An Interpretable Neural Model with Interactive Stepwise Influence

Yin Zhang, Ninghao Liu, Shuiwang Ji, James Caverlee, and Xia Hu

Texas A&M University, TX, USA
{zhan13679, nhliu43, sji, caverlee, xiahu}@tamu.edu

Abstract. Deep neural networks have achieved promising prediction performance, but are often criticized for the lack of interpretability, which is essential in many real-world applications such as health informatics and political science. Meanwhile, it has been observed that many shallow models, such as linear models or tree-based models, are fairly interpretable though not accurate enough. Motivated by these observations, in this paper, we investigate how to fully take advantage of the interpretability of shallow models in neural networks. To this end, we propose a novel interpretable neural model with Interactive Stepwise Influence (ISI) framework. Specifically, in each iteration of the learning process, ISI interactively trains a shallow model with soft labels computed from a neural network, and the learned shallow model is then used to influence the neural network to gain interpretability. Thus ISI could achieve interpretability in three aspects: importance of features, impact of feature value changes, and adaptability of feature weights in the neural network learning process. Experiments on both synthetic and two real-world datasets demonstrate that ISI could generate reliable interpretation with respect to the three aspects, as well as preserve prediction accuracy by comparing with other state-of-the-art methods.

Keywords: Neural network · Interpretation · Stepwise Influence.

1 Introduction

Neural networks (NNs) have achieved extraordinary predictive performance in many real-world applications [19]. Despite the superior performance, NNs are often regarded as black-boxes and difficult to interpret, due to their complex network structures and multiple nested layers of non-linear transformations. This "interpretability gap" poses key roadblocks in many domains – such as health informatics, political science, and marketing – where domain experts prefer to have a clear understanding of both the underlying prediction models as well as the end results [5]. In contrast, many "shallow" models, such as linear regression or tree-based models, do provide easier interpretability [3] (e.g., through inspection of the intermediate decision nodes) but may not achieve accuracy on par with deep models. To bridge this gap, we investigate how to take advantage of the interpretability of shallow models in developing interpretable deep neural networks.

Recently, several efforts have been devoted to enable interpretability of deep models, including visualization for feature selection in computer vision area [2, 24], prediction-level interpretation [18] and attention models [7] in medical and other areas. These and

related methods typically focus on *results interpretability* which explains results of each individual sample separately [18]. In contrast, we focus on *model interpretability* which can show the features influences to response variables regardless of individual samples; that is, we aim to identify for each feature its importance (the contribution to the result) and its influence (the impact of changes in the feature on changes in the result) [5]. Additionally, we aim to uncover aspects of the internal mechanism of the NN "black box" by capturing how each feature adapts over training iterations. Recently, a widely-used way to build such an interpretable neural network is to firstly train a complex but accurate deep NN, and then transfer its knowledge to a much smaller but interpretable model [6]. However, this approach has several limitations. First, it makes use of the soft labels computed from the deep model to train another shallow model, which ignores the fact that the "dark" knowledge [1] learned at the end may or may not be the best to train an effective shallow model. Second, parameters in NN are usually learned by complex process, which makes NN hard to be understood while the method does not consider that. So if we could show how each features is learned in NN, it can help interpret NN.

Motivated by these observations, we propose a novel framework ISI – an Interactive Stepwise Influence model, that can interactively learn the NN and shallow models simultaneously to realize both interpretability and accuracy. Specifically, ISI first uses a shallow model to approximate the neural network's predictions in a forward propagation. Then, ISI uses fitted values of the shallow model as prior knowledge to train the next learning step. In sum, the two parts in ISI – shallow models and the NN, interactively influence each other in each training iteration.

During the process, ISI can be interpreted in three aspects: (i) *Importance:* ISI calculates the contribution of each input feature; (ii) *Impact:* ISI gives the value changes of predicted variable based on different feature value changes by a relatively simple relationship; and (iii) *Adaptability:* ISI shows variations of feature weights changes in learning process of NN. In experiment, we evaluate ISI on both synthetic and two real-world datasets for classification problems. Specifically, we first evaluate the reliability of ISI interpretability based on the correctness of feature importance and feature influence. We also show the variations of feature weights changes in ISI updating process. At last, we compare the prediction accuracy of ISI with traditional machine learning methods and state-of-the-art methods such as CNN and MIMIC learning [6]. Our results show that ISI can give utility interpretations from the three aspects and outperforms all the other interpretable state-of-the-art methods in AUPRC and AUROC.

2 Related Work

NNs are widely used because of their extraordinary performance in fitting non-linear relationships and extracting useful patterns [12]. However, in some real world applications, such as health care, marketing, political science and education, interpretability provides significant insights behind the predictions. In such situations, interpretation can be more important than prediction accuracy. NNs are limited used [4, 6, 7, 9] in those areas because they are hard to interpret.

Some researchers have been working on the interpretability of models [8]. There is an overview about making traditional classification models more comprehensible [10].

