Interpret Neural Networks by Identifying Critical Data Routing Paths

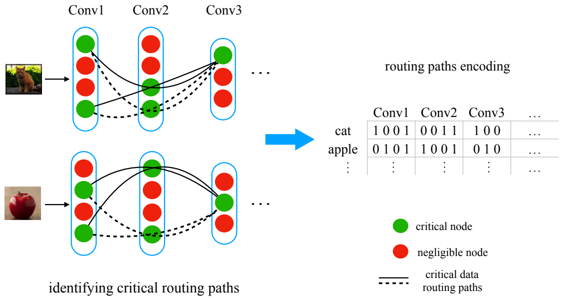
Yulong Wang Hang Su Bo Zhang Xiaolin Hu*
Tsinghua National Lab for Information Science and Technology
Beijing National Research Center for Information Science and Technology, BNRist Lab
Department of Computer Science and Technology, Tsinghua University
{wang-yl15@mails,suhangss@mail,dcszb@mail,xlhu@mail}.tsinghua.edu.cn

Abstract

Interpretability of a deep neural network aims to explain the rationale behind its decisions and enable the users to understand the intelligent agents, which has become an important issue due to its importance in practical applications. To address this issue, we develop a Distillation Guided Routing method, which is a flexible framework to interpret a deep neural network by identifying critical data routing paths and analyzing the functional processing behavior of the corresponding layers. Specifically, we propose to discover the critical nodes on the data routing paths during network inferring prediction for individual input samples by learning associated control gates for each layer's output channel. The routing paths can, therefore, be represented based on the responses of concatenated control gates from all the layers, which reflect the network's semantic selectivity regarding to the input patterns and more detailed functional process across different layer levels. Based on the discoveries, we propose an adversarial sample detection algorithm by learning a classifier to discriminate whether the critical data routing paths are from real or adversarial samples. Experiments demonstrate that our algorithm can effectively achieve high defense rate with minor training overhead.

1. Introduction

With the availability of large-scale databases and recent improvements in deep learning methodologies, deep neural network has become an indispensable tool or even exceeded the human level on an increasing number of complex tasks [12, 25, 31]. In general, most of these algorithms lack the capability to make themselves understandable to users. A machine learning model tends to create nonlinear, non-monotonic and non-polynomial functions that approximate the relationship between variables in a dataset, which makes it highly non-transparent. The opaqueness of their inner working mechanism is recognized as one major drawback in the task-critical applications where the reliance of the model must be guaranteed, such as medical diagnosis [32] or self-driving cars [16].



	Conv1	Conv2	Conv3	...
cat	1 0 0 1	0 0 1 1	1 0 0	...
apple	0 1 0 1	1 0 0 1	0 1 0	...
⋮	⋮	⋮	⋮	...

Figure 1: Overview of our proposed method. **Left**: we develop a Distillation Guided Routing method to identify the critical data routing paths for each input sample. Each layer's output channel is associated with a scalar control gate to decide whether the channel is critical for the decision. The activated channels are termed as critical nodes on the routing paths. **Right**: the learned control gates from all layers are sequentially concatenated to form an encoding feature of routing paths, which can be used to analyze the functional process and dissect the working mechanism of a DNN.

An interpretable machine learning algorithm has the capability to explain or to present in understandable terms to a human [7]. It has attracted an increasing attention to develop methods for visualizing, explaining and interpreting deep learning models [3, 14, 17]. However, there is generally an inherent tension between the performance and interpretability. It may sacrifice accuracy to pursue the interpretability, which is undesirable in settings with critical consequences [10]. In this paper, we focus on the post-hoc interpretability, i.e., given a trained model, our goal is to understand what and how the model achieves this decision by analyzing its working process, which does not damage the performance of a model itself. Some other efforts are also made to provide explanations for each individual pre-

* indicates corresponding author

diction. Representative works include influence functions to trace a model's prediction back to its training data [17], SHAP to determine the most influential features [20] and network dissection to quantify the interpretability of latent representations [3]. These approaches are still far from interpreting the overall functional process of a model.

A lot of efforts have been made to understand the internal representations of deep neural networks through visualization [26, 34, 27], in which the behavior of a DNN can be visualized by sampling image patches or attributing saliency through gradient ascent. Nevertheless, the visualization-based methods generally fail to quantitatively analyze the influence of each component to the decision. Besides, it is non-trivial to understand the internal representations of DNNs due to the sizes involved, which motivates us to compress the redundancy of a DNN, and select the essential components that contribute significantly to the decision.

The black-box property of DNNs also brings out several other defects for the secure application of an algorithm. Recent research has demonstrated that a deep architecture is highly vulnerable to adversarial examples, which are generated by adding small but purposeful modifications [9]. The adversarial samples lead to incorrect outputs while imperceptible to human eyes, which pose security concerns on machine learning systems [23]. It would be a promising direction to defend the adversarial attacking if the decision process is interpretable to human users.

