Information Sciences 547 (2021) 568–578

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Understanding adversarial robustness via critical attacking route

Tianlin Li [a,1], Aishan Liu [a,1], Xianglong Liu [a,b,*], Yitao Xu [a], Chongzhi Zhang [a], Xiaofei Xie [c]

[a] *State Key Lab of Software Development Environment, Beihang University, Beijing, China*
[b] *Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, Beijing, China*
[c] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*

ARTICLE INFO

Article history:
Received 22 April 2020
Received in revised form 20 July 2020
Accepted 13 August 2020
Available online 29 August 2020

Keywords:
Critical attacking route
Adversarial robustness
Model interpretation

ABSTRACT

Deep neural networks (DNNs) are vulnerable to adversarial examples which are generated by inputs with imperceptible perturbations. Understanding adversarial robustness of DNNs has become an important issue, which would for certain result in better practical deep learning applications. To address this issue, we try to explain adversarial robustness for deep models from a new perspective of critical attacking route, which is computed by a gradient-based influence propagation strategy. Similar to rumor spreading in social networks, we believe that adversarial noises are amplified and propagated through the critical attacking route. By exploiting neurons' influences layer by layer, we compose the critical attacking route with neurons that make the highest contributions towards model decision. In this paper, we first draw the close connection between adversarial robustness and critical attacking route, as the route makes the most non-trivial contributions to model predictions in the adversarial setting. By constraining the propagation process and node behaviors on this route, we could weaken the noise propagation and improve model robustness. Also, we find that critical attacking neurons are useful to evaluate sample adversarial hardness that images with higher stimulus are easier to be perturbed into adversarial examples.

1. Introduction

Deep learning is progressing at an astounding rate with a wide range of real-world applications, such as computer vision [13], speech recognition [22] and natural language processing [31]. Despite these successful applications, the emergence of adversarial examples [32,8], images containing perturbations imperceptible to human but misleading to DNNs, poses potential security threats to deep learning systems especially practical applications, e.g., face recognition system, automatic check-out, and auto driving [18,14,7,19]. In the past years, great efforts have been devoted to adversarial attack [8,2,27,17,26,3,34], adversarial defense [8,15,29,16], and adversarial example detection [11,21,9,6].

Though challenging to deep learning, from another point of view, understanding and interpreting adversarial examples provide insights into the weaknesses and blind-spots of models, which in turn offers us clues to build robust deep learning

* Corresponding author at: State Key Lab of Software Development Environment, Beihang University, Beijing, China.
E-mail address: xlliu@nlsde.buaa.edu.cn (X. Liu).
[1] Equal contributions.

https://doi.org/10.1016/j.ins.2020.08.043
0020-0255/© 2020 The Author(s). Published by Elsevier Inc.

models. Several studies have been proposed to understand model robustness towards adversarial noises from different perspectives [38,12,5].

In social networks, rumors spread hastily between nodes through connections, which may present massive social threats. Nodes delivering higher information capacities are more critical than others, which are much likely to convey rumors and be contained in critical routes. To effectively restrain rumor spreading, immunization strategies have been carried out to spot and block critical routes within the graph [36]. Similar to rumor spreading in social networks, small residual adversarial perturbations are progressively amplified and propagated from bottom to top layers, which causes incorrect model behaviors. With above observations, a question emerges: *Is it possible to find a similar neuron-wise attacking route to stop adversarial noise propagation in DNNs?*

In this paper, we propose a gradient-based influence propagation strategy to get critical attacking neurons, which would further help to compose a neuron-wise critical attacking route for a neural network. In sum, our paper makes the following contributions:

- We propose a gradient-based influence layer-wise propagation method to extract critical attacking route at multiple levels, including instance level and model level. The critical attacking route is responsible for conveying both adversarial noises and important semantic information in a deep learning model. This route can be beneficial to understand model behaviors in adversarial setting.
- The critical attacking route reveals some adversarial defects of a deep model. Thus, we can directly improve the robustness against adversarial examples and corruption by constraining and correcting the behaviors of neurons within the critical attacking route.
- Further, based on the critical attacking route, we propose a metric to evaluate the adversarial hardness of samples, which measures the intensity of adversarial perturbations we need to apply on a sample to fool a model. This metric is promising since it provides us an opportunity to judge how hard it is for an image to be perturbed into an adversarial example.

2. Related work

Adversarial robustness. Recently, several studies have been proposed to understand model robustness towards adversarial examples from different perspectives. Zhang et al. [38] believed that adversarially trained models gain robustness by learning more shape-biased representations. Dong et al. [5] re-examined the internal representations of DNNs using adversarial examples and further improved their interpretability with an adversarial training scheme. Meanwhile, Xu et al. [35] attempted to explore the weakness of models in adversarial setting by analyzing the effects of different regions within a specific image. Interestingly, Ilyas et al. [12] held the opinion that adversarial examples are parts of the features rather than bugs for DNNs. They believed that training models with robust features would increase adversarial robustness. Recently, Zhang et al. [37] introduced the concept of neuron sensitivity and blamed the adversarial vulnerability of deep models on the sensitive neurons.

Model data route. From the view of data route, several works have been devoted to interpreting deep neural models. Wang et al. [33] developed a Distillation Guided Routing (DGR) method to extract a neuron-wise data route by learning associated control gates for each layer's output channel. They further used the route to detect adversarial examples. Besides, Qiu et al. [24] tried to extract paths through network profiling. Synapse-wise critical route for each class was extracted and used to defend adversarial examples.

Previous studies mainly focused on semantic information data route for deep models. In contrast to them, we mainly concentrate on the perturbations propagation process and model behaviors in adversarial setting. We propose a very simple but effective method to calculate the critical attacking route, which could be used to understand adversarial robustness of deep neural networks (see Fig. 1).

3. Critical attacking route

In social networks, nodes delivering more information are more likely to make huge impacts to the whole graph when encountering small changes. Motivated by the fact, this paper proposes the concept of *Critical Attacking Route*, which is a neuron-wise attacking route by which most of the adversarial perturbations are propagated and amplified.

