

# How to Explain Neural Networks: A perspective of data space division

Hangcheng Dong<sup>1</sup>, Bingguo Liu<sup>1</sup>, Fengdong Chen<sup>1</sup>, Dong Ye<sup>1</sup> and Guodong Liu<sup>1\*</sup>

**Abstract**—Interpretability of intelligent algorithms represented by deep learning has been yet an open problem. We discuss the shortcomings of the existing explainable method based on the two attributes of explanation, which are called completeness and explicitness. Furthermore, we point out that a model that completely relies on feed-forward mapping is extremely easy to cause inexplicability because it is hard to quantify the relationship between this mapping and the final model. Based on the perspective of the data space division, the principle of complete local interpretable model-agnostic explanations (CLIMEP) is proposed in this paper. To study the classification problems, we further discussed the equivalence of the CLIMEP and the decision boundary. As a matter of fact, it is also difficult to implementation of CLIMEP. To tackle the challenge, motivated by the fact that a fully-connected neural network (FCNN) with piece-wise linear activation functions (PWLs) can partition the input space into several linear regions, we extend this result to arbitrary FCNNs by the strategy of linearizing the activation functions. Applying this technique to solving classification problems, it is the first time that the complete decision boundary of FCNNs has been able to be obtained. Finally, we propose the DecisionNet (DNet), which divides the input space by the hyper-planes of the decision boundary. Hence, each linear interval of the DNet merely contains samples of the same label. Experiments show that the surprising model compression efficiency of the DNet with arbitrary controlled precision.

## I. INTRODUCTION

Deep learning has achieved impressive success in various important fields, just like computer vision [1], natural language processing [2], and graphs [3]. Despite tremendous achieved progress, deep neural networks are often been criticized for that it is used as a black box in practice. In recent years, interpretability and its guiding significance for the design of deep learning models, known as explainable AI (XAI) [4], [5], [6], [7], [8], [9], has become increasingly important, which is crucial to the ability of neural network-based algorithms to be widely applied to industrial practice. The theory of deep learning contains three main areas of research, approximation power, the dynamics of optimization, and good out-of-sample performance [10]. While the generalized interpretability problem encompasses more than just theoretical issues. Although the interpretability of deep learning has been still under debate, some studies summarize that interpretability consists of two aspects, transparency and post-hoc interpretability [11]. Transparency concerns how to explain the way model works, and post-hoc interpretability attempts to derive knowledge which can be understood by

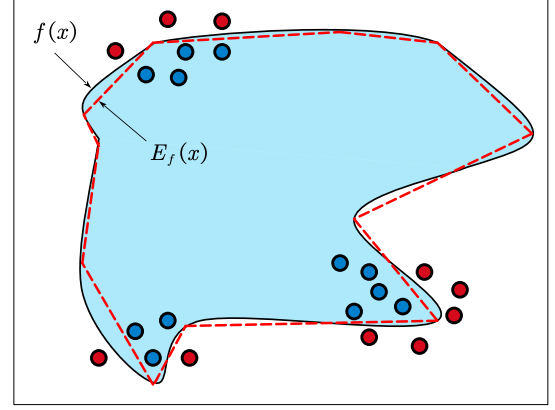


Fig. 1. The neural network is essentially an extremely complex mapping, denoted as  $f : \mathbb{R}^d \rightarrow \mathbb{R}^c$ . Disregarding various assumptions about intelligence, the problem is transformed into how to explain a complex function. Similar to the Fourier transform, we can decompose a complex function, but with the difference that we decompose the domain of the function into different parts and then find a simple function  $E_{f_i}(x)$  on each part so that it remains equal to or approximates the original function on each part. The set of these simple functions is defined as the explanation of the original complex function, denoted as  $E(f) \triangleq E_f(x) = \{E_{f_i}(x)\}_{i=1}^n$ , where  $n$  is the number of local regions.

humans from the model. Although the number of studies on interpretability is large and continuously growing, there is still no substantial progress. In this paper, we discuss the definition of the interpretability of neural networks and point out the phenomenon that most machine learning models are constructed as feed-forward mappings, which leads to difficulties in explanation. Naturally, we propose that models should be interpreted in terms of a division of the data space, named the principle of complete local interpretable model-agnostic explanations (CLIMEP). Our goal is to decompose a complex function into several simple functions, which is similar to the idea of LIME [12], but we emphasize the completeness of the decomposition, which is illustrated in Figure 1.

Some well-known approaches in recent years are concerned with hierarchical semantic features in deep learning models, such as saliency maps [13]. There are also approaches that are expected to simplify models and replace complex deep learning algorithms with more concise and easy-to-understand models, such as knowledge distillation [14], LIME [12]. However, the current studies all have a fundamental flaw in being unable to clearly define what is interpretability of deep neural networks and are satisfied with finding only a certain understandable indication, such as accuracy and semantic images. In this study, we propose that

<sup>1</sup>Hangcheng Dong (hunsen\_d@hit.edu.cn), Bingguo Liu, Fengdong Chen, Dong Ye and Guodong Liu (lgd@hit.edu.cn) are with School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin, 150001, China

completeness is the fundamental property of an explanation. In other words, the explanation should be consistent with the model being explained. Therefore, the fact that partial structures in the model have a certain semantic character is not the point. Furthermore, it is not feasible to use accuracy to measure the consistency of explanation due to the limited sampling. In addition, we consider explicitness to be another indispensable property of the interpretability. Explicitness implies universality and simplicity of explanation, which avoids the problem of "explaining your explanations".

