DRIB: Interpreting DNN with Dynamic Reasoning and Information Bottleneck

Yu Si, Keyang Cheng(✉), Zhou Jiang, Hao Zhou, and Rabia Tahir

School of Computer Science and Communication Engineering, Jiangsu University,
Zhenjiang 212013, China
{siyu,2222008027,2212108025}@stmail.ujs.edu.cn,
kycheng@ujs.edu.cn

Abstract. The interpretability of deep neural networks has aroused widespread concern in the academic and industrial fields. This paper proposes a new method named the dynamic reasoning and information bottleneck (DRIB) to improve human interpretability and understandability. In the method, a novel dynamic reasoning decision algorithm was proposed to reduce multiply accumulate operations and improve the interpretability of the calculation. The information bottleneck was introduced to the DRIB model to verify the attribution correctness of the dynamic reasoning module. The DRIB reduces the burden approximately 50% by decreasing the amount of computation. In addition, DRIB keeps the correct rate at approximately 93%. The information bottleneck theory verifies the effectiveness of this method, and the credibility is approximately 85%. In addition, through visual verification of this method, the highlighted area can reach 50% of the predicted area, which can be explained more obviously. Some experiments prove that the dynamic reasoning decision algorithm and information bottleneck theory can be combined with each other. Otherwise, the method provides users with good interpretability and understandability, making deep neural networks trustworthy.

Keywords: Dynamic reasoning · Information bottleneck · Interpreting DNNs

1 Introduction

Deep neural networks (DNNs) have achieved superb performance and widespread acceptance in many application areas, for example, in image classification [1], object detection [2], sentiment analysis [3], and other applications [4, 5]. It is well known that the success of deep learning models comes from efficient mathematical learning algorithms and a large parameter weights. This huge parameter space makes DNNs a complex black-box model, as the parameter space contains hundreds of network layers and millions of parameters. We use many data and GPUs to train the deep learning model. As illustrated in Fig. 1, deep learning models are called black-boxes and are also referred to as end-to-end systems. The input is the original data, and then the output is directly the final goal. The intermediate process is unknown and difficult to know. It is also be

Supported by National Natural Science Foundation of China [61972183].

Y. Wang et al. (Eds.): ICPCSEE 2022, CCIS 1628, pp. 178–189, 2022.
https://doi.org/10.1007/978-981-19-5194-7_14

expressed as uninterpretable and untrustworthy. However, why do we need interpretability? Although the performance of DNNs is solid and even outperforms humans on many specific tasks, we cannot understand these models. We do not apply them, especially in high-risk domains such as medical diagnosis [6], financial risk forecasting [7], and automatic driving [8]. Therefore, it is essential to open the black box and raise trust and transparency to DNNs.

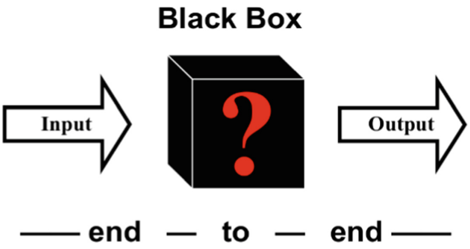


Fig. 1. The DNN is a black box trained from end to end. We cannot understand the mechanism of the inner process in it.

This paper proposes the dynamic reasoning and information bottleneck (DRIB) technique to construct an attribution interpretable convolution model. The contributions of our model are as described below:

1. A novel dynamic reasoning decision algorithm was proposed to reduce multiply accumulate operations and improve the interpretability of calculations.
2. The information bottleneck was introduced to the DRIB model to verify the attribution correctness of the dynamic reasoning module.
3. Some experiments prove that the dynamic reasoning decision algorithm and information bottleneck theory can be combined with each other. Otherwise, the method provides users with good interpretability and understandability, making deep neural networks trustworthy.