To obtain the critical attacking route, we propose a gradient-based influence propagation strategy. Neurons with higher gradients to the next layer are more likely to make higher contributions to the hidden representations at the next layer. We name these neurons as critical attacking neurons. The critical attacking route is composed of these critical attacking neurons and all connections between them. Thus, we can fairly believe, the route composed by these critical attacking neurons is more critical and more likely to make huge changes to the final prediction when encountering small perturbations. In other words, neurons with higher contributions to the model predictions are more skeptical to be involved in the critical attacking route, which is responsible for the model vulnerabilities against adversarial perturbations. Thus, it's intuitive for us to calculate the influence between neurons to figure out the critical attacking neurons and the critical attacking route. Obviously, gradients can help us to evaluate the influence of neurons between layers.

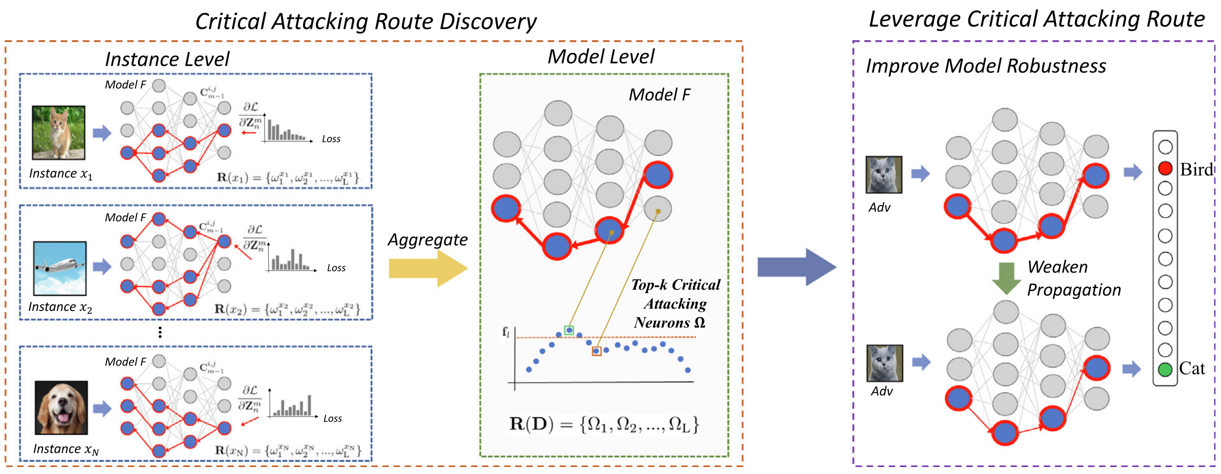

Critical Attacking Route Discovery
Instance Level
Model Level
Leverage Critical Attacking Route
Improve Model Robustness
Model F
Instance x_1
Instance x_2
Instance x_N
Loss
Aggregate
Top-k Critical Attacking Neurons Ω
$\mathbf{R}(\mathbf{D}) = \{\Omega_1, \Omega_2, ..., \Omega_L\}$
Adv
Bird
Cat
Weaken Propagation

Fig. 1. Our framework of computing instance-level critical attacking routes and model-level critical attacking route. Model robustness can be improved by weakening the propagation process.

We first extract critical attacking routes per image via gradients computed backward. After extracting instance-level critical attacking routes for each image, we then aggregate instance-level critical attacking routes into the model-level route.

3.1. Preliminaries

Given a dataset $\mathbf{D}$ with data sample $x \in \mathcal{X}$ and label $y \in \mathcal{Y}$, the deep supervised learning model tries to learn a mapping or classification function $F : \mathcal{X} \rightarrow \mathcal{Y}$. The model F consists of L serial layers. For the l-th layer F_l, where $l = 1, \ldots, \mathrm{L}$, it contains a neuron set. We use the superscript m to denote the m-th neuron satisfying $F_l^m \in F_l$. The output Z_l^m of one neuron F_l^m is equivalent to the m-th channel of the feature map in layer l. This paper chooses the popular deep convolutional neural networks (CNNs) for image classification task.

With imperceptible perturbations, an adversarial example [32] x' looks similar to its clean counterpart x, but are able to mislead model F:

$$F(x') \neq y \quad s.t. \quad \|x - x'\| < \epsilon.$$

3.2. Compute the influence between neurons

To build the critical attacking routes, we pick up critical attacking neurons in each layer and connect them together. Obviously, the key challenge is calculating the influence between neurons of two close layers. The problem can be efficiently addressed by computing the gradients between neurons during backward-propagation. In fact, the gradients largely depend on kernel weight. Reversely, in the last layer, neurons owning greater gradients with respect to loss are selected. Then, critical attacking neurons at each layer are chosen in the form of backward-propagation, based on their influences to the latter layers. We calculate the influence of i-th neuron F_{l-1}^i at the $l-1$-th layer to the j-th neuron F_l^j at the l-th layer through backward-propagation.

Specifically, for a single element z in a neuron output Z_l^j, we want to calculate the influence of i-th neuron output Z_{l-1}^i at the layer $l-1$ to this single element. We use the following formula:

$$\mathbf{C}_{l-1}^{i,z} = \sum_{s \in \mathbb{S}_{l-1}^i} \left| \frac{\partial z}{\partial \mathcal{A}(s, Z_{l-1}^i)} \right|, \tag{1}$$

where $\mathbb{S}_{l-1}^i$ is the set of elements in the i-th neuron at layer $l-1$, and $\mathcal{A}(\cdot)$ extracts the elements at specific positions. Note that, the influence of a neuron is computed as the sum of the absolute value of gradients with respect to the element at each position within a neuron. To best depict the influence, we believe, the $|\cdot|$ operation should be added after taking all the element changing directions of Z_{l-1}^i into consideration. This gradient calculating process between two intermediate layers is based on the convolutional kernel parameters, and we do not need to calculate their gradients towards loss $\mathcal{L}$ by chain rules.

