Stability, $\frac{1}{m} \sum_{j=1}^m (\hat{y}_j(\theta) - f(x))^2$, as measured by
 the variance of the model in the neighborhood, N_x^{reg} .

as the regularizer. Note that Algorithm 3, instead of estimating local explanations and computing the neighborhood-stability as given in Eq. 3, uses a faster, heuristic approximation, akin to the variance of the model in a neighborhood.⁴

⁴We note that a similar procedure was explored previously by Zheng et al. (2016) for adversarial robustness.

Limitations. From a computational perspective, Algorithm 2 can be fairly expensive since the number of samples from the neighborhood, m , has to be chosen at least larger than the dimensionality of x . Instead, we could fit a regularized linear model using a smaller number of points, somewhat mitigating the computational cost while potentially affecting generalization of our metrics. In Algorithm 3, we trade off the ability for the model to freely change across the local neighborhood (as long as it does so linearly) for the computational benefit of not having to fit the linear model or estimate its error. Because N_x^{reg} tends to be a ‘narrow’ distribution for models with non-semantic inputs, the loss in explanation flexibility is not a substantial trade-off (we do not observe such effects empirically). Other potential approaches to improving computational properties of our regularizers are left for future research.

3.1. Understanding the Properties of ExpO

The goal of this section is to compare the behavior of local linear explanations and our regularizer to some existing theoretical function approximations and measures of variance to help develop an intuitive understanding of ExpO. First, we compare the neighborhood based local linear explanations to first order Taylor approximations to show that they can have fundamentally very different behaviors. Second, we compare our fidelity-regularizer to the Lipschitz Constant (LC) and Total Variation (TV) of the learned function.

Local Explanation vs Taylor Approximations. A natural question to ask is: *Why should we sample N_x in order to locally approximate f when there are theoretically motivated ways of doing this?* One possible way to do this is via the Taylor approximation, which is closely related to SENN’s regularization approach. The primary downside of a Taylor approximation-based approach is that such an approximation cannot readily be adjusted to different neighborhood scales and their fidelity/stability are strictly a function of the learned function. This can be seen in Figure 2 where the Taylor approximations at two nearby points are both radically different and not faithful to the model outside of a small neighborhood. Note that, because their model was regularized for Taylor approximation-based explanations, it is unlikely that SENN suffers from the dramatic failures demonstrated here.

Fidelity-Regularization and the Model’s LC or TV. From a theoretical perspective, our regularizer is similar to controlling the Lipschitz Constant or Total Variation of f across N_x after removing the part of f explained by $e(x, f)$. From an interpretability perspective, there is nothing inherently wrong with having a large LC or TV, which is demonstrated in Figure 3. However, once we take into account what can be explained by $e(x, f)$, then upper bounding any

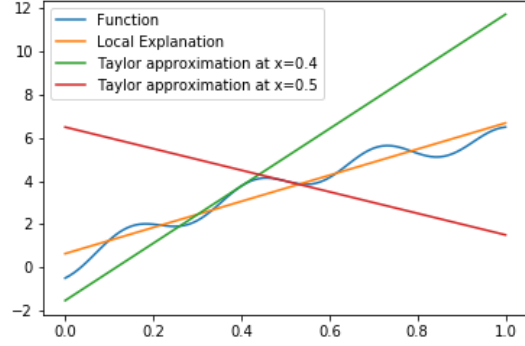


Figure 2. A function (blue), its first order Taylor approximations at $x = 0.4$ (green) and $x = 0.5$ (red), and a local explanation of the function (orange) computed with $x = 0.5$ and $N_x = [0, 1]$. Note that the Taylor approximations are (i) accurate only within a very small region around the point and (ii) highly unstable.

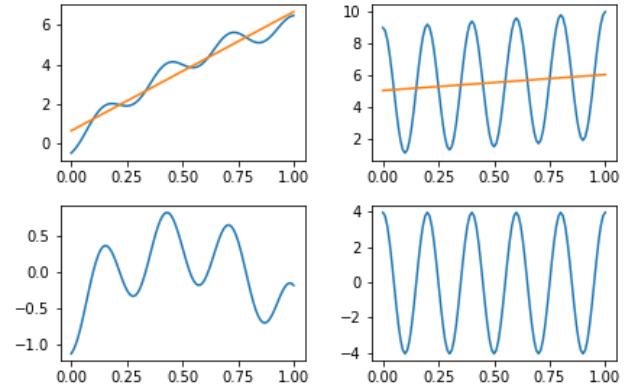


Figure 3. **Top row:** Two functions (blue) and their local linear explanations (orange). The local explanations were computed with $x = 0.5$ and $N_x = [0, 1]$. **Bottom row:** The unexplained portion of the function (residuals). **Conclusion:** Despite the fact that both of the functions have large Lipschitz constants (LC) or Total Variation (TV), the one on the left is well explained while the one on the right is not. This is evident from the variance, TV, or LC of the residuals.

one of our regularizer, the LC, or the TV will upper bound the remaining ones.

