Deep Adversarially-Enhanced k-Nearest Neighbors

Ren Wang  
University of Michigan  
renwang@umich.edu

Tianqi Chen  
University of Michigan  
qtch@umich.edu

Alfred Hero  
University of Michigan  
hero@eecs.umich.edu

Abstract—Recent works have theoretically and empirically shown that deep neural networks (DNNs) have an inherent vulnerability to small perturbations. Applying the Deep k-Nearest Neighbors (DkNN) classifier, we observe a dramatically increasing robustness-accuracy trade-off as the layer goes deeper. In this work, we propose a Deep Adversarially-Enhanced k-Nearest Neighbors (DAEkNN) method which achieves higher robustness than DkNN and mitigates the robustness-accuracy trade-off in deep layers through two key elements. First, DAEkNN is based on an adversarially trained model. Second, DAEkNN makes predictions by leveraging a weighted combination of benign and adversarial training data. Empirically, we find that DAEkNN improves both the robustness and the robustness-accuracy trade-off on MNIST and CIFAR-10 datasets.

Index Terms—robustness, deep neural networks, robustness-accuracy trade-off, deep adversarially enhanced k-nearest neighbors

I. INTRODUCTION

Deep learning has become the backbone of many applications, such as machine translation [1], image classification [2], and object detection [3]. Advanced deep neural network (DNN) models like convolutional neural network (CNN) [4] and recurrent neural network (RNN) [5] result in superiority over conventional models [6]. As a side effect, these DNN-based applications only perform well when the input is similar to the examples in the training set, i.e., when the covariate shift of the representation is small compared with the training distribution. A large shift will result in a significant accuracy degradation [7]. More seriously, recent studies have shown that DNNs are easily fooled by adversarial examples, which are well-designed with imperceptible (to the human eyes) perturbations on benign inputs [8; 9]. The lacking of robustness and the big trade-off between the standard performance and the robustness have become bottlenecks of deep learning.

To tackle the problems, various approaches have been proposed. One category of methods is robust training, which mainly focuses on hardening model weights [10; 11; 12] via adversarial training data. Empirically, we find that DAEkNN improves both the robustness and the robustness-accuracy trade-off on MNIST and CIFAR-10 datasets.

Throughout this paper, we evaluate different classifiers by measuring three key indices: (i) Standard Accuracy (SA) on benign test examples (ii) Adversarial Accuracy (AA) on the adversarial test examples generated by 10-step Projected Gradient Descent (PGD) attack [21] and (iii) Harmonic Mean (HM) of AA and SA, which can be calculated by \( \frac{2 \times AA \times SA}{SA + AA} \). HM is determined by both SA and AA, reflecting the robustness-accuracy trade-off to a certain extend.

a) Projected Gradient Descent (PGD) attack: PGD attack generates adversarial examples \( x_\delta = x + \delta \) to flip the prediction via maximizing the cross-entropy loss \( L \). The generator implements gradient ascent with projection, with the \( i \)-th iterations defined as

\[
\begin{align*}
    x_\delta^{i+1} &\leftarrow \prod_{C(x, \epsilon)} \left[ x_\delta^i + \kappa \text{sign} (\nabla_{x, y} L(f_\theta, x_\delta, y)) \right] 
\end{align*}
\]
Accuracy

0.2

0.4

0.6

0.8

1

Convolutional layer

Fig. 1. Deep Adversarially-Enhanced k-Nearest Neighbors (DAEkNN) can achieve higher robustness and alleviate the robustness-accuracy trade-off than applying Deep k-Nearest Neighbors (DkNN) on a model trained by standard training. All experiments are conducted on CIFAR-10. Adversarial accuracy is evaluated using 10-step PGD attack [21] with attack power $\epsilon = 8$. The blue curves show the results of DkNN. The red curves show the results of DAEkNN.

where $f_\theta$ is the mapping from DNN input to output. $\kappa$ is the gradient step size. $\prod$ denotes a projection operator, and $\mathcal{C}(x, \epsilon) = \{x_{\delta} \mid \|\delta\|_p \leq \epsilon, x_{\delta} \in [v_{lb}, v_{ub}]\}$ represents the projection set. Here $\epsilon$ is a positive constant, and we use $\|\cdot\|_p = \|\cdot\|_\infty$ in this paper. If not otherwise specified, we use the attack power $\epsilon = 8/80$ for CIFAR-10/MNIST. $d$ is the input dimension. $[v_{lb}, v_{ub}]$ is the constraint on the data domain. In this paper, we consider $x_{\delta}$ as an image vector, thereby $x_{\delta} \in [0, 255]^d$.

