Exploring Layerwise Adversarial Robustness
Through the Lens of t-SNE

Inês Valentim
University of Coimbra, CISUC/LASI,
DEI
Coimbra, Portugal
valentim@dei.uc.pt

Nuno Antunes
University of Coimbra, CISUC/LASI,
DEI
Coimbra, Portugal
nmsa@dei.uc.pt

Nuno Lourenço
University of Coimbra, CISUC/LASI,
DEI
Coimbra, Portugal
naml@dei.uc.pt

ABSTRACT

Adversarial examples, designed to trick Artificial Neural Networks (ANNs) into producing wrong outputs, highlight vulnerabilities in these models. Exploring these weaknesses is crucial for developing defenses, and so, we propose a method to assess the adversarial robustness of image-classifying ANNs. The t-distributed Stochastic Neighbor Embedding (t-SNE) technique is used for visual inspection, and a metric, which compares the clean and perturbed embeddings, helps pinpoint weak spots in the layers. Analyzing two ANNs on CIFAR-10, one designed by humans and another via NeuroEvolution, we found that differences between clean and perturbed representations emerge early on, in the feature extraction layers, affecting subsequent classification. The findings with our metric are supported by the visual analysis of the t-SNE maps.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; • **Security and privacy → Software and application security**.

KEYWORDS

Adversarial Examples, Latent Space Visualization, NeuroEvolution, Robustness

1 INTRODUCTION

Adversarial examples [8, 14] are a threat to the robustness of Artificial Neural Networks (ANNs). They are carefully crafted to fool these models by adding perturbations, often small and imperceptible, to benign data samples [3]. There is a vast literature showing that manually-designed ANNs [4, 10, 13], as well as ANNs designed in an automated way [7, 15], suffer from this vulnerability.

The robustness of a model against adversarial examples can be estimated by performing attacks and computing their success rate [5]. This gives us a general idea of how the model would perform in similar conditions, but does not provide any insight into its inner workings. To tackle this, we propose a method to visualize and examine changes in the representation of the input data as it goes through the different layers of an ANN.

Our proposal is based on the t-distributed Stochastic Neighbor Embedding (t-SNE) [17] technique. Relying on a visual analysis to quantify differences between original and altered data in many layers is a daunting task. Thus, we suggest a metric for measuring clean-perturbed data overlap in the t-SNE space.

We focused on Convolutional Neural Networks (CNNs) designed to solve image classification tasks. We inspected pre-trained models for the CIFAR-10 dataset [11], namely a manually-designed Wide Residual Network (WRN) [18] and a CNN designed by NeuroEvolution (NE) [1], trained without any defense against adversarial perturbations. Following recent works [5], we considered L_2 and L_∞-robustness, using three variants of the Auto-PGD (APGD) method [6] as attacks.

The metric shows that network deterioration begins in the feature extraction layers, affecting how CNNs distinguish between clean and perturbed images. This is also visible in their separation on the t-SNE maps.

The paper is organized as follows. Section 2 provides some background. The proposed approach and metric are described in Section 3. Section 4 details the general setup of our experiments and Section 5 presents the main findings. Section 6 concludes the paper and points toward future directions.

2 BACKGROUND AND RELATED WORK

An adversarial example [8, 14] is an input similar to a valid data point to which a model gives a highly different prediction [14]. In the image domain, it is common to add small L_p-norm perturbations, bounded by a budget ϵ, to the benign sample [2]. An attack can cause a misclassification of a sample as a specific class (targeted) or as any class as long as it is not the right one (untargeted) [3].

AutoAttack [6] is an ensemble of white-box and black-box attacks that can be used as a heuristic evaluation method of the adversarial robustness of a model. It is adopted by the RobustBench [5] benchmark, which uses standardized evaluation methodologies to keep track of the progress made in adversarial robustness. The APGD method [6], a variation of the Projected Gradient Descent (PGD) method [13] used by AutoAttack, progressively reduces the step size in an automated way, based on how the optimization is proceeding. Croce and Hein [6] also propose the Difference of Logits Ratio (DLR) loss, an alternative to the cross-entropy (CE) loss that is both invariant to shifts of the logits and to rescaling.

