Residual Error: a New Performance Measure for Adversarial Robustness

Hossein Aboutalebi[1], Mohammad Javad Shafiee[1]
Michelle Karg[2], Christian Scharfenberger[2]
Alexander Wong[1]
[1]Waterloo AI Institute, University of Waterloo, Waterloo, Ontario, Canada
[2]ADC Automotive Distance Control Systems GmbH, Continental, Germany
[1]{haboutal, mjshafiee, a28wong}@uwaterloo.ca
[2]{michelle.karg, christian.scharfenberger}@continental-corporation.com

Abstract

Despite the significant advances in deep learning over the past decade, a major challenge that limits the widespread adoption of deep learning has been their fragility to adversarial attacks. This sensitivity to making erroneous predictions in the presence of adversarially perturbed data makes deep neural networks difficult to adopt for certain real-world, mission-critical applications. While much of the research focus has revolved around adversarial example creation and adversarial hardening, the area of performance measures for assessing adversarial robustness is not well explored. Motivated by this, the current study presents the concept of residual error, a new performance measure for not only assessing the adversarial robustness of a deep neural network at the individual sample level, but also can be used to differentiate between adversarial and non-adversarial examples to facilitate for adversarial example detection. We also introduce a hybrid model for approximating the residual error in a tractable manner. Experimental results using the case of image classification demonstrates the effectiveness and efficacy of the proposed residual error metric for assessing several well-known deep neural network architectures. The results demonstrate the capability of the proposed measure as a useful tool for not only assessing the robustness of deep neural networks used in mission-critical scenarios, but also in designing adversarially robust models.

1. Introduction

Deep learning models over the past few years have yield new records in several fields such as different computer vision applications [6, 7, 5, 9, 9], machine translation [13, 12], and medicine [3, 2]. Although these achievements bring new level of accuracy, recent research especially within computer vision applications have shown that deep neural networks are unable to detect some of the intuitive underlying concepts in datasets [11, 4, 8]. These findings generally follow the idea of adding small perturbation ϵ to the input image causes the model to classify the image incorrectly. The perturbation to the input image is imperceptible to human eyes most of the time. This phenomenon was first discovered by szegedy *et al.* in their seminal paper [11]. They observed that the state-of-the-art deep neural networks act poorly with high confidence when an imperceptible non-random perturbation is added to the input image. The perturbed examples so-called *"adversarial examples"* are generated by adding a targeted noise calculated based on the loss value and projected gradient. They also discovered that the adversarial examples are shared between different network architectures and training data. In other words, if a set of adversarial examples are generated for one specific network, it is possible that these adversarial examples will still be missclassified by another network with different architecture even if the new network is trained on different training data.

Here, we formulate the problem of deep neural networks facing adversarial attacks from a different point of view. Given the trained model, we argue that due to the existence of adversarial examples, the test error is not a sufficient measure to indicate the accuracy of the trained model. As such, we propose a new measure that can be used besides the test error to estimate the model error for the individual input, so-called residual error. Specifically our contributions in this paper are as follows:

- The introduction of a new performance measure for adversarial robustness called residual error which provides a more precise measure of error at the individual sample level.
- Proposing a novel prediction model learning method for approximating the residual error.

- Designing a novel hybrid model which can estimate residual error in an accurate manner while being an order of magnitude faster to execute.
- The Introduction of a novel strategy to harnessing residual error prediction model for detecting adversarial examples.
- Comprehensive experimental results to evaluate the efficacy of the proposed residual error measure and associated prediction models for the task of image classification on the CIFAR-10 and CIFAR-100 datasets.

The paper is organized as follows. The underlying theory behind residual error along with the methodology for approximating the residual error by learning a hybrid prediction model is described in detail in Section 2. The experimental results are presented and discussed in detail in Section 3. Finally, conclusions are drawn and future directions are discussed in Section 4.

2. Methodology

In this section, we will describe in detail the underlying theory behind the proposed residual error performance metric. Furthermore, we will describe in detail how the residual error can be approximated using a prediction model followed by the introduction a hybrid model for estimating residual error.