We first compute $C_{l-1}^{i,z}$ for every element z in Z_l^j. Then, we calculate the neuron influence between Z_l^j and Z_{l-1}^i by summing up $C_{l-1}^{i,z}$ for all elements z in Z_l^j. Note that, the results are summed without $|\cdot|$, because all changes of element values in Z_l^j are caused by the change of the union Z_{l-1}^i, which can be viewed as changing in just one direction. That is,

$$\mathbf{C}_{l-1}^{i,j} = \sum_{z \in \mathbb{Z}_l^j} \mathbf{C}_{l-1}^{i,z}. \tag{2}$$

3.3. Instance-level critical attacking route

Since the critical attacking route is formed using neurons with the highest contributions to next layer (or loss), we can make a key assumption that this route contains the most critical information. Small amount of changes in the route would make huge impact to the model prediction. In other words, when a model makes correct predictions, this route conveys most non-trivial semantics. On the contrary, this route also contains most of the adversarial mistakes when the model is being attacked.

Given a sample x, we first derive the gradient of the m-th neuron at the last convolutional layer L with respect to the loss $\mathcal{L}$ yielding their contributions to model decision:

$$g_{\mathrm{L}}^{m} = \frac{\partial \mathcal{L}}{\partial Z_{\mathrm{L}}^{m}}, \tag{3}$$

where Z_{L}^{m} represents the output of m-th neuron at the L-th layer.

Then, we can recursively calculate the influence of i-th neuron F_{l-1}^{i} at the layer $l-1$ to the j-th neuron F_{l}^{j} at the layer l through Eqs. (1) and (2).

Thus, we can further pick up the critical attacking neurons by considering the top k percent of neurons according to their contributions. For neurons at the last convolutional layer, we have critical attacking neurons:

$$\omega_{\mathrm{L}}^{x} = \text{top-}k(F_{\mathrm{L}}, |g_{\mathrm{L}}|), \tag{4}$$

where top-$k(\cdot)$ represents top k percent instances of the input set based on certain metric. We call k the *width* of the critical attacking route. In this case, we choose g as metric and select ω_{L} from the set F_{L}. Similarly, we choose top k percent of neurons at the l-th layer based on their influences $\mathbf{C}$ to the critical attacking neurons at the $l+1$-th layer, recursively:

$$\omega_{l}^{x} = \text{top-}k(F_{l}, \tilde{\mathbf{C}}_{l}^{i}), \tilde{\mathbf{C}}_{l}^{i} = \sum_{F_{l+1}^{j} \in \omega_{l+1}^{x}} \mathbf{C}_{l}^{i,j}. \tag{5}$$

Then, the critical attacking route $\mathbf{R}(x)$ of sample x can be depicted by the connections of the critical attacking nodes from different layers:

$$\mathbf{R}(x) = \{\omega_{1}^{x}, \omega_{2}^{x}, \ldots, \omega_{\mathrm{L}}^{x}\}. \tag{6}$$

3.4. Model-level critical attacking route

To fully explore model behaviors towards adversarial noises, we further propose the model-level critical attacking route derived from the aggregation of instance-level routes.

Given a dataset $\mathbf{D}$ with N samples, we can obtain instance-level critical attacking route for each image as $\{\mathbf{R}(x_1), \mathbf{R}(x_2), \ldots, \mathbf{R}(x_{\mathrm{N}})\}$. Then, for the specific model, we can calculate the occurrence frequency of each critical attacking neuron at each layer for all instances as $\mathbf{f}$. It is intuitive that these neurons make the most non-trivial contributions since they have been witnessed in multiple instance-level routes for the specific model. We choose the top k percent of most frequent neurons as model-level critical attacking neurons for each layer l as below:

$$\Omega_{l} = \text{top-}k(F_{l}, \mathbf{f}_{l}). \tag{7}$$

Thus, the model-level critical attacking route is obtained:

$$\mathbf{R}(\mathbf{D}) = \{\Omega_1, \Omega_2, \ldots, \Omega_{\mathrm{L}}\}. \tag{8}$$

It's important to note that the model-level critical attacking route is not just a simple add-up of many different critical attacking routes. Instance-level critical attacking routes share great similarities among instances, which can be certificated by neurons occurring frequency in Section 4.1.

Our calculating process is a layer-wise process, and we pick up neurons according to their influence on next layer's critical neurons. Hence our method is more suitable for more-linear models. In the future, we will investigate ways to apply our methods to non-linear structures (e.g., using the sum of gradients in a branch structure).

3.5. Time complexity

Since the influence of each neuron needs to be calculated layer by layer, we analyze time complexity of this process here.

In the calculating process, the time is mainly spent on figuring out gradients between neurons. We use t^l to denote the time consumption for the calculation of the influence of one neuron to another. The time consumption for a specific layer l depends on n_a^l and n_p^{l+1}, which denote the number of all neurons in the former layer and the number of neurons picked in the latter layer, respectively. Then, it takes a total $T^l = n_a^l \ * n_p^{l+1} \ * t^l$ time for a layer. For one image, the whole model's route calculating time is $T = \sum_l^{l \in F} T^l$. On CIFAR-10 dataset, it took us 7.4 h to calculate the model-level critical attacking route for a VGG-16 model for 10000 images with a GTX1080 GPU. However, in our implementation, we will improve the efficiency of route calculation process and keep the same performance by using small amount of samples as illustrated in Section 5.1.3.

4. Understanding model behaviors via critical attacking routes

In this section, we try to understand model behaviors through the critical attacking route. We first demonstrate that perturbations mainly propagate via this route in adversarial setting. In other words, neurons on the route bear most of noise information and are vulnerable to adversarial attacks. Besides, critical attacking routes also convey important semantic information.