1.1. Our proposal

To resolve the above issues, we introduce a new perspective on interpreting neural network behavior by *identifying the critical data routing paths of each given input and tracing the functional processing behavior of the intermediate layers.* Specifically, we denote the critical nodes on the routing paths as the important channels of intermediate layer's output that if they were suppressed to zeros, the final test performance would deteriorate severely.

To efficiently discover the Critical Data Routing Paths (*CDRPs*), we develop a Distillation Guided Routing (*DGR*) method, which can be applied on all the classical deep neural networks without the need to retrain the whole model from scratch. Specifically, we associate a scalar control gate to each layer's output channel to learn the optimal routing paths for *each individual sample*. Inspired by the idea of knowledge distillation [15], we optimize the control gates by leveraging the criterion that the subnetwork outlined by the CDRPs preserves the knowledge of the original model. We conduct the technique to interpret the deep neural networks on ImageNet dataset with 1,000 categories, including the present popular models of AlexNet [18], VGG [26] and ResNet [12]. Our method largely outperforms other baseline methods while achieving highly sparse and interpretable routing paths.

We further propose a straightforward encoding scheme to represent the critical data routing paths. Specifically, we sequentially concatenate the learned control gates from all layers (see Figure 1). By applying hierarchical clustering and embedding visualization on these routing paths representations, we discover that 1) the intra-layer routing nodes display increasing categorical discriminative ability with ascending layer level, and 2) the whole CDRPs reflect consistent input patterns in intra-class samples , which can help identify complex examples in the dataset. Human evaluations further validate CDRPs are more efficient to capture consistent intra-class similarity than the Feedback Network [5], which uses control gates to model top-down feedback selectivity.

An interpretable technique of deep learning network provides powerful tools to verify and improve the models. In this paper, we ground our proposed algorithms on a major application in robust representation learning and detection of adversarial samples. We discover that the CDRPs of adversarial images diverge from those of real images at intermediate layers and follow the typical routing paths of adversarial target class samples at high-level layers. Based on the above observation, we propose an adversarial sample detection algorithm by learning the binary classifier to discriminate whether the CDRPs are from real or adversarial samples. Experiments demonstrate that our algorithm can effectively detect adversarial images solely based on inconsistency of CDRPs with a few training samples.

In summary, our paper makes the following contributions:

- We propose a novel and flexible frame to interpret neural networks by analyzing the CDRPs identified by the proposed distillation guided routing method. Our method largely outperforms other baseline routing methods while achieving highly sparse routing paths and preserving the performance of the original full model.

- We further propose a straightforward encoding scheme to represent CDRPs, which can be regarded as a new form of activations displaying more detailed function process during network inferring prediction. Our analysis on the new representations reveals the prevalence of consistent and interpretable semantic concepts across nodes on the routing paths.

- We further apply the CDRPs representation on adversarial sample detection problem. Our proposed representation not only effectively detects adversarial images with minor training cost, but also reveals that the model failure is mainly caused by the divergence of CDRPs between adversarial and real images.

2. Methodology

In this section, we introduce our proposed method, which is mainly inspired by model pruning [19, 13, 2]. However in our method, we do not change the original weights of pretrained neural network, and only identify important layer's output channels on the routing paths, by associating a scalar control gate with each layer's output channel. The control gates are learned to find the optimal routing decision in the network, while the final prediction remains unchanged. The critical data routing paths are consequentially identified by analyzing the response of each control gate, yielding a network-based representation for each sample.

2.1. Channel-wise Control Gates

In this section, we identify the nodes on the CDRPs by distilling the subnetwork outlined by routing paths without which the performance degenerates severely. To this end, we introduce the control gate of scalar value, λ, associated with each layer's output channel. During inference forward pass, a group of control gates $\boldsymbol{\lambda}_k$ will be multiplied to the k-th layer's output channel-wise, resulting in the actual routing nodes. Each layer's routing nodes are connected to form the routing paths. The problem of identifying the critical data routing paths reduces to optimize $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \cdots, \boldsymbol{\lambda}_K\}$, which are all the control gates for the K layers in the network. Figure 2 shows the above concepts.

For valid and reasonable critical data routing paths, we consider λ's should satisfy these two conditions: (1) λ's should be non-negative. From the functional definition of control gate, λ should only suppress or amplify the output channel activations. The negative value of λ would negate the original output activations in the network, which drastically changes the activations distribution and introduces unexpected influence during interpretation of original model, and (2) λ's should be sparse and most of them are close to zeros. This accords with common claims that sparse models [29, 30] with disentangled attributes are more interpretable than dense models.