Specifically, Wang *et. al* built an oblique treed sparse addictive model to make the interpretable model more accurate [22]. [3] analyzed tree-based models by using a training selected set to make the original model interpretable. [7] proposed an end-to-end interpretable model RETAIN by using reverse time attention mechanism. Some methods use visualization to find the good qualitative interpretations of intermediate features [15]. [18] proposed LIME to learn an interpretable model locally around each prediction. [9] investigated a guided feature inversion framework which could show the NN decision-making process for interpretation. Another approach for the interpretation methods are based on calculating the sensitivity of the output in terms of the input. For example, if an input feature change can bring a significant prediction difference, it means the feature is important to the prediction, such as [20]. Among those methods, "distilled" methods [1, 11, 13] become popular because of their extraordinary performance and strong interpretability. [1] "distilled" a Monte Carlo approximation in Bayesian parameter estimation to consider the dark knowledge inside the deep NN. Meanwhile, recent work showed that by distilling the knowledge, models not only gained a good accuracy, but also maintained interpretability in the shallow models [6].

A popular interpretable "distilled" method [6] uses a shallow model as the mimic model to interpret the final neural network results. However, since only final results are learned, there could be a large gap between soft prediction score of NN and results of the mimic shallow model, which may have an influence on the interpretation. Secondly, parameters in NN are calculated by complicated training process (propagation) which makes it harder to understand while traditional methods could not interpret that. If we can show how the influence changes of input features in the training process, it can help users better understand the neural network.

3 Preliminaries

Before we introduce the interpretable framework ISI, it is important to clarify the kind of interpretability that we aim to achieve. Specifically, following previous work [6], we focus on three aspects of interpretability which is the input feature importance, their impacts and the adaptability for neural networks.

Formally, given a supervised neural network $f : \mathcal{X} \rightarrow \mathcal{Y}$. We assume all input features in $\mathcal{X}$ are explainable. x_i represents i^{th} input feature variable and $i \in \{1, 2, ...q\}$, where q is the number of input features. Let $\mathbf{x} = [x_1, x_2, ...x_q] \in \mathcal{X}$ represents corresponding feature vector, and $\mathbf{y} = f(\mathbf{x})$. Our proposed NN targets the following three aspects of interpretation:

- *Importance*: For each feature X_i, f can provide the corresponding contribution $\beta_i \in \mathcal{R}$ to y;
- *Impact*: If feature X_i changes $\triangle X_i$, f can provide the change $\triangle \mathbf{y}$ of $\mathbf{y}$ in a linear/tree based relationship;
- *Adaptability*: Since f is a NN, f has a learning process to update its parameters. f can provide how each β_i changes in each iteration.

Here we target to perform "model interpretability" rather than "results/local interpretability" since latter explains results of each example separately [18] while the former shows the impact of features to response variable and the interpretation is not constrained by a single sample. For example, the interpretable linear models [21] can be used to explain the relationship between diabetes and lab test variables. Furthermore, humans are limited to understand complex associations between variables [14]. Shallow models are considered as more interpretable since they have simple structures explicitly expressing how features influence the prediction [6]. So for the second aspect, we are tying to find similar variable associations as shallow models to explain the feature impact. By combining the first two aspects of interpretation, f can identify features that are highly related to response variable. For the third aspect, we target to learn the changes of each input feature influence during the NN parameter updating process.

Notations	Definitions
$\mathbf{X} \in R^{n \times k}$	input matrix for sample $\mathbf{x}_1, ... \mathbf{x}_n$
$\mathbf{y} \in R^n$	output vector for sample $\mathbf{x}_1, ... \mathbf{x}_n$
$g(\cdot)$	ground true relationship from $\mathcal{X}$ to $\mathcal{Y}$
$\theta_N = \{\mathbf{w}_N^1, \dots\}$	parameters set of neural network
$f(\mathbf{X}; \theta_N^{(i)})$	learned neural network in i^{th} iteration
$\mathbf{w}_N^i \in R^i$	weight in layer i of $f(\mathbf{X}; \theta_N)$, $i \in 1, \dots h$
$\tilde{\mathbf{y}}^{(i)}$	output of $f(\mathbf{X}; \theta_N^{(i)})$
$\pi_S = \{\mathbf{w}_S^1, \dots\}$	parameters of mimic shallow model
$\xi(\mathbf{X}; \pi_S^{(i)})$	mimic shallow model of $f(\mathbf{X}; \theta_N^{(i)}, \mathbf{y})$
$\tilde{\mathbf{y}}^{(i)}$	output of $\xi(\mathbf{X}; \pi_S^{(i)})$

Table 1. Notations.

4 Interpretable Neural Networks with Interactive Stepwise Influence

The key idea of the proposed framework ISI is to use an interpretable model to approximate the NN outputs in the forward propagation, and then, update NN parameters according to the output of the interpretable model. So in ISI , a NN f for gaining prediction accuracy, and the shallow but interpretable model ξ for tuning f parameters to make it interpretable. In this section, we first introduce our proposed framework ISI and show how to utilize the ISI framework to gain the three interpretation aspects.Then we provide details of ISI optimization.

