Interpretability of Deep Learning

Zhenlin Huang, Fan Li, Zhanliang Wang, and Zhiyuan Wang

Abstract—**Deep Learning achieves surprising performance in many real-world tasks. However, on a black-box approach, computational techniques have been applied without a strong critical understanding of their properties. In this paper, we review the current methodologies and techniques about improving the interpretability of Deep Learning from different research directions. Some works are based on analysis of the learning process, some lay more emphasis on interpreted network architecture, and others intend to design self-interpretable Deep Learning models. This article analyzes the popular and advanced works in these fields and provides a future look for Deep Learning researchers.**

Index Terms—**Interpretability, proxy model, salience map, separate representation method, multimodality.**

I. Introduction

As deep learning models play an increasingly important role in many real applications for people's daily life, the interpretability of models has become a key factor to determine whether users can trust these models. Interpretability aims to describe the internal structure of the system in a way that human beings can understand.

According to the action time of interpretable methods, the matching relationship between interpretable methods and models, and the action scope of interpretable methods, we can divide the interpretable methods of machine learning into different categories, including essential interpretability and post interpretability, interpretability for specific models, model independent interpretability, local interpretability, and global interpretability.

Common deep networks use a lot of basic operations to make decisions. The basic way to explain this complex model is to reduce its complexity. This can be accomplished by designing agent models, such as linear agent model, and decision tree model. Additionally, it can also be achieved through constructing a salience map to highlight the most relevant part of the calculation, so as to provide interpretability.

A lot of neural network operations can be used, and the deep neural network is composed of a few sub components. The interpretation of deep network representation aims to understand the role and structure of data flowing through these information bottlenecks. It can be divided into two sub categories, including layer-based interpretation and neuron-based interpretation, according to its granularity. Layer-based interpretation considers all the information flowing through the layer together, while neuron-based interpretation is used to explain the situation of a single neuron or a single filter channel. In addition, based on the interpretation of other representation vectors, other directions in the representation vector space formed by the linear combination of single units are used as their representation vectors.

The network model itself can also be explained through different design methods and training. Three common methods are involved, which are attention mechanism network, separation representation, and generative interpretation. Networks based on attention mechanism can learn some functions that provide weighting of input or internal features to make the information accessible to other parts of the network. The representation of the separation method can use a separate dimension to describe meaningful and independent change factors. In application, deep network can be used to train the separation representation of explicit learning. In generative interpretation methods, deep neural network can also generate human understandable interpretation as a part of system explicit training.

II. Interpretability of Learning Process

A. Proxy Model

Currently, most of the widely used deep learning algorithms are still "black box models". It is necessary to understand the reasoning process behind the prediction so as to evaluate the reliability of the model when deciding whether to deploy the new model. One possible approach is to approximate the "black box model" with a linearly interpretable model.

Ribeiro et al. proposed a creative technique“LIME” that can interpret the predictions of arbitrary classifier in an explainable way by learning an interpretable model around the predicted results [1]. The authors also design the task as a sub-module optimization problem by simply showing the representative individual prediction results and their interpretation.

Another method of proxy model is decision tree. The work of decomposing neural networks into decision trees began in the 1990s. This work can explain shallow networks and gradually generalize to the calculation process of deep neural networks. A new rule extraction [2] method “CRED” is proposed. The neural network is decomposed by decision tree, and the generated branches are combined by c/d-rule algorithm to produce the interpretation of neural network input and output with different classification granularity, which can consider continuous and discrete values. DeepRED extends cred's work to multi-layer networks and

Manuscript received Mach 9, 2022; revised May 11, 2022.

Zhenlin Huang is with the Department of Artificial Intelligence, Central China Normal University, Wuhan, China (e-mail: huangzhenlin_666@163.com.

Fan Li is with Department of Computer Science, Tongji University, Shanghai, China (e-mail: 1950670@tongji.edu.cn).