Based on CLIMEP, we describe a way to explain fully connected neural networks (FCNN). Once an FCNN with ReLU activation function has been well-trained, the output of the FCNN is a linear function when the activation state maintains the same [15], [16], [17], [18]. Thus the ReLU-FCNN can be naturally decomposed into several linear intervals, and on each interval, the decomposed function is exactly the same as the original function. Generally, for any FCNN, by approximating the activation function with a piece-wise linear function, we can transform it into the piece-wise linear case. Due to the universal approximation property of piece-wise linear functions, we can obtain the explanation of FCNNs. It is worth pointing out that piece-wise approximate neural networks are also proposed in the literature [19], but the difference in this paper is the use of a different piece-wise linear approximation method. Moreover, we point out interpretability of classification problems is equivalent to finding decision bounds of the trained model and study the scheme for looking for decision boundaries, while literature [19] evaluates the model complexity. In addition, we propose decision neural networks to compress the primitive model.

In summary, the main contributions of our paper are:

1. We propose the principle of explanation, named CLIMEP, from the perspective of data space division, and give the definition of interpretation based on the completeness and explicitness of explanation.
2. We show that in classification problems, CLIMEP necessarily leads to the corresponding decision boundaries, and propose a method for seeking decision boundaries by approximating arbitrary activation functions through piece-wise linear functions.
3. We develop a novel model compression method called DecisionNet(DNet) for FCNNs, which can compress the FCNNs with an arbitrary controlled precision.

## II. RELATED WORK

To highlight the contributions in this paper, we would like to summarize the related results.

1)A great deal of work [4] on the interpretability of deep learning has focused on feature visualization [13], which is straightforward and explicable. One of the most intuitive ideas is to observe activation maps [20]. However, this method only provides little information that lacks semantics. A further consideration is to explore the relationship between activation maps and its input image space, which are

well known as activation maximization [21] and deconvolution [22]. Both of them show that in a well-trained network, semantic information that neural units are interesting in is enriched as the depth of layers increases. To quantify the importance of features, series of gradient-based methods [23], [24], [25], [26], [27], [28], [29], [30] are proposed, and perturbation-based [31], [32], [33] methods are also received attention. These technologies reveal that semantic information can indeed be learned by neural network classifiers, although they are vulnerable and sensitive to slight changes in the input space [34]. There is also a notable category of visualization methods, namely CAM (Class Activation mapping) [35] and its variants [36], [37], [38]. CAM-like approaches use linear combinations of activation maps from convolutional layers to produce interpretive heatmaps, and the weights are generated in different ways, mirroring the different measures of importance among different activation maps in different ways. However, the above methods can not satisfy both completeness and weak dependence on input [39], based on that relu networks could be completely expressed by the sum of input-gradients and bias-gradients, full-gradient representation is proposed. It is regretful that the full-gradient map is not satisfied either.

2)Another methodology is the proxy method, which trained smaller, simpler, easier to understand models instead of deep neural networks. Hinton et al. showed a technology called knowledge distillation that using the output of a large neural network, known as soft targets, to train another model (such as a small neural network or decision trees) can improve the generalization performance [14], [40]. knowledge distillation uses model "knowledge" computed based on whole samples. In addition to the global approach, M.T.Ribeiro et al. propose a local framework named Local Interpretable Model-agnostic Explanation (LIME) [12]. LIME generates several new samples in the neighborhood of a specific sample and computes corresponding outputs of the model. Then a linear model will be trained to fit the samples with coefficients that reflect the importance of the features. It is, however, incomplete to rely on samples to transform model knowledge, for that samples do not fully characterize the decisions of a model, as shown in Figure 2.

3)Studies directly related to ours include [41], [42], [39], which shows the neural networks with arbitrary piece-wise linear activation functions can split the input-space into several linear regions. Xia Hu [19] et al. proposed linear approximation neural network and suggested that using piece-wise linear functions instead of curve activation functions. The upper bound on the number of the linear regions has been analyzed in [16], [17], [42]. Literature [18] further verified the effect of BN and dropout techniques on the linear interval of ReLU networks. However, there is no common definition and reliable way about the interpretability of deep learning. Inspired by the studies mentioned above, this paper attempts to point out the nature of why interpretability of neural networks is difficult and to propose a principle of the explanation named CLIMEP.