4.1 The Proposed ISI Framework

In this section, we first introduce our novel ISI framework (shown in Figure 1) in details. Suppose $g : \mathcal{X} \to \mathcal{Y}$ denote the prediction function, where $\mathcal{X}, \mathcal{Y}$ are its domain and codomain, respectively. Samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_n, y_n) \in (\mathcal{X}, \mathcal{Y})$ constitute the dataset $(\mathbf{X}, \mathbf{y})$. The goal is to train a traditional NN $f(\mathbf{X}; \theta_N)$, which is parameterized by $\theta_N = \{\mathbf{w}_N^1, \mathbf{w}_N^2, ...\mathbf{w}_N^h\}$, $\mathbf{w}_N^j$ is the j^{th} hidden layer parameter for f. Parameters in θ_N are get by minimizing the loss function $L_P(f(\mathbf{X}; \theta_N), \mathbf{y})$. For example, it can be the cross entropy loss function $L_P(f(\mathbf{X}; \theta_N), \mathbf{y}) = -\sum_i y_i log\hat{y}_i$, and we minimize it to get the optimal solution θ'_N.

Based on the interpretation that we target, we dig into the neural network learning process (backpropagation for parameters updating). For traditional neural network, the optimized parameters θ'_N is calculated from:

$$\theta'_N = \underset{\theta_N^{(i)}}{\operatorname{argmin}} L_P(f(\mathbf{X}; \theta_N), \mathbf{y}), \tag{1}$$

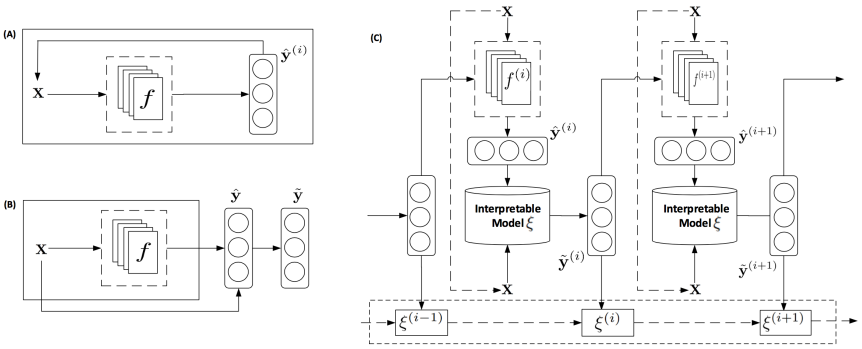


Fig. 1. (A) The standard NN learning architecture: Update parameters through backpropagation from a NN output in each iteration. (B) MIMIC learning: First train a NN, and then train an interpretable model using the output of the NN as soft labels. (C) ISI architecture: The first module is a NN f used to gain accuracy. The second module is interpretable models $\xi^{(i)}$ embedded in f. Instead of using the difference between forward propagation and ground truths for backpropagation, we use forward propagation output as soft labels to train ξ, and then use the fitted output of ξ to replace forward propagation output in backpropagation. ξ can be used to adjust f to provide interpretations for f.

by backpropagations of iterations until it converges. Specifically, for each iteration $i > 1$ of backpropagation, it includes two parts:
(1) A forward pass to use learned $\theta_N^{(i)}$ in i^{th} iteration and generate the current prediction output: $\hat{\mathbf{y}}^{(i)} = f(\mathbf{X}; \theta_N^{(i)})$;
(2) Then a backward pass to update $\theta_N^{(i)}$ in f by minimizing the current loss function value: $\theta_N^{(i+1)} = \theta_N^{(i)} - \eta \nabla_{\theta_N} L(\hat{\mathbf{y}}^{(i)}, \mathbf{y})$, where γ is the learning rate;
Repeat (1)(2), we can get a sequence of $\theta_N^{(1)} \rightarrow \theta_N^{(2)} \rightarrow ... \rightarrow \theta_N^{(k)}, ...$ until to a stable state that $|L_P(\hat{\mathbf{y}}^{(i+1)}, \mathbf{y}) - L_P(\hat{\mathbf{y}}^{(i)}, \mathbf{y})| < \epsilon$, where $\epsilon \in R$ is the threshold.

As shown above, the training process is complicated and it is hard to find how each input feature in $\mathbf{X}$ influences θ_N during the two parts of backpropagation, which also makes the final neural network model hard to be interpreted. In our proposed ISI (shown in Figure 1(c)), a mimic shallow model $\xi(\mathbf{X}; \pi_S)$ is embedded in f training process to adjust parameter updates in each iteration of f, where π_S denote the parameters of the shallow model $\xi(\mathbf{X}; \pi_S)$, respectively. Based on that, we propose a new loss function that can jointly train the shallow model $\xi(\mathbf{X}; \pi_S)$ and neural network $f(\mathbf{X}; \theta_N^{(i)})$ to gain the interpretation:

$$\theta_N^*, \pi_S^* = \underset{\theta_N, \pi_S}{\operatorname{argmin}} L_P(\xi(\mathbf{X}; \pi_S, f(\mathbf{X}; \theta_N, \mathbf{y})), \mathbf{y}), \tag{2}$$

where $L_P(\cdot)$ is the total loss function. Specifically, Equation 2 includes three parts: neural network $f(\mathbf{X}; \theta_N)$ is trained based on ground truth $\mathbf{y}$ to ensure the accuracy of ISI. Then different from mimic learning where shallow model $\xi(\mathbf{X}; \pi_S)$ is fitted by the final results of f and is used to interpret f, we jointly train $\xi(\mathbf{X}; \pi_S)$ in the training process of f, to ensure the close connection between mimic model and neural network,

since it decreases the differences between fitted $\hat{\xi}$ and f. Therefore, the shallow model can better approximate and interpret the NN than mimic learning model. Finally, we trained our joint model ISI by $L(\cdot, \mathbf{y})$ to gain interpretation. Details of ISI training process is explained in Section 4.2.