Zhanliang Wang is with the Department of Mathematics New York University, New York, USA (correspondent author; e-mail: zw3342@nyu.edu).

Zhiyuan Wang is with Department of Mathematics, Ohio State University Columbus, USA (e-mail: wang.11193@osu.edu).

doi: 10.18178/ijfcc.2022.11.2.585

adopts a variety of structures to optimize the structure of spanning trees. Another decision tree structure is ANN-DT [3], which also uses the node structure of the model to establish a decision tree and divide the data. The difference is that the judgment node uses positive and negative methods to judge whether the function of the position is activated, so as to divide the data. After the decision tree is generated, the rules of the neural network are obtained by sampling and performing experiments in the sample space.

Automatic rule generation is another method to summarize the rules of model decision-making. In the 1980s, Gallant regarded neural network as a database for storing knowledge. In order to explore information and rules from the network, A method was proposed to extract rules from simple networks [4], which can be regarded as the origin of the application of rule extraction in neural networks. Nowadays, the rule generation technology in neural network mainly regards the output as a set of rules, and obtains rules from it by means of propositional logic. Hiroshi tsukimoto [6] proposed a method to extract feature extraction rules from trained neural networks. This method belongs to decomposition method and can be applied to neural networks with monotonic output, such as the sigmoid function. This method does not depend on the training algorithm, and the computational complexity is polynomial.

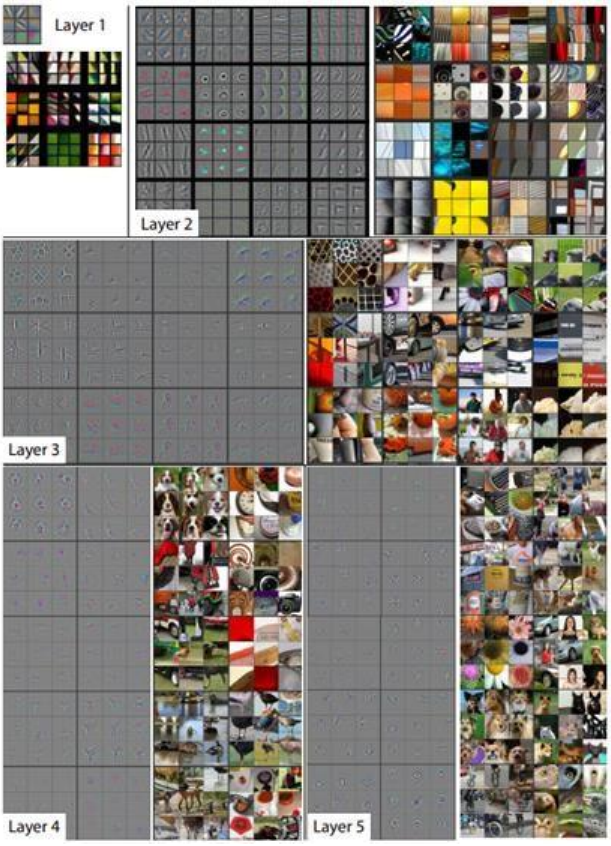

Fig. 1. Visualization of features in a fully trained model [5].

B. Transposed Convolution

The convolution process of CNN model is essentially not different from that of an ordinary neural network, which combines some common parameters into a filter form. They can be transformed into an operation of matrix multiplication, that is a sparse matrix with high parameter repetition for CNN model. The relationship between the latter layer and the previous layer can be expressed as Equation 1:

$$CX^{l} = X^{l+1} \tag{1}$$

where *C* represents the sparse matrix. In the process of back propagation, *CT* can be regarded as the gradient of the convolution layer to the input layer. In other words, the product of the corresponding gradient matrix and the current convolution layer can be obtained by properly adding 0 to the convolution layer and using the convolution kernel after the transpose of the original convolution kernel. Unpooling feature, which contains most of the zeros, is used to indicate the effect of the original input on the pooled features. Since all values are guaranteed to be non-negative during deactivation, the sign does not change during deconvolution.