2 Related Works

2.1 Explain the Existing Deep Learning Models

Some of the post-hoc explainable methods work after building the deep neural networks. The sensitive features of DNNs can be identified by the feature analysis method. This method analyzes the explainability of the model through a specific example by modifying the local input and observing the impact on the prediction [9–11]. Network dissection [12] depends on the emergence of disentangled or human-explainable units during training [10, 13, 14]. However, when this method is applied to large and deep networks, its analysis is ineffective. In addition, the class activation mapping method shines brilliantly in the explainability field. They obtain the visualization graph by multiplying the feature graph by the weight. The difference between them is that CAM [15] obtains the weight from the fully connected layer; Grad-CAM [16] flows the gradient of a specific class to each feature graph, and then uses the average gradient as the weight; Score-CAM [17] does not need the gradient, but generates the weight for each feature graph through its forwarding score. All the above methods generate class activation features from the convolution layer, but our DRIB uses the information bottleneck theory to compress information and does not need to perform additional calculations for visualization.

2.2 Construction of Interpretable Deep Learning Models

Another part of ad hoc interpretable methods can avoid bias in post hoc explainability analysis. An interpretable representation technique is used to make the constructed neural network more interpretable [18, 19]. These methods replace parts of DNNs with interpretable machine learning models, such as decision trees [20, 21] and rule systems [22, 23]. Other approaches [24, 25] fuse deep learning into each decision tree node. Zhu Songchun and Zhang Quanshi proposed an interpretable CNN [26], which allows filters at a high level to represent specific local objects, which assists us in understanding the internal logic of CNNs. However, the method of building interpretable models is not universal. Therefore, the DRIB is attached to each convolution layer to make the convolution layer interpretable, and layer by layer to achieve model interpretability.

3 Method

An overview of the dynamic reasoning and information bottleneck (DRIB) is illustrated in Fig. 2. It shows the l-th convolution layer. Two modules are attached to each convolution layer. One is the Dynamic Reasoning Decision; the other is Information Bottleneck Verification. When the neural network performs a feedforward operation on a single input, the dynamic reasoning decision module first outputs the weight channel masks. It then determines the weight used to perform the actual calculation. Then the information bottleneck verification module verifies whether the feature selected by the dynamic reasoning decision module is within the result of the information bottleneck. If so, the dynamic reasoning decision module result is adopted and sent to the next round of calculation.

This section describes in detail the dynamic reasoning decision module and the information bottleneck verification module in the following sections. This paper will first introduce the dynamic reasoning decision module. We introduce the principle of information bottleneck theory for deep neural network validation on the basis of dynamic inference decision module.

3.1 Dynamic Reasoning Decision Module

In the feedforward calculation of a convolutional neural network with an L layer, the l-th convolutional layer computes the output features $\boldsymbol{x}_l \in \mathbb{R}^{N\times C_o\times H_o\times W_o}$ with $\boldsymbol{x}_{l-1} \in \mathbb{R}^{N\times C_i\times H_i\times W_i}$ and weights $\theta_l \in \mathbb{R}^{C_o\times C_i\times k\times k}$ by Formula 1.

