LEARNING GLOBAL ADDITIVE EXPLANATIONS FOR NEURAL NETS USING MODEL DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability has largely focused on local explanations, i.e. explaining why a model made a particular prediction for a sample. These explanations are appealing due to their simplicity and local fidelity. However, they do not provide information about the general behavior of the model. We propose to leverage model distillation to learn global additive explanations that describe the relationship between input features and model predictions. These global explanations take the form of feature shapes, which are more expressive than feature attributions. Through careful experimentation, we show qualitatively and quantitatively that global additive explanations are able to describe model behavior and yield insights about models such as neural nets. A visualization of our approach applied to a neural net as it is trained is available at `https://youtu.be/ErQYwNqzEdc`.

1 INTRODUCTION

Recent research in interpretability has focused on developing *local* explanations: given an existing model and a sample, explain why the model made a particular prediction for that sample (Ribeiro et al., 2016). The accuracy and quality of these explanations have rapidly improved, and they are becoming important tools to understand model decisions for individual samples. However, the human cost of examining multiple local explanations can be prohibitive with today's large data sets, and it is unclear whether multiple local explanations can be aggregated without contradicting each other (Ribeiro et al., 2018; Alvarez-Melis & Jaakkola, 2018).

In this paper, we are interested in *global* explanations that describe the overall behavior of a model. While usually not as accurate as local explanations on individual samples, global explanations provide a different, complementary view of the model. They allow us to clearly visualize trends in feature space, which is useful for key tasks such as understanding which features are important, detecting unexpected patterns in the training data and debugging errors learned by the model.

We propose to use model distillation techniques (Bucilua et al., 2006; Hinton et al., 2015) to learn global additive explanations of the form

$$\hat{F}(\mathbf{x}) = h_0 + \sum_i h_i(x_i) + \sum_{i \neq j} h_{ij}(x_i, x_j) + \sum_{i \neq j} \sum_{j \neq k} h_{ijk}(x_i, x_j, x_k) + \cdots \tag{1}$$

to approximate the prediction function of the model, $F(\mathbf{x})$. Figure 1 illustrates our approach. The output of our approach is a set of p feature shapes $\{h_i\}_1^p$ that can be visually inspected, used for feature attribution, and composed to form an explanation model that can be quantitatively evaluated. Through controlled experiments, we empirically validate that these feature shapes provide accurate and interesting insights into the behavior of complex models. In this paper, we focus on interpreting F from fully-connected neural nets trained on tabular data.

Our goal is not to replace local explanations, nor to explain how the model functions internally. What we claim is that we can complement local explanations with global additive explanations that visualize the input-output relationship between features and predictions. Our contributions are:

- We propose to *learn* global additive explanations for complex, non-linear models such as neural nets. These explanations do not aim at competing with local explanations, and instead complement them to shed a different light into the models.

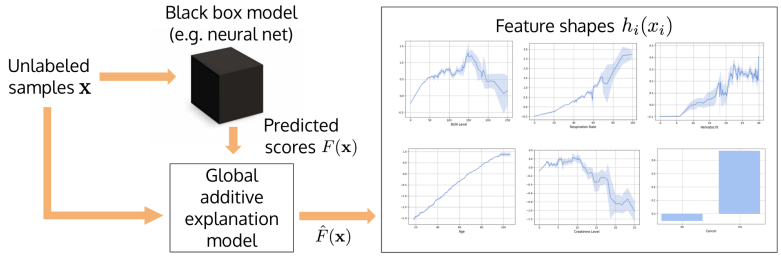

Figure 1: Given a black box model and unlabeled samples (new unlabeled data or training data with labels discarded), our approach leverages model distillation to learn feature shapes that describe the relationship between input features and model predictions.

- We leverage powerful tree- and spline-based additive models in a model distillation setting to learn global feature shapes that are more expressive than feature attributions.
- We quantitatively evaluate different *global* explanation methods in terms of fidelity to the model being explained and accuracy on independent test data.
- Through controlled experiments, we show that these global explanations can provide accurate and interesting insights into the behavior of complex models.

2 Learning Global Additive Explanations

Although our approach of using model distillation with powerful additive models of the form in equation 1 is new, our work is based on two previous research threads: (1) *decomposing* F into additive $\hat{F}$ to understand how F is affected by its inputs (e.g. Hooker (2007)), and (2) *learning* an interpretable model (often some form of decision tree) to mimic F (e.g. Craven & Shavlik (1995)).

2.1 Additive $\hat{F}$

Global additive explanations have been used to analyze inputs to complex, nonlinear mathematical models and computer simulations (Sobol, 2001), analyze how hyperparameters affect the performance of machine learning algorithms (Hutter et al., 2014), and decompose prediction functions into lower-dimensional components (Hooker, 2004). They are determined by the choice of metric L between F and its approximation $\hat{F}$, degree d of highest order components ($d = 3$ in equation 1, and type of base learner h. One common theme shared by these methods is that they *decompose* F into $\hat{F}$ using numerical or computational methods (e.g. matrix inversion, quasi Monte Carlo).

Rather than approximately decomposing $\hat{F}$ (which can be prohibitively expensive with large n or p)[1], we propose to *learn* $\hat{F}$ using model distillation. This is equivalent to choosing L that minimizes the empirical risk between the prediction function F and our global additive explanation $\hat{F}$ on the training data. By limiting $d = 1$, we investigate how much of F can be approximated by feature shapes of main components h_i alone, and study the impact of each input feature on F. To minimize approximation error $||F - \hat{F}||_L$, we select two flexible, nonparametric base learners for h: splines (Wood, 2006) and bagged trees. This gives us two global additive explanation models: **Student Bagged Additive Boosted Trees (SAT)** and **Student Additive Splines (SAS)**. Other choices of h are possible. We describe our distillation setup to learn these models in Section 2.3.

2.2 Interpretable Building Blocks of $\hat{F}$: Feature shapes

Our global additive explanation models, SAT and SAS, can be visualized as feature shapes (Figure 1). These are plots with x-axis being the domain of input feature x_i and y-axis being the feature's contribution to the prediction $h_i(x_i)$. This class of model is also used in other work in interpretability that learn models from the original data (i.e. without distillation) with feature shapes that fulfill monotonicity (Gupta et al., 2016) or concavity/convexity (Pya & Wood, 2015) constraints.