2.2. Distillation Guided Routing

To efficiently find the control gates in a pretrained network $f_\theta(\cdot)$ for an input image $\boldsymbol{x}$, we develop Distillation Guided Routing(*DGR*) method, which is inspired by knowledge distillation technique [15] to transfer the original full model's knowledge to the new subnetwork outlined by the routing paths. Specifically, the optimization objective for all the control gates $\boldsymbol{\Lambda}$ is

$$\begin{aligned} \min_{\boldsymbol{\Lambda}} \quad & \mathcal{L}(f_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x}; \boldsymbol{\Lambda})) + \gamma \sum_k |\boldsymbol{\lambda}_k|_1 \\ s.t. \quad & \boldsymbol{\lambda}_k \succeq 0, k = 1, 2, \cdots, K, \end{aligned} \tag{1}$$

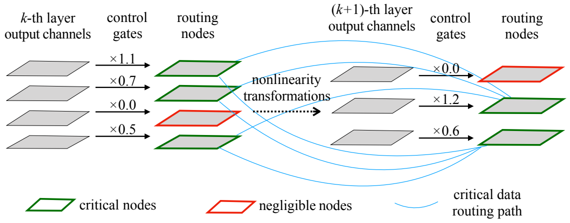

k-th layer output channels
control gates
routing nodes
(k+1)-th layer output channels
control gates
routing nodes
×1.1
×0.7
×0.0
×0.5
nonlinearity transformations
×0.0
×1.2
×0.6
critical nodes
negligible nodes
critical data routing path

Figure 2: The control gates are multiplied to the layer's output channel-wise, resulting in the actual routing nodes. In this demonstration, we identify those nodes whose responses of the control gates are larger than 0. The layerwise routing nodes are linked together to compose the routing paths.

where $\mathcal{L}$ is the cross entropy loss between the original full model's prediction probability $f_\theta(\boldsymbol{x}) = [p_1, p_2, \cdots, p_m]$ and the new prediction probability $f_\theta(\boldsymbol{x}; \boldsymbol{\Lambda}) = [q_1, q_2, \cdots, q_m]$, which is $\mathcal{L} = \sum_i^m -p_i \log q_i$, where m is the category number, and γ is the balanced parameter.

Note that here we do not need the ground-truth label for the $\boldsymbol{x}$ during optimization. The learned control gates try to make the prediction consistent with that of the full model, even if the original prediction is incorrect.

To encourage $\boldsymbol{\lambda}$ to be sparse, we use ℓ_1 norm as the sparsity penalty function. The subgradient descent method can be adopted to optimize the objective, and all the training procedure is similar to the usual stochastic gradient descent (SGD) method.

2.3. Routing Paths Representation

Denote $\boldsymbol{\Lambda}^* = \{\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*, \cdots, \boldsymbol{\lambda}_K^*\}$ as the optimized control gates, and the corresponding identified CDRPs can be represented by

$$\boldsymbol{v} = \text{concatenate}([\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*, \cdots, \boldsymbol{\lambda}_K^*]). \tag{2}$$

To obtain the actual critical routing paths selection from the representation, the routing nodes can be selected based on a binary mask, generated by thresholding $\boldsymbol{v}$ with some given value. However, the CDRPs representation contains more abundant information than the binary mask, since the former weighs routing nodes with different importance coefficients, reflecting the network's semantic selectivity for the input patterns. Moreover, the CDRPs representation can be regarded as a new form of *activations*, compared to the usual high-level feature extractor's responses. The new representation displays more detailed functional process during network inferring prediction than the mere final result. Compared to other methods which probe the network's intermediate responses [33, 34, 1], our method results in a succinct and effective representation. More results are presented in Section 4 demonstrating the close relationships of CRDP representation with input semantic patterns and model's functional process.

2.4. Implementation Details

Before optimizing the objective in Equation (1), all control gates in $\mathbf{\Lambda}$ are initialized with 1, which activates all the nodes. After calculating the original full model's prediction probability for the given input data. the gradients for control gates are computed by

$$\frac{\partial Loss}{\partial \mathbf{\Lambda}} = \frac{\partial \mathcal{L}}{\partial \mathbf{\Lambda}} + \gamma * \mathrm{sign}(\mathbf{\Lambda}), \tag{3}$$

which are used for performing stochastic gradient descent on control gates.

As for the implementation, we perform SGD on the same input $\boldsymbol{x}$ for $T = 30$ iterations, with learning rate of 0.1, momentum of 0.9 and no weight decay. After finishing the iterations, the optimized CDRPs representation $\boldsymbol{v}$ is formed by concatenating $\mathbf{\Lambda}$, which can result in the lowest loss value while retaining the exact same top-1 prediction with the original model. If no routing paths can satisfy the condition, we denote the CDRPs as the original model's all plausible routing paths, which means all control gates in $\boldsymbol{\lambda}$ are reset to 1.