$$\mathbf{x}_l = conv(\mathbf{x}_{l-1}, \theta_l) \tag{1}$$

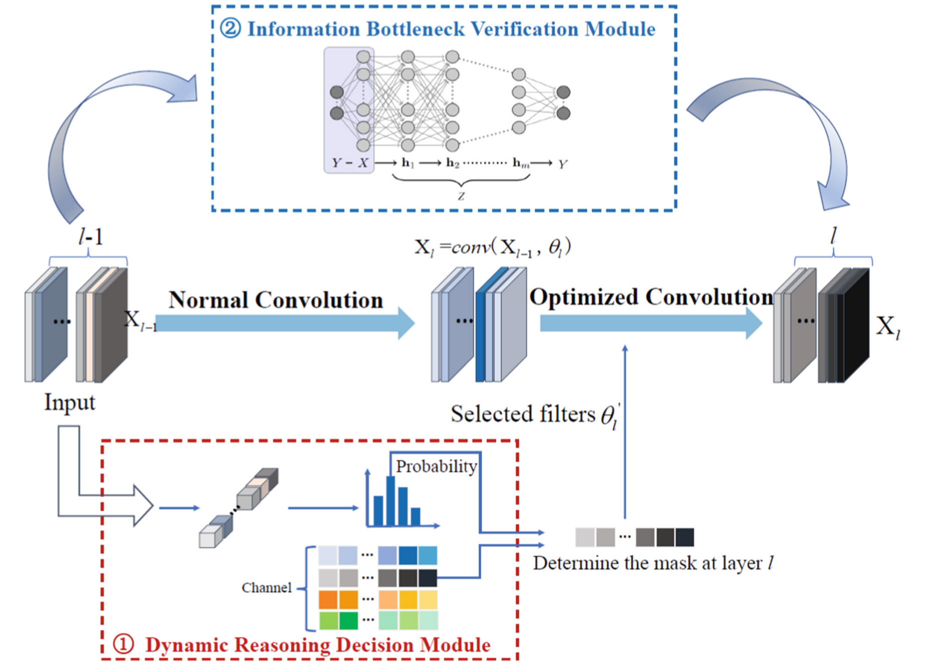

② Information Bottleneck Verification Module
Y - X → h₁ → h₂ ·········· hₘ → Y
Z
l-1
X_{l-1}
Normal Convolution
X_l =conv(X_{l-1}, θ_l)
Optimized Convolution
l
X_l
Input
Selected filters θ_l'
Probability
Channel
Determine the mask at layer l
① Dynamic Reasoning Decision Module

Fig. 2. The proposed DRIB overview model. The dynamic reasoning decision module optimizes the features with the best contribution in each convolution layer to reduce the amount of computation. The information bottleneck verification module verifies the accuracy of the whole network attribution and provides explanation and analysis.

where (N, C, H, W) are the corresponding batch, channel, height and width dimensions, respectively. The subscripts i, o denote variables for the input and output, respectively. Besides, the variable k is kernel size, and the symbol $conv$ is the convolution operation.

Suppose that $\boldsymbol{x}_{l-1}$ is the input features of the l-th layer. As indicated in Formula 2. Through the $ReLU$ layer and $AvgPool$ layer, we obtain an m-dimensional vector, $A(\boldsymbol{x}_{l-1}) \in \mathbb{R}^m$.

$$\begin{aligned} A(\mathbf{x}_{l-1}) &= Linear\ (GlobalAvgPool(Relu(\mathbf{x}_{l-1}))) \\ &= \left[p_1, p_2, \ldots, p_m\right] \end{aligned} \tag{2}$$

In addition, m kinds of masks are saved as $G_l \in \mathbb{R}^{m \times C_l}$ in layer l. $\pi(\boldsymbol{x}, \phi)$ is associated with each convolution layer. We take the largest $G_l[i]$ as the mask of layer l to run, where $i = argmax(A(x))$. How can the value of $G_l[i]$ be obtained? Given an m-dimensional output channel vectors with probability $P = \left[p_1, p_2, \ldots, p_m\right]$. The selection function I of the channel mask is represented formally as a categorical random variable. Gumbel random variables sampled from the Gumbel distribution $G = -log(-log(X))$ with $X \sim U[0, 1]$. Nevertheless, the problem of nondifferentiability to the underlying probability p is very troublesome. To solve this problem, the index I of $G_l[i]$ is substituted with a softmax form [27], as shown in Formula 3.

$$I_i = \frac{exp\left(\left(log\, p_i + G_i\right)/\tau\right)}{\sum_{j=1}^{m} exp\left(\left(log\, p_j + G_j\right)/\tau\right)}, \quad \forall i = 1, \cdots, m \tag{3}$$

where τ is a temperature parameter that governs the probability.

Finally, we can select weights with nonzero channel selection values $\theta_l^{'} = \{\theta_l[j] \mid \boldsymbol{G}[i][j] \neq 0\}$, and compute the final output for the l-th convolution layer by Formula 4.

$$\boldsymbol{x}_l = conv\left(\boldsymbol{x}_{l-1}, \theta_l^{'}\right) * G_l[i] \tag{4}$$

Therefore, the dynamic reasoning decision can be summarized as Algorithm 1 from the above steps.

