INTERPRETABILITY EVALUATION FRAMEWORK FOR DEEP NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) have attained surprising achievement during the last decade due to the advantages of automatic feature learning and freedom of expressiveness. However, their interpretability remains mysterious because DNNs are complex combinations of linear and nonlinear transformations. Even though many models have been proposed to explore the interpretability of DNNs, several challenges remain unsolved: 1) The lack of interpretability quantity measures for DNNs, 2) the lack of theory for stability of DNNs, and 3) the difficulty to solve nonconvex DNN problems with interpretability constraints. To address these challenges simultaneously, this paper presents a novel intrinsic interpretability evaluation framework for DNNs. Specifically, Four independent properties of interpretability are defined based on existing works. Moreover, we investigate the theory for the stability of DNNs, which is an important aspect of interpretability, and prove that DNNs are generally stable given different activation functions. Finally, an extended version of deep learning Alternating Direction Method of Multipliers (dlADMM) are proposed to solve DNN problems with interpretability constraints efficiently and accurately. Extensive experiments on several benchmark datasets validate several DNNs by our proposed interpretability framework.

1 INTRODUCTION

The last decade has witnessed the tremendous success of deep neural networks (DNNs) in a variety of domains since Alexnet ranked first in the ImageNet Large Scale Visual Recognition Challenge in 2012 (Krizhevsky et al., 2012). The main advantages of DNNs over traditional machine learning models are two folds: firstly, DNNs can learn features automatically through linear and nonlinear transformations, while traditional machine learning models require prior knowledge and feature engineering; secondly, DNNs usually outperform traditional machine learning models tremendously. This is because DNNs have more flexibility and freedom of expressiveness than traditional machine learning models.

Even though DNNs have achieved outstanding performance in many applications, interpreting DNNs is very challenging because DNNs are a combination of linear and nonlinear transformation and hence lack transparency, whose decision making process is still not well understood by human beings. Without sufficient interpretability, their applications in specialized domains such as finance are largely limited. For example, it is useful for applicants to know the reason why their mortgage applications are denied, to get the loan when applied again in the future.

A surge of works have been proposed to explore the interpretability of DNNs recently, they are mainly classified as two categories: the post-hoc approach and the intrinsic approach. The post-hoc approach proposes an interpretable model to interpret a pre-trained DNN. This approach is independent of structures of DNNs. Different from the post-hoc approach, the intrinsic approach explains DNNs directly by imposing regularization and constraints. Please refer to (Du et al., 2018) for more information.

Although many approaches have been proposed to interpret DNNs intrinsically, several challenges remain unsolved including **1. The lack of interpretability quantity measures for DNNs.** Existing literature focuses mainly on the qualitative investigation: For example, Melis and Jaakkola proposed three desiderata for self-explaining models: explicitness, faithfulness, and stability (Melis & Jaakkola, 2018); Vaughan et al. and Yang et al. presented an additive index model with architecture constraints including sparsity, orthogonality and smoothness (Vaughan et al., 2018; Yang et al., 2019). However, there still lacks an investigation of the quantitative measures of interpretability. **2. The lack of theory for the stability of DNNs.** Even though previous literature has proposed practical measures for the stability of DNNs (Alvarez-Melis & Jaakkola, 2018), which is considered as an important issue of interpretability, the theoretical analysis of stability for DNNs has not yet been established. This is because stability analysis requires the smoothness of DNNs while DNNs can be nonsmooth, which makes theoretical analysis rather difficult. **3. The difficulty to solve nonconvex DNN problems with interpretability constraints.** Many recent works have imposed interpretability constraints on DNNs like an orthogonality constraint (Yang et al., 2019), but such hard constraints are nonlinear and nonconvex such that common state-of-the-art optimizers such as Stochastic Gradient Descent (SGD) are not suitable for such problems (Márquez-Neila et al., 2017). As a result, an optimization framework to solve these problems is demanding.

To address these challenges simultaneously, we propose a novel intrinsic interpretability evaluation framework for DNNs. Our interpretability evaluation framework is established on the formulation of fully-connected neural networks. Specifically, we define four interpretability properties based on existing literature and give quantitative measures for interpretability evaluation. These properties are shown to be conceptually independent. We show that many previous works are special cases of our interpretability evaluation framework. Moreover, we prove that the fully-connected neural network is theoretically globally stable given smooth activation functions, and locally stable given nonsmooth activation functions. Finally, the deep learning Alternating Direction Method of Multipliers (dlADMM) (Wang et al., 2019) is extended to solve DNN problems with interpretability constraints, and global convergence to a critical point is maintained. Our contributions in this paper include:

- We present a novel framework to evaluate the interpretability of a fully-connected neural network, which can be extended to other network structures. Four interpretability properties are introduced, and quantitative measures are given.
- The stability aspect of the interpretability for DNNs is analyzed theoretically. Two types of stabilities are discussed, and we prove that DNNs are globally stable for smooth activation functions, and locally stable for nonsmooth activation functions.
- An extended version of deep learning Alternating Direction Method of Multipliers (dlADMM) is proposed to handle interpretability constraints. This optimization framework is efficient, and convergence to a critical point can be guaranteed.
- We conduct experiments on several benchmark datasets to evaluate the interpretability of different layers of fully-connected neural networks by our proposed evaluation framework.

The rest of this paper is organized as follows. In Section 2, we summarize recent research related to this topic. In Section 3, we present our interpretability framework to evaluate a fully-connected neural network. In Section 4, an extended version of dlADMM algorithm is proposed to handle interpretability constraints. The experimental results are reported in Section 5, and Section 6 concludes this paper by summarizing the research.

2 Related Work

The previous works on intrinsically interpretable models are related to this paper, which can be summarized as follows:

Sparsity-based Interpretable Models. Sparsity is considered as an important factor of interpretability in many machine learning models. Sparse models indicate small subsets of useful features and reflect good interpretability. Many models impose ℓ_1 penalty on the objective function in order to restrict the number of useful features: Bouchard et al. proposed a Union of Intersection (UoI) model for model selection and compression (Bouchard et al., 2017); Yang et al. added ℓ_1 penalties in their interpretable neural network model to ensure sparsity in both the scales of ridge functions and the projection weights (Yang et al., 2019); Wang et al. proposed an interpretable model to detect vaccine adverse events (Wang et al., 2018). Some works utilized a ℓ_2 regularized term to ensure feature sparsity, because ℓ_2 is differentiable and models can be solved by gradient-based optimization methods (Ross et al., 2017; Wu et al., 2018; Tong et al., 2018; Bansal et al., 2018). Other papers aimed to impose sparsity on group structures: for instance, Scardapane et al. proposed a $\ell_{2,1}$ regularization to ensure the sparsity on the input groups, hidden groups, and bias groups of neural networks (Scardapane et al., 2017); Tsang et al. proposed a disentangled group regularizer to disentangle feature interactions (Tsang et al., 2018).