For the regularization term, we set $\gamma = 0.05$, which reaches a balance between performance and sparsity in our experiments. Though there is no upper limit for λ, for numerical stability consideration, we constrain λ's to be in $[0, 10]$ after each iteration, which is a quite loose bound. We allow λ larger than 1 to compensate the distribution variation of output channels after multiplied by control gates. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Distillation Guided Routing

```
Require: Input $\boldsymbol{x}$, pretrained network $f_\theta(\cdot)$, control gates
    $\mathbf{\Lambda}$ initialized with 1, balanced parameter $\gamma$. Max itera-
    tions $T$, SGD optimizer
Ensure: identified CDRPs representation $\boldsymbol{v}$

original prediction class $i \leftarrow \arg\max f_\theta(\boldsymbol{x})$
for $t \leftarrow 1$ to $T$ do
    compute loss $cur\_loss$ by Equation (1)
    compute control gates gradients $\mathbf{\Lambda}$ by Equation (3)
    update $\mathbf{\Lambda}$ by SGD optimizer and clip $\mathbf{\Lambda}$ to be non-negative
    new prediction class $j \leftarrow \arg\max f_\theta(\boldsymbol{x}; \mathbf{\Lambda})$
    if $i = j$ then                    ▷ keep the prediction same
        if $cur\_loss$ is minimum then
            $\boldsymbol{v} \leftarrow \mathrm{concatenate}(\mathbf{\Lambda})$
        end if
    end if
end for
```

3. Adversarial Samples Detection

Adversarial samples [28, 9], which are generated by adding indistinguishable noise to human eyes onto the real images, but are misclassified by the deep architecture become an intriguing property and pose concerns on the robustness of neural network. In this section, we utilize the identified CDRPs representation to analyze the adversary phenomenon.

For a given real image $\boldsymbol{x}$ and corresponding adversarial image $\hat{\boldsymbol{x}}$, the CDRPs for each image can be identified as $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$. Since the difference between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is small, it is expected that the CDRPs $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ on the low-level layers are similar. However, the drastic change in the final prediction should be attributed to the increasing divergence between $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ at high-level layers. Based on the above reasoning, adversarial sample can be detected by recognizing the CDRPs difference through a binary classifier f, which optimizes the following objective as

$$\min_{f} \quad \sum_{i} L(f(\boldsymbol{v}_i), y_i) + L(f(\hat{\boldsymbol{v}}_i), \hat{y}_i), \tag{4}$$

where L is loss function, $y_i = 1$ for real images and $\hat{y}_i = 0$ for adversarial images. yulongThe loss function in Equation (4) can be any scoring rules to encourage the binary classifier to distinguish real and adversarial samples. For adaboost classifier, the loss function is Huber loss. For gradient boosting classifier, the loss function is squared error loss. This framework is very general and flexible. Compared to feature-inconsistency detection method [6], our method requires much more succinct feature representation and less computation overhead. In Section 4.3, we validate the above reasoning by comparing the correlation coefficients of CDRPs between real image and its corresponding adversarial image layer-wise. Experiments also demonstrate that our CDRPs-based adversarial sample detection algorithm is effective with a few training samples.

4. Experiments

In this section, we first implement the quantitative analysis on the performance of the critical data routing paths, and then elaborate on the semantic concepts emergence of the nodes in the paths; finally, we demonstrate that our proposed method is effective in detecting the adversarial samples. Since our method focus on *post-hoc* prediction interpretation for each single input, we use ImageNet validation dataset with 50,000 images and VGG-16 network [26] for all the experiments. More results for ResNet [12] and AlexNet [18] are provided in the supplementary material.

4.1. Quantitative Analysis

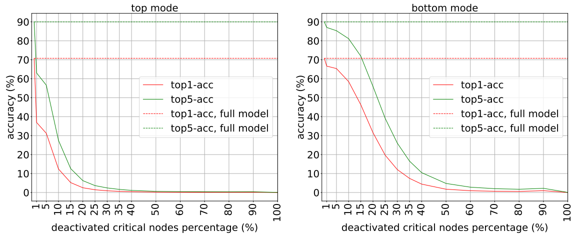
In this section, we report classification accuracy results of the subnetwork outlined by identified critical data routing paths. To demonstrate our method's effectiveness, we compare our method with other two baseline methods, which are 1) *Weight Routing*, which decides the control gates solely based on weights norm, and 2) *Activation Routing*, which decides the control gates based on layer's output activations magnitude. Instead of directly selecting routing nodes by thresholding on weights norm or activations norm, we adopt a greedy strategy to iteratively prune the weights or output channels, which result in *Adaptive Weight Routing* (AWR) and *Adaptive Activation Routing* (AAR) policies. In each iteration, the remaining weights or output channels are ranked in ℓ_1-norm order and 2% of them with least norms are pruned (not based on threshold but on ranking order). The pruning iteration is halted as long as the top-1 prediction is altered. The final CDRPs selection criterion is exactly the same with the description in 2.4