b) Deep k-Nearest Neighbors (DkNN): Different from DNN classifiers, DkNN [17] combines kNN classifiers that are embedded into selected layers of a DNN, and makes predictions via majority vote among all kNNs. For any test input $x$, DkNN classifies it to

$$y = \arg \max_c \sum_{l \in \mathcal{L}} p_l^c(x; D^b_l), \ c \in [C],$$

(2)

where $\mathcal{L}$ is the set of the selected layers and $l \in \mathcal{L}$ is the $l$-th layer of a DNN. $f_\theta$ denotes the mapping from input to layer $l$. Here $p_l^c(x)$ denotes the confidence score assigned to class $c$ predicted by the corresponding kNN in layer $l$. We use $[C]$ to denote the set $\{1, 2, \ldots, C\}$, where $C$ is the number of classes. $D^b_l = (X^b_{tr}, Y^b_{tr})$ is the selected training dataset containing all benign examples $X^b_{tr}$ together with their labels $Y^b_{tr}$, and $D^b_l(j) = (X^b_{tr}(j), Y^b_{tr}(j)) = (x_j, y_j)$ represents the $j$th example in $D^b_l$. Let $K^l_c$ be the number of data points belonging to class $c$ among the $K$ nearest neighbors found via measuring $\text{dist}(f_\theta(x), f_\theta(X^b_{tr}(j)))$, $\forall j \in |X^b_{tr}|$, where $|\cdot|$ means the cardinality of a set. We use $\ell_2$ distance for $\text{dist}$. The confidence score can be calculated by $p_l^c(x; D^b_l) = \frac{K^l_c}{K} \times \frac{1}{C}$.

By default, DkNN is applied on a model trained by standard training. From Figure [1] one can see that the adversarial accuracy obtained by DkNN drops dramatically as the layer goes deeper. As the standard accuracy obtained by DkNN monotonically increases, the robustness-accuracy trade-off also rises. To improve both the robustness and robustness-accuracy trade-off, we introduce a Deep Adversarially-Enhanced k-Nearest Neighbors (DAEkNN) method containing two key components: (i) A DNN model obtained by adversarial training (ii) A joint prediction mechanism leveraging benign and adversarial training data. We introduce each of them in detail below.

B. Improving Robustness Through Adversarially Trained Model

From Figure [1] one can see that the adversarial accuracy of DkNN drops dramatically as the layer goes deeper, indicating the features learned by DNN lack robustness. In order to robustify k-nearest neighbor classifiers in deep layers, we need the features extracted from those layers to be robust beforehand. Recent works have demonstrated that models trained by standard training tend to learn non-robust features, while models trained by adversarial training [21] are able to extract more robust features [22] [23] [24].

As shown in Figure [2], the adversarial training is performed by updating model weights with adversarial examples generated from benign data in each epoch, and can be characterized by the following optimization problem.

$$\arg \min_{\theta} \frac{1}{|S|} \sum_{(x, y) \in S} \max_{\|\delta\|_\infty \leq \epsilon_{tr}} L(f_\theta, x + \delta, y),$$

(3)

where $\epsilon_{tr}$ is the attack power of the adversarial generator in the training process, and in this paper we fix it to $\epsilon_{tr} = 4$ and $\epsilon_{tr} = 60$ for CIFAR-10 and MNIST, respectively. $S$ denotes a subset of $D^b_l$. We remark that there are different robust training methods [21] [41] that can be leveraged to harden DNNs, and we only focus on the adversarial training method [21].
C. Improving Robustness Through Adversarial Training Data

Note that adversarial perturbations are intended to enlarging the gap between the adversarial test distribution and the distribution of the original benign class. Using benign training distribution as the baseline, the covariate shift of the adversarial test distribution is larger than the covariate shift of the benign test distribution. For this reason, when all k-nearest neighbors are found from benign training data, the kNN classifiers are not able to correctly capture the information of the adversarial distribution. Spurred by that, we ask: Is it possible to better handle the covariate shift as well as obtaining more robust classifiers by changing the baseline (benign) distribution to a benign-adversarial combined distribution (in other words, by adding adversarial examples into the kNN searching space)?