Stability-based Interpretable Models. Aside from sparsity, stability is also one of the considerations of interpretability: stability refers to the idea that similar inputs generate similar interpretations, in other words, the interpretation should not change much when the input changes little. Previous literature mentioned stability explicitly in their models: for example, Zhang et al. presented an algorithm to generate stable corrections, which are a useful way to provide feedback to users (Zhang et al., 2018); Melis and Jaakkola required basic interpretable concepts to be stable (i.e. difference-bounded in their definition) in their self-explaining models (Melis & Jaakkola, 2018); Melis and Jaakkola also proposed a quantity to gauge stability (Alvarez-Melis & Jaakkola, 2018). For other papers, stability was a byproduct of their interpretable models. For more information, Please see (Chen et al., 2016; Yeh et al., 2017).

Other Regularized Interpretable Models. Apart from sparsity and stability, many models considered other aspects of interpretability: as an example, Melis and Jaakkola proposed faithfulness and explicitness as two additional properties of interpretability (Melis & Jaakkola, 2018); In the algorithm presented by Zhang et al, they guaranteed that the correction is minimal and symbolic (Zhang et al., 2018); Yang et al. imposed tow additional orthogonality and smoothness constraints on the interpretable additive index model (Yang et al., 2019); Hsu et al. presented a factorized hierarchical variational autoencoder to learn disentangled and interpretable representations from sequential data (Hsu et al., 2017). However, to the best of our knowledge, there still lacks an quantitative investigation of interpretability for DNNs.

3 Interpretability of a Fully-connected Neural Network

In this section, the interpretability of a fully-connected neural network is discussed in detail. Table 1 introduces necessary mathematical notations. We consider a typical fully-connected neural network of L layers, which generally is composed of by multiple linear mappings and nonlinear activation functions. A linear mapping for the l-th layer is defined by a weight matrix $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$ and an intercept $b_l \in \mathbb{R}^{n_l}$, where n_l is the number of neurons for the l-th layer; a nonlinear mapping for the l-th layer is defined by a continuous nonlinear activation function $f_l(\bullet)$. a_l is denoted as the output of the l-th layer, or equivalently the input of the $(l+1)$-th layer, then the relation between the a_l and a_{l-1} is shown as $a_l = f_l(z_l) = f_l(W_l a_{l-1} + b_l)$, where $z_l = W_l a_{l-1} + b_l$ is an auxiliary variable. Using this fact, the relation between the output a_{L-1} and the input a_0 is shown as $a_{L-1} = f_{L-1}(W_{L-1} f_{L-2}(W_{L-2} \cdots f_1(W_1 a_0 + b_1) + \cdots + b_{L-2}) + b_{L-1})$.

Now we introduce four interpretability properties for a fully-connected neural network in this section, all of them have appeared in the previous literature, but are not defined formally. First of all, the sparsity is defined as follows:

Definition 1 (Sparsity)**.** *The l-th layer is ε-sparse in $\ell_{p,q}$-norm if $\|W_l\|_{p,q} \le \varepsilon$.*

Sparsity controls the number of nonzero weights. Intuitively, sparse models mean that predictions of models are closely related to a small subset of features. Such models usually provide users with good interpretations, which can be expressed by several concise sentences for people to easily understand. For example, a grade prediction model shows that grades of a course are only positively correlated to the time spent on homework, which provides good feedback for the instructor, who may assign more homework to students to improve course grades.

Table 1: Important Notations and Descriptions

Notations	Descriptions
L	Number of layers.
W_l	The weight matrix for the l-th layer.
$W_{l,i}$	The i-th row of W_l.
$W_{l,i,j}$	The i-th row, j-th column of W_l.
b_l	The intercept for the l-th layer.
$f_l(\bullet)$	The nonlinear activation function for the l-th layer.
a_l	The output of the l-th layer.
a_0	The input of a fully-connected neural network.
y	The predefined label vector.
$\Omega_l(W_l)$	The regularization term for the l-th layer.
n_l	The number of neurons for the l-th layer.

The definition of sparsity is well-generalized: it includes common regularization terms like ℓ_1 or ℓ_2 penalties, and group sparsity penalties like $\ell_{2,1}$. ε quantifies the degree of sparsity: the less ε is, the more sparse the l-th layer is.

The second property, stability, is another important property of interpretability, which is defined as follows:

Definition 2 (Stability)**.**
(a). The fully-connected neural network is globally zeroth-order stable if there exists a constant

$H_1 > 0$ *such that for any two inputs* $a_0^{'}$ *and* $a_0^{''}$, $\|a_{L-1}^{''}-a_{L-1}^{'}\| \leq H_1\|a_0^{''}-a_0^{'}\|$. *The fully-connected neural network is locally first-order stable if there exist a constant* $H_2 > 0$ *and a neighborhood* $N(a_0)$ *such that for any two inputs* $a_0^{'}, a_0^{''} \in N(a_0)$, $\|a_{L-1}^{''} - a_{L-1}^{'}\| \leq H_2\|a_0^{''} - a_0^{'}\|$.
(b). Assume $f_l(\bullet)$ *is subdifferentiable such that* $\partial a_{L-1}/\partial a_0$ *exists, then the fully-connected neural network is globally first-order stable if there exists a constant* $M_1 > 0$*, for any two inputs* $a_0^{'}$ *and* $a_0^{''}$, $\|\partial a_{L-1}^{''}/\partial a_0^{''} - \partial a_{L-1}^{'}/\partial a_0^{'}\| \leq M_1\|a_0^{''} - a_0^{'}\|$. *the fully-connected neural network is locally first-order stable if there exist a constant* $M_2 > 0$ *and a neighbor of* a_0 $N(a_0)$*, such that for any two inputs* $a_0^{'}, a_0^{''} \in N(a_0)$, $\|\partial a_{L-1}^{''}/\partial a_0^{''} - \partial a_{L-1}^{'}/\partial a_0^{'}\| \leq M_2\|a_0^{''} - a_0^{'}\|$.