As shown as the experimental results in Fig. 1, the second layer corresponds to some edges, corners, and color features. The third layer corresponds to texture features. The fourth layer corresponds to some local parts such as dog faces and wheels. The fifth layer has a strong ability to recognize the overall object. Through the work of visualizing and understanding convolutional neural networks, the types of concepts learned by neural networks with the naked eye can be roughly judged.

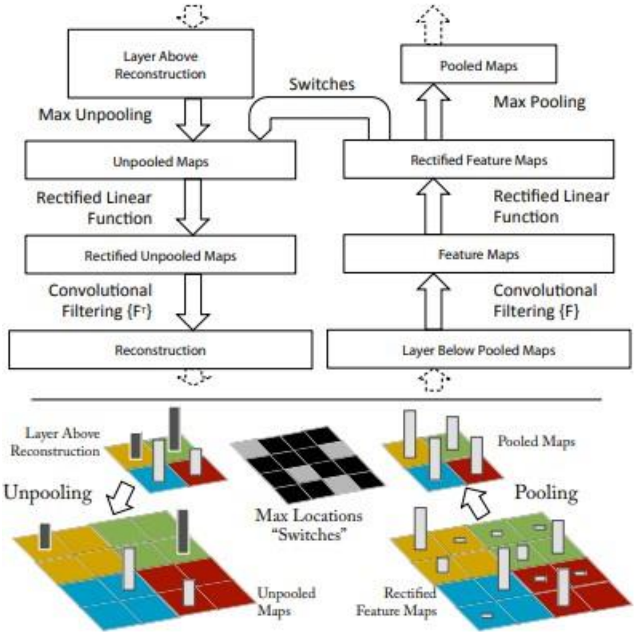

Fig. 2. Top: A deconvnet layer (left) attached to a convnetlayer (right). Bottom: An illustration of the unpoolingoperation in the deconvnet [5].

Each layer of the deconvolution network can be regarded as the inverse process of the corresponding layer in the deconvolution network. They have the same convolution kernel and pooled index. Therefore, deconvolution inversely maps the eigenvalue back to the pixel space of the input image, thereby indicating which pixels in the image participate in activating the eigenvalue. Fig. 2 combines the processes of convolutional network and deconvolution network to show the relationship between the two layers. Both are mutually inverse processes overall. First, the convolutional network takes a picture as input and calculates the feature representation of each layer. To verify a specific eigenvalue of a certain layer, all values other than the eigenvalue of this layer are zeroed and taken as the input of the deconvolution network. After each layer of deconvolution network operation, the eigenvalue is mapped back to the pixel space of the input image.

Unpooling operation [5]: Theoretically, maximum pooling operation in convolutional networks is irreversible. However, it can be nearly reversible by pooling indexes.

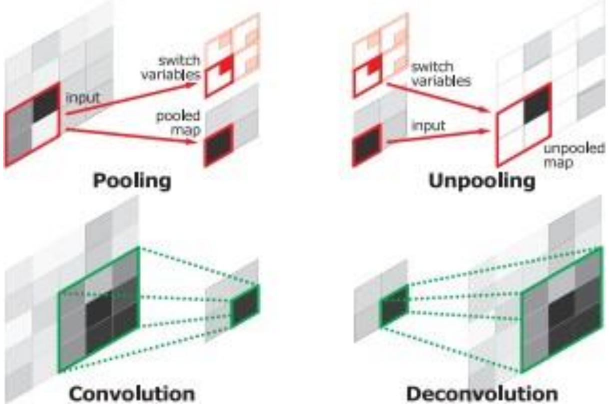

Fig. 3. Illustration of deconvolution and unpooling operations [6].

Rectification [5]: ReLU is adopted in convolutional networks to ensure non-negative eigenvalues. To ensure the consistency of forward and inverse processes, the reconstruction features of each layer of the deconvolution network are also obtained non- negative values through ReLU.