4.1. Empirical settings

Datasets and Models. In our experiments, we use CIFAR-10, SVHN and ImageNet datasets. For models, we adopt three typical DNNs for image classification tasks, i.e., AlexNet [13], VGG-16 [28], and ResNet-18 [10].

Adversarial Attack and Defense. We apply several state-of-the-art adversarial attack methods including FGSM [8], Step-LL [15], MI-FGSM [4], and PGD [20]. We set the parameters following the guidance [1] as below. For FGSM and Step-LL, we set the parameter $\epsilon = 8/255$. For MI-FGSM and PGD attack, we set $\epsilon = 8/255$, step size $\alpha = \epsilon/10$, and iteration $k = 10$. As for adversarial defense, we choose the state-of-the-art defense method PGD-based adversarial training (PAT [20]).

Critical Attacking Route Extraction. Empirically, we choose top-20% neurons as critical attacking neurons every layer for instance-level critical attacking routes. When aggregation, we choose top-30% neurons for the model-level critical attacking route. We further calculate the mean occurrence frequency $\mathbf{O}_l$ of critical attacking neurons at each layer for all instance-level critical attacking routes. As shown in Table 1, the similarities of neurons at different layers are very high in all instance-level critical attacking routes. More importantly, such a high mean occurrence frequency makes it possible for us to utilize a small amount of instance-level critical attacking routes to aggregate an overall attacking route, which is close to the model-level critical attacking route.

4.2. Adversarial perturbations are propagated and amplified through attacking route

In this section, we introduce a very important characteristic of the critical attacking route that perturbations are mainly propagated via this route in adversarial setting.

4.2.1. Critical attacking neurons are sensitive and influential

We first try to investigate the behaviors of neurons within the critical attacking route when facing adversarial noises.

Neuron sensitivity is measured by neuron behavior variation intensity against benign and adversarial examples, and has been used to depict adversarial robustness for deep models [37]. Thus, we try to explore the sensitivity of both critical attacking neurons and neurons outside the route. In Fig. 2(a), our experiments show that neurons on critical attacking route are significantly more sensitive than regular neurons outside the route especially at bottom layers. Intuitively, critical attacking neurons are more vulnerable and unstable to adversarial noises, which should be responsible for adversarial weakness of deep neural networks.

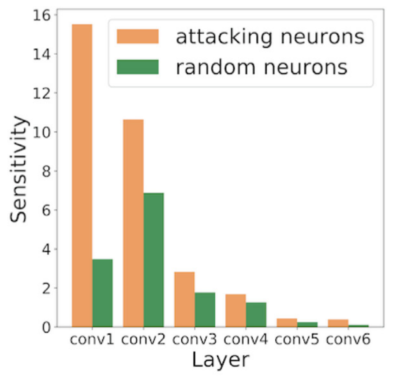
Further, we study the neuron contributions to model decisions by exploring their influences to the model loss. Given the dataset $\mathbf{D}$, we calculate the average gradient of every neuron towards the loss based on all test samples. As shown in Fig. 2(b), it is obvious that critical attacking neurons have greater gradients with respect to the loss compared to other regular neurons. In contrast to regular neurons, similar degrees of changes on the activation value for critical attacking neurons will cause more stimulus to the final classification. Thus, it's fair to say that the critical attacking route is more likely to convey and propagate perturbations.

4.2.2. Adversarial feature transplant

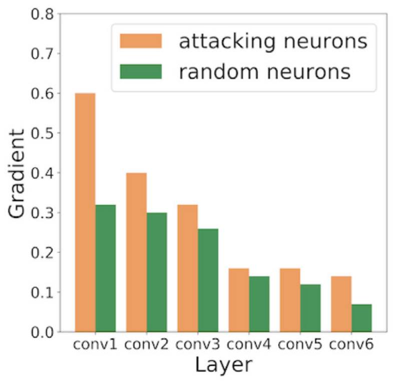
With the above analysis, we further investigate the contribution of critical attacking route to model robustness by feature transplanting. Given a dual pair data set $\bar{\mathbf{D}} = \{(x_i, x_i')\}$ with benign example x_i and the corresponding adversarial example x_i', where $i = 1, \ldots, \mathrm{N}$, we replace the adversarial features $Z_l^j(x_i')$ with the benign features $Z_l^j(x_i)$ at specific position to see the model behavior difference. As shown in Fig. 3(a), with the increasing of proportions of the transplanted critical attacking neurons (the width of the critical attacking route) on the first conv layer, adversarial robustness rises significantly. When we transplant nearly 30% of critical neurons at the first conv layer, the accuracy of adversarial examples is promoted from 0% to 53.68%. We further transplant attacking neurons at multiple layers. As shown in Fig. 3(b), after transplanting 30% of the

Table 1
Mean occurrence frequency $\mathbf{O}_l$ for each layer. Specifically, the neurons in the bottom layers are more similar while those for top layers are slightly lower. Intuitively, the phenomenon demonstrates that the bottom layers mainly focus on low-level features (i.e., textures, lines, etc.), while top layers are required to classify images and make predictions.

		AlexNet			
layer	conv1	conv2	conv3	conv4	conv5
$\mathbf{O}_l$	97%	97%	97%	94%	45%
		VGG16			
layer	conv1	conv4	conv7	conv10	conv13
$\mathbf{O}_l$	70%	48%	60%	67%	38%

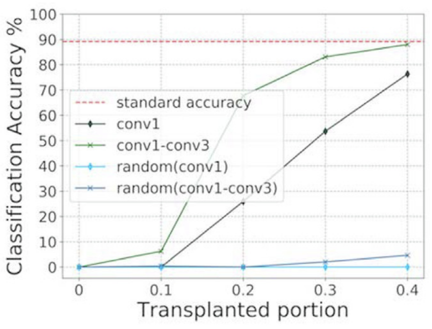


(a) Sensitivity

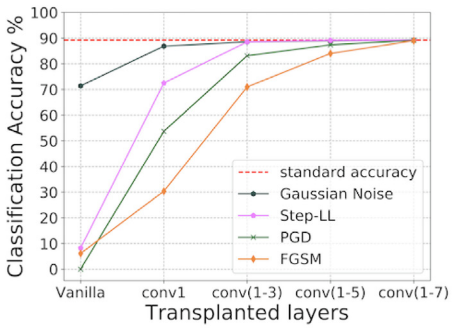


(b) Gradient

Fig. 2. Special characteristics of critical attacking neurons. (a) Attacking neurons own greater sensitivities than regular neurons. (b) Attacking neurons own greater gradients towards loss.