Stability measures the consistency of the interpretability, which implies that two similar inputs should lead to similar outputs and interpretations. For example, two applicants who both have excellent credit records and high salaries should both approved by the banking decision support system to get the loan.
In the above definition, we consider two types of stability: zeroth-order stability and first-order stability. Zeroth-order stability means that the change of a_{L-1} (i.e. output) is bounded by the change of a_0 (i.e. input); while first-order stability means that the change of $\partial a_{L-1}/\partial a_0$ (i.e. interpretation) is bounded by the change of a_0 (i.e. input). The zeroth-order stability is weaker than the first-order one: if a fully-connected neural network is locally/globally first-order stable, then it is also locally/globally zeroth-order stable. Similarly, local stability is also weaker than the global one. This means that if a fully-connected neural network is globally zeroth-order/first-order stable, then it is also locally zeroth-order/first-order stable. The following theorem guarantees that a fully-connected neural network is guaranteed to be zeroth-order stable for common activation functions $f_l(\bullet)$:

Theorem 1 (Zeroth-order Stability)**.** *If the activation function* f_l *is either sigmoid, tanh, ReLU or leaky ReLU, then the fully-connected neural network is globally zeroth-order stable, and hence is also locally zeroth-order stable.*

The proof of Theorem 1 is in the Appendix. Theorem 1 provides a upper bound of C. The case is more complex for first-order stability, which is summarized in the following theorem:

Theorem 2. *(First-order Stability) If the activation function* f_l *is either sigmoid or tanh, then the fully-connected neural network is globally first-order stable. Moreover, if the activation function* f_l *is either ReLU or leaky ReLU and* $b_l \neq 0$*, then the fully-connected neural network is locally first-order stable almost surely (i.e. with probability 1).*

The fully-connected neural network is guaranteed to be zeroth-order/first-order locally stable from Theorems 1 and 2, this shows that individual interpretation of a_0, which can be defined by $\partial a_{L-1}/\partial a_0$, remains consistent when a_0 is in a small neighbourhood $N(a_0)$. However, for nonsmooth activation functions, the global first-order stability is not necessarily achieved, due to the abrupt change of derivative directions at nonsmooth points, while it can be achieved for smooth activation functions, as mentioned in Theorem 2. In this sense, smooth activation functions are interpretable than nonsmooth ones.
Previous papers have proposed quantitative measures to evaluate the zeroth-order and first-order stability, which are shown as follows (Alvarez-Melis & Jaakkola, 2018; Melis & Jaakkola, 2018):

$$C_1 = \arg\max_{\forall a_0^{'}, a_0^{''}} \|a_{L-1}^{''} - a_{L-1}^{'}\|/\|a_0^{''} - a_0^{'}\| \tag{1}$$

$$C_2 = \arg\max_{\forall a_0^{'}, a_0^{''}} \|\partial a_{L-1}^{''}/\partial a_0^{''} - \partial a_{L-1}^{'}/\partial a_0^{'}\|/\|a_0^{''} - a_0^{'}\| \tag{2}$$

where $C_1, C_2 > 0$ are stability quantities.

The third property of interpretability is faithfulness, which is defined as follows:

Definition 3 (Faithfulness)**.** *Assume* u_l *is a vector of performance gain whose* i*-th element is* $u_{l,i} = R|_{W_{l,i}=0} - R$*, where* R *is a risk function of fully-connected neural network, and* $R(z_l)|_{W_{l,i}=0}$ *is the risk function such that* $W_{l,i} = 0$ *while fixing other weights. The* l*-th layer is* β*-faithful if the correlation between the performance gain* u_l *and feature importance* $\|W_{l,i}\|$ *is no less than* β*, i.e* $corr(u, \|W_{l,i}\|)) \geq \beta$*, where* $corr(\bullet, \bullet)$ *is a correlation function, and* $\beta > 0$ *is a threshold.*

Faithfulness is a quantitative measure to assess the reliability of fully-connected neural networks. DNNs should be faithful if weights truly reflect the importance of features. That is, if important features are removed from DNNs in ablation study, performance should decrease in proportion to the importance of such features. Interpretations which faithful models provide reflect true importance of features and thus are reliable.

Finally, the definition of disentanglement is given below:

Definition 4 (Disentanglement)**.** *Assume $W_{l,i}$ is the i-th row of the W_l, i.e. the vector representation of the i-th neuron for the l-th layer, then the i-th neuron and the j-th neuron are α-disentangled if $\cos(W_{l,i}, W_{l,j}) = W_{l,i}W_{l,j}^T / \|W_{l,i}\|_2 \|W_{l,j}\|_2 \leq \alpha$, where cos($\bullet, \bullet$) is a cosine similarity function, and $\alpha > 0$ is a threshold.*

The disentanglement is defined to measure the independence of neurons in the same layer. If the i-th neuron and j-th neuron are different (i.e. $\cos(W_{l,i}, W_{l,j})$ is small) , then they are more likely to represent independent concepts. For example, assume the i-th neuron, the j-th neuron, and the k-th neuron for the l-th layer represent a human face, eye, and nose, respectively. Then $\cos(W_{l,i}, W_{l,j})$ should be large because face representation includes eye while $\cos(W_{l,j}, W_{l,k})$ should be small because eye and nose are independent of each other. Disentangled models learn independent features which represent atomic components of data. As an example, a disentangled model to learn a human face should return atomic features such as eye, nose, and ear, and therefore increases the interpretability of a model.

However, disentanglement is not a necessary condition of interpretability. In other words, even though a model is not disentangled, it may be still an interpretable model. For example, a neural network all whose neurons behave the same is easy to interpret, but it does not satisfy disentanglement.

Now the relationship between interpretability and interpretability properties are summarized in Figure 1: an interpretable model implies sparsity, stability and faithfulness, but not disentanglement, and every interpretability property is independent of others. Due to space limit, the concept independence among interpretability properties and the relation between our proposed interpretability framework and previous works are detailed in the supplementary materials.

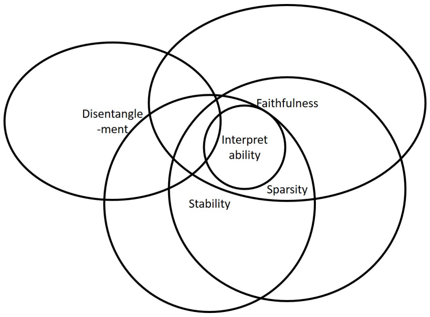


Figure 1: The relation between interpretability and interpretability properties.

4 An Optimization Framework to Solve Interpretability-constrained Problems

We adapt a novel optimization framework in this section to DNN problems with interpretability constraints, which can be formulated as follows (Wang et al., 2019):

$$\min_{W_l,b_l,z_l,a_l} F(\mathbf{W},\mathbf{b},\mathbf{z},\mathbf{a}) = R(z_L;y) + \sum\nolimits_{l=1}^{L} \Omega_l(W_l) + (\nu/2)\sum\nolimits_{l=1}^{L-1}(\|z_l - W_l a_{l-1} - b_l\|_2^2 + \|a_l - f_l(z_l)\|_2^2)$$

$$s.t.\ z_L = W_L a_{L-1} + b_L, G_l(W_l) = 0\ (l = 1, \cdots, L)$$

where $\mathbf{W} = \{W_l\}_{l=1}^L$, $\mathbf{b} = \{b_l\}_{l=1}^L$, $\mathbf{z} = \{z_l\}_{l=1}^L$, $\mathbf{a} = \{a_l\}_{l=1}^{L-1}$, $R(z_L;y)$ is a risk function, y is a predefined label vector, $\nu > 0$ is a tuning parameter, and $G_l(W_l) = 0 (l = 1, \cdots, L)$ are any interpretability constraints for the l-th layer.