C. Gradient Based Method

The correspondence between the real data distribution pdata(x|y) and the quasi-conditional density function pθ(x|y) is what makes the input-gradients interpretation meaningful. Strengthening the correspondence between the two density functions makes estimated distribution more consistent with the real data distribution. Input-gradients should do better. On the contrary, weakening the correspondence will make the interpretability meaningless and even wrong. To find the correspondence between the real data distribution and the quasi-conditional density function, it is equivalent to find the correspondence between $\nabla xpdata(x|y)$ and $\nabla xlogp\theta(x|y)$. Intuitively, it is just to estimate the expected distance between the two functions. The target function of score-matching, which makes expected square distance between the model score function and data score function small, is presented in Equation 2 [7]:

$$J(\theta) = Ep_d(x)\frac{1}{2}||\nabla_x logp_\theta(x|y) - \nabla_x p_{data}(x|y)||_2^2 \qquad (2)$$

This formula is almost unsolvable in higher dimensional space. Scholars have given a simplification. Under very loose conditions, the formula can be obtained as below [7]:

$$J(\theta) = Ep_d(x)\left\{trace(\nabla_x^2 \log p_\theta(x)) + \frac{1}{2}||\nabla_x p_\theta(x|y)||_2^2\right\} + const \qquad (3)$$

The input gradients can vary arbitrarily without changing. The prediction can be regarded as gradients of a quasi-conditional density function $p\theta(x|y)$. $p\theta(x|y)$ reflects the real data distribution $pdata$ $(x|y)$. The close correspondence between the two functions makes the input-gradients interpretability meaningful.

III. Interpretability Based on Deep Neural Network

The aim of interpretability is to present some of the characteristics of an Machine Learning (ML) model in comprehensible terms to a human. Interpretations can be obtained by way of comprehensible proxy models, which approximate the predictions of a more complex approach [8]. Our task in this section is to provide a reasonable and understandable summary and overview of the interpretability of deep neural networks in machine learning and the related work that has been done to improve the interpretability of the DNNs.

Deep Neural Networks are based on simple mathematical operations, however, the combination of neurons with nonlinear activation at several hidden layers results in that models cannot be evaluated by mathematical formulas. This characteristic of Deep Neural Networks is lack of transparency[9]. The lack of transparency may lead to unreliability since uncertain problems could exist even when the best deep learning models are applied in common unexplainable ways.

A. The Interpretability Based on Layers

Understanding the application of deep neural network has become an important research goal for artificial intelligence scientists in recent years, and remarkable theoretical progress has been made to a certain extent. A focal point of those studies stems from the success of excessively large networks which defy the classical wisdom of uniform convergence and learnability [10].

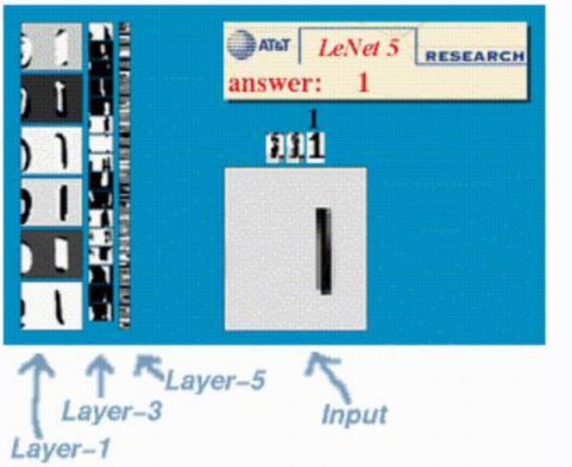

Fig. 4. The working process of LeNet and MNIST [4].