3.2. Generalization of Local Linear Explanations

We conclude this section with the analysis of the quality of local linear explanations in terms of generalization. Note that ExpO regularization encourages learning models that are explainable in the neighborhoods of each *training point*. However, how would this property generalize?

We answer this question by providing a generalization bound in terms of neighborhood-fidelity metric. First, we assume that local linear explanations, β_x , are obtained by solving the ordinary least squares regression problem (as given in

Algorithm 2). The fidelity of the explanation in expectation over the neighborhood N_x can be computed analytically:

$$\begin{aligned} r(f, x) = & \mathbb{E}_{N_x} [f(x')^2] - \\ & \mathbb{E}_{N_x} [f(x')x']^\top \mathbb{E}_{N_x} [x'x'^\top]^{-1} \mathbb{E}_{N_x} [f(x')x'] \end{aligned} \quad (5)$$

where expectation $\mathbb{E}_{N_x} [\cdot]$ is taken with respect to x' over the neighborhood N_x . Note the expression in (5) is the expected value of the squared residual between $f(x)$ and the optimal local linear explanation, which is upper-bounded by the variance of the model in the corresponding neighborhood:

$$\begin{aligned} 0 \leq r(f, x) & \leq \mathbb{E}_{N_x} [f(x')^2] - \mathbb{E}_{N_x} [f(x')]^2 \\ & = \text{Var}_{N_x} [f(x')] \end{aligned} \quad (6)$$

For instance, if $f(x)$ is L -Lipschitz and the neighborhood N_x is defined a uniform distribution within a σ -ball centered at x , then the variance of $f(x)$ within the neighborhood can be further bounded by $4L^2\sigma^2$, hence $r(f, x) \leq 4L^2\sigma^2$.

For the explanations to generalize, we would like to make sure that the gap between the average fidelity on the training set and the expected fidelity is small with high probability. More formally, the following inequality should hold:

$$\mathbb{P} \left(\mathbb{E} [r(f, x)] - \frac{1}{n} \sum_{i=1}^n r(f, x_i) > \varepsilon \right) < \delta_n(\varepsilon) \quad (7)$$

Under certain mild assumptions on the local behavior of $f(x)$, the following proposition specifies a particular bound.

Proposition 1 *Let the neighborhood sampling function N_x be characterized by some parameter σ (e.g., the effective radius of a neighborhood) and the variance of the trained model $f(x)$ across all such neighborhoods be bounded by some constant $C(\sigma) > 0$. Then, the following bound holds with at least $1 - \delta$ probability:*

$$\mathbb{E} [r(f, x)] \leq \frac{1}{n} \sum_{i=1}^n r(f, x_i) + \sqrt{\frac{C^2(\sigma) \log \frac{1}{\delta}}{2n}} \quad (8)$$

Proof. By assumption, the variance of the model $f(x)$ is bounded in each local neighborhood specified by N_x . Then (6) implies that each residual is bounded as $0 \leq r(f, x) \leq C(\sigma)$. Applying Hoeffding’s inequality, we get:

$$\mathbb{P} \left(\mathbb{E} [r(f, x)] - \frac{1}{n} \sum_{i=1}^n r(f, x_i) > \varepsilon \right) < \exp \left\{ \frac{-2n\varepsilon^2}{C^2(\sigma)} \right\}$$

Inverting the inequality gives us the bound in (8). ■

Remark 2 *The obtained bound tells us that explainable models with smaller local variances across the neighborhoods are likely to have high-fidelity explanations on the held out points as well. This additionally motivates the approximation used in Algorithm 3.*

Table 1. Statistics of the various datasets used in the experiments.