General DNN problems are conventionally solved by state-of-the-art Stochastic Gradient Descent (SGD) and its variant. However, they are not applicable for the case where the hard nonconvex constraint $G_l(W_l) = 0$ is imposed on the DNN problems. This is because SGD and its variants can not guarantee the feasibility of $G_l(W_l) = 0$. On the other hand, the recently proposed deep learning Alternating Direction Method of Multipliers (dlADMM) by Wang et al. can be adapted to solve the above interpretability-constrained DNN problem. To achieve this, the augmented Lagrangian function L_ρ is formulated mathematically as follows:

$$L_\rho(\mathbf{W},\mathbf{b},\mathbf{z},\mathbf{a},u) = F(\mathbf{W},\mathbf{b},\mathbf{z},\mathbf{a}) + \mathbb{I}(G_l(W_l) = 0)$$
$$+ u^T(z_L - W_L a_{L-1} - b_L) + (\rho/2)\|z_L - W_L a_{L-1} - b_L\|_2^2$$

where u is a dual variable and $\rho > 0$ is a parameter. $\mathbb{I}(G_l(W_l) = 0)$ is an indicator function such that its value is 0 if $G_l(W_l) = 0$ and $+\infty$ for otherwise. The strategy of the dlADMM framework is to update W_l,b_l, z_l and a_l backward and then forward. The above problem is solved by the same routine of Algorithm 1 in (Wang et al., 2019). The only modification is to solve W_l: the interpretability constraints $G_l(W_l) = 0$ must be satisfied before jumping out of loops in Line 2 of Algorithm 3 and Algorithm 4. The Algorithms 1 and 2 of the extended dlADMM are shown in the Appendix, which correspond to Algorithm 3 and 4 in (Wang et al., 2019), respectively. The theoretical guarantee of the dlADMM algorithm still holds: the extended dlADMM algorithm is guaranteed to converge globally to a critical point of the above problem when ρ is sufficiently large.

5 Experiments

In this section, we evaluate our proposed interpretability evaluation framework using benchmark datasets. The effectiveness and scalability of the extended dlADMM algorithm are also examined. All experiments were conducted on 64-bit Ubuntu16.04 LTS with Intel(R) Xeon processor and GTX1080Ti GPU.

5.1 Experiment Setup

5.1.1 Dataset

In this experiment, two benchmark datasets were used for performance evaluation: MNIST (LeCun et al., 1998) and Fashion MNIST (Xiao et al., 2017). The MNIST dataset has ten classes of handwritten-digit images, which was firstly introduced by Lecun et al. in 1998 (LeCun et al., 1998). It contains 55,000 training samples and 10,000 test samples with 784 features each, which is provided by the Keras library (Chollet, 2017). Unlike the MNIST dataset, the Fashion MNIST dataset has ten classes of assortment images on the website of Zalando, which is Europes largest online fashion platform (Xiao et al., 2017). The Fashion-MNIST dataset consists of 60,000 training samples and 10,000 test samples with 784 features each.

5.1.2 Experiment Settings

We set up three DNN architectures: (1). The DNN contained four hidden layers with 300 neurons each. (2). The DNN contained three hidden layers with 500 hidden units each. (3). The DNN contained two hidden layers with $1,000$ hidden units each For each DNN architecture, we consider two different problem constraints: (a). no regularization term, and (b). The regularization term is set as $\Omega_l(W_l) = \lambda\|W_l\|_1$ where $\lambda > 0$ is a tuning parameter and was set to 10^{-5}. Therefore, Four problem formulations are used for interpretability evaluation altogether. DNN (1)+(a) denotes that the DNN architecture (1) with problem constraint (a) and so on.

For other common settings, the Rectified Linear Unit (ReLU) was used for the activation function for all network structures. The loss function was set as the cross-entropy loss. ν was set to 10^{-6}. ρ was initialized to be 10^{-6}. The number of iteration was set to 100.

In the experiments, several metrics were utilized to evaluate model interpretability including sparsity, zeroth-order stability, disentanglement, and faithfulness. Besides, the accuracy is used to evaluate model performance, which is the ratio of accurately labeled samples to all samples.

5.2 Experimental Results

The results of experiments are detailed in this section.

5.2.1 Sparsity

Table 2 shows the average ℓ_1 norm of weight matrices W_l on the MNIST and Fashion MNIST datasets from DNN architectures (1), (2) and (3), respectively. Overall, constraint (b) reduces the sparsity of all DNN architectures because the ℓ_1 regularization is imposed in the constraint (b). However, the degree of reduced sparsity is different: the reduced sparsity from architectures (1) and (2) is much more significant than architecture (3). For example, while the average ℓ_1 norm of layer 1 from architecture (1) on the Fashion MNIST dataset is

Dataset	Constraint	DNN architecture (1)		DNN architecture (2)			DNN architecture (3)			
		Layer 1	Layer 2	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3	Layer 4
MNIST	(a)	0.0798	0.0798	0.0798	0.0798	0.0799	0.0798	0.0796	0.0796	0.0796
MNIST	(b)	0.0602	0.0344	0.0609	0.0263	0.0385	0.0770	0.0754	0.0758	0.0764
Fashion MNIST	(a)	0.0798	0.0798	0.0797	0.0798	0.0798	0.0798	0.0796	0.0796	0.0796
Fashion MNIST	(b)	0.0240	0.0384	0.0242	0.0362	0.0402	0.0739	0.0764	0.0768	0.0770

Table 2: The average ℓ_1 norm of weight matrices from three DNN architectures: Constraint (b) reduces the average ℓ_1 norm on two datasets significantly.

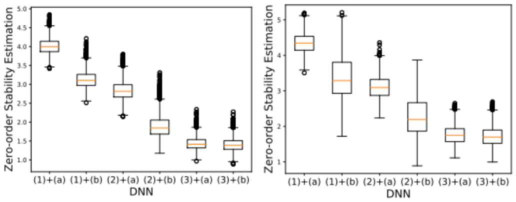
Dataset	Constraint	DNN architecture (1)		DNN architecture (2)			DNN architecture (3)			
		Layer 1	Layer 2	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3	Layer 4
MNIST	(a)	0.0707	0.0026	0.0383	0.0165	0.0119	0.0274	-0.0246	-0.0322	0.0486
MNIST	(b)	0.0951	0.3808	0.0108	0.3976	0.2033	0.0380	0.0074	-0.0032	0.0743
Fashion MNIST	(a)	0.0011	0.0092	0.0526	0.0190	0.0090	0.0284	-0.0016	-0.0372	0.0175
Fashion MNIST	(b)	0.01	0.1383	0.3148	0.1743	0.2355	0.0579	0.0100	-0.0300	0.0256

Table 3: The faithfulness of DNN architectures (1) (2) and (3).

around 0.08 when no constraint is imposed (i.e. constraint (a)), and this value drops drastically to 0.0240 when constraint (b) takes effect, this value changes little for architecture (3).