2.1. Residual Error

Let us assume the hypothesis $h \in \mathcal{H}$[1] is a mapping function to model $\mathcal{X} \rightarrow \mathcal{Y}$. As such, h tries to estimate the target function $f(\cdot)$ based on the training data $S \subset \mathcal{X} \times \mathcal{Y}$; therefore, the error of h with loss function $l\big(h(x), f(x)\big)$ is defined as follows:

$$L_{D,f}(h) = \mathbb{E}_{x \sim D}\left[l(h(x), f(x)) \right] \tag{1}$$

where D is the distribution of domain space. For a regression problem with MSE error, we can rewrite (1) as follows:

$$L_{D,f}(h) = \mathbb{E}_{x \sim D}\left[\big(h(x) - f(x)\big)^2 \right] \tag{2}$$

and the error can be calculated for a classification problem using the indicator function:

$$L_{D,f}(h) = \mathbb{E}_{x \sim D}\left[\mathbf{1}_{h(x) \neq f(x)} \right] \tag{3}$$

where $\mathbf{1}_{h(x) \neq f(x)}$ is equal to 1 only when the predicted label for input x by hypothesis h is different from the target label $f(x)$.

[1] $\mathcal{H}$ is the set of hypotheses

However, the exact value of (1) cannot be calculated and as such, a test set is used to estimate its value. Assuming S' is our test set, the test error becomes:

$$L_{S',f}(h) = \frac{1}{|S'|} \sum_{x \in S'} l(h(x), f(x)) \tag{4}$$

which is the empirical error approximates the final accuracy of the trained model in practice.

Considering the existence of adversarial examples, in this paper, we argue that the test error may not be the sufficient measure of empirical error of the model. In this regard, (1) is the value of the expected error over the domain space $\mathcal{X}$ and (1) does not provide any further information for individual input x. This also holds for the test error as it is the empirical estimation of (1). For example, although the trained model behaves differently under adversarial attack, the test error does not provide an insight to differentiate between adversarial and non-adversarial example. As a result, we need to provide a precise estimate of the model error for an individual input x:

$$R_h(x) = \mathbb{E}\left[l\big(h(x), f(x)\big) \right] \tag{5}$$

which (5) is called the residual error of the trained model h for input x.

The benefit of having residual error $R_h(x)$ to the test error $L_{S',f}(h)$ is that now we can decide based on the input if the model is able to make the correct prediction or not. While test error provides a general error estimation on the whole domain, residual error gives us an error measure on each individual input data from the domain. A particular interpretation that we can have is that if $R_h(x)$ is a high value and the model has a low error on the test set, we can expect that the input x might be an adversarial example. This way, we will be able to detect adversarial examples. As such, the main remaining challenge here is that how we want to estimate the value of $R_h(x)$. We will cover this in the next section.

2.2. Residual Error Prediction Model

Estimating $R_h(x)$ is different from estimating the test error since the training set used for this estimation is only a subset which represents the domain partially. As such, approximating $R_h(x)$ is highly desirable. Given the prediction model h, the residual error of trained model h can be approximated using prediction function $g(\cdot)$. Here, we call h the primary model and g as the residual error prediction model. The residual error prediction model is trained given the primary model has been already trained.

We now describe the dataset used to train the residual error prediction model. For each pair of $(x, y) \in S$ in our training set, it is replaced with $(x, r(x))$, where

$$r(x) = l\big(h(x), y\big). \tag{6}$$

Algorithm 1: Resedual Network Training

Data: $\Big\{\big(x, r(x)\big) | x \in D\Big\}$
Result: g^*: trained residual error prediction model g
input: validation set: S_{valid}
primary model: h
residual error prediction model :g
loss function primary model: l
loss function residual prediction model: l'
begin
S=[]
for (x, y) in S_{valid}:
$r(x) = l\big(y, h(x)\big)$
Insert $(x, r(x))$ into S
g^* = $\underset{g}{argmin}\ \sum_{x \in S} l'\big(g, r(x)\big)$
return g^*
end