A classical example is the LeNet-5 network for the MNIST digit recognition task [12]. A network constructed in this way is called an "excessively large network" because the depth of the network is much greater than the number of training examples, and the number of parameters in the network is also much greater than the number of training examples. In these cases, the network is over-parameterized and has many degrees of freedom. As a result, the training process is highly sensitive to initial conditions, and the network can learn a very large number of patterns that are not useful for the task at hand. The classical wisdom about the learnability of such networks has indicated that their capacity is limited, and they can only learn a small number of patterns.

In contrast, the classical wisdom about the learnability of small networks is that they are not capable of learning complex patterns. However, we have not seen small networks with the capacity to learn complex patterns. Thus, scientists have developed a new approach to train neural networks that can learn complex patterns. This approach is based on the idea of "interpretability and interpretability-based learning" [13]. If a network is capable of learning complex patterns, the network to develop a new interpretation of the data will be applied. This interpretation provides us with a way to understand the data, and we can use this understanding to improve our ability to learn complex patterns.

B. The Interpretability Based on the Convolutional Neural Networks

Deep convolutional neural networks(CNNs) have been successfully used in a variety of image processing and have become one of the most important tools in many fields like video processing and computer vision. However, the mechanism on how CNNs learn and work has not been clearly revealed and understood. A key barrier to the adoption of deep models in many applications is lack of interpretability. Due to this reason, developing the interpretability of the CNNs becomes a significant work to make systems be more consummate.

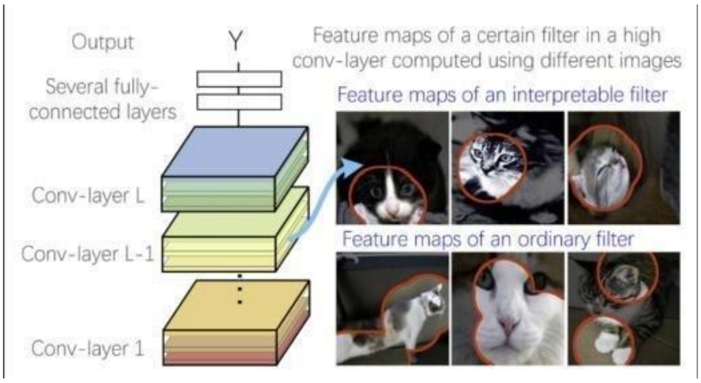
Output
Y
Several fully-connected layers
Conv-layer L
Conv-layer L-1
Conv-layer 1
Feature maps of a certain filter in a high conv-layer computed using different images
Feature maps of an interpretable filter
Feature maps of an ordinary filter

Fig. 5. Comparison of filter's feature maps in an interpretable CNN and those in a traditional CNN [7].

With the help of deconvolution[15] and guided back propagation[16] the interior of the CNN model can be obtained, however, it is not used to explain the classification results. They are not sensitive to the category and show directly all the features that can be extracted. A high-level filter can describe a mixture of patterns, that is, the filter may be activated by the head and legs of an object [14]. According to the example shown in Figure5, the filter can be activated and recognized by a body part of the cat, which means that the filter can be triggered by the head or the legs of the cat. Therefore, this representation in the high convolutional layer is complex and makes the model difficult to understand. Based on the reduction of the CNN interpretability, it is necessary to use interpretable CNN algorithms to solve this problem. In this way, parts of the object can be recorded in the classification without ambiguity. Class Activation Map (CAM) is proposed to solve the problem of class insensitivity in deconvolution and guided back propagation.

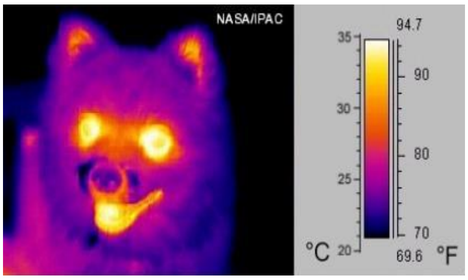

Fig. 6. Image of a Pomeranian taken in mid-infrared ("thermal") light (false- color)[17].