In sum, compared with the other interpretable methods, there are two major benefits of ISI: (1) The mimic shallow model $\xi(\mathbf{X}; \pi_S)$ is jointly trained with neural network f to ensure the close connection between them, rather than use the final results of f and directly fitted $\xi(\mathbf{X}; \pi_S)$ in traditional mimic learning process. Then $\xi(\mathbf{X}; \pi_S)$ can better be used for interpretation of f; (2) We can use the trained $\xi(\mathbf{X}; \pi_S^{(i)})$ in each parameter updating process to explain the feature influence in each iteration since they are jointly trained. Specifically, we can record ξ in each iteration: instead of representing the learning process as complex $f^{(1)} \rightarrow f^{(2)} \rightarrow ... \rightarrow f^{(k)}...$, it can be represented by shallow models as $\xi^{(1)} \rightarrow \xi^{(2)} \rightarrow ... \rightarrow \xi^{(k)}...$ which is easier to show the feature influence in each iteration. For example, if the shallow models are linear models, their corresponding parameters $\pi_S^{(1)} \rightarrow \pi_S^{(2)} \rightarrow ... \rightarrow \pi_S^{(k)}...$ represent variations of feature contributions of each input feature; if the shallow models are tree-based models, we can use Gini importance to calculate the variations.

4.2 Optimization of ISI

Directly optimizing Equation 2 is hard and time-consuming. In this section, we discuss how to optimize it. Specifically, we divide each learning iteration in three parts for Equation 2 and formulate them as below:

1. **Train the shallow model with soft labels**: At the i^{th} iteration, we utilize a loss function $\pi_S^{(i)} = \arg\min_{\pi_S} L_I(\xi(\mathbf{X}; \pi_S), \hat{\mathbf{y}}^{(i)})$ to train the shallow model part $\xi(\mathbf{X}; \pi_S)$. Here $\hat{\mathbf{y}}^{(i)}$ is the i^{th} iteration output of f, so it contains the knowledge acquired by f.
2. **Obtain predictions from the shallow model**: The fitted output of the shallow model is obtained by computing $\tilde{\mathbf{y}}^{(i)} = \xi(\pi_S, \mathbf{X})$ with optimized π_S. The interpretable patterns are contained in ξ, and it can also be used as a snapshot of the learning process.
3. **Update parameters of the neural network**: We use the outputs $\tilde{\mathbf{y}}^{(i)}$ from the shallow model, instead of $\hat{\mathbf{y}}^{(i)}$ from the neural network, as an approximation of NN forward prediction to compute errors and update NN parameters: $\theta_N = \arg\min_{\theta_N} L_P(\tilde{\mathbf{y}}^{(i)}, \mathbf{y})$. Due to the relatively simple structure of $\tilde{\mathbf{y}}^{(i)}$, $\tilde{\mathbf{y}}^{(i)}$ makes NN easier to be interpreted [6].

The procedure above is formally presented in Algorithm 1. We first initialize parameters $\mathbf{w}_k, \mathbf{b}_k$ in each hidden layer k, then select a shallow model to be trained in line 2 and 3. From line 4 to 7, we optimize parameters in the shallow model ξ based on loss function $L_I(\xi(\mathbf{X}; \pi_S), \hat{\mathbf{y}}^{(i)})$. From line 8 to 14, we update the parameters in f using backward propagation. We use gradient descent as an example. Note here, if traditional gradient descent is used in $L_P(\tilde{\mathbf{y}}_{(S)}^{(i)}, \mathbf{y})$ to update parameters $\mathbf{w}_N$ in f, we should calculate the derivative of ξ trained by $L_I(\xi(\mathbf{X}; \pi_S), \hat{\mathbf{y}}^{(i)})$. Even if

Algorithm 1: Interactive Stepwise Influence (ISI) Model

Input : Data $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, ...\mathbf{x}_n^T]$, $\mathbf{y}$ is the true label, C is the number of class $\mathbf{y}$ has, η is the stepsize, $\gamma \in (0, 1]$ is the fitting parameter, T is the maximum number of iterations, h is the number of hidden layer