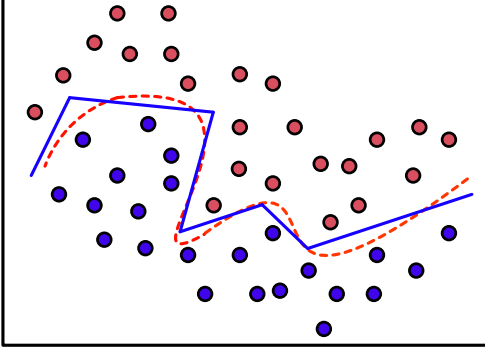


Fig. 2. The metrics calculated based on the sample set, such as the accuracy, cannot completely reflect the function expressed by the model. For example, in the binary classification problem in this figure, the blue solid line and the red dashed line are consistent in terms of the accuracy, but the functions expressed are completely different.

### III. PROPERTIES OF INTERPRETABILITY

In this section, firstly, we introduce the definition of interpretability, and then discuss two of its important properties, namely completeness and explicitness.

#### A. definition of interpretability

The interpretability of deep learning has not made substantial progress in recent years. One fundamental reason is that the definition of interpretability is intractable. In this section, we try to discuss the following questions:

1. what is the interpretability of a neural network?
2. how to explain a neural network?

For the first question, our point is that interpretability of neural networks is the understanding of the function represented by models, which consists of two aspects, one is the constructing process of the function, and the other is the result of the construction. Modern neural networks are generally designed layer-by-layer. Hence, a fully-connected neural network (FCNN) can be represented by a composite function  $f: \mathbb{R}^d \rightarrow \mathbb{R}^c$ , where  $d$  denotes the input dimension, as well as,  $c$  denotes the output dimension. For an FCNN with  $L$  layers, let the output of the  $i$ -th layer be  $h_i(\cdot)$ , then  $\forall x \in \chi \subseteq \mathbb{R}^d$ , we have:

$$f(x) = h_L(h_{L-1}(\dots h_1(x))) \quad (1)$$

It is commonly believed that the interpretability of deep learning requires formalizing the association between the construction process of the network and the final result. Whereas we argue that it is formidable and even impossible to explain it in terms of function mapping. A classic example is the SVM [43], which achieves good generalization performance by reasoning from the perspective of decision boundaries. However, when the model is introduced with a kernel trick [43], it is still considered as a black-box model.

For the second question, our solution is to explain the result of the complex mapping with simple ones, based on the consideration discussed above. For the sake of clarity, we need to find out what kind of functions are explainable

and what kind of functions are not. For example, denoted by  $g(x) = wx + b$  a linear model, we can easily understand and further ascribe meaning to the parameters. Despite that for the neural network, whose function is compounded from several simple functions, as shown in Eq. (1), we would consider the function is difficult to understand, even if we can explicitly know the expression of it. Therefore, we believe that the essential reason why interpretability is so difficult is due to the complexity of the function. Since the complexity of functions is hard to define, we need to draw support from human feelings. In this article, we do not strictly define explainable functions as basic elementary functions.

Mathematically, we define the interpretability as follows:

**Definition 1:** For any given precision  $\delta > 0$ , if there exist a function or a set of functions  $E(f) \triangleq E_f(\cdot)$  with an explicit expression, we say  $E_f(\cdot)$  is an interpretation of a function  $f(\cdot)$ , implies that it satisfies the following conditions:

$$\forall x \in \chi \subseteq \mathbb{R}^d, |f(x) - E_f(x)| < \delta \quad (2)$$

where  $\chi$  is the input space,  $x$  and  $d$ , respectively, are an instance within the input space and the number of input features.

**Definition 2:** (Human-friendly interpretation) In this article, we do not strictly set that, if a interpretation  $E_f(\cdot)$  is Human-friendly, implies that it is composed of the Basic Elementary Functions.

In particular, a function is its own interpretation, but it is not human-friendly.

#### B. the principle of explanations

To highlight the advantages of our approach, firstly, we discuss the shortcomings of the current method of interpretability and then introduce our proposed the principle of complete local interpretable model-agnostic explanations (CLIMEP).

According to Definition 1, we find that most of the existing methods cannot satisfy it, and the reasons can be generalized into two aspects, namely completeness and explicitness.

Completeness is a natural property of an Explanation, which suggests it should satisfy Eq.(2). However, many methods do not fulfill this characterization, including CAM-based approaches [35], activation maximization [21], deconvolution [22], vallina gradient maps [23] and so on. Knowledge distillation [40] and LIME [12] try to dig alternative models for the trained complex neural networks, nevertheless fail to meet completeness, as they are trained on only a limited number of samples in the input space.

Explicitness means that an Explanation should have an explicit expression. Attribution methods assign a contribution score to each input feature as a measure of its importance, and the contribution score is often gradient-based. Several attribution methods, such as integrated gradients [28], deep Taylor decomposition [27], DeepLIFT [30] and Shapley value [44] are complete but not explicit because they only provide an explanation for a single input sample at each time. The lack of explicitness makes it difficult to develop a comprehensive and systematic understanding of the model.