(a) Transplant experiment1



(b) Transplant experiment2

Fig. 3. Adversarial feature transplant experiment. (a) denotes the model performance with different transplant proportions at different layers. (b) depicts the transplant results on different single layers or multiple layers with fixed transplant proportion as 30%.

critical attacking neurons at the first three conv layers, the model basically corrects all the adversarial examples. However, the amount of neurons we transplant only accounts for 1.8% of the whole model's neurons.

To further confirm our observations, we also transplant neurons outside the critical attacking route randomly. Surprisingly, with the same amount of neurons transplanted, the adversarial robustness is still extremely low. The classification accuracy is no more than 4.63% as shown in Fig. 3(a).

Since corruption noises such as snow and blur pose strong challenges to robust deep learning models, we also demonstrate the effectiveness of our critical attacking route for corruption. As shown in Fig. 3(b), the critical attacking route is not only important for gradient-based adversarial attack methods, e.g., FGSM and PGD, but also critical for corruption like Gaussian noise, etc.

4.3. Attacking route conveys strong semantic information

In this section, we give more insights into the roles of critical attacking neurons. Firstly, we deactivate the output of neurons. As shown in Fig. 4(a), the model classification accuracy degenerates significantly when critical attacking neurons are increasingly deactivated. On the contrary, the model performance is just slightly influenced when we deactivate regular neurons. Then, we try to weaken the connection between neurons with an coefficient β. We directly multiply β to connections (corresponding weights) between neurons to weaken the conveying process on this route. According to the results in Fig. 4 (b), with the decreasing of β, severer weakened connections between critical attacking neurons witness a significant drop of model accuracy. However, model is nearly not influenced when the connections between regular neurons are weakened.

To better visualize the roles of critical attacking neurons, we adapt Grad-CAM [25] to investigate the attention field of different neurons (i.e., critical attacking neurons and regular neurons) in different settings (i.e., adversarial and clean). In the adaption, we separate critical attacking neurons and regular neurons for visualization. We adopt unified normalization for these two cams to compare their contributions to final predictions. As shown in Fig. 5, after adversarial attack, critical attacking neurons tend to pay more attention to the noisy backgrounds compared with their subtle detection regions on clean examples. However, the changes for normal neurons are not so obvious. Interestingly, for clean examples, attention regions of attacking neurons are focused on the key part of the class (i.e., the body of the bird), while those for normal neurons are meaningless.

Thus, it's reasonable to assume that the critical attacking route contains rich semantic features and is responsible for model predictions.

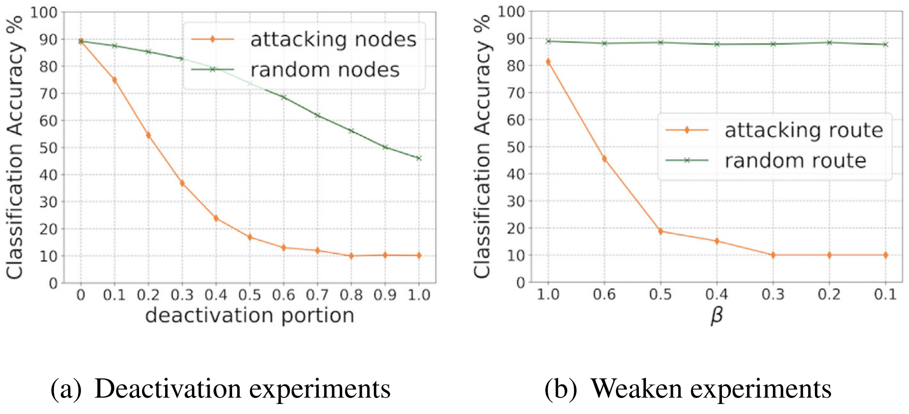


Fig. 4. Model performance degradation when we deactivate critical attacking neurons and weaken weight on this attacking route. Ablation on critical attacking neurons leads severer degradation compared with ablation on random neurons.

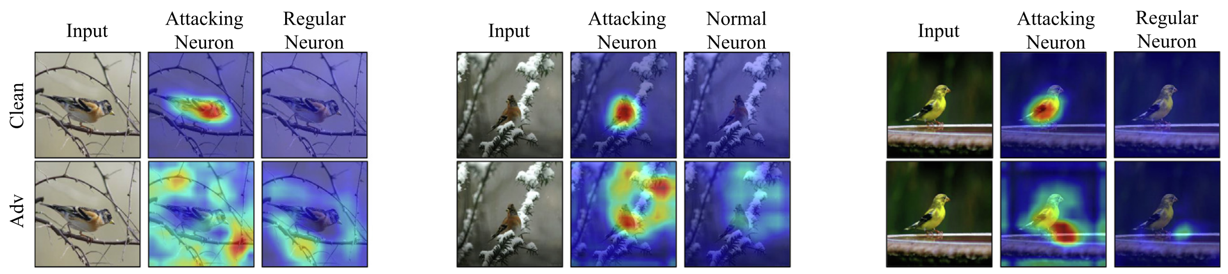


Fig. 5. Grad-CAM of neurons on critical attacking route(middle column) and Grad-CAM of neurons not on critical attacking route(right column) of the last conv layer using pretrained VGG16 on ImageNet. Traditional Grad-CAM aggregates all neurons' features. We separately merge two kinds of neurons to view their contributions to classification. Upper row is the grad-cam saliency map on benign samples, showing that neurons on attacking route contain more semantic information. Below row is the grad-cam saliency map on adversarial samples, demonstrating that attacking neurons bear more misleading information.