Instead of using (2), we propose a benign-adversarial joint prediction mechanism that robustifies classifiers through incorporating the adversarial training data, as illustrated in the inference stage in Figure 2. Mathematically, the benign-adversarial joint prediction is expressed by the following formula.

\[ y = \arg \max_c \sum_{l \in L} [\omega^l_a p^l_f (x; D^a_u) + \omega^l_b p^l_f (x; D^b_u)], \quad c \in [C], \quad (4) \]

where \( D^a_u \) denotes the adversarial training data obtained by applying PGD attack on \( D^b_u \).

\[ D^a_u = \text{PGD}(\theta, \epsilon_r, D^b_u), \quad (5) \]

Here \( D^a_u \) serves as the hardening dataset. We therefore can leverage the hardening strength. We remark that \( \epsilon_r \) is not as large as better. For DAEkNN to be useful, we need to ensure that the features extracted by deep layers contain useful class information. For example, a model trained by standard training cannot tolerate large perturbations, and therefore using a large \( \epsilon_r \) will hurt the performance. In contrast, an adversarially trained model can extract class-relevant features from both benign data and perturbed data. In this sense, an adversarially trained model results in narrower gap between the representations of benign examples and the adversarial examples. \( \omega^l_a \) and \( \omega^l_b \) are weights of the two classifiers. Note that (4) is not a simple mixed up of the benign and adversarial training data. Instead, the new classifier in a single layer is a weighted combination of two classifiers. The weights \( \omega^l_a, \omega^l_b \) are calculated based on the relative distance of the kNN found by the two classifiers.

\[ \omega^l_a = \frac{e^{\text{dist}(f_{\theta_l}(x); f_{\theta_l}(X^b_l))}}{e^{\text{dist}(f_{\theta_l}(x); f_{\theta_l}(X^b_l))} + e^{\text{dist}(f_{\theta_l}(x); f_{\theta_l}(X^a_l))}}, \quad (6) \]

\[ \omega^l_b = \frac{e^{\text{dist}(f_{\theta_l}(x); f_{\theta_l}(X^b_l))}}{e^{\text{dist}(f_{\theta_l}(x); f_{\theta_l}(X^b_l))} + e^{\text{dist}(f_{\theta_l}(x); f_{\theta_l}(X^a_l))}} \]

where \( X^b_l \) denotes the kNN of \( x \) found from \( D^b_u \) on layer \( l \), and similarly, \( X^a_l \) denotes the kNN of \( x \) found from \( D^a_u \) on layer \( l \). We use the softmax version of the distance-based weights. One can see that the weight is large when the corresponding distance is small, and \( \omega^l_a + \omega^l_b = 1 \).

(4) refines the decision boundaries of (2) by leveraging the combined and weighted classifiers. The implementation details can be viewed in Algorithm 1. We also refer readers to Section III for the ablation study and comparisons.

**Algorithm 1 Deep Adversarially-Enhanced k-Nearest Neighbors (DAEkNN)**

**Input:** Model \( \theta \) trained by adversarial training with \( \epsilon_u \); the selected layers \( L \); hardening strength \( \epsilon_r \); test input \( x' \); number of nearest neighbors \( K \).

Generate \( D^b_u \) from \( D^b_u \) through (5).

for All \( l \in L \) do

Find \( K \) nearest neighbors \( X^b_l \) and \( X^a_l \) from \( D^b_u \) and \( D^a_u \), respectively.

Count the number of data belonging to class \( c \) (\( K^c_l \)) among each of \( X^b_l \) and \( X^a_l \).

Calculate \( p^l_f (x'; D^b_u) \) and \( p^l_f (x'; D^a_u) \) using \( K^c_l / K \) obtained from \( X^b_l \) and \( X^a_l \).

Calculate \( \omega^l_a, \omega^l_b \) through (6).

end for

Obtain the final prediction by (4).

Output: \( y \).
In this work, we propose a new deep neural network (DNN)-based classifier, called Deep Adversarially-Enhanced k-Nearest Neighbors (DAEkNN), which enjoys high robustness and can mitigate the robustness-accuracy trade-off. We first study the performance of single-layer Deep k-Nearest neighbors (DkNN) on each layer of a DNN. The results indicate that DkNN has poor robustness and big robustness-accuracy trade-offs. The proposed DAEkNN improves robustness through two key adversarial components. DAEkNN extracts features from an adversarially trained model that can provide more robust features. DAEkNN utilizes both the benign training data and adversarial training data in a benign-adversarial joint prediction classifier. The experiments on MNIST and CIFAR-10 demonstrate the effectiveness of DAEkNN.