How are feature shapes different from feature attribution? A classic way to interpret black-box models is feature attribution/importance measures. Examples include permutation-based measures

[1]E.g. Hooker (2007)'s decomposition procedure requires computing $X^T X$, where X is an n by p feature matrix and solving a system of equations involving 2^{p+1} samples.

(Breiman, 2001), gradients/saliency (see Montavon et al. (2017) or Ancona et al. (2018) for a review), and measures based on variance decomposition (Iooss & Lemaitre, 2015), game theory (Datta et al., 2016; Lundberg & Lee, 2017), etc. We highlight that *feature shapes are different from and more expressive than feature attributions*. Feature attribution is a single number describing the feature's contribution to either the prediction of one sample (local) or the model (global), whereas our feature shapes describe the contribution of a feature, *across the entire domain of the feature*, to the model. Nonetheless, feature attribution, both global and local, can be automatically derived from feature shapes: global feature attribution by averaging feature shape values at each unique feature value; local feature attribution by simply taking one point on the feature shape.

2.3 Learning $\hat{F}$ using Model Distillation

Model distillation was originally proposed to transfer knowledge from a large, complex model (teacher) to a faster, simpler model (student) without significant loss in prediction accuracy (Bucilua et al., 2006; Ba & Caruana, 2014; Hinton et al., 2015). We use model distillation for a different purpose: to learn global explanations for the teacher model. Neural nets and other black-box teachers have been distilled into interpretable models such as trees (Craven & Shavlik, 1995; Che et al., 2016; Frosst & Hinton, 2017; Bastani et al., 2017), rules (Ribeiro et al., 2018) and sets (Lakkaraju et al., 2017). An advantage of using additive student models over these models is that our feature shapes have automatic feature attribution, unlike e.g. decision trees (White & Liu, 1994).

Training teacher neural nets. Our teacher models are fully-connected nets with ReLU nonlinearities. We use the Adam optimizer (Kingma & Ba, 2015) with Xavier initialization (Glorot & Bengio, 2010) and early stopping based on validation loss. At each depth, we search for optimal hyperparameters (number of hidden units, learning rate, weight decay, dropout probability, batch size, enabling batch norm) based on average validation performance on multiple train-validation splits and random initializations. The most accurate nets we trained are fully-connected models with 2-hidden layers and 512 hidden units per layer (2H-512,512); nets with three or more hidden layers had lower training loss, but did not generalize as well and had worse validation loss. In some experiments we also use a restricted-capacity model with 1 hidden layer of 8 units (1H-8) to compare explanations.

Training student additive explanation models. To train SAT and SAS, we find optimal feature shapes $\{h_i\}_1^p$ that minimize the mean square error between the teacher F and the student $\hat{F}$, i.e.

$$L(h_0, h_1, \ldots, h_p) = \frac{1}{T}\sum_{t=1}^{T} \|F(x^t) - \hat{F}(x^t)\|_2^2 = \frac{1}{T}\sum_{t=1}^{T} \|F(x^t) - (h_0 + \sum_{i=1}^{p} h_i(x_i^t))\|_2^2, \quad (2)$$

where $F(x)$ is the output of the teacher model (scores for regression tasks and logits for classification tasks), T is the number of training samples, x^t is the t-th training sample, and x_i^t is its i-th feature. The exact optimization details depend on the choice of h. For trees we use cyclic gradient boosting (Buhlmann & Yu, 2003; Lou et al., 2012) which learns the feature shapes in a cyclic manner. As trees are high-variance, low-bias learners (Hastie et al., 2001), when used as base learners in additive models, it is standard to bag multiple trees (Lou et al., 2012; 2013; Caruana et al., 2015). We follow that approach here. For splines, we use cubic regression splines trained using penalized maximum likelihood in R's `mgcv` library (Wood, 2011) and cross-validate the splines' smoothing parameters.

In most of this paper, our learned explanations $\hat{F}$ are composed of main components h_i. Higher order components h_{ij}, h_{ijk} can increase the accuracy of $\hat{F}$, but make interpretation more difficult because we no longer get one shape per input feature and some shapes now have three or more dimensions. When $\hat{F}$ consists of only main components h_i, any pairwise or higher order interactions in F are expressed as a best-fit additive approximation added to main components h_i, plus a pure-interaction residual. We show examples of this expression in Section 4.1, and in Section 4.5 show an example of an explanation $\hat{F}$ that includes higher-order interaction components h_{ij} and h_{ijk}.

3 Evaluating Global Explanations

Lundberg & Lee (2017) suggested the perspective of viewing an explanation of a model's prediction as a model itself. With this perspective, we propose to *quantitatively evaluate explanation models as if they were models*. Specifically, we evaluate not just fidelity (how well the explanation matches

the teacher's predictions) but also accuracy (how well the explanation predicts the original label). Note that Lundberg & Lee (2017) and Ribeiro et al. (2016) evaluated local fidelity (called local accuracy by Lundberg & Lee (2017)), but not accuracy. A similar evaluation of global accuracy was performed by Kim et al. (2016) who used their explanations (prototypes) to classify test data. In our case, we use the feature shapes generated by our approach to predict on independent test data.

Baselines. We compare to three other explanation methods commonly used for tabular data: partial dependence (Friedman, 2001) as well as two local methods that we first adapt to the global setting: Shapley additive explanations (Lundberg & Lee, 2017) and linearization through gradients.

Partial dependence (PD) is a classic global explanation method that estimates how predictions change as feature x_j varies over its domain: $PD(x_j = z) = \frac{1}{T}\sum_{t=1}^{T} F((x_1^t, \ldots, x_j^t = z, \ldots, x_p^t)$ where the neural net is queried with new data samples generated by setting the value of their x_j feature to z, a value in the domain of x_j. Plotting $PD(x_j = z)$ by z returns a feature shape.