Output : $\mathbf{y}^{(f,\xi)}$ is the output

1 Initialized $\mathbf{W}^{total} = \{\mathbf{W}_1, \mathbf{W}_2, ...\mathbf{W}_h\}, \mathbf{b}^{total} = [\mathbf{b_1}, \mathbf{b_2}...\mathbf{b_h}]$;
2 Pick explainable model ξ;
3 **for** i *from* 1 *to* T **do**
4 Assign $\hat{\mathbf{y}}^{(i)}$ by using forward-propagate of the inputs over the whole unfolded network;
5 **for** $c \in Class$ **do**
6 Optimize objective function of $L_I(\mathbf{X}; \xi(\pi_S), \hat{\mathbf{y}}^{(i)})$ based on ξ to get $\hat{\xi}_c$
7 Calculate the fitted value $\tilde{\mathbf{y}} \leftarrow \hat{\xi}_c(\mathbf{X}, \hat{\mathbf{y}}^{(i)})$
8 Calculate gradient $\frac{dL_P(\mathbf{y}, \hat{\mathbf{y}}^{(i)})}{d\mathbf{W}^{total}}$, $\frac{dL_P(\mathbf{y}, \hat{\mathbf{y}}^{(i)})}{d\mathbf{b}^{total}}$;
9 Update $\mathbf{W}^{total} \leftarrow \mathbf{W}^{total} - \eta \tilde{d} L_P(\mathbf{y}, \tilde{\mathbf{y}}^{(i)}) / \tilde{d}\mathbf{W}^{total}$;
10 $\mathbf{b}^{total} \leftarrow \mathbf{b}^{total} - \eta \tilde{d} L_P(\mathbf{y}, \tilde{\mathbf{y}}^{(i)}) / \tilde{d}\mathbf{b}^{total}$ based on previous step;
11 Assign $\hat{\mathbf{y}}^{(i+1)}$ by using forward-propagate using updated parameter $\mathbf{W}^{total}, \mathbf{b}^{total}$;
12 Calculate loss function $L_P(\mathbf{y}, \hat{\mathbf{y}}^{(i+1)})$;
13 **if** $L_P(\mathbf{y}, \hat{\mathbf{y}}^{(i+1)})$ *increase* **then**
14 Update $\eta \leftarrow \gamma\eta$;
15 Use updated $\mathbf{W}^{total}$, $\mathbf{b}^{total}$ or π_S to calculate $\mathbf{y}^{(f,\xi)}$ based on performance.

ξ is differentiable, calculating its gradient is time consuming. So instead of letting $\theta_N \leftarrow \theta_N - \eta\, dL_P(\hat{\mathbf{y}}^{(i)}, \mathbf{y})/d\theta_N$, we first calculate the derivative of $dL_P(\hat{\mathbf{y}}^{(i)}, \mathbf{y})/d\theta_N$. Then we replace $\hat{\mathbf{y}}^{(i)}$ with $\tilde{\mathbf{y}}^{(i)}$ in the calculated gradient equations in line 8 and 9. We denote the procedure as $\theta_N \leftarrow \theta_N - \eta\, \tilde{d} L_P(\tilde{\mathbf{y}}^{(i)}, \mathbf{y})/\tilde{d}\theta_N$. Thus, ISI would not be limited by non-differential shallow models.

5 Experiments

We conduct comprehensive experiments to evaluate the performance of ISI on the three interpretation aspects and accuracy. In particular, we aim to answer the following questions: (1) Can ISI provide reliable interpretations for its predictions, in terms of giving proper feature contributions and unveiling feature influences? (2) Can ISI provide reasonable interpretations for feature adaptability in its learning process? (3) Does ISI at the same time have a good precision compared to the state-of-art methods?

5.1 Data and Setup

We use three datasets including one synthetic data (SD) and two real-world datasets, i.e., MNIST and the default of credit card clients (D_CCC) [17] for classification tasks. Parametric distributions of different classes in SD are known as the basis to assess the faithfulness of the three interpretation results from ISI. The two real-world datasets are used to evaluate ISI interpretation utility and accuracy. Specifically, MNIST [16] is for

handwritten digit classification, and D_CCC is to explore features that have an influence on the occurrence of default payment (DP). D_CCC is randomly partitioned into 80% for training and validation, and 20% for testing. We use widely used and relatively robust interpretable shallow models [6]: Logistic regression (LR), Decision Trees (DT), linear SVM, the state-of-art interpretable neural network model mimic learning [6], and also neural networks ANN and CNN as baselines. Specifically, for the neural network module in ISI and mimic learning, we use the same structure of ANN with three layers where $tanh$ and $sigmoid$ are used as activation functions with considering the trade-off between performance and computation complexity as well as for fair comparison. Cross-entropy is used as the loss function. The CNN with two convolution layers, a pooling layer and a densely-connected layer are used. Hyperparameters for all methods are tuned by five-fold cross validation. Prediction accuracy is measured by AUPRC (Area Under Precision-Recall Curve) and AUROC (Area Under receiver operating Characteristic Curve) [6]. Results are reported by averaging over 100 random trails.