Algorithm 1 Dynamic Reasoning Decision Algorithm

```
Input: feature $\boldsymbol{x}_{l-1} \in \mathbb{R}^{N \times C_i \times H_i \times W_i}$, weight $\theta_l \in \mathbb{R}^{C_o \times C_i \times k \times k}$
Output: feature $\boldsymbol{x}_l \in \mathbb{R}^{N \times C_o \times H_o \times W_o}$
Get an $m$-dimensional vector from $Linear\left(GlobalAvgPool\left(Relu(\mathbf{x}_{l-1})\right)\right)$;
for $k$ in [$0$, …, $C$-$1$] do
    Get $m$ kinds of masks $G_l \in \mathbb{R}^{m \times C_l}$ in layer $k$;
    $I_i = \frac{exp((log\, p_i + G_i)/\tau)}{\sum_{j=1}^{m} exp((log\, p_j + G_j)/\tau)}$;
    $i = argmax(A(x))$;
    $\theta_k^{'} = \{\theta_k[j] \mid \boldsymbol{G}[i][j] \neq 0\}$;
    $conv(\boldsymbol{x}_{l-1}, \theta_l^{'}) * G_l[i]$;
    $k \leftarrow k + 1$;
end for
return
```

3.2 Information Bottleneck Verification Module

The theory of an information bottleneck is based on mutual information. Given the variable X and label Y, the mutual information $p(x, y)$ is defined as Formula 5. For the distributions p and q, $D_{KL}[p \| q]$ is their Kullback-Liebler divergence. $H(X)$ is the entropy of X and Y. $H(X \mid Y)$ is the conditional entropy of X and Y. The mutual information $I(X; Y)$ is an analytical representation of the degree of correlation between two variables. This method quantifies the total amount of information represented by another variable through one variable.

$$\begin{aligned} I(X; Y) &= D_{KL}[p(x, y) \| p(x)p(y)] \\ &= \sum_{x \in X, y \in Y} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right) \\ &= H(X) - H(X \mid Y) \end{aligned} \tag{5}$$

The information bottleneck verification module is used to take a proficient and compacted representation of the unique information X without losing the ability to predict the label Y. This means that we want to obtain the minimum sufficient statistic. However, the detailed minimal sufficient statistics only exist for extraordinary distributions. The information bottleneck can be taken as a measure of the solution to rate-distortion theory. In addition, it provides an idea to obtain an approximate minimum sufficient statistic. Minimal sufficient statistics $Z(X)$ are maps or partitions that compress all the information that X has on Y. We can formulate minimal sufficient statistics through the Markov chain: $Y \to X \to Z(X)$. Z indicates the key information between the data and the predicted label.

Suppose $x \in X$ is the input and $z \in Z$ is the compressed and effective representation output of x. $p(\mathrm{z} \mid x)$ denotes the probabilistic mapping of the compressed representation. The information bottleneck can be formulated by Formula 6.

$$\min_{p(z|x), Y \to X \to Z}\{I(X; Z) - \beta I(Z; Y)\} \tag{6}$$

where $I(X; Z)$ and $I(Z; Y)$ denote the mutual information, and β determines those tradeoffs in the process of anticipating these labels, guaranteeing that the minimum sufficient statistic is obtained.

4 Experiments

The experimental settings will be introduced first in this section. The first part includes datasets, pretraining models and evaluation metrics. Then, comparisons of the amount of calculation displayed by the dynamic reasoning decision of the DRIB are made. Furthermore, a series of attribution results and pixel disturbances shows the faithfulness of the information bottleneck in the DRIB. Finally, some SOTA visualization methods are used to verify the interpretability of the DRIB model.