The CAM image is similar to the image above. When we need the model to explain the reasons for classification, it shows the basis of its decision in the form of a Saliency Map, which tells us where a hot object in the dark is. For a deep convolutional neural network, after multiple convolution and pooling, the last layer of the convolutional layer contains the richest spatial and semantic information. The fully connected and softmax layers, containing information that is difficult for humans to understand and display in a visual way. Therefore, to make the convolutional neural network give a reasonable explanation of its classification results, it is necessary to make full use of the last convolutional layer. This is one of the most important tasks of interpretable neural network models.

IV. Self-interpretable Deep Learning System

Through designing different training ways and network architectures, we can make deep learning model acquire better representation ability and interpretability by themselves.

A. Separate Representation Method

From the view of representation learning, a good model should learn representations of the data that make it easier to extract useful information when building classifiers or predictors. Encoding data more explicitly is beneficial to use these features to represent samples independently.

InfoGAN is a milestone in feature learning [18]. One of the biggest problem of GANs in prior work is that on how generator G(z) use noise z, z can be used in a highly entangled way and each of dimension of z may not represent any salient feature of the training data. To solve this, Chen et.al proposed to decompose the input noise vector into two parts. One is incompressible noise z which contains implicit semantic information that is hard to understand, the other is latent code c which will target the salient structured semantic features of data distribution. A new generator G(z,c) can be obtained and adjusted to maximize the influence imposed to distribution of generated data by latent code c. Inspired by information theory, the authors proposed an information-theoretic regularization: there should be a high mutual information between c and G(z,c). Thus I(c,G(z,c)) should be high [18]. The new minimax game played by D and G can be modified as below:

$$\min \max V_I(D, G) = V(D, G) - \lambda I(c, G(z, c)) \quad {}_{G,D} \tag{4}$$

However, posterior *P(c|x)* is hard to be accessed in practice, The authors defined an auxiliary distribution Q(c|x) to approximate it and use the Variational information Maximization technique to find the lower bound of the mutual information as follow:

$$\begin{aligned} L_I(G, Q) &= E_{c\sim P(c),x\sim G(z,c)}[\log Q(c \;/\; x)] + H(c) \\ &= E_{x\sim G(z,c)}\,[E_{c'\sim P(c\,/\,x)}[\log Q(c' \;/\; x)]] + H(c) \\ &\leq I(c; G(z, c)) \end{aligned} \tag{5}$$

It is easy to optimize LI(G, Q) , when LI(G, Q) reaches its maximum H(c). The max value of mutual information can be accessed. The final objective function is as below:

$$\min_{G,Q} \max_{D} V_{InfoGAN}\ (D, G, Q) = V(D, G) - \lambda L_I(G, Q) \tag{6}$$

Futhermore, the authors did many experiments on different datasets such as MNIST with InfoGAN and verified that mutual information can be effectively maximized through training and the model can learn interpretable features separated from entangled data. This indicates that an uncontrollable black-box model can be converted into a self- interpretable system.

Explanatory graph mentioned by Zhang *et al*. is another significant work [19]. An explanatory graph represents the knowledge hierarchy hidden in Conv-layers of CNN. Considering each filter in a CNN can be activated by different objects parts. Zhang et al. used graph nodes in different layers to represent part patterns and stored the information in feature maps. Besides, co-activation logics and spatial relationships between nodes were encoded by graph edges that connect them. In a word, this method disentangles part patterns from each filter in an unsupervised manner.

B. *Multimodal Approach*

Multimodal Representation is devoted to digging the complementary information among different modalities and eliminates redundant features to learn better feature representations [20]. In this way, it is easier for us to explain the mechanism of a deep model.