Linearization through gradient approximation (GRAD). We construct the additive function G through the Taylor decomposition of F, defining $G(x) = F(0) + \sum_{i=1}^{p} \frac{\partial F(x)}{\partial x_i} x_i$, and defining the attribution of feature i of value x_i as $\frac{\partial F(x)}{\partial x_i} x_i$. This formulation is related to the "gradient*input" method (e.g. Shrikumar et al. (2017)) used to generate saliency maps for images.

Shapley additive explanations (SHAP). SHAP is a state-of-the-art local explanation method that satisfies several desirable local explanation properties (Lundberg & Lee, 2017). Given a sample and its prediction, SHAP decomposes the prediction additively between features using a game-theoretic approach. We use the python package by the authors of SHAP.

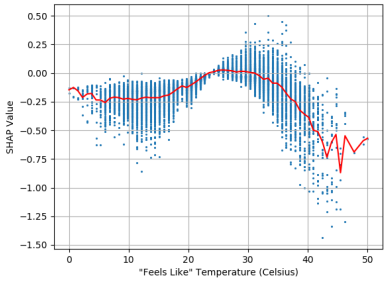

Figure 2: From SHAP to gSHAP. Blue points are individual SHAP values; red line is gSHAP feature shape.

Both GRAD and SHAP provide local explanations that we adapt to a global setting by averaging the generated local attributions at each unique feature value. For example, the global attribution for feature "Temperature" at value 10 is the average of local attribution "Temperature" for all training samples with "Temperature=10". This is the red line passing through the points in Figure 2. Applying this procedure to GRAD and SHAP's local attributions, we obtain global attributions **gGRAD** and **gSHAP** that we can now plot as feature shapes and evaluate quantitatively.

4 Experimental Results

We divide our experimental evaluation into four parts. First, we generate global additive explanations of synthetic functions with known ground-truth feature shapes (Section 4.1). In the second part, we quantitatively evaluate our global additive explanations against other explanations (Section 4.2). Next, we further validate our explanations with controlled experiments on real data (Section 4.3). Finally, we discuss insights obtained from our explanations (Section 4.4).

4.1 Validation Using Synthetic Data with Known ground-truth

For this experiment, we simulate data from synthetic functions with *known* ground-truth feature shapes, which allows us to test our predicted shapes. We are particularly interested in observing how predicted feature shapes differ for neural nets of different capacity trained on the same data. Our expectation is that for neural nets that are accurate, our predicted shapes would match the ground-truth feature shapes, independent of how the features are used internally by the net. On the other hand, predicted shapes of less accurate models should less accurately match ground-truth shapes.

We design an additive, highly nonlinear function combining components from synthetic functions proposed by Hooker (2004), Friedman & Popescu (2008) and Tsang et al. (2018): $F_1(\mathbf{x}) = 3x_1 + x_2^3 - \pi^{x_3} + \exp(-2x_4^2) + \frac{1}{2+|x_5|} + x_6 \log(|x_6|) + \sqrt{2|x_7|} + \max(0, x_7) + x_8^4 + 2\cos(\pi x_8)$. Like Tsang et al. (2018), we set the domain of all features to be $\mathcal{U}(-1, 1)$. Like Friedman & Popescu

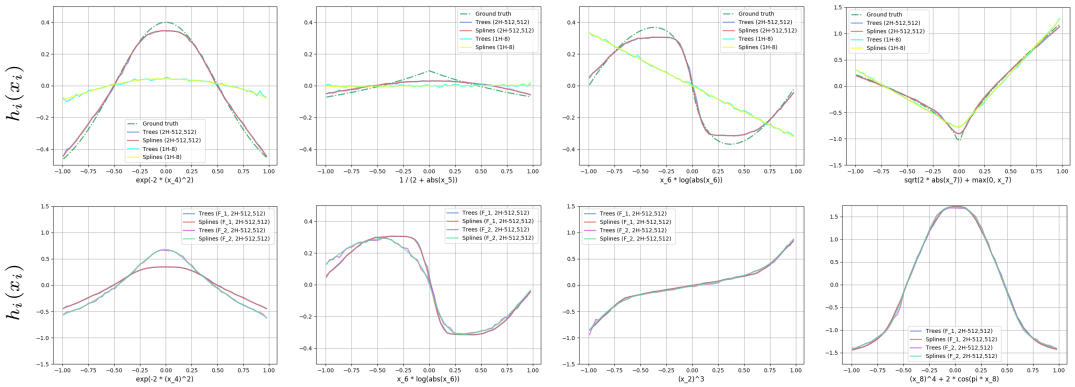

Figure 3: Four of ten feature shapes learned for F_1 (top row) and F_2 (bottom row) by small (1H-8) and large (2H-512,512) nets. To view shapes learned for all features, see the Appendix.

(2008), we add noise features to our samples that have no effect on $F_1(x)$ via two noise features x_9 and x_{10}. Over 50,000 samples, the mean of $F_1(x)$ is 1.15, maximum is 8.65 and minimum is -6.62.

We started by training two teacher neural nets, 2H-512,512 and 1H-8 as described in Section 2.3. The high-capacity 2H neural net obtained test RMSE of 0.14, while the low-capacity neural net obtained test RMSE of 0.48, more than 3x larger. For each neural net, we used our approach to generate two global additive explanation models, SAT and SAS. These explanation models are faithful: the reconstruction RMSE of SAT is 0.14 for the 1H model and 0.08 for the 2H model, while the reconstruction RMSE of SAS is 0.14 for the 1H model and 0.07 for the 2H model. This suggests that both student methods should accurately represent the teacher, and that they probably will be very similar to each other.

Do SAT and SAS explain the teacher model, or just the original data? The top row of Figure 3 compares the feature shapes of our global explanation models SAT and SAS to function F_1's analytic ground-truth feature shapes. SAT and SAS' feature shapes are almost identical. More importantly, it is clear that the feature shapes for the 2H model are different from shapes for the 1H model, and that the shapes for the 2H model better match ground-truth shapes. In general, the shapes of the 2H model are very faithful to the ground-truth shapes, but sometimes fall short when there are sharp changes in the ground-truth, highlighting the limitations of a 2-hidden-layer neural net (which achieves 0.14 test RMSE, as noted before). On the other hand, both SAT and SAS' feature shapes for the 1H neural net show a less accurate teacher model that captures the gist of the ground-truth function but not its details, which is consistent with the original teacher RMSE of 0.48. This showcases that our methods fit what the teacher model has learned, and not the original data, and that when the teacher model is accurate the learned shapes match the ground-truth shapes.