Dataset	# samples	# dimensions
autmpgs	392	8
communities	1993	103
day	731	15
happiness	578	8
housing	506	12
music	1059	70
winequality-red	1599	12
SUPPORT2	9104	51
MNIST	60000	784

4. Experimental Results

In the first of our two experimental sections, we demonstrate the effectiveness of our local-fidelity regularizer on datasets with semantic features.⁵ We do this on several regression problems from the collection of UCI (Dheeru & Karra Taniskidou, 2017) as well as in-hospital mortality classification problem.⁶ We also discuss 1) how to interpret our metrics, 2) how to apply them to models with multi-dimensional outputs, and 3) how soft-max cross-entropy as a loss function, when used without ExpO, may lead to poorly explainable models.

Our second experiment demonstrates the effectiveness of our stability metric for creating saliency maps (Simonyan et al., 2013) on MNIST (LeCun, 1998). We show, quantitatively, that the explanations become far more stable and, qualitatively, that they make more sense.

4.1. Local-Fidelity Regularization

In this section, we fit small neural networks to several regression problems from the UCI datasets and an in-hospital mortality classification problem. We compare models with and without our regularizer and report accuracy and three interpretability metrics: 1) Point-Fidelity (PF), 2) Neighborhood-Fidelity (NF), 3) Stability (S) for explanations generated by LIME and MAPLE. In addition to these metrics, we also report the variance of the model within the neighbourhood, N_x , averaged across the test points.⁷

Experimental Setup. Due to computational considerations, we slightly modified the training process in comparison to the idealized approach outlined in Algorithm 1. Specifically,

⁵The code for our regularizers and all experiments is at: <https://github.com/Anonymous/ExpO>

⁶<http://biostat.mc.vanderbilt.edu/wiki/Main/SupportDesc>

⁷The variance is typically smaller since the evaluation neighborhood, N_x , is smaller than the one used by LIME and fundamentally different from the one used by MAPLE.

we drew separate minibatches for the model training loss and for the regularizer loss and we also used a stopping condition rather than a fixed number of epochs. The network architectures and hyperparameters were chosen by a simple grid search. All inputs were standardized to have mean zero and variance one (including the response variable for regression problems). Both N_x and N_x^{reg} were set to be $\mathcal{N}(x, 0.1)$. The stability metric was defined using $d(\cdot, \cdot)$ as the ℓ_2 distance between the coefficients of the linear explanations. The reported numbers are averaged across ten trials.

UCI Regression Experiments. The effects of the Local-Fidelity regularizer on model accuracy and interpretability metrics are in Table 2. Generally, it had a slight positive effect on accuracy: on the ‘communities’ dataset it made it worse while on the ‘happiness’ and ‘winequality-red’ datasets it made it better. With the exception of the fidelity metrics on the ‘music’ dataset, almost all of the explanation metrics were decreased by 30% or more.

Point-Fidelity vs Neighborhood-Fidelity. Plumb et al. (2018) presented a theoretical argument as to why the Neighborhood-Fidelity metric may be more meaningful than the Point-Fidelity metric. Our results confirmed that their algorithm (MAPLE) can sometimes have substantially different values for these two, as seen on ‘communities’ for both unregularized and regularized models. In contrast, given that LIME is trained based on the Neighborhood-Fidelity metric, we would not expect these two metrics to differ and our experimental results support that.

Medical Classification Experiments. The SUPPORT2 dataset is used for in-hospital mortality classification. The output layer of our models is softmax over logits for two classes. Consequently, we run our explanation systems on the logits for each class. Table 3 presents the results. We observe that the regularizer did not affect the accuracy but did dramatically improve the interpretability metrics.

The Dangers of Interpreting Soft-Max Activations. At a high level, using soft-max with cross-entropy does not necessarily eliminate the random variations in the learned function that are the result of the initialization process. Consequently, local explanations should either 1) explain the difference between the logits of ‘Class A’ and ‘Class B’, 2) explain the soft-max activation of ‘Class A’, or 3) be regularized for smoothness.

One of the properties of soft-max is that it is shift-invariant. Consequently, cross-entropy only directly supervises the difference between the logits at any given point. Subsequently, any ‘wiggleness’ in the logit functions that was the result of the random initialization can potentially be preserved throughout the training process. We observe that this is the case experimentally by looking at the average variance of

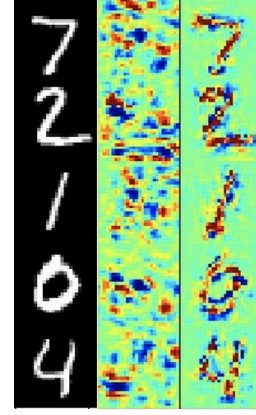


Figure 4. Left) Original MNIST image, Center) Unregularized Model’s Saliency Map, Right) Regularized Model’s Saliency Map.

the unregularized model, which ranges from 0.04 to 4.77 with a standard deviation of 2.02. In comparison, the standard deviation of the variance is 0.002 for the regularized model.