Most machine learning models are built from the perspective of function mapping, which makes the model easy to get complicated and difficult to interpret. We propose to understand the model from the perspective of the division of the data space, which satisfies the Definition 1, as described as follows:

**Definition 3:** (CLIMEP):  $\forall x_i \in \chi$ , if the explanation  $E(f)$  can always provides an explicit explanation  $E_f^{(i)}(\cdot)$  such that it is complete in a certain neighborhood  $U(x_i)$ , formally, for any given precision  $\delta > 0$ , satisfying:

$$\forall x \in U(x_i), |f(x) - E_f^{(i)}(x)| < \delta \quad (3)$$

then we say  $E(f)$  is the complete local interpretable model-agnostic explanation (CLIME) of  $f(\cdot)$ .

**Proposition 1:** CLIME is compatible, which means:

$$\forall x \in U_i \cap U_j, |E_f^{(i)}(x) - E_f^{(j)}(x)| \leq |E_f^{(i)}(x) - f(x)| + |E_f^{(j)}(x) - f(x)| < 2\delta \quad (4)$$

**Discuss:** Through CLIME, we can significantly reduce the complexity of the functions. If the complex function corresponding to the deep network is transformed into a simple format in the subdivided local space, especially as the human-friendly function in Definition 2, then the interpretability of the model can be greatly increased.

#### IV. DECISION BOUNDARY FOR CLASSIFICATION MODELS

In this section, we introduce the relationship between CLIME and the decision boundary corresponding to the classification models.

Firstly, we define the decision boundary of the classification model, which enables us to analyze the classification problem from the perspective of the data space division.

**Definition 4:** (decision boundary) Given a trained classifier  $f : \mathbb{R}^d \rightarrow \mathbb{R}^c$ , denote by  $C = \{C_i | i = 1, 2, \dots, n\}$  the class labels. The decision boundary between class  $C_i, C_j (i \neq j)$  corresponding to the model  $f$  is defined as follows:

$$DB_{ij} = \{x | \forall \delta > 0, \exists x_i, x_j \in U(x, \delta), f(x_i) \neq f(x_j)\} \quad (5)$$

where  $U(x, \delta)$  means a open region satisfying  $U(x, \delta) = \{z | \text{dist}(x, z) < \delta\}$ ,  $\text{dist}(\cdot, \cdot)$  is the distance metric.

Secondly, we define interpretability under the sense of classification, which is the bridge associated with the decision boundary.

**Definition 5:** (interpretability in classification problem) Given a trained classifier  $F(\cdot)$ , if there exist a function or a set of functions  $E(F) \triangleq E_F(\cdot)$  with an explicit expressions, we say  $E_f(\cdot)$  is a complete interpretation of  $F(\cdot)$  in the sense of classification, implying that it satisfies the following conditions:

$$\forall x \in \chi \subseteq \mathbb{R}^d, F(x) = E_F(x) \quad (6)$$

where  $U(x, \delta)$  means a open region satisfying  $U(x, \delta) = \{z | \text{dist}(x, z) < \delta\}$ ,  $\text{dist}(\cdot, \cdot)$  is the distance metric.

**Proposition 2:** For a classification problem, if an explanation fulfills Definition 1, then it must fulfills Definition 5.

**Proof:** Let the classifier be  $F(x) = \text{argmax}(f(\vec{x}))$ , where  $f(\cdot)$  is the logit output. Let the  $i$ -th output be the maximum, namely  $f_i - f_j > 0, (j = 1, 2, \dots, n, j \neq i)$ . Denote by  $E_F(x) = \text{argmax}(E_f(x))$  the explanation, we have:

$$E_{f_i} - E_{f_j} = (E_{f_i} - f_i) + (f_i - f_j) + (f_j - E_{f_j}) \quad (7)$$

Notice the completeness by Definition 1 of  $f(\cdot)$ , we have:

$$\begin{aligned} \lim_{\delta \rightarrow 0} (E_{f_i} - E_{f_j}) &= \\ \lim_{\delta \rightarrow 0} [(E_{f_i} - f_i) + (f_i - f_j) + (f_j - E_{f_j})] &= f_i - f_j > 0 \end{aligned} \quad (8)$$

Then, due to the sign-preserving theorem of limit, we have  $E_{f_i} - E_{f_j} > 0, (j = 1, 2, \dots, n, j \neq i)$ , namely,  $E_F(x) = \text{argmax}(E_f(x)) = F(x)$ , which concludes the proof.

Finally, we show that in the scene of classification, CLIME is equivalent to the related decision boundary, which shows that to obtain the decision boundary of the classification model is essential for understanding the model from the perspective of data space division.

**Proof:** In a classification problem, the necessity is that, for a trained classifier  $F : \mathbb{R}^d \rightarrow \mathbb{R}^c$ , given the decision boundary, the input space  $\chi \subseteq \mathbb{R}^d$  can be divided into several maximum decision regions  $R_i = \{x | F(x) = i\}$ ,  $i = 1, 2, \dots, n$ , then CLIME  $E_F(\cdot)$  can be described as  $\{R_i\}_1^n$ . The sufficiency is that,  $\forall x \in \chi$ , by the Remark 2, CLIME  $E_F(\cdot)$  is completeness in the sense of classification, that is  $F(x) = E_F(x)$ , then the decision boundary is  $\{DB_{ij} | i, j = 1, 2, \dots, n, i \neq j\}$ , where  $DB_{ij} = \bigcup_{x \in R_i \cap R_j} (x)$ , which concludes the proof.