Model	**Easy**	**All**	**Hard**
1H-8	0.42	0.48	-
2H-512,512	-	0.14	0.17

Table 1: RMSE error of the teacher models on "easy" and "hard" samples chosen through the predicted attribution.

Do SAT and SAS' feature shapes match the real behavior of the model? To further validate this we use the feature shapes to predict which samples will be inaccurately predicted by the teacher model. Specifically, we sample testing points with feature values where the feature shape of the 2H model is less accurate according to the feature shape ground-truth (for example, with $x_4, x_5, x_7 = 0$ and $x_6 = 0.3$) and evaluate them using the teacher model. If the learned feature shapes correctly represents the teacher model, the teacher should also be less accurate on those points than on other points where the learned and ground-truth feature shapes match. Similarly, by sampling points where the feature shapes of the 1H model and the ground-truth overlap, we would expect the error of the 1H teacher to be low. Indeed, as shown in Table 1, points sampled to be easy or hard guided by the feature shapes lead to lower and higher RMSE error, respectively, providing more evidence that our learned feature shapes are faithful.

How do interactions between features affect feature shapes? We design an augmented version of F_1 to investigate how interactions in the teacher's predictions are expressed by feature shapes: $F_2(\mathbf{x}) = F_1(\mathbf{x}) + x_1 x_2 + |x_3|^{2|x_4|} + \sec(x_3 x_5 x_6)$. We again simulate 50,000 samples. The mean of $F_2(x)$ is 2.74, maximum is 11.48 and minimum is -4.46. Note that this function is much harder

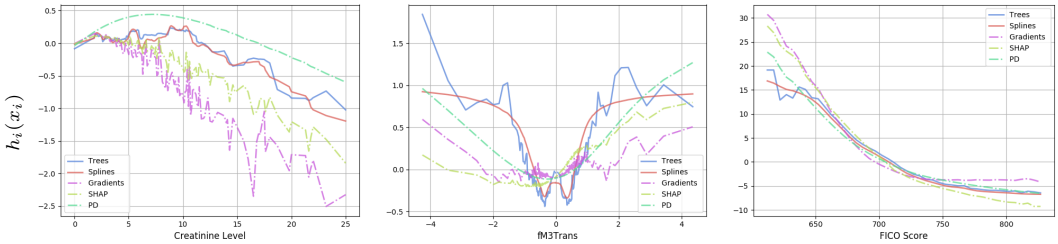

Figure 4: Example feature shapes from Pneumonia (left), Magic (center), and Loan (right). SAT and SAS tend to agree. gSHAP, PD, and gGRAD capture the trend of the shape but not the details. Best seen on a screen.

to learn (the 2H model obtained an RMSE of 0.21) and also harder for students that do not model interactions to mimic (SAT and SAS obtain fidelity RMSEs of 0.35). The bottom row of Figure 3 displays features with interactions (x_4, x_6, x_2) and a feature without interactions (x_8), and compares them with the shapes from F_1. For features x_4 and x_6, the part of the interactions that can be approximated additively by h_i's has been absorbed into the h_i feature shapes, changing their shapes as expected. On the other hand, we were still able to recover perfectly the feature shapes of features without interactions (e.g. x_8). An interesting case study is x_2, where, despite the interaction, its feature shape has not changed. This is less surprising if we understand the feature shapes as the *expected importance* of the feature, learned in a data-driven fashion. The interaction term is $x_1 x_2$, which, for $x_1 \sim \mathcal{U}(-1, 1)$, has an expected value of zero, and therefore does not affect the feature shape. Similarly, the expected value of $|x_3|^{2|x_4|}$ when $x_3 \sim \mathcal{U}(-1, 1)$ is $1/(2|x_4| + 1)$, an upward pointing cusp, which modifies the feature shape as shown in Figure 3 (bottom left figure).

4.2 Comparing Explanation Methods on Real Data

We selected five data sets to evaluate our approach: two UCI data sets (Bikeshare and Magic), a Loan risk scoring data set from an online lending company (LendingClub, 2011), the 2018 FICO Explainable ML Challenge's credit data set (FICO, 2018), and the pneumonia data set analyzed by Caruana et al. (2015). Table 2 provides details about the datasets and performance of the 1H and 2H teacher models.

					Performance	
Data	n	p	**Type**		1H	2H
Bikeshare	17,000	12	Reg	RMSE	0.60	0.38
Loan	42,506	22	Reg	RMSE	2.71	1.91
Magic	19,000	10	Class	AUC	92.52	94.06
Pneumonia	14,199	46	Class	AUC	81.81	82.18
FICO	9,861	24	Class	AUC	79.08	79.37

Table 2: Performance of neural net teachers

4.2.1 Quantitative Evaluation of Explanation Methods

Table 3 presents the fidelity (how well does the student reproduce the teacher scores) and accuracy (how well does the student perform on the original task on independent test data) results for different global explanations of the 2H neural nets. We discuss results for the 1H neural nets in the Appendix. Accuracy is measured in terms of RMSE for regression tasks and AUROC for classification tasks, while fidelity is always measured as the RMSE between the student's predictions and the teacher's scores or logits (equation 2).