5.2 Interpretation Evaluation

We first test the interpretation ability of ISI in SD since ground truth is known. The task is a binary classification where data samples are generated from a mixture of multivariate Gaussian distributions $\{\mathcal{N}(\mu_{\mathbf{1}}, \mathbf{\Sigma_1}), \mathcal{N}(\mu_{\mathbf{2}}, \mathbf{\Sigma_2})\}$ of two classes. For each sample $\mathbf{x}_i \in \mathbb{R}^{(d_1+d_2)}$, d_1 and d_2 are dimensions for informative and noise features respectively. Informative features are used to separate the two classes. Noise features are appended to evaluate interpretations, as those features are not expected to affect classification. $N_1 = 1200$ and $N_2 = 1500$ denote the number of samples in each class. $d_1 = 6$, $d_2 = 6 \times 20 = 120$ and $\mathbf{\Sigma_1} = \mathbf{\Sigma_2}$ are identity matrices. Each noise feature is generated from independent standard normal distribution $\mathcal{N}(0, 1)$. To distinguish contributions among different features, we set $\mu_{\mathbf{1}} = [6, 5, 4, 3, 2, 1]^T$, $\mu_{\mathbf{2}} = [-1, -1, -1, -1, -1, -1]^T$, so the contributions of features are already sorted in a descending order according to their importance. Figure 2(a) shows a 3-D visualization of SD.

ISI	AUPRC	AUROC
ANN + LR	0.8567 ± 0.0000	0.8850 ± 0.0000
ANN + DT	0.7200 ± 0.0438	0.7357 ± 0.0438
ANN + SVM	0.8731 ± 0.0016	0.9018 ± 0.0004
ANN + LASSO	**0.8802** ± 0.0096	**0.9082** ± 0.0067

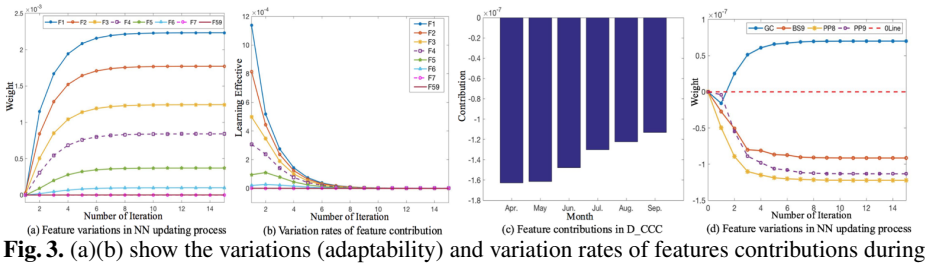
Table 2. ISI performance of different interpretable models.

	Selected features indices	NM	NP
LASSO	1, 2, 3, 4, 5, (10, 15, 31, 30)	18%	0.20
MIMIC	1, 2, 3, 4, 5, (50, 68, 99, 103, 122)	22%	0.26
ISI	1, 2, 3, 4, 5, (71)	**3%**	**0.03**

Table 3. Feature selection performances of different methods.

Table 2 shows the prediction accuracy of ISI embedded with different shallow models. When the mimic shallow model part ξ uses a linear model such as LR, linear SVM and LASSO, ISI has higher AUPRC and AUROC than that of tree-based models. This indicates that linear classifiers are preferred, which matches the features associations in synthetic data. The best accuracy performance is achieved by using LASSO in ISI, so we use it for subsequent interpretation analysis. For the first interpretation aspect "importance"of each feature, we first test the percentage of selected noise features for

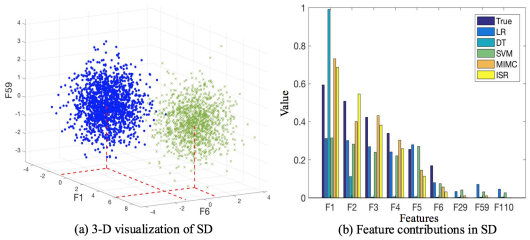
different models in Table 3 where parameters are tuned based on their best accuracy in Table 4. Indices in the parameter represent noise features. "NM" in the table denotes the possibility that the corresponding model contains noise features out of 100 trails. "NP" is the average ratio of noise features in each model. Specifically, noise features appear in 22% and 18% models over 100 random trails for LASSO and MIMIC respectively, while noise features appear in only 3% of the models for ISI. Moreover, we calculate the contributions of each feature by normalized coefficients of each linear model and Gini importances of DT. The results of each interpretable method is shown in Figure 2(b). We notice feature importance calculated by ISI are close to the true value. For second aspect "impact", since LASSO is selected in ISI shallow model part (shown in Table 2), if feature F_i changes $\triangle F_i$, the probability that it belongs to a certain group changes $\alpha_i \triangle F_i$, where α_i is the coefficient of F_i in LASSO. Those results indicate ISI can provide more reliable interpretations.



(a) Feature variations in NN updating process (b) Variation rates of feature contribution (c) Feature contributions in D_CCC (d) Feature variations in NN updating process

Fig. 3. (a)(b) show the variations (adaptability) and variation rates of features contributions during the learning process in SD. (c) illustrates contributions for some features in D_CCC. (d) depicts some parameter variations (adaptability) for D_CCC.