Fundamentally, this ‘wiggleness’ is a problem from an interpretability perspective because it does not correspond to any real world phenomena, as it is the result of the random initialization of the network, and yet it may be the overwhelming signal that our local explanation systems are modeling. One potential solution is to normalize the logits against a reference class, so that the explainer is explaining why the prediction was ‘Class A’ and not ‘Class B’; this compares the difference of logits across the neighborhood (rather than their actual value) and is what Soft-max Cross-Entropy supervises. However, this is not the same as estimating what features contribute to ‘Class A’ directly and one of the additional benefits of our regularizer is that it allows users to safely make this type of explanation.

4.2. Stability Regularization

In this section we fit a convolutional neural network to MNIST and then evaluate the stability of its saliency maps to perturbations where $N_x = N_x^{reg} = \text{Uniform}(x - 0.05, x + 0.05)$. Both the unregularized and the regularized and regularized models achieved an accuracy of 99%. This demonstrates one of the practical differences between SENN and our regularizers: SENN places strict structural constraints on the network and subsequently achieves a testing accuracy of roughly 97% while our model can be applied to any network. Further, our regularizer decreased the average ℓ_2 distance between the explanation at x and some $x' \sim N_x^{reg}$ from 6.94 to 0.0008. Finally, it qualitatively makes the resulting saliency maps look much better as seen in Figure 4.

Table 2. The results of comparing a model trained with ExpO-Fidelity to one trained with it on seven UCI regression datasets. Values shown are ‘mean (standard deviation)’ across 10 trials; bold indicates a statistically significant difference at $p = 0.05$. In addition to a slight increase in accuracy, ExpO-Fidelity almost always substantially increased the quality of the local explanations.

	Reg	autompgs	communities	day	happiness	housing	music	winequality.red
MSE	N	0.16 (0.033)	0.43 (0.039)	0.00051 (7e-04)	0.00038 (0.00015)	0.13 (0.052)	0.69 (0.1)	0.83 (0.067)
	Y	0.15 (0.034)	0.52 (0.078)	0.00016 (0.00015)	0.00013 (0.00015)	0.16 (0.047)	0.74 (0.084)	0.66 (0.042)
Variance	N	0.0072 (0.0013)	0.0093 (0.00078)	0.0071 (0.00029)	0.044 (0.004)	0.012 (0.0014)	0.0074 (0.00069)	0.027 (0.01)
	Y	0.0066 (0.00086)	0.0047 (0.0011)	0.0072 (0.00029)	0.047 (0.0037)	0.0066 (0.0017)	0.0041 (0.00062)	0.0075 (0.0021)
MAPLE-PF	N	0.039 (0.021)	0.16 (0.014)	5e-04 (7e-04)	0.00037 (0.00016)	0.092 (0.036)	0.19 (0.048)	0.22 (0.084)
	Y	0.025 (0.01)	0.12 (0.047)	8.7e-05 (8.8e-05)	9.5e-05 (0.00014)	0.04 (0.021)	0.19 (0.052)	0.056 (0.019)
MAPLE-NF	N	0.042 (0.022)	0.31 (0.052)	0.00054 (0.00073)	0.00039 (0.00015)	0.1 (0.033)	0.2 (0.049)	0.22 (0.086)
	Y	0.028 (0.012)	0.21 (0.082)	1e-04 (9.7e-05)	9.2e-05 (0.00013)	0.045 (0.024)	0.19 (0.054)	0.058 (0.02)
MAPLE-S	N	0.033 (0.012)	1.3 (0.26)	1.4e-06 (1.9e-06)	1.1e-07 (9.2e-08)	0.21 (0.16)	0.14 (0.066)	0.019 (0.0067)
	Y	0.021 (0.0083)	0.66 (0.22)	7.7e-05 (7.4e-05)	8.1e-07 (1.1e-06)	0.09 (0.058)	0.069 (0.033)	0.011 (0.0044)
LIME-PF	N	0.061 (0.012)	0.12 (0.0095)	0.00034 (3e-04)	0.00034 (0.00011)	0.16 (0.023)	0.12 (0.021)	0.34 (0.14)
	Y	0.044 (0.011)	0.097 (0.034)	0.00013 (6e-05)	0.0011 (0.00055)	0.07 (0.017)	0.1 (0.024)	0.088 (0.031)
LIME-NF	N	0.062 (0.012)	0.12 (0.01)	0.00036 (3e-04)	0.00037 (0.00011)	0.16 (0.022)	0.12 (0.021)	0.34 (0.14)
	Y	0.045 (0.011)	0.098 (0.034)	0.00013 (6.2e-05)	0.0011 (0.00059)	0.071 (0.017)	0.1 (0.024)	0.09 (0.031)
LIME-S	N	0.0026 (0.0015)	0.037 (0.0037)	6e-05 (4.4e-05)	2.4e-05 (9.6e-06)	0.0072 (0.001)	0.02 (0.0031)	0.024 (0.0099)
	Y	0.0016 (0.00053)	0.019 (0.0067)	1.1e-05 (4.7e-06)	0.00021 (0.00011)	0.0032 (0.0013)	0.011 (0.002)	0.0042 (0.0015)