We draw several conclusions. First, SAT and SAS yield similar results in all cases, both in terms of accuracy and fidelity. In some cases, such as Magic, SAT (which uses tree base learners) can be more accurate, while in some others such as FICO, SAS (which uses spline base learners) may have the edge. Our interpretation is that trees are able to adapt better to sudden changes in shape than splines, but that also gives them more capacity to slightly overfit. We also see this in the feature shapes, where trees may be slightly more jagged than the splines, particularly in regions with fewer points. Figure 4 displays a few feature shapes for Pneumonia, Magic, and Loan. The feature shapes produced by PD tend to be much too smooth, which hurts its fidelity and accuracy. Second, in all cases, trees and splines have similar feature shapes and obtain equal or better accuracy and fidelity than the other methods. This is not surprising as the other methods are either local methods adapted to the global setting (gSHAP, gGRAD), or are global explanations that are not optimized to learn the teacher's predictions (PD). For reference, gSHAP when used as a local method (i.e. individual SHAP values, not global feature shapes) achieved a lower RMSE of 0.37 compared to 1.02 on

Accuracy Global Explanation	Bikeshare RMSE	Loan score RMSE	Magic AUC	Pneumonia AUC	FICO AUC
SAT	0.98 ± 0.00	2.35 ± 0.01	90.75 ± 0.06	82.24 ± 0.05	79.42 ± 0.04
SAS	0.98 ± 0.00	2.34 ± 0.00	90.58 ± 0.02	82.12 ± 0.04	79.51 ± 0.02
gGRAD	1.25 ± 0.00	6.04 ± 0.01	80.95 ± 0.13	81.88 ± 0.05	79.28 ± 0.02
gSHAP	1.02 ± 0.00	5.10 ± 0.01	88.98 ± 0.05	82.31 ± 0.03	79.36 ± 0.01
PD	1.00 ± 0.00	4.31 ± 0.00	82.78 ± 0.00	82.15 ± 0.00	79.47 ± 0.00
Fidelity Global Explanation	Bikeshare RMSE	Loan score RMSE	Magic RMSE	Pneumonia RMSE	FICO RMSE
SAT	0.92 ± 0.00	1.74 ± 0.01	1.78 ± 0.00	0.35 ± 0.00	0.15 ± 0.00
SAS	0.92 ± 0.00	1.71 ± 0.00	1.75 ± 0.00	0.35 ± 0.00	0.14 ± 0.00
gGRAD	1.20 ± 0.00	5.93 ± 0.01	2.93 ± 0.01	0.43 ± 0.00	0.16 ± 0.00
gSHAP	0.96 ± 0.00	4.83 ± 0.00	2.15 ± 0.00	0.46 ± 0.00	0.16 ± 0.00
PD	0.94 ± 0.00	3.85 ± 0.00	3.17 ± 0.00	0.47 ± 0.00	0.16 ± 0.00

Table 3: Accuracy and fidelity of global explanation models for 2H-512,512 neural nets on different datasets. For RMSE, lower is better. For AUC, higher is better. Results for 1H-8 nets in Appendix.

Bikeshare, and a lower RMSE of 1.99 compared to 5.10 on Loan, which is comparable to its 2H teacher's RMSE on test data (Table 2). The conclusion is that methods such as gSHAP excel at local explanations and should be used for those, but, to produce global explanations, global model distillation methods optimized to learn the teacher's predictions should be used instead.

4.3 Validation Using Controlled Experiments on Real Data

In this section we demonstrate the utility of global additive explanations on real data. Although here we do not have an analytic solution for the ground-truth feature shapes, we can still design experiments where we modify data in ways that will lead to expected known changes to the ground-truth feature shapes and then verify that these changes are captured in the learned feature shapes.

Label modification. In the bikeshare data, we added 1.0 to the label (the number of rented bikes) for samples where one of the features (humidity) is between 55 and 65. We then retrained a 2H neural net on the modified data, and applied our approach to learn feature shapes from the 2H net. Ideally, the feature shapes of that new neural net should be almost identical to those of the original net except in that particular range of the humidity feature, where we should see an abrupt "bump" that increases its feature shape value by one. Figure 5 (left) displays the feature shapes. Our method was able to recover the change to the label for the neural net in the new feature shape.

Data modification: expert discretization. Sometimes features are transformed before training. For example, in medical data, continuous variables such as body temperature may be discretized by domain experts into bins such as normal, mild fever, moderate fever, high fever, etc. In this experiment we test if our additive explanation models can recover these discretizations from the neural net without access to the discretized features. We train our student additive models using as input features *the original un-discretized features*, but using as labels the outputs of a neural net that was trained on discretized features. Our expectation is that if the student models are an accurate representation of what the neural net learned from the discretized features, they will detect the discretizations, even if they never have access to the discretized features or to the internal structure of the neural-net teacher. We study the feature shapes of two features in the Pneumonia data (Blood pO_2 and Respiration Rate) in Figure 5, where we compare the feature shapes learned from teachers trained on the original continuous data (dotted lines) with those from teachers trained on discretized features (solid lines). Recall that in both cases the student models only saw non-discretized features to generate feature shapes. Our approach captures the expected discretization intervals (in yellow) as described in Cooper et al. (1997).

4.4 Insights from Global Additive Explanations

Checking for monotonicity. Domains such as credit scoring have regulatory requirements that prescribe monotonic relationships between predictions and some features (Federal Reserve Governors, 2007). For example, the 2018 FICO Explainable ML Challenge encouraged participants to impose monotonicity on 16 features (FICO, 2018). We use feature shapes to see if the function learned by the

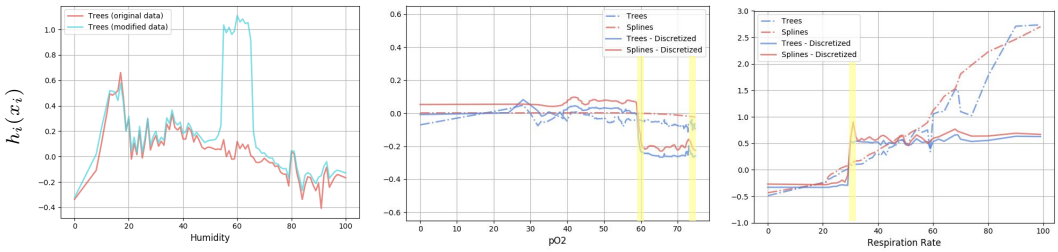

Figure 5: Feature shapes from controlled experiments on real data. Left: Label modification experiment. Center and right: Data modification experiment. See details in Section 4.3.

neural net is monotone for these features. 15 of 16 features are monotonically increasing/decreasing as required. One feature, however, "Months Since Most Recent Trade Open" was expected to decrease monotonically, but actually increased monotonically. This is true not just in our explanations, but also in PD, gGRAD, and gSHAP (Figure A3). Note that testing for monotonicity requires global explanations or checking and aggregating many local explanations. See Appendix for details.