In future work, we will (i) consider more attack types; (ii) explore the combinations of adversarial training data generated by different hardening strength and different attack types, which could potentially improve the robustness under stronger and unforeseen attacks; (iii) provide a theoretical analysis of the DAEkNN.

### IV. CONCLUSION

We then study how the hardening strength $\epsilon_r$ affects the performance of DAEkNN. As shown in Figure 3, increasing $\epsilon_r$ from 0–8 improves the robustness without appreciable loss of standard accuracy.

![Fig. 3. Increasing the hardening strength $\epsilon_r$ within a certain range (≤ $\epsilon$ in this test) can improve the robustness of Deep Adversarially-Enhanced k-Nearest Neighbors (DAEkNN) without appreciable loss of standard accuracy. All experiments are conducted on CIFAR-10. Adversarial accuracy is evaluated using a 10-step PGD attack [21] with attack power $\epsilon = 8$.

| Hardening strength $\epsilon_r$ | Standard accuracy | Adversarial accuracy |
|---------------------------------|------------------|---------------------|
| 0                               | 0.45             | 0.45                |
| 2                               | 0.46             | 0.46                |
| 4                               | 0.47             | 0.47                |
| 6                               | 0.48             | 0.48                |
| 8                               | 0.49             | 0.49                |

### REFERENCES

[1] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghavrininejad, M. Lewis, and L. Zeltlemoyer, “Multilingual denoising pre-training for neural machine translation,” Transactions of the Association for Computational Linguistics, vol. 8, pp. 726–742, 2020.

[2] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, “Deep learning for hyperspectral image classification: An overview,” IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 9, pp. 6690–6709, 2019.

[3] X. Wu, D. Sahoo, and S. C. Hoi, “Recent advances in deep learning for object detection,” Neurocomputing, vol. 396, pp. 39–64, 2020.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[5] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013, pp. 6645–6649.

[6] J. Diederik, N. O’Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Kpalkova, D. Riordan, and J. Walsh, “Deep learning vs. traditional computer vision,” in Science and Information Conference. Springer, 2019, pp. 128–144.

[7] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” Machine learning, vol. 79, no. 1, pp. 151–175, 2010.

[8] A. Korakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” arXiv preprint arXiv:1607.02533, 2016.

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” 2014 ICLR, vol. arXiv preprint arXiv:1312.6199, 2014.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” arXiv preprint arXiv:1706.06083, 2017.

[11] H. Zhang, Y. Yu, J. Jiao, B. Xing, L. E. Ghaoui, and M. I. Jordan, “Theoretically principled trade-off between robustness and accuracy,” International Conference on Machine Learning, 2019.

[12] J. Cohen, E. Rosenfeld, and Z. Kolter, “Certified adversarial robustness via randomized smoothing,” in International Conference on Machine Learning, 2019, pp. 1310–1320.

[13] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, “On detecting adversarial perturbations,” in International Conference on Learning Representations, 2017.

[14] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, “Detecting adversarial samples from artifacts,” arXiv preprint arXiv:1703.00410, 2017.

[15] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, “On the (statistical) detection of adversarial examples,” arXiv preprint arXiv:1702.06280, 2017.
N. Carlini and D. Wagner, “Adversarial examples are not easily detected: Bypassing ten detection methods,” in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

N. Papernot and P. McDaniel, “Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning,” *arXiv preprint arXiv:1803.04765*, 2018.

A. Dubey, L. v. d. Maaten, Z. Yalniz, Y. Li, and D. Mahajan, “Defense against adversarial images using web-scale nearest-neighbor search,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8767–8776.

R. Wang, T. Chen, S. Lindsly, C. Stansbury, A. Rehemtulla, I. Rajapakse, and A. Hero, “Rails: A robust adversarial immune-inspired learning system,” *arXiv preprint arXiv:2107.02840*, 2021.

D. Bahri, H. Jiang, and M. Gupta, “Deep k-nn for noisy labels,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 540–550.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.

L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry, “Adversarial robustness as a prior for learned representations,” *arXiv preprint arXiv:1906.09945*, 2019.

R. Wang, K. Xu, S. Liu, P.-Y. Chen, T.-W. Weng, C. Gan, and M. Wang, “On fast adversarial robustness adaptation in model-agnostic meta-learning,” 2021.

R. Wang, G. Zhang, S. Liu, P.-Y. Chen, J. Xiong, and M. Wang, “Practical detection of trojan neural networks: Data-limited and data-free cases,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.