#### V. INTERPRET AND COMPRESS FCNNs

In this section, we show how to explain a fully-connected neural network (FCNN) from the perspective of the division on data space. First, we introduce the FCNN with piece-wise linear activation function (PLNN) and its piece-wise linear property. Second, we propose CLIME for FCNN with a non-linear activation function by linearizing it (LFNN). Next, we show how to calculate the decision boundary corresponding to a LFNN. Finally, we discuss how to compress LFNNs with no loss of precision by the proposed the DecisionNet (DNet).

##### A. linear regions of PLNNs

An FCNN with piece-wise linear activation functions (PLNN) can be partitioned into several linear regions. Once a PLNN has been well-trained, the sample will be activated by a certain linear part of the piece-wise linear function when it is calculated forward through the PLNN. Samples with the same activation state can be expressed by a same linear model, activation state means that on each neuron, the sample is activated by the same linear region of the activation function, as illustrated in Figure 3.

According to the linear regions of PLNN, we can naturally obtain the CLIME. On each linear interval, CLIME can be expressed as the following formula:

$$\forall x \in \Omega_i, E_{f_i}(x) = (\partial f / \partial x)^T x + \Sigma_i (\partial f / \partial b_i)^T \odot b_i \quad (9)$$

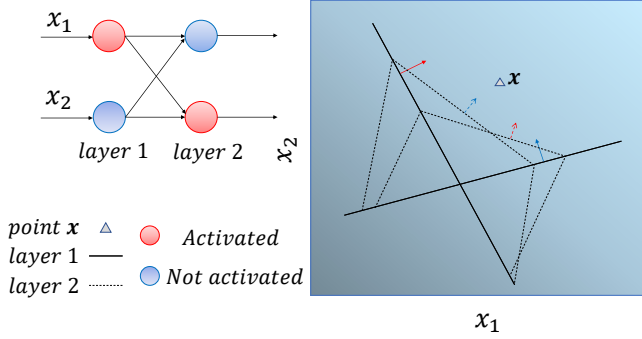


Fig. 3. The activation state of the neuron corresponds to a hyper-plane in the data space. According to different activation states, the position of the sample relative to the hyper-plane can be determined. The first layer of neurons will divide the entire space, the second layer will divide all the subspaces of the first layer except the dead zone of the previous layer (all neurons are not activated), and so on.

### B. complete explanations of FCNNs

Motivated by the piece-wise linear characteristics of PLNN, for FCNNs that use other non-linear activation functions, we can transform them into PLNNs by linearizing the activation functions. In fact, there are two reasons to support the operation, one is that the non-linear factors in FCNNs are totally derived from the activation functions, and the other is that most of the activation functions can be completely approximated by piece-wise linear functions with a limited number of pieces. "Completely" means that as the number of pieces of the functions increases, the approximation error will approach zero.

The activation functions used in deep networks are often with bounded derivative function [45], [46], which means they are Lipschitz continuity. More importantly, these activation functions often have asymptotic lines. According to these characteristics, we can approximate the activation functions by piece-wise linear functions. In order to meet the requirements of completeness, we require the approximation scheme to fulfill the condition that as the number  $n$  of the pieces of the piece-wise linear function  $p_n(\cdot)$  increasing, the error  $\delta$  between the target activation function  $\sigma(\cdot)$  and  $p_n(\cdot)$  gradually decreases and tends to zero, formalized as follows:

$$\lim_{n \rightarrow \infty} (\sigma(x) - p_n(x)) = 0 \quad (10)$$

Algorithm 1 summarizes the construction process of  $p_n(x)$ . The linear approximation of some classical activation functions is further described in Figure 4. By the linear approximation, the CLIME of FCNNs can be depicted as the same as preceding section. The remaining question is that, whether this linear approximation explanation is complete, as described in Definition 3. We will illustrate the completeness of the method in proposition 3 as follows.

**Proposition 3:** (completeness analysis) Consider an FCNN  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , for the sake of simplicity, suppose that there are only two layers in the network, noted as  $f(x) = f_1(f_2(x))$ . The activation function and its linear

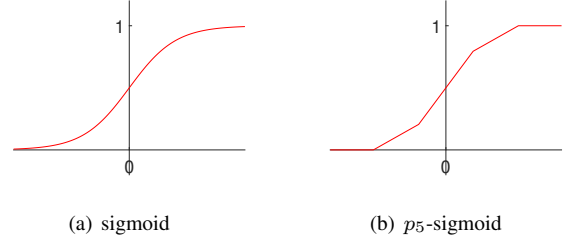


Fig. 4. The  $n$ -piece linear approximation function of the sigmoid function calculated by Algorithm 1.

### Algorithm 1 Linear approximate activation function

**Input:**  $\sigma(x)$  := the target activation function,  $n$  := the number of pieces, ( $n \geq n_0$ ),  $n_0$  := the number of asymptotic lines of  $\sigma(x)$ .