4.1 Experimental Settings

Two public and popular datasets were applied in our experiments: (1) **CIFAR-10** is a sub-group of the Tiny Images dataset. It comprises 60000 32 × 32 color images. Those pictures were named with a standout amongst 10 fundamentally unrelated classes. There are 6000 pictures for every class for 5000 preparations and 1000 testing pictures. (2) **MNIST** is an expansive gathering about written by hand digits. It has a training set of approximately 60,000 samples, and a test set of 10,000 samples. The size of these images is 20 × 20 pixels. (3) ILSVRC2012 is a subset of the abundant hand-labeled ImageNet dataset arranged by the WordNet hierarchy, at which point each bud of the hierarchy is described by large group and thousands of images. The dataset includes 1000 leaf node categories, all of which consist of 1860 nodes. The pretraining models used in the experiments in this paper include VGG16 and ResNet-50. We set the learning rate to 0.01, the training batch to 128, and the training epoch to 100 for both datasets and models. The computing equipment used for the work was four TITAN-RTXs at 3.60 GHz, 32 GB of RAM and 500 GB of hard disk.

The evaluation has two aspects. One is for dynamic reasoning decisions, and the other is for information bottleneck assessment. (1) **Multiple Accumulate Operations (MACs)**. In network computing, multiply-accumulation is a common step to calculate the product of two numbers and add them to the accumulator. One MAC is calculated twice, assuming that both the input channel and the output channel are 1, the kernel size is $k \times k$, the input map size is (H_{in}, W_{in}), and the output map size is (H_{out}, W_{out}). The input channel C_{in} and the output channel C_{out} are considered; then $MACs = C_{in} * K * K * H_{out} * W_{out} * C_{out}$. (2) **The reduction ratio** measures the percentage of reduction in MACs. Let M_{before} be the MAC value before reduction, and M_{after} be the MAC value after reduction. Then Reduction Ratio $= ((M_{before} - M_{after})/M_{after}) * 100\%$. (3) **Accuracy** indicates the number of correct samples divided by the number of all samples. Its value represents the model effect. (4) **A bounding box (Bbox)** is used to quantify how sound attribution methods identify and localize the object of interest. Assuming that the bounding box contains n pixels, we assess how many of the top k pixels in the predicted mask P, where $k \leq n$. We can take the Bbox score by $Bbox = (k/n) \times 100$.

Table 1. Comparisons of MACs, reduction ratio and accuracy for pruning methods with dynamic reasoning decisions of the VGG16 model on the CIFAR10 dataset.

Method	MACs (M)	Reduction ratio (%)	Accuracy (%)
Baseline	313.2	–	93.50
Luo and Wu [28]	156.5	50	93.36
Li and Kadav [29]	206.6	34	93.00
He and Zhang [30]	156.5	50	93.18
Liu and Li [31]	153.4	51	93.31
Lin and Rao [32]	156.5	50	92.65
Gao and Zhao [33]	156.5	50	93.03
Wang and Zhang [27]	155.2	50.4	93.45
DRIB	**156.5**	**50.2**	**93.30**

4.2 Interpretability of Calculation in Dynamic Reasoning Decision

The excessive and complex parameters are the reasons for the inexplicability of deep neural networks. We reduce the calculation of parameters through the dynamic reasoning decision module in this experiment. By comparing some of the related methods mentioned in the SOTA work [27], we find that we can improve 156.7 M in MACs, with the reduction ratio increasing 0.2% and up to 93.30% accuracy compared with the baseline in Table 1. It also has obvious improvement compared with other methods, especially in the logical calculation. Especially in the ResNet50 model as shown in Table 2, on the basis of the reduction ratio reaching 51.4%, the accuracy can reach 93.27%. This is enough to prove that the dynamic reasoning decision is highly understandable and explainable in the calculation.

Table 2. Comparisons of MACs, reduction ratio and accuracy for pruning methods with dynamic reasoning decisions of the ResNet-50 model on the CIFAR10 dataset.