5. Leveraging critical attacking routes

Since adversarial perturbations are mainly propagated through the critical attacking route, it can be viewed as a defect of a model. It is natural for us to consider to improve model robustness by repairing this defect. We try to respectively constrain the behaviors of neurons or weaken the adversarial noises propagation within the critical attacking route to weaken the noise propagation. Besides, inspired by the phenomenon that samples with bigger gradients on critical attacking neurons are easier to be adversarially attacked, we also propose a metric to evaluate the adversarial hardness of samples.

5.1. Model adversarial robustness enhancement

Motivated by our observations, in this section, we try to improve model robustness by constraining the behaviors of critical attacking neurons, e.g., gradients, propagation process.

We design two simple but effective methods to promote model robustness based on the critical attacking route. All the experiments in this section are conducted on CIFAR-10 and SVHN using VGG-16. We use Vanilla and PAT to denote the original model and the PGD-based adversarial training model, respectively.

5.1.1. Constraining gradients towards loss

During the experiment, we observe that critical attacking neurons own greater gradients towards loss. Thus, it is intuitive to reduce the impact caused by these neurons when facing adversarial inputs and directly punish their gradients towards loss. Thus, we directly obtain a loss term by constraining the gradients of critical attacking neurons:

$$g_l^m = \frac{\partial \mathcal{L}}{\partial Z_l^m}, \tag{9}$$

$$\mathcal{L}_g(x,y;\theta) = \sum_{l}^{l\in F} \sum_{m}^{F_l^m \in \Omega_l} |g_l^m|. \tag{10}$$

For Vanilla, we can simply add the loss term to the cross entropy loss:

$$\mathcal{L} = \mathcal{L}_c(x,y;\theta) + \lambda \mathcal{L}_g(x,y;\theta). \tag{11}$$

For PAT, combining with adversarial training, we design a new loss function as:

$$\mathcal{L} = \mathcal{L}_{adv}(x,x',y;\theta) + \lambda(\mathcal{L}_g(x,y;\theta) + \mathcal{L}_g(x',y;\theta)), \tag{12}$$

where $\mathcal{L}_c$ denotes the cross entropy loss and $\mathcal{L}_{adv}$ denotes the adversarial training loss. λ is a hyper-parameter balancing these loss terms. We use Vanilla$_g$ and PAT$_g$ to denote models trained with gradient punishment based on Vanilla and PAT models, respectively.

5.1.2. Constraining propagation process

In addition, we try to improve model robustness by weakening the connections in the critical attacking route. Simply, we punish the weight of every neuron in this route to weaken the adversarial perturbation propagation process:

$$\mathcal{L}_w(x;\theta) = \sum_{l}^{l\in F} \sum_{m,n}^{F_l^m \in \Omega_l, F_{l-1}^n \in \Omega_{l-1}} ||\mathbf{w}_l^{m,n}||_1, \tag{13}$$

where $\mathbf{w}_l^{m,n}$ is the weight of channel n of neuron m in the l-th layer.

For Vanilla, we can simply add the loss term to the cross entropy loss:

$$\mathcal{L} = \mathcal{L}_c(x,y;\theta) + \lambda \mathcal{L}_w(x;\theta). \tag{14}$$

Similarly, for PAT, we design a new loss as:

$$\mathcal{L} = \mathcal{L}_{adv}(x,x',y;\theta) + \lambda \mathcal{L}_w(x;\theta). \tag{15}$$

We use Vanilla$_w$ and PAT$_w$ to denote models trained with weight punishment based on Vanilla and PAT models.

5.1.3. Experimental results and discussion

According to the results in Tables 2 and 3, either constraining the gradients or the propagation process improves model robustness towards both adversarial examples and Gaussian noise. Our methods can also be combined with state-of-the-art adversarial training strategies to further improve model robustness.

To improve the efficiency of the critical attacking route calculation process, we use a small amount of images to generate instance-level critical attacking routes, and then aggregate the model-level critical attacking route as introduced in Section 4.1. In our implementation, we use 10% samples to get the overall attacking route, which shares a 94.6% occurrence fre-

Table 2
Performance of models trained with our methods on Vanilla and PAT trained on CIFAR-10. All the values are the classification accuracy (%). For Vanilla models, we use MI-FGSM to replace PGD. PGD's attacking ability is too strong for Vanilla to observe obvious improvement.

Vanilla				
Model	Clean	FGSM	MI-FGSM	Gaussian
Vanilla	89.19	4.88	3.10	25.49
Vanilla$_g$	**92.60**	9.26	**7.48**	**28.54**
Vanilla$_w$	91.94	**10.01**	5.52	26.08
PAT				
Model	Clean	FGSM	PGD	Gaussian
PAT	84.95	47.88	46.71	76.63
PAT$_g$	84.93	49.16	**48.39**	77.16
PAT$_w$	**85.06**	**49.61**	48.20	**78.18**

Table 3
Performance of models trained with our methods on Vanilla and PAT on SVHN. All the values are the classification accuracy (%). The numbers in bold represent the best performance in each setting.

Vanilla				
Model	Clean	FGSM	MI-FGSM	Gaussian
Vanilla	94.42	28.10	1.86	38.76
Vanilla$_g$	95.38	**56.97**	**14.95**	**45.47**
Vanilla$_w$	**95.48**	55.38	14.29	43.12
PAT				
Model	Clean	FGSM	PGD	Gaussian
PAT	92.23	55.25	46.21	78.34
PAT$_g$	93.56	57.07	**52.50**	**83.15**
PAT$_w$	**95.40**	**57.80**	48.74	80.23

quency with the model-level critical attacking route. For adversarial defense, using 10% samples achieves similar results (differences are not more than 0.5%) compared to using all samples. In other words, we can use 10% samples to achieve similar defense performance. In fact, the PAT training time is more than 12 h. However, it just take us 42 min to calculate critical attacking route with 10% samples and an extra 2.3 h to fine-tune the model with the route. Thus, in the following parts of the paper, we use 10% samples to get the model-level critical attacking route.