Table 1 summarizes the performance of our method in terms of top-1 and top-5 accuracy and sparsity. All the three methods achieve the same top-1 accuracy due to the selection criterion. However, our method achieves the highest top-5 accuracy compared to other two baseline methods, and only suffers about 1.4% top-5 accuracy degradation compared to the full model. We also compare the resulting routing paths sparsity of each method. We define the sparsity as the ratio of selected critical routing nodes in the total nodes. More sparse routing paths indicate that less redundant and irrelevant nodes are included. Our method achieves far more sparse routing paths compared to the baseline methods. We attribute this to the distillation procedure to keep performance comparable, and ℓ_1 norm regularization to encourage sparsity.

Table 1: Adaptive routing methods comparison with same top-1 prediction requirement. For sparsity, lower is better

Methods	Top-1	Top-5	Sparsity
VGG-16 Full Model (%)	70.79	89.99	100.00
AWR (%)	70.79	85.42	89.23 ± 2.52
AAR (%)	70.79	84.85	88.77 ± 0.68
DGR (Ours) (%)	70.79	**88.54**	**13.51 ± 4.19**

Ablation Study We also further validate the CDRPs through ablation study. The procedure is to partially deactivate the critical nodes on the identified CDRPs in the original full model, while keeping other non-critical nodes unchanged. The critical nodes' corresponding control gate values weigh their importance in the CDRPs. We experiment with two schemes to deactivate the critical nodes, which are (1) *Top Mode* that deactivates the critical nodes with larger control gates values first, and (2) *Bottom Mode* that deactivates the critical nodes with smaller control gates values first. Figure 3a and Figure 3b show the model accuracy degradation with different fractions of critical nodes being deactivated. In *Top Mode*, when only 1% most critical nodes are deactivated in the original full model, the top-1 and top-5 accuracy drop 33.84% and 26.92%. Note that the number of these most critical nodes only accounts for 0.13% of the total nodes amount. When the nodes on CDRPs are completely pruned out in the network, the model performance deteriorates severely, reaching nearly zero. Through our ablation study, we validate the CDRPs identified by our method are effective.



(a) (b)

Figure 3: The accuracy degradation when critical nodes are deactivated in the original full model, with (a) *Top Mode* and (b) *Bottom Mode*. Only small fractions of critical nodes being deactivated will lead severe performance degradation, which validates the effectiveness of the identified CDRPs obtained by our method.

4.2. Semantic Concepts Emerge in CDRPs

Functional process of intra-layer routing nodes. In this section, we want to explore the intermediate layer's functional process through the lens of intra-layer routing nodes. We regard all the individual critical nodes in a certain layer composing the intra-layer routing nodes. The encoding representation is simply the optimized control gates $\boldsymbol{\lambda}_k^*$ for the k-th layer. We use t-SNE [21] method to display features in 2D embedding. Figure 4 shows 5 typical convolutional layers in VGG-16 network. Each point on the embedding stands for a single image. Each class consists of 50 validations images, which are painted in the same color according to their ground-truth label. From the figure, we can discover the degree of embedding discriminative ability increases in ascending layers. For high-level layers, like 'Conv4_3' layer has already reached a level of classification ability from the perspective of learned control gates. This implies that the intermediate layers have reached a certain level of classification ability.

To further validate the routing paths' discriminative ability, we apply K-means and agglomerative clustering on each layer's optimized control gates, and measure whether clustering separations of the data are similar to ground truth set of classes in homogeneity score, completeness score, and V-measure score [24]. The value close to 1.0 indicates better

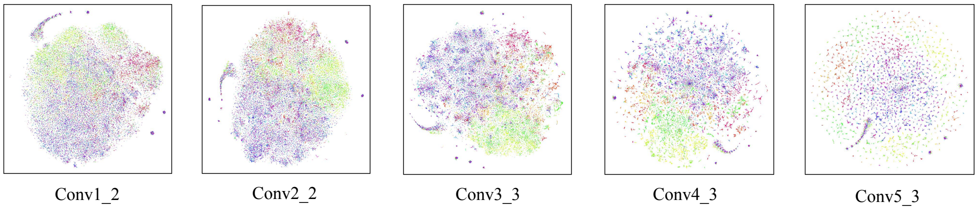
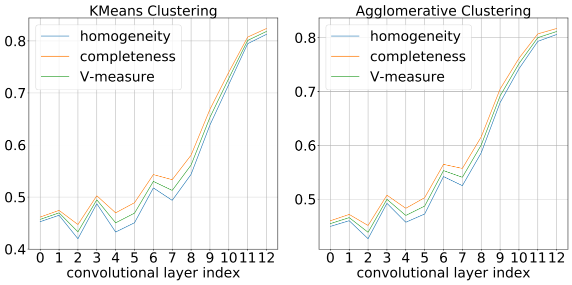


Figure 4: t-SNE 2D embedding of 50,000 ImageNet validation images' intra-layer routing nodes representations on 5 typical convolutional layers in VGG-16 network. Each point stands for a single image. Points with same ground-truth labels are painted in the same color for visual effect. From the figure, we can discover the degree of embedding discriminative ability is increasing with ascending the layer level.