**Output:**  $p_n(x)$  := a piece-wise linear function with  $n$  pieces, the maximum error  $\epsilon := |p_n(x) - \sigma(x)|$

- 1: Initialization: calculate the hard- $\sigma(x)$  which is consist of the asymptotic lines of  $\sigma(x)$ ,  $p_0(x) = \text{hard} - \sigma(x)$
- 2: **for**  $i$  in range( $1, n - n_0 + 1$ ) **do**
- 3:   compute the point by  $x \leftarrow \text{argmax}(\sigma(x) - p_n(x))$
- 4:   **if**  $\#x > 1$  **then**
- 5:      $x \leftarrow \min(x)$
- 6:   **end if**
- 7:   compute tangent line  $l_i$  at point  $x$
- 8:   compute the new function by  $p_n \leftarrow (p_n, l_i)$
- 9: **end for**
- 10: **return**  $p_n(x)$ ,  $\epsilon = \max(p_n(x) - \sigma(x))$ .

approximation are respectively noted as  $\sigma(\cdot)$ ,  $p_n(\cdot)$ , where  $n$  is the number of pieces. Denoted by  $g(x) = g_1(g_2(x))$  the linearized FCNN, the error analysis is as follows:

$$\begin{aligned} |f_1(f_2(x)) - g_1(g_2(x))| &= \\ |f_1(f_2(x)) - g_1(f_2(x)) + g_1(f_2(x)) - g_1(g_2(x))| & \\ \leq |f_1(f_2(x)) - g_1(f_2(x))| + |g_1(f_2(x)) - g_1(g_2(x))| & \end{aligned} \quad (11)$$

By Eq.(10), we have:

$$\begin{aligned} |g_1(f_2(x)) - g_1(g_2(x))| &\leq L \cdot |w| \cdot |f_2(x) - g_2(x)| \\ &= C(L, w) \cdot \delta(n) \end{aligned} \quad (12)$$

where  $L$  and  $w$  are respectively the lipschitz constant of  $p_n(\cdot)$  and the parameter of the affine transformation in layer  $f_1$ . Thus, we have:

$$|f_1(f_2(x)) - g_1(g_2(x))| \leq (C(L, w) + 1) \cdot \delta(n) \quad (13)$$

Which concludes the proof.

### C. Decision boundary of FCNNs

As discussed in the preceding section, the decision boundary is an important representation for classification models but difficult to obtain. Literature [47], [48], [49], [50], [51] has discussed the methods for calculating the



approximate decision boundary. In [41], for the first time to our best knowledge, a method for computing the consistent decision boundary has been given. Motivated by this, we proposed to extend the algorithm in [41] to arbitrary FCNNs by the technology discussed in previous section.

Algorithm 2 outlines the method of computing the approximate decision boundary related to FCNNs. As discussed in preceding section, the LFNN meets the Definition 1. Consequently, LFNN meets Definition 5 which means the decision boundary representation is also complete. Thus, we can explicitly calculate the decision boundary of the model by algorithm.

Figure 5 illustrates the results of decision boundary extraction on the toy dataset. We trained a 4-layer neural network with sigmoid and tanh as activation functions, respectively, and used Algorithm 1 to compute their linear approximation by taking the number of pieces  $n = 3$  (noted as  $p_3$ -sigmoid/tanh). The results in the figure show that our method is highly efficient.

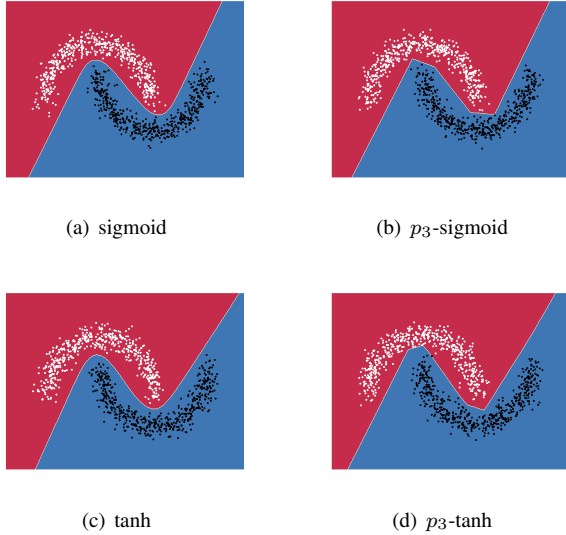


Fig. 5. (a) and (b) are the decision boundary of the FCNN with sigmoid and tanh functions respectively, (b) and (d) show the decision boundary of the FCNN with  $p_3$  linear approximate function

However, if we want to traverse the activation state of all neurons, the computational complexity is  $O(n^a)$ , where  $a, n$  is the number of activation state and neurons, respectively. As an alternative, the algorithm traverses the train set.

Another question is that, how can we use the decision boundary consisting of a set of hyper-planes? we will talk about it in the following part.