Method	MACs (M)	Reduction ratio (%)	Accuracy (%)
Baseline	125.8	–	92.80
Luo and Wu [28]	62.9	50	91.98
He and Zhang [30]	62.9	50	91.80
He and Kang [34]	65.4	48	92.56
He and Lin [35]	62.9	50	90.20
Wang and Zhang [27]	59.6	52.6	92.57
DRIB	**60.2**	**51.4**	**93.27**

4.3 Explainability of Attribution in the Information Bottleneck

The theory of information bottlenecks measures the effectiveness of the residual feature attribution of the above dynamic reasoning decision module after reducing the amount of computation. $I[Z, Y]$ denotes the amount of information that Z learns about label Y. $I[X, Z]$ indicates the degree of simplification of information in the learning process X. The transition point refers to the point at which information is formed during the transition, and the convergence point is a parameter included after excessive stability. The data in Table 3 for VGG16 and Table 4 show that the degree of information attribution is 82.15% and 88.34%, respectively, and fewer epochs are needed. This result proves that the DRIB model has an excellent effect on the attribution function.

Table 3. The key parameters of the VGG16 model.

	$I[X, Z]$	$I[Z, Y]$	Epochs	Accuracy
Transition point	8.753	2.285	7	45.56
Convergence point	4.673	2.732	69	82.15

Table 4. The key parameters of the ResNet-50 model.

	$I[X, Z]$	$I[Z, Y]$	Epochs	Accuracy
Transition point	9.375	2.376	2	55.47
Convergence point	4.895	2.983	55	88.34

In addition, we quantify the influence of input samples on classification accuracy in the attribution process. As depicted in Fig. 3, the input sample is compressed by the information bottleneck to obtain the mask after attribution. Finally, the mask was covered on the original image to obtain the attributed picture. To further verify the effect of attribution, we view the classification confidence by deleting and inserting pixels in the attribution weight area. We perform 113 significant pixel deletion or insertion operations on each sample, each of which performs 0.9% of the pixel range in the graph. The figure contains three parts: the dog, the cat, and the bird. The horizontal axis of the coordinates represents the number of pixels inserted or deleted in each table. The number of steps is counted a total of 113 times. The vertical axis represents the classified confidence of each step. The blue value is the confidence of the deleted pixels, and the orange value is the confidence of the inserted pixels.

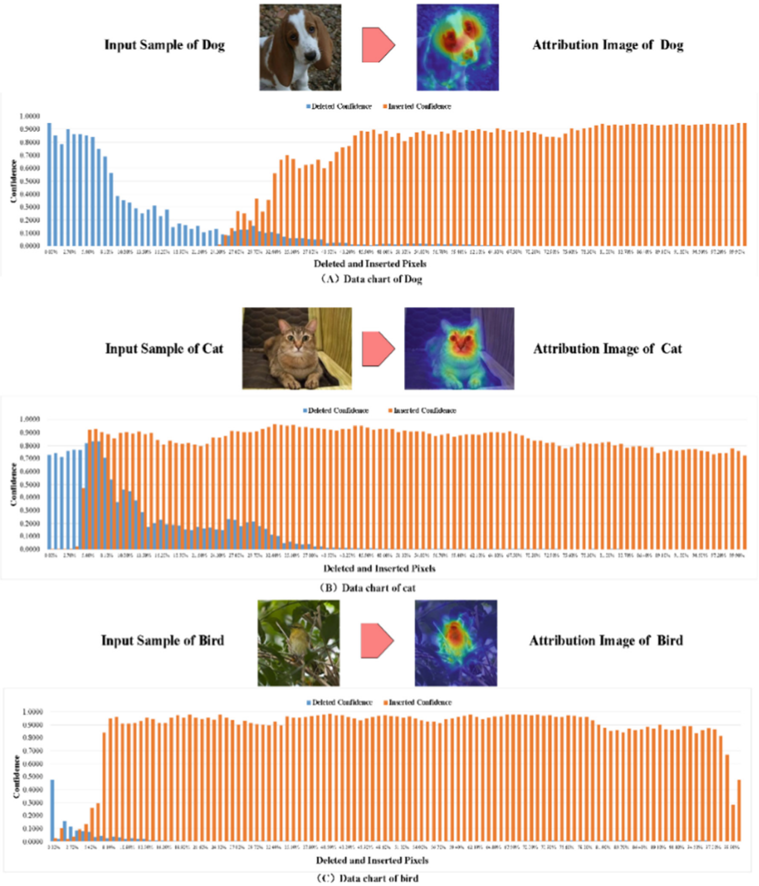
In Fig. 3(A), because the picture of the dog is much larger than that of the bird, we can see from the table that the change of in confidence of the classified dog is more evident than that of the bird. The proportion of cats (B) in the image is similar to that of dogs in (A), so there is little change in the table data. The middle bird (C) environment is more complex than other samples. The table data show that the environment significantly influences the confidence of classification. The results of this experiment can help us to more intuitively understand the application of information bottleneck theory in interpretability.