As a result, the training dataset for the residual error prediction model is constructed as follow:

$$S_{res} = \Big\{\big(x, r(x)\big)\ |\ (x, y) \in S\ \ \wedge\ \ r(x) = l\big(h(x), y\big)\Big\}. \tag{7}$$

For a regression problem, the labels of training data model is represented by the MSE error and the residual error prediction model training is still a regression problem. On the other hand, the labels of the residual error prediction model become 0 or 1 (i.e., the primary model classifies the sample correctly or not) and the training of the residual error prediction model is a classification problem, and it is formulated as a binary classification problem. In this regard, although the primary model training might be done with a multi-class classification training data, the residual error prediction model training becomes a simpler task of binary classification.

The loss function for residual error prediction model is the Cross-entropy loss for the classification problem, while the MSE loss can be used to train the residual error prediction model for a regression problem. It is worth to note that other variation of loss functions with regularization can be used for training the residual error prediction model if the loss function can fit the learning problem of the residual error prediction model. The details of the training of the residual error prediction model is described in Algorithm 1.

2.3. Hybrid Residual Error Prediction Model

In this section, we propose a hybrid architecture design which takes advantage of deep neural network macroarchitecture to extract useful features from the input image while the decision tree macroarchitecture tries to discriminate the decision-making of the primary network in whether it classifies the input image correctly or not. Our observations have shown that using the proposed hybrid structure as the residual error prediction model can improve the performance of the proposed measure. The proposed hybrid model can use the primary model architecture to extract useful features followed by a decision tree for the classification purposes. As such, the output layer (classifier) of the primary module is substituted by a decision tree where the output of the last layer in the primary model is fed into the decision tree. During the training of residual error prediction model, the primary model weights are frozen and the decision tree parameters are only updated.

This approach can benefit from the feature representation of a deep neural network while reducing the whole problem of training the residual error prediction model with only constructing a decision tree. Experimental results showed that the training of the proposed hybrid residual model is an order of magnitude faster than training a deep network model from scratch and use as the residual error prediction model. In our experiments, we used XGBoost [1] decision tree structure. The architecture of our model is depicted in the Supplementary.

3. Experimental Results

The proposed method is evaluated based on different neural network architectures and with two datasets of CIFAR-10 and CIFAR-100. Different neural network architectures including ResNet-18, ResNet-34 [5], LeNet and MobileNetV2 [10] are used to measure the effectiveness of the proposed method. The proposed hybrid model is compared with other approaches as well. To examine the performance of the proposed residual method the input samples are perturbed by FGSM with $\epsilon = 8$ adversarial attacks. Most of the experimental results are included in the Supplementary. Also, the details regarding the training of the network is also included in the Supplementary.

3.1. CIFAR-10

Table 1 shows the experimental results of evaluating the proposed residual error prediction model on different deep neural network architecture on CIFAR-10 dataset. To better evaluate the effectiveness of the proposed method, the performance of the residual error prediction model is assessed on two different situations, i) the normal dataset where the examined samples are clean and without any adversarial perturbation, ii) the adversarial dataset which is the samples are perturbed by FGSM adversarial attacks. The goal here is to determine what is performance of the residual error prediction model in detecting whether the primary network is classifying the samples correctly or not. The training data for the residual error prediction model is created based on the residual validation set explain in (7). In order to increase the size of training data for training the residual error

Table 1: Results of residual error prediction model accuracy on CIFAR-10. Here adversarial refers to FGSM attack with $\epsilon = 8$, and normal dataset is the dataset without any adversarial examples.