For the third aspect "adaptability", the NN f can be intuitively explained using the embedded shallow models in ISI. The approximated variation of each feature contribution is shown in Figure 3(a)(b). They are calculated by using features weights (coefficients) of embedded shallow models in each iteration, since LASSO is selected as shallow models. The variation rates in Figure 3(b) are the weight differences of two adjacent iterations. As the learning process proceeds, contributions to the final results of each informative feature becomes more clear, at a fast rate especially in the early stages of training. The weight of noise features approaches 0. It also matches the converge process in NN

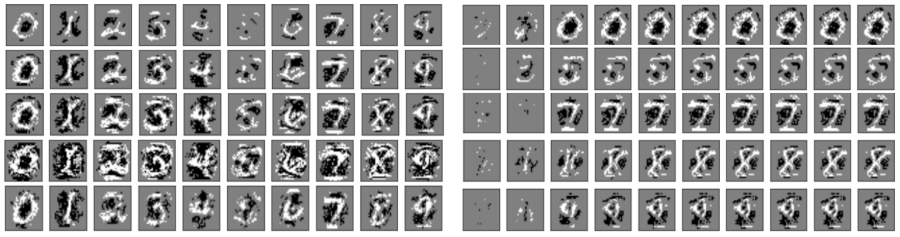


(a) 3-D visualization of SD (b) Feature contributions in SD

Fig. 2. Fi is the i^{th} input feature of SD. (a) uses three features to give a 3-D visualization of SD. Different colors mean different groups. (b) calculates feature contributions of different interpretable methods based on the accuracy in Table 4.

learning iterations. Such information may help people understand the final parameters of neural network.

For real-word dataset, ISI also shows extraordinary and reliable interpretability in terms of the three interpretable aspects that we targets. For MNIST, we select LASSO as the shallow model part of ISI to interpret classification results according to best accuracy. Figure 4(a) shows input pixels contributions measured by corresponding coefficients in LASSO. The darker area means that the corresponding pixels have higher negative relations to the class, while the lighter area means the weights have more positive relations. Specifically, gray area means that the coefficients of corresponding pixels in the shallow part are approximate to zero. For example, for pixels in an image with high values, if they are in the lighter area, there is a higher probability that the image would be classified to the corresponding digit. While if those pixels are in the darker area, the image is less likely to be classified to the corresponding digit. Gray area means the corresponding pixels have little contribution to detecting digit. Here we can observe that the white areas sketch the outline of each digit and dark areas are near them. Gray areas are far from the outline of digits. Figure 4(b) shows five examples of feature variations interpretation results from ISI in the first 100 iterations. The interval between two columns is 10 iterations. The results show that there are no specific patterns at the beginning regarding how to classify a digit. But after more iterations, we can see that the sketch of each digit highlighted by white areas becomes more obvious. For D_CCC, LR is selected as the shallow model ISI to explain the three interpretation aspects based on the accuracy performance. Figure 3(c) shows the contribution of the amount of previous payment in each month. It indicates that the amount of previous payments in April and May strongly influence DP. Since linear model LR is selected, each feature influence is the product of the corresponding coefficient of LR and the changes of the feature. Figure 3(d) shows the feature adaptability. It also gives reasonable explanation of each features to the final DP [23].



(a) Selected features for different λ (b) Feature variations in NN updating process

Fig. 4. (a) Selected features by ISI with LASSO in MNIST dataset. The first four rows are calculated with $\mathcal{L}_1$ regularization parameter $\lambda = 0.5, 0.1, 0.05, 0.01$ and 100 hidden units. The results in the last row are calculated with $\lambda = 0.01$ and 500 hidden units. (b) Variations of feature weights in NN learning process of five examples with $\lambda = 0.1$ and 100 hidden units.

5.3 Prediction Accuracy Evaluation

In this section, we evaluate the prediction capability of ISI in AUPRC and AUROC, compared with other classification models as baselines cross the three different datasets.

	MNIST		D_CCC		SD	
Method	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
LR	0.8159 ± 0.0113	0.9589 ± 0.0021	0.5954 ± 0.0017	0.6482 ± 0.0030	0.8721 ± 0.0109	0.9007 ± 0.0075
DT	0.7570 ± 0.0140	0.9189 ± 0.0019	0.5177 ± 0.0451	0.5321 ± 0.0091	0.8595 ± 0.0079	0.8128 ± 0.0109
SVM	NA	NA	0.5349 ± 0.0400	0.5661 ± 0.0048	0.8626 ± 0.0072	0.8944 ± 0.0072
ANN	0.9726 ± 0.0023	0.9946 ± 0.0006	0.6792 ± 0.0189	0.6133 ± 0.0049	0.8891 ± 0.0139	0.9119 ± 0.0093
CNN	0.9894 ± 0.0007	0.9982 ± 0.0001	0.6002 ± 0.0597	0.5010 ± 0.0018	0.8706 ± 0.0104	0.8987 ± 0.0104
MIMIC	0.7219 ± 0.0086	0.9261 ± 0.0029	0.5446 ± 0.0028	0.5790 ± 0.0028	0.8789 ± 0.0151	0.9062 ± 0.0123
ISI	0.8722 ± 0.0033	0.9710 ± 0.0003	0.6066 ± 0.0101	0.6553 ± 0.0083	0.8802 ± 0.0096	0.9082 ± 0.0067

Table 4. Accuracy performance on the three datasets. MIMIC learning uses SVM, LASSO, LASSO respectively for the three datasets. ISI is embedded with LASSO, LR and LASSO for MNIST, D_CCC and SD, respectively. Here different shallow models are used for different datasets because we choose the best NN-shallow models combination for each case. ISI outperforms all the interpretable models. The performance of ISI is comparable to that of ANN and CNN, and sometimes is even better.