#### D. model compression based on decision boundary

For the decision boundary consisting of several hyper-planes, it is still not conducive to further use and understanding. For example, if the decision boundary is not one simple convex set, we need to recalculate the topological relationship of samples of different labels. However, it is computationally expensive to search in high-dimensional

---

#### Algorithm 2 Calculating the Decision Boundary

---

**Input:**  $f(x)$ :=the target FCNN,  $\sigma(x)$ :=activation function used in  $f(x)$ ,  $\delta$ :=given precision

**Output:**  $DB$ :=the decision boundary of  $f(x)$

```

1: Initialization:  $n = n_0, \epsilon = \epsilon_0, DB = \emptyset$ 
2: while  $\epsilon > \delta$  do:
3:    $n \leftarrow n + 1$ 
4:    $p_n(x), \epsilon \leftarrow \text{Algorithm 1}(\sigma(x), n)$ 
5: end while
6: compute the activation state set  $AS \leftarrow \{AS_i : x, f(x)\}$ 
7: for each  $AS_i$  in  $AS$  do
8:   if the label  $f(x)$  in  $AS_i$  are not the same do
9:     compute the CLIME by Eq.(9)
10:    compute the  $DB_i$  by Definition 4
11:   end if
12:    $DB \leftarrow DB \cup DB_i$ 
13: end for
14: return  $DB$ .
```

---

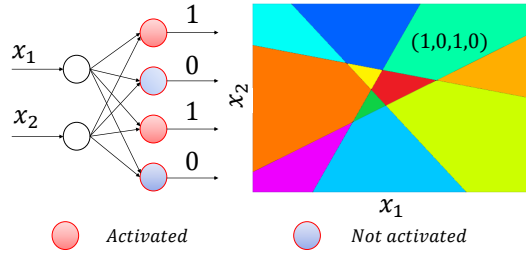
space. To solve this problem, we propose the DecisionNet (DNet) as an alternative to the decision boundary set.

Figure 6(a) shows the details of the DNet. We use the parameters of the hyper-planes in the decision boundary set as the weights and bias of a neural network with a single layer. For the arbitrary samples in the input space, if the output of DNet is exactly the same, then the label of such samples must be the same. According to this nature of the DNet, firstly, we use the train set to label all output states. When inferring a new sample, we can obtain the label of the new sample by comparing the output of the DNet. When there is a new state not covered by the train sample, we can rely on the original network for recalibration.

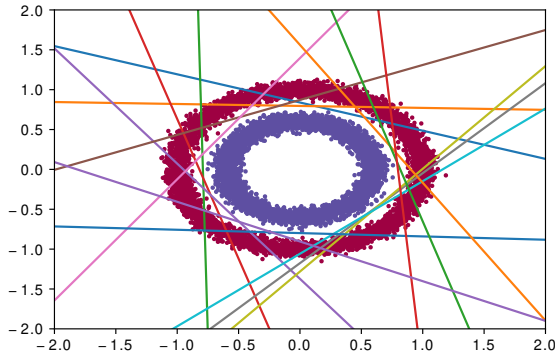
In addition to the convenience of calculation, another obvious benefit is that we can use the network to perform model compression. The quantity of parameters of DNet is much smaller than the original network. Theoretically, for any given accuracy, we can get the DNet corresponding to the FCNN. As the linear interval of the activation function grows, the number of decision boundaries also grows. Considering the trade-off between model size and accuracy, we can adjust it as needed. We test the DNet in synthetic data illustrated in figure 6(b), which shows that the compression rate of model parameters reaches 2%.

## VI. CONCLUSION

This work gives a perspective of the data space division for understanding the full-connected neural networks. Firstly, we believe that there are two basic properties that should belong to the interpretability, which are, respectively, completeness and explicitness. Based on this point, we propose the principle of complete local interpretable model-agnostic explanations (CLIME). Next, we further define the classification-complete intertability, under the concept of which we illustrate the equivalence of decision boundary and CLIME. To implement CLIME in FCNNs, a method according to linearizing the activation functions has been



(a) DNet



(b) DNet on toy dataset

Fig. 6. (a) DNet is constructed with parameters of decision boundaries. The decision boundary hyper-plane set will divide the space into multiple regions, and since the labels of the data in each region are the same, it is only necessary to label each region in advance, and when new samples are coming, the labels of the samples can be known by comparing the active region. (b) Since only the parameters of the decision boundary are used, the DNet is with single layer and thus can compress the model efficiently.

proposed. After approximating the activation functions by piece-wise linear functions (PWL), the decision boundary corresponding to FCNNs in classification problem can be calculated efficiently. Last but not least, we propose the DecisionNet(DNet) to simplify the calculation and compress the original model. Experiments show the DNet achieves almost lossless compression ratios under a controllable scale.

In the future, we will dedicate ourselves to developing the principle of CLIME to the general machine learning algorithms, including the calculation of the complete decision boundary corresponding to classification models, as well as the general DNet.

## REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 6000–6010.
- [3] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81.
- [4] F.-L. Fan, J. Xiong, M. Li, and G. Wang, "On Interpretability of Artificial Neural Networks: A Survey," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 7311, no. c, pp. 1–1, 2021.
- [5] Y. Zhang, P. Tiño, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *arXiv preprint arXiv:2012.14261*.
- [6] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, no. October 2019, pp. 82–115, 2020.
- [7] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.
- [8] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications," *Proc. IEEE*, vol. 109, no. 3, pp. 247–278, 2021.
- [9] G. Vilone and L. Longo, "Explainable Artificial Intelligence: a Systematic Review," *arXiv preprint arXiv:2006.00093*, no. September, may 2020.
- [10] "Theoretical issues in deep networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 117, no. 48, pp. 30 039–30 045, 2020.
- [11] Z. C. Lipton, "The mythos of model interpretability," *Communications of the ACM*, vol. 61, no. 10, pp. 36–43.
- [12] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?" Explaining the predictions of any classifier," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-August-2016, pp. 1135–1144, 2016.
- [13] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," in *International Conference on Learning Representations*, 2018.
- [14] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," *arXiv preprint arXiv:1711.09784*, 2017.
- [15] S. Avner, "Extraction of comprehensive symbolic rules from a multi-layer perceptron," *Engineering Applications of Artificial Intelligence*, vol. 9, pp. 137–143, 1996.
- [16] R. Pascanu, G. Montúfar, and Y. Bengio, "On the number of inference regions of deep feed forward networks with piece-wise linear activations," in *Second international conference on learning representations - ICLR 2014: 14-16 April 2014, Banff, Canada*. Banff: ICLR, 2014.
- [17] B. Hanin and D. Rolnick, "Deep relu networks have surprisingly few activation patterns," *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.
- [18] X. Zhang and D. Wu, "Empirical studies on the properties of linear regions in deep neural networks," in *International Conference on Learning Representations*, 2020.
- [19] X. Hu, W. Liu, J. Bian, and J. Pei, "Measuring model complexity of neural networks with curve activation functions," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.
- [20] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *Computer Science*, 2015.
- [21] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.
- [22] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision, ECCV 2014 - 13th European Conference, Proceedings*, no. PART 1, 2014, pp. 818–833.
- [23] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Computer Science*, 2013.
- [24] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR (workshop track)*, 2015.
- [25] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.
- [26] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

- [27] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K. Müller, “Explaining nonlinear classification decisions with deep taylor decomposition,” *Pattern Recognition*, vol. 65, pp. 211–222, 2016.
- [28] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.
- [29] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “Smoothgrad: removing noise by adding noise,” *arXiv preprint arXiv:1706.03825*, 2017.
- [30] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3145–3153.
- [31] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3449–3457.
- [32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene cnns,” *arXiv preprint arXiv:1412.6856*, 2014.
- [33] J. Zhang, S. A. Bargal, L. Zhe, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1084–1102, 2018.
- [34] P.-J. Kindermans, S. Hooker, J. Adebayo, K. T. Schütt, M. Alber, S. Dähne, D. Erhan, and B. Kim, “The (un)reliability of saliency methods,” 2018. [Online]. Available: <https://openreview.net/forum?id=r1Oen--RW>
- [35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2921–2929.
- [36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2017, pp. 618–626. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.74>
- [37] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 839–847.
- [38] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Score-cam: Score-weighted visual explanations for convolutional neural networks,” *arXiv preprint arXiv:1910.01279*, 2019.
- [39] S. Srinivas and F. Fleuret, “Full-gradient representation for neural network visualization,” *Advances In Neural Information Processing Systems 32 (Nips 2019)*, 32, 2019.
- [40] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *Computer Science*, vol. 14, no. 7, pp. 38–39, 2015.
- [41] L. Chu, X. Hu, J. Hu, L. Wang, and J. Pei, “Exact and consistent interpretation for piecewise linear neural networks: A closed form solution,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, pp. 1244–1253.
- [42] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S.-T. Yau, and X. Gu, “A geometric understanding of deep learning,” *Engineering*, vol. 6, no. 3, pp. 361–374, 2020.
- [43] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [44] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4768–4777.
- [45] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” *arXiv preprint arXiv:1710.05941*, 2017.
- [46] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv:1811.03378*, 2018.
- [47] G. Vlassopoulos, T. van Erven, H. Brighton, and V. Menkovski, “Explaining predictions by approximating the local decision boundary,” *arXiv preprint arXiv:2006.07985*, 2020.
- [48] H. Karimi, T. Derr, and J. Tang, “Characterizing the decision boundary of deep neural networks,” *arXiv preprint arXiv:1912.11460*, 2019.
- [49] Y. Li, L. Ding, and X. Gao, “On the decision boundary of deep neural networks,” *arXiv preprint arXiv:1808.05385*, 2018.
- [50] O. Melnik, “Decision region connectivity analysis: A method for analyzing high-dimensional classifiers,” *Mach. Learn.*, vol. 48, no. 1–3, p. 321–351, Sep. 2002. [Online]. Available: <https://doi.org/10.1023/A:1013968124284>
- [51] D. Mickisch, F. Assion, F. Greßner, W. Günther, and M. Motta, “Understanding the decision boundary of deep neural networks: An empirical study,” *arXiv preprint arXiv:2002.01810*, 2020.