5.2.2 ZEROTH-ORDER STABILITY

The zeroth-order stability is estimated from Equation equation 1. The smaller a C is, the more stable a DNN is. Figure 2(a) and (b) show the estimation of the zeroth-order stability on the MNIST and the Fashion MNIST datasets, respectively. X-axis and Y-axis reflect different DNN formulations and the estimated zeroth-order stability, respectively. The estimated zeroth-order stability is larger from architectures (1) and (2) than that from architecture (3). For example, the estimated value is around 4 or more from architecture (1) on two datasets, while it is only 1.5 from architecture (3). Moreover, constraint (b) again has a more significant effect on the stability from architectures (1) and (2) than from architecture (3). But the variance of the estimation is also enlarged: for instance, the rectangles in Figure 2(b) when constraint (b) is imposed on DNN architectures are wider than those when constraint (a) is imposed.



(a). The MNIST dataset.

(b). The Fashion MNIST dataset.

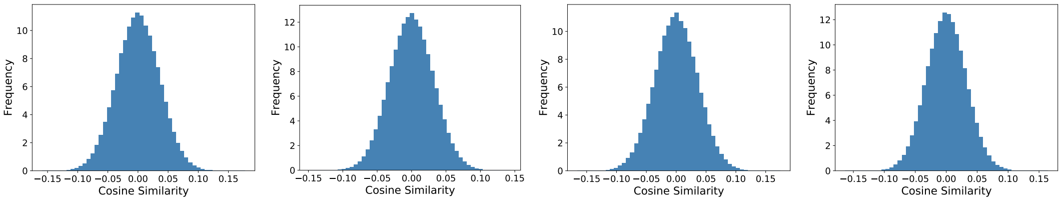
Figure 2: The estimation of the zeroth-order stability for different DDN formulations: constraint (b) enhances zeroth-order stability.

5.2.3 FAITHFULNESS

Table 3 shows the value of faithfulness of architectures (1), (2) and (3). Generally, the interpretability constraint (b) does improve the faithfulness for all architectures, but the degrees are more obvious for architectures (1) and (2) than (3). For example, the faithfulness of layer 2 from architecture 1 on the MNIST dataset increases by 0.38 when constraint (b) is imposed on the problem. As another example, the value of layer 3 from architecture 1 on the MNIST dataset improves by 0.21 as well, and little improvement can be seen from architecture 3: all values are below 0.1 on two datasets either constraint (a) or (b) is imposed on the problem.

5.2.4 DISENTANGLEMENT

Figure 3 illustrates the disentangled degree of every pair of neurons from DNN architecture (1). X-axis and Y-axis reflect the cosine similarity measure and frequency, respectively. It can be concluded that every pair of neurons is disentangled because the maximum cosine similarity is around 0.15. The cosine similarity measure generally follows the Gaussian distribution. Most pairs of cosine similarity measures are around 0. This indicates that they are orthogonal to each other. We also find that a fully-connected neural network seems to be naturally disentangled. This is because this disentanglement can be formed without any imposed constraints (i.e. constraint (a)), as shown in Figure 3.

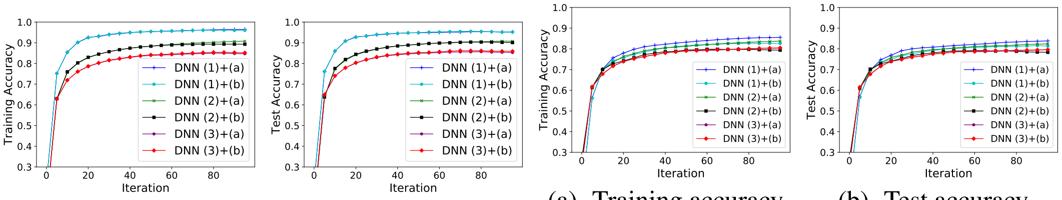


(a).DNN(1)+(a):Layer1. (b).DNN(1)+(a):Layer2. (c).DNN(1)+(b):Layer1. (d).DNN(1)+(b):Layer2.

Figure 3: The histogram of cosine similarity between every pair of neurons on the MNIST dataset: every pair of neurons is disentangled.

5.2.5 Accuracy

Previous interpretability evaluation has shown that the fully connected DNN is interpretable in terms of sparsity, zeroth-order stability, and disentanglement. Specifically, constraint (b) improves sparsity and zeroth-order stability to a great extent. However, the accuracy of a DNN model may decrease according to the no free lunch theorem (Xu et al., 2011). To evaluate the accuracy loss, we test the performance of DNN architectures (1), (2) and (3) with constraints (a) or (b). The performance curves are shown in Figure 4. X-axis and Y-axis reflect the number of iterations and accuracy, respectively. We find that the regularization (i.e. constraint (b)) generally has a tiny impact on accuracy loss. Most of the curves overlap from the same DNN architecture. DNN architecture (1) performs the best while DNN architecture (2) is the secondary. The performance on the MNIST dataset is better than that on the Fashion dataset from the same DNN architecture.



(a). Training accuracy on the MNIST dataset. (b). Test accuracy on the MNIST dataset. (a). Training accuracy on the Fashion-MNIST dataset. (b). Test accuracy on the Fashion-MNIST dataset.

Figure 4: Performance of all DNN formulations on the MNIST and Fashion MNIST datasets: the regularization has a tiny effect on accuracy.

6 Conclusion

With the popularity of deep neural networks (DNNs), their interpretability has attracted the attention of the machine learning community. In this paper, we propose a novel interpretability evaluation framework. Specifically, we propose Four interpretability properties and quantitative measures. Moreover, we theoretically prove that DNNs are zeroth-order stable and first-order stable. Last but not least, We adapt an extended version of deep learning Alternating Direction Method of Multipliers (dlADMM) to solve DNN problems with interpretability constraints, which is guaranteed convergence to a critical point. Experiments on two benchmark datasets validate our proposed interpretability evaluation framework.

References

David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.

Yamini Bansal, Madhu Advani, David D Cox, and Andrew M Saxe. Minnorm training: an algorithm for training over-parameterized deep neural networks. *stat*, 1050:21, 2018.

Kristofer Bouchard, Alejandro Bujan, Farbod Roosta-Khorasani, Shashanka Ubaru, Mr Prabhat, Antoine Snijders, Jian-Hua Mao, Edward Chang, Michael W Mahoney, and Sharmodeep Bhattacharya. Union of intersections (uoi) for interpretable data driven discovery and prediction. In *Advances in Neural Information Processing Systems*, pp. 1078–1086, 2017.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Francois Chollet. *Deep learning with python*. Manning Publications Co., 2017.

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *arXiv preprint arXiv:1808.00033*, 2018.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in neural information processing systems*, pp. 1878–1889, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Imposing hard constraints on deep networks: Promises and limitations. *arXiv preprint arXiv:1706.02025*, 2017.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pp. 7775–7784, 2018.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Alexander Tong, David van Dijk, Jay S Stanley III, Matthew Amodio, Guy Wolf, and Smita Krishnaswamy. Graph spectral regularization for neural network interpretability. *arXiv preprint arXiv:1810.00424*, 2018.