Here MIMIC learning uses the same NN structure as ISI for fair comparison. For the shallow part in MIMIC and ISI methods, we try difference shallow models (LR, DT, SVM, LASSO) in each dataset and reports the best performed ones. Table 4 shows the accuracy of ISI compared with baseline methods. "NA" here means the corresponding method takes more than 10 times longer than the other methods.

Overall, we see the full-blown ISI improves upon all the other interpretable models cross the three different datasets. Moreover, the performance of ISI is comparable to that of ANN and CNN while ISI is also easier to interpret. From the first three rows of LR, DT and SVM in Table 4, ISI improves versus the next-best alternative an average of 3.24% in AUPRC and 1.06% in AUROC. It may contributes to ISI neural network structure. Comparing with traditional NN, the AUROC of ISI is significantly higher than the AUROC of ANN in D_CCC dataset. Based on the row of MIMIC method, ISI outperforms the state-of-the-art interpretable model MIMIC 10.78% on average in AUPRC and 6.08% in AUROC. It shows by jointly training shallow models and neural network, ISI can gain a higher accuracy. Moreover, based on the standard deviation of each experiment, ISI is also more robust than MIMIC learning in terms of stability. This further shows that ISI has desirable discriminative power after being incorporated into the shallow model to enable interpretability.

6 Conclusions and Future Work

We have proposed a novel interpretable neural network framework ISI which embeds shallow interpretable models in NN learning process, and they are jointly trained to gain both accuracy and interpretability. Through experiments over different datasets, ISI not only outperforms the state-of-the-art methods, but also can be reasonably explained in three aspects: feature importance, feature impact and the adaptability of feature weights in NN learning process. Notice here ISI is mainly applied in areas where interpretability is necessary and traditional models are still widely used [6, 22], such as political and economics area. For the future work, how to choose proper interpretable shallow models and applying ISI to more complex data and other neural network architectures are promising directions for future explorations.

References

1. Balan, A.K., Rathod, V., Murphy, K.P., Welling, M.: Bayesian dark knowledge. In: NIPS (2015)
2. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. arXiv preprint arXiv:1704.05796 (2017)
3. Cano, J.R., Herrera, F., Lozano, M.: Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability. Data & Knowledge Engineering **60**(1) (2007)
4. Che, Z., Liu, Y.: Deep learning solutions to computational phenotyping in health care. In: ICDMW. IEEE (2017)
5. Che, Z., Purushotham, S., Khemani, R., Liu, Y.: Distilling knowledge from deep networks with applications to healthcare domain. arXiv preprint arXiv:1512.03542 (2015)
6. Che, Z., Purushotham, S., Khemani, R., Liu, Y.: Interpretable deep models for ICU outcome prediction. In: AMIA Annual Symposium Proceedings. vol. 2016. American Medical Informatics Association (2016)
7. Choi, E., Bahadori, M.T., Sun, J., Kulas, J., Schuetz, A., Stewart, W.: RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In: NIPS (2016)
8. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. arXiv preprint arXiv:1808.00033 (2018)
9. Du, M., Liu, N., Song, Q., Hu, X.: Towards explanation of dnn-based prediction with guided feature inversion. In: KDD (2018)
10. Freitas, A.A.: Comprehensible classification models: a position paper. ACM SIGKDD explorations newsletter **15**(1) (2014)
11. Frosst, N., Hinton, G.: Distilling a neural network into a soft decision tree. arXiv preprint arXiv:1711.09784 (2017)
12. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW. International World Wide Web Conferences Steering Committee (2017)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Jennings, D., Amabile, T.M., Ross, L.: Informal covariation assessment: Data-based vs. theory-based judgments. udgment Under Uncertainty: Heuristics and Biases (1982)
15. Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078 (2015)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998)
17. Merz, C.J., Murphy, P.M.: {UCI} repository of machine learning databases (1998)
18. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you ?": Explaining the predictions of any classifier. In: SIGKDD. ACM (2016)
19. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61** (2015)
20. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. arXiv preprint arXiv:1703.01365 (2017)
21. Ustun, B., Rudin, C.: Methods and models for interpretable linear classification. arXiv preprint arXiv:1405.4047 (2014)
22. Wang, J., Fujimaki, R., Motohashi, Y.: Trading interpretability for accuracy: Oblique treed sparse additive models. In: SIGKDD (2015)
23. Yeh, I.C., Lien, C.h.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications (2009)
24. Zhang, Q., Wu, Y.N., Zhu, S.C.: Interpretable convolutional neural networks. In: CVPR. pp. 8827–8836 (2018)