Park et al. claimed that considering multimodal explanations results in better visual and textual explanations in the 2018 CVPR [21]. Through reorganizing two new datasets, which are ACT-X and VQA-X datasets, they generated a dataset that includes visual information and textual justification offered by humans and trained a novel model called Pointing and Justification Explanation(PJ-X)on it

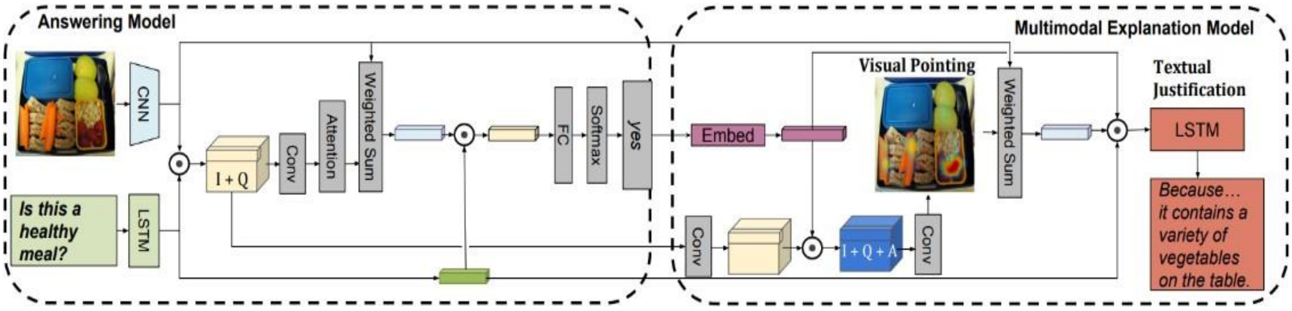

Fig. 7. Pointing and Justification (PJ-X) architecture [21].

A Q&A task can be taken as an example. Using the attention mechanism [22]. first, the model acquires the importance of different pixels in an image and output the Visual Pointing. After that, textual justification is generated to further explain the prediction. The authors argued that two modalities will help us to train a model and provide a complementary explanatory strength [21]. The Architecture of PJ-X is shown in Fig. 7.

V. Conclusion

This article introduces the interpretability of deep learning from three aspects: the interpretability of the deep learning process, the interpretability of the deep network representation and the deep learning system. The agent model that is easier to explain can reduce the complexity of the original model or construct the most relevant part of the calculation to provide interpretability. The decision tree can explain shallow networks and gradually generalize to the calculation process of deep neural networks. Transposed Convolution inversely maps the eigenvalue back to the pixel space of the input image, thereby indicating which pixels in the image participate in activating the eigenvalue. Multimodal Representation is devoted to digging the complementary information among different modalities and eliminates redundant features to learn better feature representations.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Fan Li collected and reorganized the papers and materials about interpretability of deep learning in recent years and designed the content and structure of paper. Fan Li also wrote the forth section of paper—self-interpretable deep learning system. Zhanliang Wang wrote the third section of paper—interpretability based on deep neural network and typeset the paper.Zhenlin Huang wrote the introduction and Proxy Model part in the paper. Zhiyuan Wang wrote the conclusion and Transposed Convolution and Gradient-based method parts in the paper.all authors had approved the final version.

REFERENCES

[1] R. M. Tulio, S. Singh, and C. Guestrin. “Why should I trust you?' Explaining the predictions of any classifier,” in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016

[2] S Makoto, and H. Tsukimoto, “Rule extraction from neural networks via decision tree induction,” in *Proc. International Joint Conference on Neural Networks*, vol. 3, 2001.

[3] P. J. S. Gregor, C. Aldrich, and F. S. Gouws, “ANN-DT: an algorithm for extraction of decision trees from artificial neural networks,” *IEEE Transactions on Neural Networks*, 1999, pp. 1392- 1401.

[4] T. Hiroshi, “Extracting rules from trained neural networks,” *IEEE Transactions on Neural Networks*, vol. 11, no. 2, 2000, pp. 377-389.

[5] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” in *Proc. European Conference on Computer Vision*, Springer International Publishing, 2013.