Michael Tsang, Hanpeng Liu, Sanjay Purushotham, Pavankumar Murali, and Yan Liu. Neural interaction transparency (nit): Disentangling learned interactions for improved interpretability. In *Advances in Neural Information Processing Systems*, pp. 5804–5813, 2018.

Joel Vaughan, Agus Sudjianto, Erind Brahimi, Jie Chen, and Vijayan N Nair. Explainable neural networks based on additive index models. *arXiv preprint arXiv:1806.01933*, 2018.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pp. 3835–3844, 2018.

Junxiang Wang, Liang Zhao, and Yanfang Ye. Semi-supervised multi-instance interpretable models for flu shot adverse event detection. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 851–860. IEEE, 2018.

Junxiang Wang, Fuxun Yu, Xiang Chen, and Liang Zhao. Admm for efficient deep learning with global convergence. *arXiv preprint arXiv:1905.13611*, 2019.

Mike Wu, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):187–193, 2011.

Zebin Yang, Aijun Zhang, and Agus Sudjianto. Enhancing explainability of neural networks through architecture constraints. *arXiv preprint arXiv:1901.03838*, 2019.

Raymond Yeh, Jinjun Xiong, Wen-Mei Hwu, Minh Do, and Alexander Schwing. Interpretable and globally optimal prediction for textual grounding using image concepts. In *Advances in Neural Information Processing Systems*, pp. 1912–1922, 2017.

Xin Zhang, Armando Solar-Lezama, and Rishabh Singh. Interpreting neural network judgments via minimal, stable, and symbolic corrections. In *Advances in Neural Information Processing Systems*, pp. 4874–4885, 2018.

Appendix

A Proof of Theorem 1

Proof. Four activation functions, which are sigmoid, tanh, ReLU and leaky ReLU, are Lipschitz continuous (Virmaux & Scaman, 2018). Then there exists $M > 0$ such that

$$\begin{aligned} \|a_l^{''} - a_l^{'}\| &\leq M\|z_l^{''} - z_l^{'}\| \\ &= M\|W_l\|\|a_{l-1}^{''} - a_{l-1}^{'}\| \end{aligned} \tag{3}$$

where $a_l^{'} = f_l(z_l^{'})$ and $a_l^{''} = f_l(z_l^{''})$. We continue this process recursively from $L-1$ to 1 to obtain

$$\|a_{L-1}^{''} - a_{L-1}^{'}\| \leq M^{l-1} \prod_{l=1}^{L-1} \|W_l\|\|a_0^{''} - a_0^{'}\|.$$

We let $H_1 = M^{l-1}\prod_{l=1}^{L-1}\|W_l\|$, then for any $a_0^{'}, a_0^{''}$, we have $\|a_{L-1}^{''} - a_{L-1}^{'}\| \leq H_1\|a_0^{''} - a_0^{'}\|$. □

B Proof of Theorem 2

Proof. Four activation functions, which are sigmoid, tanh, ReLU and leaky ReLU, are categorized as two cases. Case (1) contains like sigmoid and tanh whose second derivatives are bounded. Case (2) includes non-smooth functions like ReLU and leaky ReLU.

Case (1). If f_l is either sigmoid or tanh, then the first derivative and the second derivative of f_l are bounded, this means that $\|\nabla f_l\| \leq S$ and $\|\nabla^2 f_l\| \leq T$ where $S > 0$ and $T > 0$ are constant. Because $a_l = f_l(z_l) = f_l(W_l a_{l-1} + b_l)$, we have

$$\begin{aligned} &\|\partial a_l/\partial a_0\| = \prod_{i=1}^{l} \|W_i^T \nabla f_i(W_i a_{i-1} + b_i)\| \\ &\leq \prod_{i=1}^{l} \|W_i\|\|\nabla f_i(W_i a_{i-1} + b_i)\| \text{(Cauthy-Schwarz inequality)} \\ &\leq S^l \prod_{i=1}^{l} \|W_i\| (\nabla f_i \text{ is bounded}) \end{aligned} \tag{4}$$

Now we need to prove that there exists $Q_l > 0$ such that for any two inputs $a_0^{'}, a_0^{''}$,

$$\|\partial a_l^{''}/\partial a_0^{''} - \partial a_l^{'}/\partial a_0^{'}\| \leq Q_l\|a_0^{''} - a_0^{'}\| (l = 1, \cdots, L-1) \tag{5}$$

Thus the fully-connected neural network is globally stable. To achieve this, we prove by induction as follows:

(a). When $l = 1$, we have the following:

$$\begin{aligned} &\|\partial a_1^{''}/\partial a_0^{''} - \partial a_1^{'}/\partial a_0^{'}\| \\ &= \|W_1\|\|\nabla f_1(W_1 a_0^{''} + b_1) - \nabla f_1(W_1 a_0^{'} + b_1)\| \\ &= \|W_1\|\|\nabla^2 f_1(\gamma_1) W_1(a_0^{''} - a_0^{'})\| \\ &\text{(Mean value theorem, } \gamma_1 \text{ is between } W_1 a_0^{'} + b_1 \text{ and } W_1 a_0^{''} + b_1.) \\ &\leq \|W_1\|^2\|\nabla^2 f_1(r_1)\|\|a_0^{''} - a_0^{'}\| \\ &\leq \|W_1\|^2 T\|a_0^{''} - a_0^{'}\| (\nabla^2 f_1(\bullet) \text{ is bounded}) \end{aligned}$$

Let $Q_1 = \|W_1\|^2 T$, then $\|\partial a_1^{''}/\partial a_0^{''} - \partial a_1^{'}/\partial a_0^{'}\| \leq Q_1\|a_0^{''} - a_0^{'}\|$ holds for $a_0^{'}, a_0^{''}$.

(b). Assume there exists Q_l such that for any two inputs $a_0^{'}, a_0^{''}$, $\|\partial a_l^{''}/\partial a_0^{''} - \partial a_l^{'}/\partial a_0^{'}\| \leq Q_l\|a_0^{''} - a_0^{'}\|$, we prove that there exists Q_{l+1} such that for any two inputs $a_0^{'}, a_0^{''}$, $\|\partial a_{l+1}^{''}/\partial a_0^{''} -$