Visualizing neural net training: from underfit to overfit. Using additive models to peek inside a neural net creates many opportunities. For example, we can see what happens in the neural net when it is underfit or overfit; when it is trained with different losses such as squared, log, or rank loss or with different activation functions such as sigmoid or ReLUs; when regularization is performed with dropout or weight decay; when features are coded in different ways; etc. The video at `https://youtu.be/ErQYwNqzEdc` shows what is learned by a neural net as it trains on a medical dataset. The movie shows feature shapes for five features before, at, and after the early-stopping point as the neural net progresses from underfit to optimally fit to overfit. We had expected that the main cause of overfitting would be increased non-linearity (bumpiness) in the fitting function, but a significant factor in overfitting appears to be unwarranted growth in the confidence of the model as the logits grow more positive or negative than the early-stopping shape suggests is optimal.

4.5 Extending $\hat{F}$ to Include Interactions

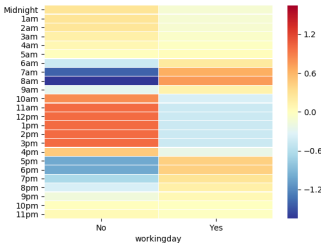

Figure 6: An important pairwise interaction in Bikeshare.

Functions learned by neural nets cannot always be represented with adequate fidelity by the additive function $\hat{F}$ in equation 1. We can improve $\hat{F}$'s expressive power by adding pairwise and higher-order components h_{ij}, h_{ijk}, and so on to account for interactions between two or more input features. In Bikeshare, RMSE decreases from 0.98 to 0.60 when we add pairwise interactions to the student model. Figure 6 shows an interesting interaction between two features: "Time of Day", and "Working Day". On working days, the highest bike rental demand occurs at 7-9am and 5-7pm, but on weekends there is very low demand at 7-9am (presumably because people are still sleeping) and at 5-7pm, and demand peaks during midday from 10am-4pm. These two features also form a three-way interaction with temperature. Because the teacher neural net learned these (and other) interactions, a global explanation method must also incorporate interactions if it is to provide high-fidelity explanations of the teacher model.

5 Conclusions

We present a method for "opening up" complex models such as neural nets trained on tabular data. The method is based on distillation with high-accuracy additive models to provide a global explanation of what a neural net has learned. Although in this paper we focus on explaining fully-connected neural nets, the method will work with any classification or regression model including CNNs and RNNs, but is not designed to work with raw inputs such as pixels where providing a global explanation in terms of the input pixels is not well-defined. We perform a battery of experiments to show that explanations generated by the method are faithful representations of the complex teacher model, and compared the method to other global explanation methods such as partial dependence, Shapley adapted to a global setting, and gradient methods. Our method is computationally efficient and requires only that the teacher neural net label a training set; it does not require repeated probing or access to the teacher model's internal structure or derivatives. We also show how our global explanations can be improved by adding pairwise and higher-order interactions to the explanations.

REFERENCES

David Alvarez-Melis and Tommi S Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NIPS*, 2018.

Marco Ancona, Enea Ceolini, Cengiz ztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction. In *FAT/ML Workshop*, 2017.

Leo Breiman. Random forests. *Machine Learning*, 2001.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

Peter Buhlmann and Bin Yu. Boosting with the l2 loss: regression and classification. *Journal of the American Statistical Association*, 2003.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*, 2015.

Zhengping Che, Sanjay Purushotham, Robinder G. Khemani, and Yan Liu. Interpretable deep models for ICU outcome prediction. In *AMIA Annual Symposium*, 2016.

Gregory F. Cooper, Constantin F. Aliferis, Richard Ambrosino, John M. Aronis, Bruce G. Buchanan, Rich Caruana, Michael J. Fine, Clark Glymour, Geoffrey J. Gordon, Barbara H. Hanusa, Janine E. Janosky, Christopher Meek, Tom M. Mitchell, Thomas S. Richardson, and Peter Spirtes. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 1997.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *NIPS*, 1995.

Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy*, 2016.

Federal Reserve Governors. Report to the congress on credit scoring and its effects on the availability and affordability of credit. 2007. URL `https://www.federalreserve.gov/boarddocs/rptcongress/creditscore/creditscore.pdf`.

FICO. Explainable machine learning challenge, 2018. URL `https://community.fico.com/s/explainable-machine-learning-challenge`.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 2001.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2008.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. *JMLR*, 2016.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

Geoff Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning Workshop*, 2015.

Giles Hooker. Discovering additive structure in black box functions. In *KDD*, 2004.

Giles Hooker. Generalized functional ANOVA diagnostics for high dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 2007.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *ICML*, 2014.

Bertrand Iooss and Paul Lemaitre. A review on global sensitivity analysis methods. In *Uncertainty Management in Simulation-Optimization of Complex Systems*. 2015.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable and explorable approximations of black box models. In *FAT/ML Workshop*, 2017.

LendingClub. Lending club loan data, 2011. URL `https://www.lendingclub.com/info/download-data.action`.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *KDD*, 2012.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *KDD*, 2013.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

Natalya Pya and Simon N Wood. Shape constrained additive models. *Statistics and Computing*, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the predictions of any classifier. In *KDD*, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 2001.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *ICLR*, 2018.

Allan P White and Wei Zhong Liu. Bias in information-based measures in decision tree induction. *Machine Learning*, 1994.

Simon N Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2006.

Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 2011.