Primary model	Dataset	Primary Accuracy	Residual error prediction model				
			ResNet-18	ResNet-34	LeNet	MobileNetV2	Hybrid
ResNet-18	Normal	0.9304	0.8747	**0.9303**	0.9262	0.8844	0.8761
	Adversarial	0.2999	0.7012	0.7067	0.7007	0.7051	**0.7503**
ResNet-34	Normal	0.9327	0.9086	0.8321	**0.9314**	0.8991	0.8882
	Adversarial	0.3302	0.6609	0.6804	0.671	0.6649	**0.7471**
LeNet	Normal	0.746	0.6946	0.3971	**0.7248**	0.3687	0.6303
	Adversarial	0.0353	0.3055	0.9483	**0.9545**	0.9454	0.8921
MobileNetV2	Normal	0.9201	0.8414	0.8402	**0.9176**	0.8730	0.8856
	Adversarial	0.3248	0.6848	0.6649	0.6752	0.6923	**0.7852**

Table 2: Results of residual error prediction model accuracy on CIFAR-100. Here adversarial refers to FGSM attack with $\epsilon = 8$, and normal dataset is the dataset without any adversarial examples.

Primary model	Dataset	Primary Accuracy	Residual error prediction model				
			ResNet-18	ResNet-34	LeNet	MobileNetV2	Hybrid
ResNet-18	Normal	0.7454	0.7454	0.7454	0.7436	0.6994	**0.7604**
	Adversarial	0.156	0.7183	0.9131	0.9124	**0.9145**	0.8755
ResNet-34	Normal	0.7579	0.7106	0.7482	0.7486	0.4441	**0.7763**
	Adversarial	0.1233	0.876	**0.8752**	0.8736	0.8063	0.8458
MobileNetV2	Normal	0.7067	0.7064	0.6992	0.7057	0.6989	**0.7092**
	Adversarial	0.0781	0.9219	**0.923**	0.9222	0.9111	0.8922

prediction model, the residual error prediction models are trained by both clean and adversarial examples.

A grid search algorithm was done to choose the best hyper-parameters for the residual error prediction models. As seen in Table 1, although the primary models perform with relatively low accuracy on adversarial examples, the residual error prediction models could identify whether the primary models is performing correctly or not with a very higher performance. The experimental results show that the proposed hybrid model which is the composite of a primary model and XGBoost outperforms other deep networks (except for LeNet as the primary model) in adversarial dataset experiments. The hybrid model provided the best average performance compared to other residual error prediction models in these experiments.

3.2. CIFAR-100

To better evaluate the effectiveness of the proposed residual error prediction model, the same experiments are conducted for the CIFAR-100 as well. Different network architectures are used for both primary and residual error prediction models. Table 2 shows the accuracy of different network architectures based on normal and adversarial datasets. Due to the very low accuracy of LeNet model (i.e., because of its low capacity) on CIFAR-100, it is excluded as a primary model in this experiment. Unlike what we observed in CIFAR-10, we can see that ResNet-34 has a higher accuracy for adversarial examples compared to other residual error prediction models (except for ResNet-18 where it is the second best model). On the other hand, Hybrid model has the best accuracy on the normal dataset.

3.3. Adversarial Example Detection

Finally, in the Supplementary, we show the effectiveness of the proposed method in discriminating adversarial examples from non-adversarial examples. The primary model used in this experiment is MobileNetV2 and ResNet-18. To better visualize the samples the image tensors are projected into a 2D space using PCA algorithm. Please refer to the Supplementary materials for more information on this.3

4. Conclusion

In this work, we presented the notion of residual error, a new performance metric for not only assessing adversarial robustness at the individual sample level but also differentiating between adversarial and non-adversarial examples, thus facilitating for adversarial example detection. A hybrid prediction model comprised of deep neural network and decision tree macroarchitectures is to improve the performance of the residual model to mitigate the lack of training data and improve the effeciency of the model. The proposed performance metric is especially useful with the existence of adversarial attacks as it can provide a confidence bound on the performance of the trained deep neural

network for each input. Experimental results showed the performance of the hybrid residual error prediction model on several different image classification networks. Building better hybrid residual error prediction models with higher accuracy is an interesting direction for future research. Furthermore, there are many applications of residual error beyond the context of adversarial robustness assessment, as it can also be harnessed as a safety measure in other domains of machine learning, which would be interesting to explore as a future direction.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[2] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie NC Shih, John Tomaszewski, Fabio A González, and Anant Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific reports*, 7:46450, 2017.

[3] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[10] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[11] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[13] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's

neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.