(a) (b)

Figure 5: Different clustering consistency evaluation score for K-means clustering and agglomerative clustering results on the intra-layer routing nodes of different layers. The layer indexes correspond to 13 convolution layers. All the metrics show a common increasing trend, which indicates intra-layer routing nodes of higher level layers have stronger correspondence to category semantic concepts.

match. Figure 5a and 5b display the clustering consistency scores for different convolutional layers. As the layer index ascending, different clustering evaluation scores show a common increasing trend, which indicates the learned control gates of higher level layers have a stronger relationship with corresponding category semantics.

Intra-class sample clustering The critical data routing paths not only reflect the functional process of intermediate layers in the network, but also reflect the input data layout patterns. Figure 6 and 7 show the agglomerative clustering results on the intra-class samples using the whole CDRPs representation. We can discover that the clustering result corresponds to input layout patterns strongly. For example in Figure 6, for the class 'Tinca', we find three typical clusters, which first consists of the lateral view of tinca in the horizontal direction, and second consists of anglers holding the tech in the squat position. The third cluster mainly consists of samples hard to classify, in which techs are not in the regular position or size. Particularly, there is a rotated image in the third cluster, which results in a drastic change in the CDRPs. Figure 7 also shows similar pattern. These results show that the identified CDRPs reflect input patterns, and help to find out hard examples or complex samples in the dataset.

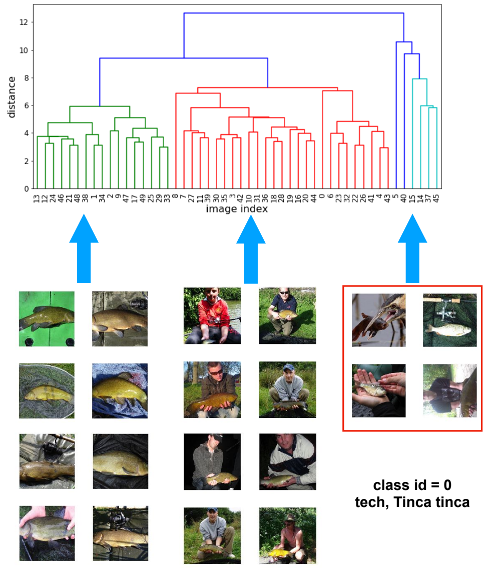


Figure 6: Intra-class sample clustering based on the whole CDRP: The first cluster shows the lateral view of tinca in the horizontal direction, and the second cluster shows anglers holding the tech in the squat position. Red bounding box indicates samples hard to classify, which even include an image rotated to horizontal direction.

Human evaluation & comparison to the Feedback network The Feedback network [5] is similar to our method, in which top-down feedback is learned by control gates on every neurons across spatial and channel-wise dimensions. However, this leads to much larger dimensional representation of Feedback network (about 13M for VGG16) com-

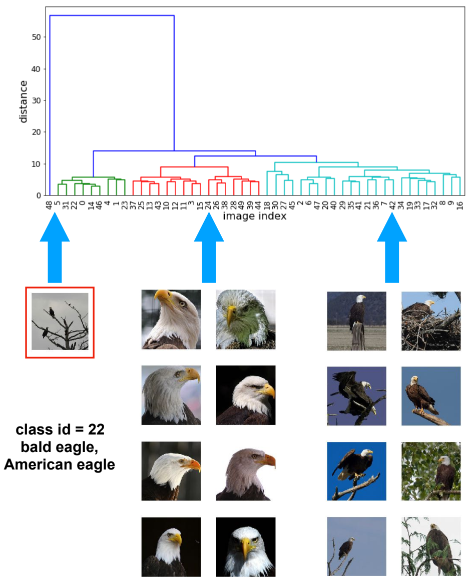


Figure 7: Intra-class sample clustering based on the whole CDRP: The first cluster includes a single hard sample, which two bald eagles perch on the distant tree. The second cluster mainly focuses on bald eagle head, and the third cluster consists of a single eagle perching with clear background contrast.

pared to CDRPs representation (about 4K for VGG16). Furthermore we conduct quantitative comparison to validate that CDRPs capture intra-class variation more consistent and interpretable than naive network activations (`pool5` activation in VGG16) and the Feedback network control gates (the spatial mean of each channel), which meets the *Explanation Continuity* [22] requirement for a good explanation method. Specifically, we randomly select 100 classes from ImageNet, and use corresponding features to perform agglomerative clustering. We then ask workers on Amazon Mechanical Turk (AMT) to identify which kind of image partitions showing more intra-cluster appearance consistency. In every round, we show images of the top 3 clusters. Each class partition comparison is evaluated by 4 different workers (400 workers in total).