$\partial a^{'}_{l+1}/\partial a^{'}_0\| \leq Q_{l+1}\|a^{''}_0 - a^{'}_0\|$. To achieve this,

$$
\begin{aligned}
&\|\partial a^{''}_{l+1}/\partial a^{''}_0 - \partial a^{'}_{l+1}/\partial a^{'}_0\| \\
&= \|(\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{''}_l/\partial a^{''}_0) - (\partial a^{'}_{l+1}/\partial a^{'}_l)(\partial a^{'}_l/\partial a^{'}_0)\| \\
&\text{(Chain rule)} \\
&= \|(\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{''}_l/\partial a^{''}_0) - (\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{'}_l/\partial a^{'}_0) \\
&+ (\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{'}_l/\partial a^{'}_0) - (\partial a^{'}_{l+1}/\partial a^{'}_l)(\partial a^{'}_l/\partial a^{'}_0)\| \\
&= \|(\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{''}_l/\partial a^{''}_0) - (\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{'}_l/\partial a^{'}_0)\| \\
&+ \|(\partial a^{''}_{l+1}/\partial a^{''}_l)(\partial a^{'}_l/\partial a^{'}_0) - (\partial a^{'}_{l+1}/\partial a^{'}_l)(\partial a^{'}_l/\partial a^{'}_0)\| \\
&\text{(Triangle inequality)} \\
&\leq \|\partial a^{''}_{l+1}/\partial a^{''}_l\|\|\partial a^{''}_l/\partial a^{''}_0 - \partial a^{'}_l/\partial a^{'}_0\| \\
&+ \|\partial a^{'}_l/\partial a^{'}_0\|\|\partial a^{''}_{l+1}/\partial a^{''}_l - \partial a^{'}_{l+1}/\partial a^{'}_l\| \\
&\text{(Cauchy-Schwarz inequality)} \\
&\leq \|\partial a^{''}_{l+1}/\partial a^{''}_l\|Q_l\|a^{''}_0 - a^{'}_0\| + \|\partial a^{'}_l/\partial a^{'}_0\| \\
&\|W^T_{l+1}(\nabla f_{l+1}(W_{l+1}a^{''}_l + b_{l+1}) - \nabla f_{l+1}(W_{l+1}a^{'}_l + b_{l+1}))\| \\
&= \|\partial a^{''}_{l+1}/\partial a^{''}_l\|Q_l\|a^{''}_0 - a^{'}_0\| + \|\partial a^{'}_l/\partial a^{'}_0\| \\
&\|W^T_{l+1}\nabla^2 f_{l+1}(\gamma_{l+1})(W_{l+1}a^{''}_l - W_{l+1}a^{'}_l)\| \\
&\text{(Mean value theorem, } \gamma_{l+1} \text{ is between} \\
&W_{l+1}a^{'}_l + b_{l+1} \text{ and } W_{l+1}a^{''}_l + b_{l+1}) \\
&\leq \|\partial a^{''}_{l+1}/\partial a^{''}_l\|Q_l\|a^{''}_0 - a^{'}_0\| + \|\partial a^{'}_l/\partial a^{'}_0\| \\
&\|W_{l+1}\|^2\|\nabla^2 f_{l+1}(\gamma_{l+1})\|\|a^{''}_l - a^{'}_l\| \\
&\text{(Cauthy-Schwarz inequality)} \\
&\leq \|\partial a^{''}_{l+1}/\partial a^{''}_l\|Q_l\|a^{''}_0 - a^{'}_0\| + \|\partial a^{'}_l/\partial a^{'}_0\| \\
&\|W_{l+1}\|^2\|\nabla^2 f_{l+1}(\gamma_{l+1})\|M^l\prod_{i=0}^{l-1}\|W_i\|\|a^{''}_0 - a^{'}_0\| \\
&\text{(Inequality equation 3)} \\
&\leq \|W_l\|SQ_l\|a^{''}_0 - a^{'}_0\| + S^l\prod_{i=1}^{l}\|W_i\| \\
&\|W_{l+1}\|^2 TM^l\prod_{i=0}^{l-1}\|W_i\|\|a^{''}_0 - a^{'}_0\| \\
&\text{(Inequality equation 4 and } \nabla^2 f_{l+1} \text{ is bounded)}
\end{aligned}
$$

Let $Q_{l+1} = \|W_l\|SQ_l + S^l TM^l\|W_{l+1}\|^2\prod_{i=1}^{l}\|W_i\|\prod_{i=0}^{l-1}\|W_i\|$, for any two inputs $a^{'}_0, a^{''}_0$, $\|\partial a^{''}_{l+1}/\partial a^{''}_0 - \partial a^{'}_{l+1}/\partial a^{'}_0\| \leq Q_{l+1}\|a^{''}_0 - a^{'}_0\|$ holds.
Based on steps (a) and (b), we prove that Inequality equation 5 holds. Therefore, the fully-connected neural network is globally stable.
Case (2). In this case, we consider nonsmooth activation functions like ReLU and leaky ReLU. We show that the fully-connected neural network is locally stable almost surely when f_l is ReLU, the same routine is applied when f_l is leaky ReLU.
The derivative of ReLU (except 0) is defined as follows:

$$
\partial a_l/\partial z_l = \begin{cases} 0 & z_l < 0 \\ 1 & z_l > 0 \end{cases}
$$

Noticeably, the ReLU is non-differentiable at 0. Now we find out all inputs that make $z_l = 0$. To achieve this, we define $S_l = \{a_0 | \ z_l = 0\}(l = 1, \cdots, L)$ as non-differentiable inputs for the l-th layer. Then the non-differentiable input set is defined as $C = \bigcup_{l=1}^{L-1} C_l = \{a_0 | \exists 1 \leq l \leq L-1, s.t. \ z_l = 0\}$, and $\overline{C} = \mathbb{R}^{n_0} - C$ is the differentiable input set, where n_0 is the number of

input features. We firstly show that the fully-connected neural network is locally stable on $\overline{C}$, then we show that $P(\overline{C}) = 1$ to prove the fully-connected neural network is locally stable almost surely. For any $a_0 \in \overline{C}$, we easily find a neighborhood $N(a_0)$ such that any $a_0^{'}, a_0^{''} \in N(a_0)$, $z_l^{'} \circ z_l^{''} > 0 (l = 1, \cdots, L-1)$ where $\circ$ is the Hadamard product. This implies that $\partial a_l^{'}/\partial z_l^{'} = \partial a_l^{''}/\partial z_l^{''}$. Therefore, for any $M_2 > 0$, $\|\partial a_l^{''}/\partial z_l^{''} - \partial a_l^{'}/\partial z_l^{'}\| = 0 \leq M_2 \|a_l^{''} - a_l^{'}\|$. In other words, the fully-connected neural network is locally stable on $\overline{C}$.
Next, we show $P(\overline{C}) = 1$. To achieve this, we prove that $P(C_l) = 0$. For W_l, we discuss two situations:
Situation a. $W_l = 0$. In this case, $z_l = b_l \neq 0$, so $C_l = \emptyset$. $P(C_l) = 0$.
Situation b. $W_l \neq 0$. In this case, $z_l \in \mathbb{R}^{n_l}$, then $P(C_l) = P(z_l = 0) = 0$ (i.e. the probability of taking 0 in a real space is 0).
So $P(C) \leq \sum_{l=1}^{L-1} P(C_l) = 0$ and $P(\overline{C}) = 1$. This means that the fully-connected neural network is locally first-order stable almost surely. □