APPENDIX

A ALL FEATURE SHAPES FOR F_1 AND F_2 SYNTHETIC FUNCTIONS (SECTION 4.1)

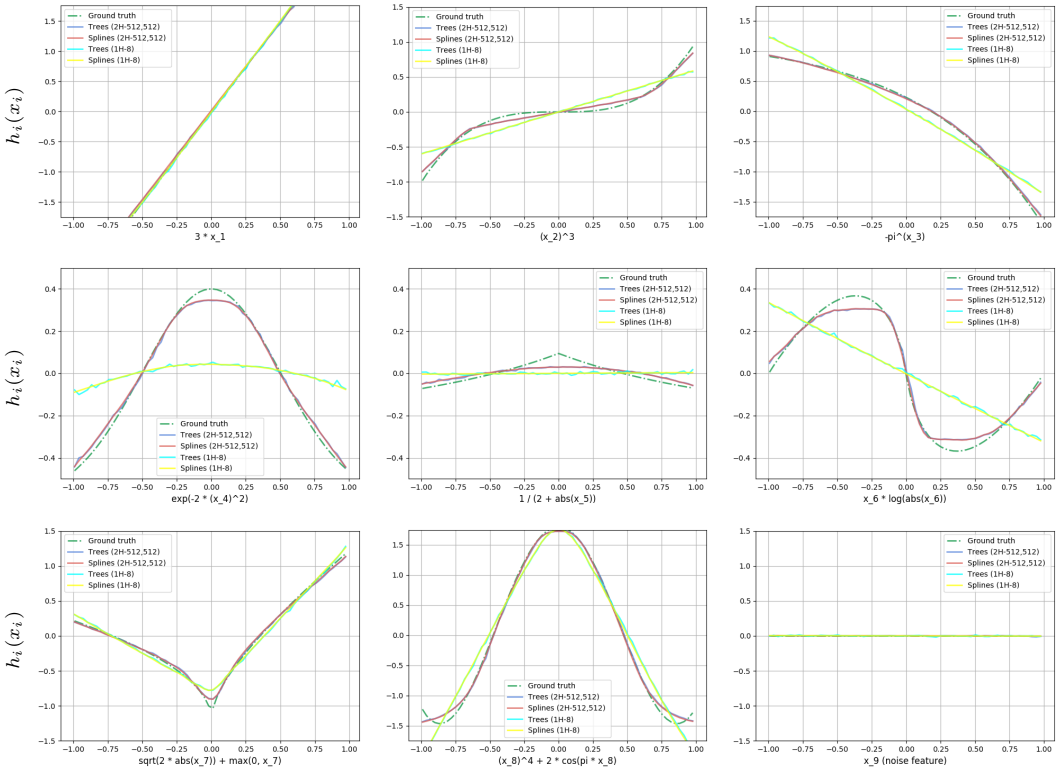

Figure A1: Feature shapes for features x_1 to x_9 of F_1 from Section 4.1. Notice how x_9, which is a noise feature that does not affect F_1, has been assigned an importance of approximately 0 throughout its range. The feature shape of x_{10}, another noise feature, is very similar to x_9 and hence not included here.

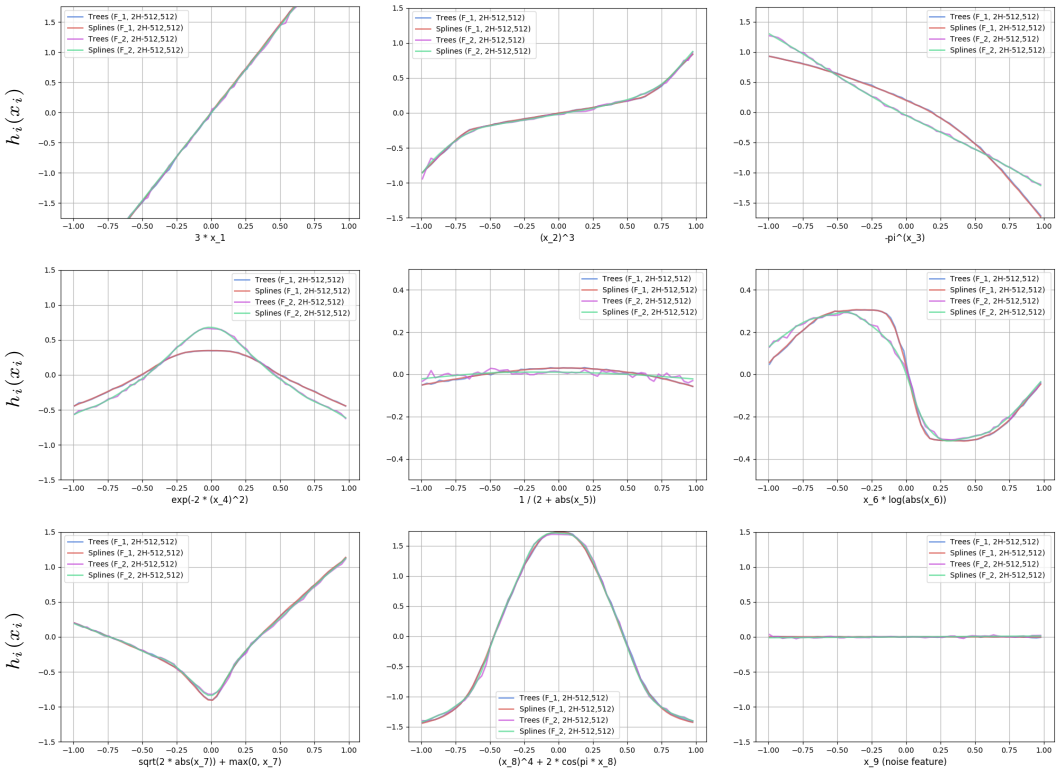

Figure A2: Feature shapes for features x_1 to x_9 of F_2 from Section 4.1. Notice how x_9, which is a noise feature that does not affect F_2, has been assigned an importance of approximately 0 throughout its range. The feature shape of x_{10}, another noise feature, is very similar to x_9 and hence not included here.

B QUANTITATIVE RESULTS FOR GLOBAL EXPLANATIONS OF 1H AND 2H NEURAL NETS ON REAL DATA (SECTION 4.2.1)