4.4 Visualization of Understandability

To better validate the effectiveness of our method in feature attribution, we further identify and localize targets through the ILSVRC2012 dataset. Because this dataset possesses borderline annotation. In the process of attribution verification using information bottleneck theory, the feature graphs in the process are randomly selected. In each iteration of the experimental step, the highest ranked value is replaced with a constant. The modified inputs are used for network prediction, and then the degree of decline in the target category scores is measured based on the prediction results. The results are exhibited in Table 5. The DRIB model achieves better performance in VGG-16 by 8 * 8 pixels and ResNet-50 by 8 * 8 pixels. In addition, VGG Bbox and ResNet Bbox are better than the other models. In addition, as demonstrated in Fig. 4, the visualization obtains the feature map attributed by the information bottleneck theory.

Table 5. The decline of the target category score and bounding box score.

Model	Vgg-16 8 * 8	Vgg-16 14 * 14	ResNet-50 8 * 8	ResNet-50 14 * 14	Vgg Bbox	ResNet Bbox
Smooth Grad	43.8%	45.5%	48.5%	50.2%	39.9%	43.9%
Grad-CAM	51.0%	51.7%	53.6%	54.1%	39.9%	46.5%
Guided Grad-CAM	55.5%	57.6%	56.5%	57.7%	41.9%	46.8%
DRIB	**56.2%**	**54.3%**	**57.1%**	**53.1%**	**53.7%**	**55.1%**



（A）Data chart of Dog

（B）Data chart of cat

（C）Data chart of bird

Fig. 3. The influence of some samples' classification confidence after deleting or inserting pixels.

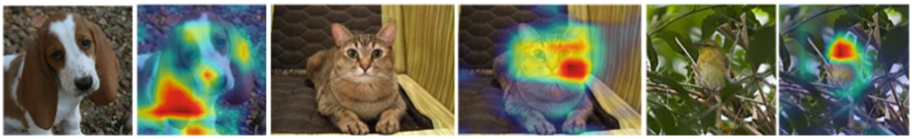


Fig. 4. Visualization of the feature maps attributed by the information bottleneck theory.

5 Conclusion

This paper proposes a new method named the dynamic reasoning and information bottleneck (DRIB). In the method, a novel dynamic reasoning decision algorithm was proposed to reduce multiply accumulate operations and improve the interpretability of the calculation. The information bottleneck was introduced to the DRIB model to verify the attribution correctness of the dynamic reasoning module. Some experiments prove that the dynamic reasoning decision algorithm and information bottleneck theory can be combined with each other. Otherwise, the method provides users with good interpretability and understandability, making deep neural networks trustworthy.