C CONCEPT INDEPENDENCE AMONG PROPERTIES

We illustrate the concept of independence among all Four properties of interpretability in this section: sparsity, first-order stability, disentanglement, faithfulness, and explicitness. To prove the independence between any two properties of interpretability P_1 and P_2, we show that there exist two cases for P_1 does not imply P_2 and P_2 does not imply P_1, respectively.
(1). sparsity and first-order stability.
The following example shows that sparsity does not imply first-order stability: there is a two-layer neural network where all weights are zeros and activation functions are ReLU. Then it satisfies sparsity, but is not first-order locally stable everywhere because of the nonsmooth ReLU.
The following example shows that first-order stability does not imply sparsity: there is a two-layer network where all weights are 9999 and activation functions are sigmoid. Then it satisfies first-order local stability, but is not sparse.
(2). sparsity and disentanglement.
The following example shows that sparsity does not imply disentanglement: there is a one-layer neural network whose weights are almost zeros except $W_{1,1,1} = 0.001$ and $W_{1,2,1} = 0.001$. Then it satisfies sparsity, but the first neuron and the second neuron are not disentangled because $W_{1,1}$ and $W_{1,2}$ are identical.
The following example shows that disentanglement does not imply sparsity: there is a one-layer neural network where W_1 is orthogonal so that every pair of neurons is disentangled, but all elements of W_1 are nonzero and hence it does not satisfy sparsity.
(3). sparsity and faithfulness.
The following example shows that sparsity does not imply faithfulness: there is a neural network whose prediction accuracy is low and almost all weights are 0. However, its faithfulness is low due to low accuracy. In other words, its weights do not truly reflect the importance of features.
The following example shows that faithfulness does not imply sparsity: there is a neural network whose prediction accuracy is high and every weight is nonzero. Obviously, its faithfulness is high but is not sparse.
(4). first-order stability and disentanglement.
The following example shows that first-order stability does not imply disentanglement: there is a neural network whose activation functions are sigmoid, and every row of the weight matrices is exactly the same. Then it is globally first-order stable, but every pair of neurons is not disentangled.
The following example shows that disentanglement does not imply stability: there is a neural network whose activation functions are ReLU, and all weight matrices are orthogonal. Then every pair of neurons is disentangled, but the neural network is not first-order stable everywhere because of the nonsmooth ReLU.
(5). first-order stability and faithfulness.
The following example shows that first-order stability does not imply faithfulness: there is a neural network whose activation functions are sigmoid, and its prediction accuracy is low. Then it is globally first-order stable, but is not faithful.
The following example shows that faithfulness does not imply first-order stability: there is a neural network whose activation functions are ReLU, and its prediction accuracy is high. Then it is faithful but is not first-order stable everywhere because of the nonsmooth ReLU.

(6). disentanglement and faithfulness.
The following example shows that disentanglement does not imply faithfulness: there is a neural network whose weight matrices are orthogonal and thus every pair of neurons is disentangled, but its prediction accuracy is low and hence is not faithful.
The following example shows that faithfulness does not imply disentanglement: there is a neural network whose prediction accuracy is high and hence is faithful, but every row of weight matrices is identical so that every pair of neurons is not disentangled.

D Relations to Previous Work

In this part, our proposed interpretability evaluation framework is compared with several previous state-of-the-art intrinsic methods. We show that they are special cases of our framework.
interpretation-constrained model (Ross et al., 2017). Ross et al. proposed a general model to penalize the input gradient and imposed ℓ_2 regularization on weight, which equivalently requires DNN to be sparse and zeroth-order stable.
Self-explaining Neural Network (Melis & Jaakkola, 2018). Melis and Jaakkola proposed three desiderata in their self-explaining neural network models: explicitness, faithfulness, and stability, two of which are shown as properties of interpretability in our framework.
Explainable neural networks with architecture constraints (Yang et al., 2019). Yang et al. imposed three constraints on the explainable neural network: ℓ_1 penalty, orthogonality constraint, and smooth constraint. These are equivalently required neural network to be sparse, disentangled and stable.

E Algorithms 1 and 2 of the extended dlADMM algorithm

All notations in Algorithms 1 and 2 follow the same dlADMM algorithm proposed by Wang et al. (Wang et al., 2019).

Algorithm 1 The Backtracking Algorithm to update $\overline{W}_l^{k+1}$

```
Require: $\overline{\mathbf{W}}_{l+1}^{k+1}$,$\overline{\mathbf{b}}_l^{k+1}$, $\overline{\mathbf{z}}_l^{k+1}$,$\overline{\mathbf{a}}_l^{k+1}$,$u^k$, $\rho$, some constant $\overline{\gamma} > 1$.
Ensure: $\overline{\theta}_l^{k+1}$,$\overline{W}_l^{k+1}$.
Pick up $\overline{\alpha}$ and $\overline{\zeta} = W_l^k - \nabla_{W_l^k} \phi / \overline{\alpha}$.
while $\phi(\{W_i^k\}_{i=1}^{l-1}, \overline{\zeta}, \{\overline{W}_i^{k+1}\}_{i=l+1}^{L}, \overline{\mathbf{b}}_l^{k+1}, \overline{\mathbf{z}}_l^{k+1}, \overline{\mathbf{a}}_l^{k+1}, u^k) > \overline{P}_l(\overline{\zeta}; \overline{\alpha})$ or $G_l(\overline{\zeta}) \neq 0$ do
    $\overline{\alpha} \leftarrow \overline{\alpha}\, \overline{\gamma}$.
    Solve $\overline{\zeta}$ by Equation (9) in (Wang et al., 2019).
end while
Output $\overline{\theta}_l^{k+1} \leftarrow \overline{\alpha}$.
Output $\overline{W}_l^{k+1} \leftarrow \overline{\zeta}$.
```

Algorithm 2 The Backtracking Algorithm to update W_l^{k+1}

```
Require: $\mathbf{W}_{l-1}^{k+1}$,$\mathbf{b}_{l-1}^{k+1}$, $\mathbf{z}_{l-1}^{k+1}$,$\mathbf{a}_{l-1}^{k+1}$,$u^k$, $\rho$, some constant $\gamma > 1$.
Ensure: $\theta_l^{k+1}$,$W_l^{k+1}$.
Pick up $\alpha$ and $\zeta = W_l^k - \nabla_{\overline{W}_l^{k+1}} \phi / \alpha$.
while $\phi(\{W_i^{k+1}\}_{i=1}^{l-1}, \zeta, \{\overline{W}_i^{k+1}\}_{i=l+1}^{L}, \mathbf{b}_l^{k+1}, \mathbf{z}_l^{k+1}, \mathbf{a}_l^{k+1}, u^k) > P_l(\zeta; \alpha)$ or $G_l(\zeta) \neq 0$ do
    $\alpha \leftarrow \alpha\, \gamma$.
    Solve $\zeta$ by Equation (11) in (Wang et al., 2019).
end while
Output $\theta_l^{k+1} \leftarrow \alpha$.
Output $W_l^{k+1} \leftarrow \zeta$.
```