Table 2 summarizes the results. First, the intra-class similarity captured by CDRPs is more significant than that of network activations. Second, CDRPs capture intra-class sample similarity slightly better than the Feedback network control gates representation. However, considering the large dimension and explicitly learned spatial selectivity of Feedback representation, CDRPs are more efficient in capturing interpretable concepts.

4.3. Adversarial Sample Detection

CDRPs divergence between real and adversarial image In this section, we analyze the DNN's adversarial phenomenon by utilizing the whole CDRPs representation. The proposed approach is to compare the correlation coefficients of CDRPs between real image and its adversarial image layer-wise.

Table 2: Human evaluation on the intra-class sample similarity captured by CDRPs and network activations. Higher percentage indicates the partition is more favorable and interpretable by the subjects.

Whether the partition is more consistent	CDRPs	`pool5` Activations	No Difference
percentage	**52%**	22%	26%

Whether the partition is more consistent	CDRPs	Feedback Weights	No difference
percentage	**38%**	35%	27%

To generate target adversarial image for a given input $\boldsymbol{x}$ and the target class y^*, we use iterative Fast Gradient Sign Method (FGSM) [9] to generate adversarial image as

$$\boldsymbol{x}_t = \text{clip}(\boldsymbol{x}_{t-1} - \epsilon{\cdot}\text{sign}(\nabla_{\boldsymbol{x}_{t-1}}\mathcal{L}(f_\theta(\boldsymbol{x}_{t-1}), y^*))), \quad (5)$$

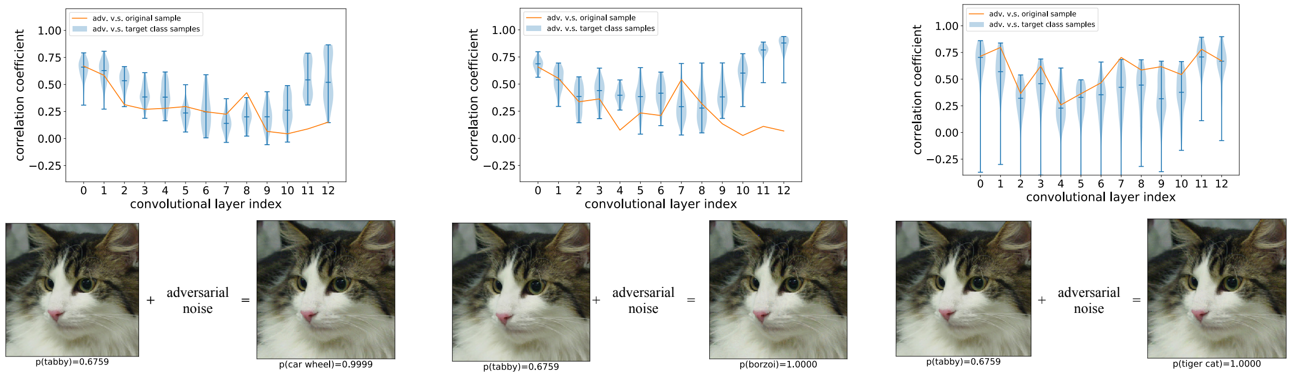
where $\boldsymbol{x}_0$ is initialized with the original image $\boldsymbol{x}$; $\mathcal{L}$ is the cross entropy loss, $f_\theta(\boldsymbol{x}_{t-1})$ is the network prediction with current input $\boldsymbol{x}_{t-1}$, clip$(\cdot)$ constrains new perturbed input to be in the range of pixel values. We also use the similar technique to generate non-target adversarial image by

$$\boldsymbol{x}_t = \text{clip}(\boldsymbol{x}_{t-1} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}_{t-1}}\mathcal{L}(f_\theta(\boldsymbol{x}_{t-1}), y))), \quad (6)$$

where y is the ground-truth label of original input image $\boldsymbol{x}$. In our experiment, we set $\epsilon = 0.01$ and achieve final adversarial image after 10 iterations.

Figure 8 summarizes the results. In Figure 8a, we show that with ascending layer level, the CDRPs of adversarial image diverge from the original image's routing paths and follow similar routing paths with those of target class images. The similar trend is found in Figure 8b, which shows the situation of non-target adversarial attacking. However, when the adversarial target class is much semantically-closer to the original image class ('tiger cat' v.s. 'tabby cat' compared to 'car wheel' v.s. 'tabby cat'), the resulting divergence between routing paths is not obvious and even indistinguishable because of the overlap of CDRPs between original class and target class. This phenomenon validates the aforementioned conclusion that adversarial images follow the typical routing paths of target class at high-level layers, leading to adversary consequences.