5.2. Sample adversarial hardness evaluation

Prior studies have revealed the fact that the difficulty for distinguishing instances from different classes are not equal [23]. With the same magnitude of perturbations, some images are easily perturbed into adversarial examples, while some are very hard.

According to the experiments, we find that images that are easier to be perturbed into adversarial examples own higher gradients towards loss on the critical attacking route. As shown in Table 4, the gradients of instances with different adversarial hardness differ significantly. Obviously, small gradients towards loss indicate that intensive perturbations are required to change model predictions, which explains the reason that these instances are harder to attack. Similarly, samples with huge gradients mean the opposite. Thus, we propose to use the gradients to evaluate sample adversarial hardness. Note that our evaluation process requires no perturbation.

Given an input sample, we utilize the average gradients of model's critical attacking neurons towards loss to calculate the adversarial hardness metric. Specifically, we adopt a voting strategy for every sample according to their gradients on critical attacking neurons. For a specific layer, samples with higher gradients are given less votes. Then, every sample gets its vote for every layer. Thus, we accumulate all layer's votes of one sample as its adversarial hardness metric. Obviously, samples with lower gradients denote more votes and higher values of the adversarial hardness metric. For a sample x, according to Eq. (9), we get g_l^m and calculate the overall gradients of a sample on a layer's critical attacking neurons.

$$g_l^x = \sum_{m \in \Omega_l} |g_l^{m,x}|, \tag{16}$$

$$v_l^x = Rank(g_l^x, D), \tag{17}$$

where $Rank(\cdot)$ returns the rank of sample x's gradient in the Dataset $\mathbf{D}$, and we use v_l^x to denote this rank. Then, we can get the adversarial hardness metric V^x as follows:

Table 4
Gradients towards loss of attacked samples. Those attacked successfully(weak instances) show greater gradients than those attacked unsuccessfully(hard instances).

Instances	conv1	conv4	conv7	conv10
Hard instances	0.0046	0.0014	0.0005	0.0001
Weak instances	4.5565	1.1183	2.1692	0.4096

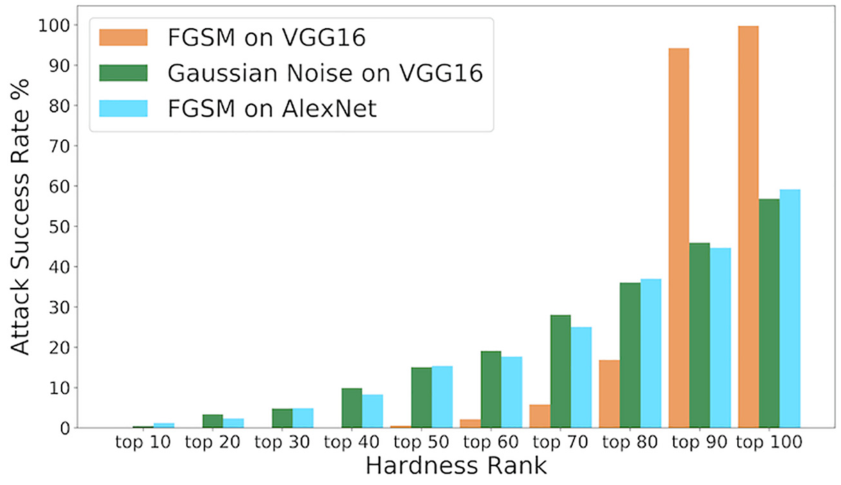


Fig. 6. Numbers of instances that are successfully attacked. We sort these samples by our hardness metric inversely (i.e., samples with higher hardness are ranking in the front). Samples with higher adversarial hardness are harder to attack. For FGSM, we set the parameter $\epsilon = 1.0/255$ to achieve fair attacking effect for observation.

$$V^x = \sum_{l \in F} v_l^x. \tag{18}$$

We use both adversarial examples (FGSM) and corruption (Gaussian noise) to verify the effectiveness of the adversarial hardness metric. In Fig. 6, it's obvious that instances with lower V^x are generally easier to attack.

Moreover, our metric can be used to select model criticisms [30] without perturbations. Stock et al. call data points sampled from regions of the input space not well captured by the model as model criticisms. They propose that the examples whose decisions change after 1 or few steps of attacking methods are more likely to be valid criticisms because they do not quite fit the model. By our metric, we can select those samples harder to be attacked successfully, which represent what model learns better.

6. Conclusion

To better understand adversarial robustness of deep models, we propose a new concept "critical attacking route" which is computed by a gradient-based influence propagation strategy. We then draw the close connection between adversarial robustness and critical attacking route, as the route makes the most non-trivial contributions to model predictions in the adversarial setting. Further, critical attacking routes also contain strong semantic information. By constraining the propagation process and node behaviors on this route, we could weaken the noise propagation process and improve model robustness. Also, critical attacking neurons can be used to evaluate sample adversarial hardness.

CRediT authorship contribution statement

Tianlin Li: Investigation, Conceptualization, Methodology, Writing - original draft. **Aishan Liu:** Methodology, Writing - review & editing. **Xianglong Liu:** Conceptualization, Supervision. **Yitao Xu:** Visualization, Investigation. **Chongzhi Zhang:** Investigation, Methodology. **Xiaofei Xie:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by The National Key Research and Development Plan of China (2019AAA0103502), National Natural Science Foundation of China (61872021), Beijing Nova Program of Science and Technology (Z191100001119050), State Key Lab of Software Development Environment (SKLSDE-2020ZX-06) and Fundamental Research Funds for Central Universities (YWF-20-BJ-J-646).