Cianfarani et al. [4] inspect the layerwise representations of CNNs using representation similarity metrics. The authors investigate the similarity between the representations of clean and adversarially perturbed images, which closely relates to our work. Their findings suggest that this similarity score is typically high in earlier layers of the networks and, for undefended models, gets close to zero once the final layer is reached. This work does not include any method to visualize the representations themselves.

3 METHODOLOGY

The proposed methodology to analyze the different layers of a CNN from an adversarial robustness perspective is presented in Figure 1. We do not use the training set of the dataset while analyzing a model. Moreover, we create several random splits from the test set so as to be able to have access to validation data (used to analyze the model), while also putting aside test data.

The first step is to select and perform an adversarial attack, only considering correctly classified images. Pixel values are normalized beforehand and any pre-processing specific to a model is included in its definition. Once the attack is applied, both the perturbed images and the clean ones are passed through the ANN up until the desired target layer to extract the hidden representations.

Due to being high-dimensional, it is not possible to visualize this latent data. Thus, for each layer that we want to inspect, we apply the t-SNE method to the extracted representations to get a two-dimensional map. We also apply the t-SNE method to the clean images incorrectly classified by the model (not used as inputs to the adversarial attacks).

Generating adversarial examples occurs once per validation set, but extracting latent layer representations and computing the t-SNE map must be repeated for each inspected layer. For each validation set and target layer, we examine the 2D maps and compute a metric that measures layer robustness by comparing clean and perturbed image embeddings.

3.1 Robustness Metric

To summarize the layer's outputs, we propose a robustness metric based on the differences between the clean and the corresponding adversarially perturbed representations on the t-SNE map. The rationale behind the proposed metric is based on the notion that, for a representation learned by a layer to be robust, the clean image and the adversarially perturbed one should be mapped to the same point in the t-SNE space.

We restrict the computation of the metric to pairs of clean-perturbed images, even though the t-SNE technique is also applied to clean images for which adversarial attacks are not generated. For each analyzed layer, we calculate the t-SNE for both clean and attacked images. We find the shortest Euclidean distance between each clean image and a clean instance from a different class. If the distance between a clean image and its perturbed counterpart is less than this minimum, we consider that they overlap.

The final metric value corresponds to the ratio of clean-perturbed pairs that overlap according to that heuristic:

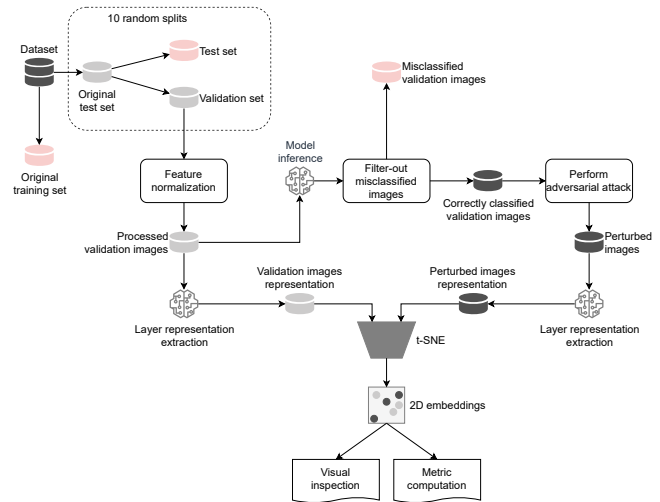$$\text{robustness metric} = \frac{\sum_{i=1}^{n} \text{OverlappingEmbeddings}\,(i, C, A, Y)}{n}$$

Figure 1: Proposed methodology to analyze the adversarial robustness of different layers of a CNN.

where OverlappingEmbeddings is the algorithm described to detect overlaps, C and A are the t-SNE representations of the clean and adversarially perturbed images, respectively, Y is the set of true labels of the clean images, and n is the number of perturbed images that were generated (and, thus, the number of clean-perturbed pairs). Metric values range from 0 to 1, with higher values indicating that more clean and perturbed images overlap on the t-SNE map, which suggests layerwise robustness. This metric requires attacking instances from multiple classes, as detecting overlaps depends on the minimum distance to an instance of a different class.