References

1. Kim, Y.J., Bae, J.P., Chung, J.W., et al.: New polyp image classification technique using transfer learning of network-in-network structure in endoscopic images. Sci. Rep. **11**(1), 1–8 (2021)
2. Fan, Q., Zhuo, W., Tang, C.K., et al.: Few-shot object detection with attention-RPN and multi-relation detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4013–4022 (2020)
3. Yadav, A., Vishwakarma, D.K.: Sentiment analysis using deep learning architectures: a review. Artif. Intell. Rev. **53**(6), 4335–4385 (2019). https://doi.org/10.1007/s10462-019-09794-5
4. Wu, M., Parbhoo, S., Hughes, M., et al.: Regional tree regularization for interpretability in deep neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 6413–6421
5. Kubara, K.J., Manczak, B., Dolicki, B., et al.: Towards transparent and explainable attention models. In: ML Reproducibility Challenge 2020 (2021)
6. Misheva, B.H., Osterrieder, J., Hirsa, A., et al.: Explainable AI in credit risk management. arXiv preprint arXiv:2103.00949 (2021)
7. Torrent, N.L., Visani, G., Bagli, E.: PSD2 explainable AI model for credit scoring. arXiv preprint arXiv:2011.10367 (2020)
8. Loquercio, A., Segu, M., Scaramuzza, D.: A general framework for uncertainty estimation in deep learning. IEEE Robot. Autom. Lett. **5**(2), 3153–3160 (2020)
9. Zhang, Q., Cao, R., Shi, F., et al.: Interpreting CNN knowledge via an explanatory graph. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
10. Bau, D., Zhou, B., Khosla, A., et al.: Network dissection: quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6541–6549 (2017)
11. Nguyen, A., Clune, J., Bengio, Y., et al.: Plug & play generative networks: conditional iterative generation of images in latent space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4467–4477 (2017)
12. Bau, D., Zhu, J.Y., Strobelt, H., et al.: Understanding the role of individual units in a deep neural network. Proc. Natl. Acad. Sci. **117**(48), 30071–30078 (2020)
13. Zhou, B., Bau, D., Oliva, A., et al.: Interpreting deep visual representations via network dissection. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2131–2145 (2018)
14. Fong, R., Vedaldi, A.: Net2Vec: quantifying and explaining how concepts are encoded by filters in deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8730–8738 (2018)
15. Zhou, B., Khosla, A., Lapedriza, A., et al.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
16. Selvaraju, R.R., Cogswell, M., Das, A., et al.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
17. Wang, H., Wang, Z., Du, M., et al.: Score-CAM: score-weighted visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 24–25 (2020)

18. Lage, I., Ross, A., Gershman, S.J., et al.: Human-in-the-loop interpretability prior. In: Advances in Neural Information Processing Systems, p. 31 (2018)
19. Subramanian, A., Pruthi, D., Jhamtani, H., et al.: SPINE: Sparse interpretable neural embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
20. Ho, D.: NBDT: neural-backed decision trees. Master's thesis, EECS Department, University of California, Berkeley (2020)
21. Nauta, M., van Bree, R., Seifert, C.: Neural prototype trees for interpretable fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14933–14943 (2021)
22. Fan, F., Wang, G.: Fuzzy logic interpretation of quadratic networks. Neurocomputing **374**, 10–21 (2020)
23. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
24. Zhang, Q., Yang, Y., Ma, H., et al.: Interpreting CNNs via decision trees. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6261–6270 (2019)
25. Shen, W., Guo, Y., Wang, Y., et al.: Deep differentiable random forests for age estimation. IEEE Trans. Pattern Anal. Mach. Intell. **43**(2), 404–419 (2019)
26. Zhang, Q., Wu, Y.N., Zhu, S.C.: Interpretable convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8827–8836 (2018)
27. Wang, Y., Zhang, X., Hu, X., et al.: Dynamic network pruning with interpretable layerwise channel selection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 6299–6306 (2020)
28. Luo, J.H., Wu, J., Lin, W.: ThiNet: a filter level pruning method for deep neural network compression. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5058–5066 (2017)
29. Li, H., Kadav, A., Durdanovic, I., et al.: Pruning filters for efficient ConvNets. arXiv preprint arXiv:1608.08710 (2016)
30. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1389–1397 (2017)
31. Liu, Z., Li, J., Shen, Z., et al.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2736–2744 (2017)
32. Lin, J., Rao, Y., Lu, J., et al.: Runtime neural pruning. In: Advances in Neural Information Processing Systems 30 (2017)
33. Gao, X., Zhao, Y., Dudziak, Ł., et al.: Dynamic channel pruning: feature boosting and suppression. arXiv preprint arXiv:1810.05331 (2018)
34. He, Y., Kang, G., Dong, X., et al.: Soft filter pruning for accelerating deep convolutional neural networks. arXiv preprint arXiv:1808.06866 (2018)
35. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., Han, S.: AMC: AutoML for model compression and acceleration on mobile devices. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 815–832. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_48