[6] H. Noh, S. Hong, and B. Han. (2015). Learning deconvolution network for semantic segmentation. [Online]. Available: https://arxiv.org/abs/1505.04366

[7] S. Srinivas and F. Fleuret. (2021, March 3). Rethinking the role of gradient-based attribution methods for model interpretability. arXiv.org. [Online]. Available: https://arxiv.org/abs/2006.09128.

[8] R. Roscher, B. Bohn, M-F. Duarte, and J. Garcke, *Explainable Machine Learning for Scientific Insights and Discoveries*, 2021.

[9] H. Hagras, “Toward human-understandable, explainable AI,” *Computer*, vol. 51, no. 9, pp. 28-36, 2018.

[10] C. Zhang, S. Bengio, and Y. Singer, *Are all Layers created Equal?*, 2019.

[11] K. Le. (2021, July 23). Lenet and mnist handwritten digit classification. Medium. [Online]. Available: https://medium.com/mlearning-ai/lenet-and-mnist-handwritten-digit-classification-354f5646c590.

[12] AWikipedia Contributors. (2021, May 5). LeNet. In Wikipedia, The Free Encyclopedia. Retrieved 09:57. [Online]. Available: https://en.wikipedia.org/w/index.php?title=LeNet&oldid=1021579098

[13] A. Bibal and B. Frénay, “Interpretability of machine learning models and representations: an introduction,” In ESANN.

[14] Q. Zhang, Y. N. Wu, and S. C. Zhu, “Interpretable convolutional neural networks,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8827-8836, 2018.

[15] C. Schmied and P. Tomancak, “Quantitative imaging in cell biology,” *In Methods in Cell Biology*, 2014.

[16] R. Draelos, “CNN heat maps: Gradients vs. deconvnets vs,” Guided Backpropagation.

[17] Wikipedia contributors. (2021). [Online]. Available: https://en.wikipedia.org/w/index.php?title=Thermographic_camera&oldi d=1022740066

[18] InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. Neural Information Processing Systems (NIPS)

[19] Q. Zhang, R. Cao, F. Shi, Y. N . Wu, and S. C. Zhu, “Interpreting cnn knowledge via an explanatory graph,” 2017.

[20] T. Baltrusaitis, C. Ahuja, and L. P. Morency, “Multimodal machine learning: a survey and taxonomy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[21] D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, and T. Darrell *et al*., “Multimodal explanations: Justifying decisions and pointing to the evidence,” in *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.R. Nicole, “Title of paper with only first word capitalized*, 2018.

[22] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Fan Li was born in Hefei, China. Fan Li majors in Data Science and Big Data Technology from 2019.9.1~present in Tongji University and now in his junior year.

His research intersts include machine learning and applied data science

Zhenlin Huang got his master degree in computer science from Central China Normal University. He was born in henan province.

During his master's period, he participated in the world robotics competition. Currently he is a software engineer. His research interests include high-dimensional statistics, probability, and machine learning.

Zhanliang Wang was born in Qingdao, Shandong Province, on February 23rd, 2001. Zhanliang Wang studied Applied Mathematics and Statistics at Stony Brook University's College of Engineering and Applied Science from 2019.9 - 2021.5 and Tandon College of Engineering at New York University from 2021.9-present, to pursue a degree in Applied Mathematics.

He was a teaching assistant in AMS 361, Applied Calculus IV: Differential Equations, and AMS210, Applied Linear Algebra at Stony Brook University from 2020.09 to 2021.05. Also, he worked at the China Center to help solve problems for international students from 2020.9 to 2021.5 in Stony Brook University. His research interest is the application of Machine Learning in the Data Science field.

Zhiyuan Wang was born in Harbin, February 1st, 2000. Zhiyuan Wang earned Actuarial Science, Bachelor of Science Degree at The Ohio State University in August 2021.

He did the internship of Financial Research Assistant in China Great Wall Securities from May to July 2021. He also did the internship of Wealth Management Analyst in Angelis Wealth Group, Inc. from July to September 2021. His research interest is machine learning in the financial field.