Adversarial sample detection Based on the above observation, we propose an adversarial sample detection scheme by learning the binary classifier to discriminate whether the CDRPs are from real or adversarial samples. Since



(a) Target adversarial attacking with target class: ‘car wheel’

(b) Non-target adversarial attacking resulting in implicit target class: ‘borzoi’

(c) Target adversarial attacking with semantic-closer target class: ‘tiger cat’

Figure 8: Layerwise correlation coefficients of CDRPs between adversarial images and original class/target class image. In the upper part of each sub-figure, the correlation coefficients between adversarial image’s routing paths and original image’s routing paths are plotted in orange color. The violinplot summarizes the correlation coefficients of CDRPs between each of 50 target class images and adversarial image. (a) and (b) show that with ascending the layer level, the CDRPs of adversarial image diverge from the original image’s routing paths and follow similar routing paths with those of target class images. However, when target class is semantic-closer to original class, the divergence between routing paths is not obvious

Table 3: The Area-Under-Curve (AUC) score for different binary classifier on adversarial detection by discriminating CDRPs of real and adversarial image. Higher is better.

Num. of training samples	1	5	10
random forest	0.879	0.894	0.904
adaboost	0.887	0.905	0.910
gradient boosting	0.905	0.919	0.915

most non-target adversarial samples result in semantic-closer class with original class, and from the observation we conclude that the CDRPs of semantic-close samples are difficult to discriminate, we focus on target adversarial sample detection problem. In our experiment, we randomly select 1, 5 or 10 images from each class in the ImageNet training dataset to organize three different scales training datasets. The test dataset remains the same, which is collected by selecting 1 image from each class in the ImageNet validation dataset. Each sample is used to generate an adversarial sample by Equation (5). The adversarial target classes are from a random permutation of original classes. We experiment with three classifiers, random forest [4], adaboost [11] and gradient boosting [8]. Each experiment is run five times independently.

Table 3 summarizes the results. Each method outperforms the feature-inconsistency method [6] by a large margin, which reports 0.847 AUC score. Moreover, due to the succinct representation of CDRPs, these methods require less computation overhead. Our results demonstrate that without complicated algorithm, the adversarial attacking can be defended based on the discriminative CDRPs representation.

5. Conclusion

In this paper, we investigate the topic of neural network interpretability from a new perspective by identifying the critical data routing paths during network inferring prediction. We propose a Distillation Guided Routing method, which is a flexible and general framework to efficiently learn the control gates associated with each output channel. By thorough analysis, we find semantic concepts contained in the CDRPs. First, the discriminative ability of intra-layer routing nodes is increasing with ascending layer level. Second, the whole CDRPs reflect intra-class samples layout patterns, which can help identify hard examples in the dataset. To improve the robustness of neural network against adversarial attacking, we propose a novel adversarial sample detection method based on the discrimination on CDRPs of real and adversarial images. Results show that our method can reach quite high defense success rate due to the property that CDRPs of adversarial image diverge at intermediate level layers with those of real image and follow the typical routing paths of adversarial target class samples at high-level layers. Future work should explore the underlying principle of critical data routing paths emergence.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61571261, 61332007, 61621136008, 61620106010 and U1611461, and partially funded by Microsoft Research Asia and Tsinghua-Intel Joint Research Institute.

References

[1] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv:1610.01644*, 2016.

[2] S. Anwar, K. Hwang, and W. Sung. Structured pruning of deep convolutional neural networks. *JETC*, 13(3):32, 2017.

[3] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017.

[4] L. Breiman. Random forests. *Machine learning*, 2001.

[5] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, 2015.

[6] Y. Dong, H. Su, J. Zhu, and F. Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv:1708.05493*, 2017.

[7] F. Doshi-Velez and B. Kim. A roadmap for a rigorous science of interpretability. *arXiv:1702.08608*, 2017.

[8] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 2001.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014.

[10] D. Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2017.

[11] T. Hastie, S. Rosset, J. Zhu, and H. Zou. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *ICCV*, 2017.

[14] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *ECCV*, 2016.

[15] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[16] J. Kim and J. Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *CVPR*, 2017.

[17] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

[20] S. Lundberg and S.-I. Lee. An unexpected unity among methods for interpreting model predictions. *arXiv:1611.07478*, 2016.

[21] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008.

[22] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[23] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[24] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, 2007.

[25] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:1312.6034*, 2013.

[27] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

[29] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[30] S. Wisdom, T. Powers, J. Pitton, and L. Atlas. Interpretable recurrent neural networks using sequential sparse recovery. *arXiv:1611.07252*, 2016.

[31] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016.

[32] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang. Mdnet: A semantically and visually interpretable medical image diagnosis network. In *CVPR*, 2017.

[33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv:1412.6856*, 2014.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.