References

[1] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705, 2019.
[2] Nicholas Carlini, David Wagner, Towards evaluating the robustness of neural networks, in: In IEEE Symposium on Security and Privacy (S&P), 2017.
[3] Igino Corona, Giorgio Giacinto, Fabio Roli, Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences 239 (08) (2013) 201–225.
[4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Xiaolin Hu, Jun Zhu, Discovering adversarial examples with momentum, CoRR, abs/1710.06081, 2017.
[5] Yinpeng Dong, Hang Su, Jun Zhu, Fan Bao, Towards interpretable deep neural networks by leveraging adversarial examples. CoRR, abs/1708.05493, 2017.
[6] Xiaoning Du, Xiaofei Xie, Yi Li, Yang Liu, Jianjun Zhao, Deepstellar: model-based quantitative analysis of stateful deep learning systems, 08 (2019) 477–487.
[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, Dawn Song, Physical adversarial examples for object detectors. arXiv preprint arXiv:1807.07769, 2018.
[8] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and harnessing adversarial examples, 2014.
[9] Feng Guo, Qingjie Zhao, Xuan Li, Xiaohui Kuang, Jianwei Zhang, Yahong Han, Yu an Tan, Detecting adversarial examples via prediction difference for deep neural networks, Information Sciences, 501 (2019) 182–192.
[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[11] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, Bastian Bischoff, On Detecting Adversarial Perturbations. arXiv e-prints, page arXiv:1702.04267, 2017.
[12] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, Aleksander Madry, Adversarial examples are not bugs, they are features. arXiv preprint arXiv:1905.02175, 2019.
[13] Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, Imagenet classification with deep convolutional neural networks, Neural Information Processing Systems 25 (2012) 01.
[14] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.
[15] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio, Adversarial machine learning at scale. CoRR, abs/1611.01236, 2016.
[16] Alex Lamb, Vikas Verma, Juho Kannala, Y. Bengio, Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy 11 (2019) 95–103.
[17] Aishan Liu, Tairan Huang, Xianglong Liu, Yitao Xu, Yuqing Ma, Xinyun Chen, Stephen Maybank, Dacheng Tao, Spatiotemporal attacks for embodied agents, in: European Conference on Computer Vision, 2020.
[18] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, Dacheng Tao, Perceptual-sensitive GAN for generating adversarial patches, in: The Thirty-Third AAAI Conference on Artificial Intelligence, 2019.
[19] Aishan Liu, Jiakai Wang, Xianglong Liu, bowen Cao, Chongzhi Zhang, Hang Yu, Bias-based universal adversarial patch attack for automatic check-out, in: European Conference on Computer Vision, 2020.
[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.
[21] Dongyu Meng, Hao Chen, Magnet: A two-pronged defense against adversarial examples, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS–17, New York, NY, USA, 2017. Association for Computing Machinery, pp. 135–147.
[22] A. Mohamed, G.E. Dahl, G. Hinton, Acoustic modeling using deep belief networks, IEEE Transactions on Audio Speech and Language Processing 20 (1) (2012) 14–22.
[23] Konda Reddy Mopuri, Utsav Garg, R. Venkatesh Babu, Fast feature fool: A data independent approach to universal adversarial perturbations. CoRR, abs/1707.05572, 2017.
[24] Yuxian Qiu, Jingwen Leng, Cong Guo, Quan Chen, Chao Li, Minyi Guo, Yuhao Zhu, Adversarial defense through network profiling based path extraction. CoRR, abs/1904.08089, 2019.
[25] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, Dhruv Batra, Grad-cam: Why did you say that? Visual explanations from deep networks via gradient-based localization. CoRR, abs/1610.02391, 2016.
[26] Yucheng Shi, Yahong Han, Quanxin Zhang, Xiaohui Kuang, Adaptive iterative attack towards explainable adversarial robustness, Pattern Recognition 02 (2020) 107309.
[27] Yucheng Shi, Siyu Wang, Yahong Han, Curls & whey: Boosting black-box adversarial attacks, CoRR, abs/1904.01160, 2019.
[28] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 09 (2014), arXiv 1409.1556.
[29] Chuanbiao Song, Kun He, Liwei Wang, John E. Hopcroft, Improving the generalization of adversarial training with domain adaptation. CoRR, abs/1810.00740, 2018.
[30] Pierre Stock, Moustapha Cissé, Convnets and imagenet beyond accuracy: Explanations, bias detection, adversarial examples and model criticism. CoRR, abs/1711.11443, 2017.
[31] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, Sequence to sequence learning with neural networks. CoRR, abs/1409.3215, 2014.
[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, Rob Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, 2014.
[33] Yulong Wang, Su. Hang, Bo Zhang, Hu. Xiaolin, Interpret neural networks by identifying critical data routing paths, IEEE Conference on Computer Vision and Pattern Recognition (2018).
[34] Yatie Xiao, Chi-Man Pun, Bo Liu, Adversarial example generation with adaptive gradient search for single and ensemble deep neural network, Information Sciences 528 (2020) 147–167.
[35] Kaidi Xu, Sijia Liu, Gaoyuan Zhang, Mengshu Sun, Pu Zhao, Quanfu Fan, Chuang Gan, Xue Lin, Interpreting adversarial examples by activation promotion and suppression. CoRR, abs/1904.02057, 2019.
[36] Yu. Yintao, Tanya Y Berger-Wolf, Jared Saia, et al, Finding spread blockers in dynamic networks, in: International Workshop on Social Network Mining and Analysis, Springer, 2008, pp. 55–76.
[37] Chongzhi Zhang, Aishan Liu, Xianglong Liu, Yitao Xu, Hang Yu, Yuqing Ma, Tianlin Li, Interpreting and improving adversarial robustness with neuron sensitivity, arXiv preprint arXiv:1909.06978, 2019.
[38] Tianyuan Zhang, Zhanxing Zhu, Interpreting adversarially trained convolutional neural networks. CoRR, abs/1905.09797, 2019.