Accuracy Teacher	Global Explanation	Bikeshare RMSE	Loan score RMSE	Magic AUC	Pneumonia AUC	FICO AUC
1H-8	SAT	1.00 ± 0.00	2.82 ± 0.00	90.44 ± 0.05	82.01 ± 0.05	79.43 ± 0.02
	SAS	1.00 ± 0.00	2.82 ± 0.00	90.43 ± 0.03	81.91 ± 0.06	79.56 ± 0.02
	gGRAD	1.08 ± 0.00	2.84 ± 0.00	84.52 ± 0.67	81.63 ± 0.06	79.34 ± 0.05
	gSHAP	1.04 ± 0.00	2.87 ± 0.00	89.94 ± 0.03	82.02 ± 0.02	79.49 ± 0.02
	PD	1.00 ± 0.00	3.00 ± 0.00	85.11 ± 0.00	82.03 ± 0.00	79.46 ± 0.00
2H-512,512	SAT	0.98 ± 0.00	2.35 ± 0.01	90.75 ± 0.06	82.24 ± 0.05	79.42 ± 0.04
	SAS	0.98 ± 0.00	2.34 ± 0.00	90.58 ± 0.02	82.12 ± 0.04	79.51 ± 0.02
	gGRAD	1.25 ± 0.00	6.04 ± 0.01	80.95 ± 0.13	81.88 ± 0.05	79.28 ± 0.02
	gSHAP	1.02 ± 0.00	5.10 ± 0.00	88.98 ± 0.05	82.31 ± 0.03	79.36 ± 0.01
	PD	1.00 ± 0.00	4.31 ± 0.00	82.78 ± 0.00	82.15 ± 0.00	79.47 ± 0.00
Fidelity Teacher	Global Explanation	Bikeshare RMSE	Loan score RMSE	Magic RMSE	Pneumonia RMSE	FICO RMSE
1H-8	SAT	0.64 ± 0.00	1.15 ± 0.00	1.12 ± 0.00	0.30 ± 0.00	0.21 ± 0.00
	SAS	0.64 ± 0.00	1.14 ± 0.00	1.11 ± 0.00	0.30 ± 0.00	0.21 ± 0.00
	gGRAD	0.71 ± 0.00	1.54 ± 0.00	35.40 ± 4.47*	0.36 ± 0.00	0.24 ± 0.00
	gSHAP	0.68 ± 0.00	1.28 ± 0.00	1.29 ± 0.00	0.38 ± 0.00	0.22 ± 0.00
	PD	0.65 ± 0.00	1.37 ± 0.00	1.94 ± 0.00	0.38 ± 0.00	0.25 ± 0.00
2H-512,512	SAT	0.92 ± 0.00	1.74 ± 0.01	1.78 ± 0.00	0.35 ± 0.00	0.15 ± 0.00
	SAS	0.92 ± 0.00	1.71 ± 0.00	1.75 ± 0.00	0.35 ± 0.00	0.14 ± 0.00
	gGRAD	1.20 ± 0.00	5.93 ± 0.01	2.93 ± 0.01	0.43 ± 0.00	0.16 ± 0.00
	gSHAP	0.96 ± 0.00	4.83 ± 0.01	2.15 ± 0.00	0.46 ± 0.00	0.16 ± 0.00
	PD	0.94 ± 0.00	3.85 ± 0.00	3.17 ± 0.00	0.47 ± 0.00	0.16 ± 0.00

Table A1: Accuracy and fidelity of global explanation models across 1H and 2H teacher neural nets and datasets. For RMSE, lower is better. For AUC, higher is better. Table 3 is a subset of this table with only 2H neural nets.

In general, the lower-capacity 1H neural nets are easier to approximate (i.e. better student-teacher fidelity), but their explanations are less accurate on independent test data. Students of simpler teachers tend to be less accurate even if they are faithful to their (simple) teachers. One exception is the FICO data, where the fidelity of the 2H explanations is better. Our interpretation is that many features in the FICO data have almost linear feature shapes (see Figure A3 for a sample of features), and the 2H model may be able to better capture fine details while being simple enough that it can still be faithfully approximated. The accuracy of the SAT and SAS for 1H and 2H neural nets are comparable, taking into account the confidence intervals.

On the Magic data, the fidelity of the gGRAD explanation to the 1H neural net (see * in Table A1) is markedly worse than other explanation methods. We investigate the individual gradients of the 1H neural net with respect to each feature ($\frac{\partial F(x)}{\partial x_i}$ in GRAD equation in Section 3). 99% of them have reasonable values (between -5.6 and 6). However, 3 are larger than 1,000 (with none between 6 and 1,000) and 13 are lower than -1,000 (with none between -1,000 and -5.6), resulting in the ensuing gGRAD explanation generating extreme predictions for several samples that are not faithful to the teacher's predictions. Because AUC is a ranking loss, accuracy (AUC) is less affected than fidelity (RMSE) by the presence of these extreme values. This shows that gGRAD explanations may be problematic when individual gradients are arbitrarily large, e.g. in overfitted neural nets.

C CONTINUATION OF FICO MONOTONICITY EXPERIMENT (SECTION 4.4)

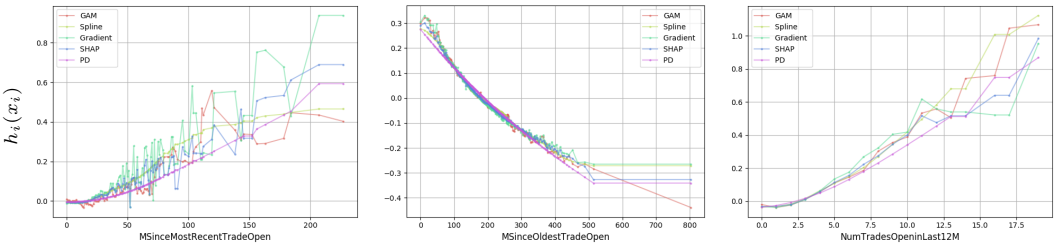

Figure A3: 3 of 16 features with expected monotonically increasing/decreasing patterns in the FICO data. "Months Since Most Recent Trade Open", the leftmost figure, was expected to decrease monotonically, but actually increased monotonically according to all explanations. The two figures on the right are two related features, "Months Since *Oldest* Trade Open" and "Number of Trades Open in Last 12 Months", both of which exhibit the expected monotonically decreasing/increasing patterns.

Continuing from Section 4.4, with the insight from the global explanations that the neural net may not be exhibiting the expected pattern for "Months Since Most Recent Trade Open", we perform a quick experiment to verify this in the neural net. We sample values of this feature across its domain, set all data samples to this value (for this feature), and obtain the neural net's predictions for these modified samples. The majority of samples (70%) had predictions that increased as this feature increased across its domain, confirming that on average, the neural net exhibits a monotonically increasing instead of decreasing pattern for this feature. Note that we could not have checked for a monotonicity pattern (which is by definition a global behavior) without checking and aggregating multiple local explanations.