Table 3. The results of comparing a model trained with ExpO-Fidelity to one trained with it on the SUPPORT2 classification dataset. Values shown are ‘mean (standard deviation)’ for the logits of ‘Class 1’ and ‘Class 2’ across 10 trials. Notice that the unregularized model has both much larger and more variable metrics.

Class	Unregularized		Regularized	
	Positive	Negative	Positive	Negative
Variance	1.84 (2.08)	1.86 (2.08)	0.014 (0.0023)	0.014 (0.0023)
MAPLE-PF	9.98 (11.9)	9.54 (11.3)	0.125 (0.021)	0.126 (0.021)
MAPLE-NF	11.2 (13.2)	10.7 (12.7)	0.135 (0.021)	0.136 (0.021)
MAPLE-S	32.2 (37)	30.4 (35.5)	0.376 (0.1)	0.374 (0.088)
LIME-PF	10.7 (12.7)	10.5 (12.3)	0.225 (0.044)	0.225 (0.042)
LIME-NF	11.1 (13.2)	10.9 (12.8)	0.231 (0.046)	0.23 (0.044)
LIME-S	2.58 (3.1)	2.54 (3.04)	0.0279 (0.0086)	0.028 (0.0087)
Accuracy	0.847 (0.0059)		0.851 (0.0021)	

5. Conclusion

In this work, we have introduced the novel idea of directly regularizing arbitrary models to be more interpretable. We contrasted these regularizers to classical approaches for function approximation and smoothing and provided a generalization bound for them. We demonstrated that this regularization could be done, surprisingly, while slightly improving the accuracy of the model and simultaneously decreasing the interpretability metrics by somewhere from 30% to orders of magnitude, across a variety of problem settings and domains. Finally, we identified a fundamental problem with interpreting individual logits of classification

models trained with soft-max activations for unregularized networks.

We believe that potential future work may focus on two areas. First, exploring alternative neighborhood functions N_x^{reg} that match those used by other black-box explanation systems, such as MAPLE. Second, methods for making our regularizer more computationally efficient, which range from using it as a secondary training stage after a model was trained normally, to adapting methods from Lipschitz constant or total variation regularization to the interpretability setting.

References

- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pp. 9525–9536, 2018.
- Al-Shedivat, M., Dubey, A., and Xing, E. P. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.
- Caruana, R. et al. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1721–1730. ACM, 2015.
- de Oliveira, L., Kagan, M., Mackey, L., Nachman, B., and Schwartzman, A. Jet-images—deep learning edition. *Journal of High Energy Physics*, 2016(7):69, 2016.
- Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pp. 2673–2682, 2018.
- LeCun, Y. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.
- Lipton, Z. C. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.
- Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.
- Melis, D. A. and Jaakkola, T. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pp. 7785–7794, 2018.
- Plumb, G., Molitor, D., and Talwalkar, A. S. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pp. 2516–2525, 2018.
- Ribeiro, M. T., Singh, S., and Guestrin, C. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM, 2016.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626. IEEE, 2017.
- Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.
- Wang, F. and Rudin, C. Falling rule lists. In *Artificial Intelligence and Statistics*, pp. 1013–1022, 2015.
- Zheng, S., Song, Y., Leung, T., and Goodfellow, I. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4480–4488, 2016.