4 EXPERIMENTAL SETUP

All experiments were run in Python 3.8, using Tensorflow 2.5.0 and PyTorch 1.10.1.

4.1 Dataset and Models

Using 10 random seeds and maintaining data balance through stratification, the test set of the CIFAR-10 dataset [11] was splitted into a validation set and a final test set, each with 5000 images. Adversarial examples were generated for validation images.

Given the remarkable results achieved by some NeuroEvolutionary approaches [1, 12], we included a CNN designed by NE [1] in our experiments. Additionally, we used a handcrafted architecture [18] as a baseline, specifically the WRN-28-10 model[1] trained by the RobustBench team. Regarding DENSER [1], we chose the best performing architecture[2] over the original evolutionary runs. To avoid introducing bias from our end, we used the pre-trained models directly, without any form of re-training.

4.2 Threat Models, Attacks, and t-SNE

We performed white-box attacks, since an attacker can easily have full access to the models. Furthermore, we considered L_∞ perturbations with $\epsilon = 8/255$, as well as L_2 perturbations with $\epsilon = 0.5$ [5].

---

[1]https://github.com/RobustBench/robustbench/tree/master/robustbench/model_zoo
[2]https://github.com/fillassuncao/denser-models/tree/master/CIFAR-10/net_1

Running the complete AutoAttack ensemble [6] on undefended models would be unnecessarily expensive. Thus, we performed some of the attacks from the ensemble in isolation: an untargeted APGD on the CE loss (APGD-CE), an untargeted APGD on the DLR loss (APGD-DLR), and an APGD on the targeted DLR loss with 9 target classes (denoted by APGD-T). The number of iterations for all the attacks is 100 and neither performs random restarts.

These attacks operate over the logits. As such, a slight modification had to be introduced in the definition of the DENSER model, whose original architecture has the softmax activation directly incorporated in the last fully-connected (FC) layer. We used the original Python implementations[3] of the attacks.

For visualization, we used a t-SNE implementation which relies on Barnes-Hut approximations of the gradient [16]. We considered the default value of 0.5 for the parameter that controls the trade-off between speed and accuracy. Instead of randomly initializing the solution (i.e., the two-dimensional embedding), the PCA method [9] is applied to the input data to get the initial low-dimensional representation. We considered 1500 iterations and a perplexity of 50 [17].

5 INSPECTING THE NETWORKS

We attack only correctly classified images, leading to varying attack counts per model. The reported post-attack accuracy includes all validation images, covering both generalization errors and adversarial robustness. This promotes a fair comparison between models.

The WRN-28-10 model has a mean accuracy of 94.84% ± 0.25%, which is slightly higher than that of DENSER (mean accuracy of 93.68% ± 0.31%). This refers to the accuracy of the pre-trained models on the validation sets of clean images from CIFAR-10.

For both models and robustness scenarios, the accuracy drops to near-zero values after performing either of the three attacks (APGD-CE, APGD-DLR, or APGD-T). Considering L_2 perturbations, not all validation images can be successfully perturbed with APGD-CE and APGD-DLR, but the accuracy is always below 0.45%, showing that neither model is robust against any of these attacks.

5.1 Robustness Metric across Layers

Next, we computed the robustness metric for some of the layers of the models. Figure 2 shows the results for APGD-CE, with each point representing a run with one of the 10 validation sets. The WRN-28-10 model comprises three groups (denoted by b1, b2, and b3) of four residual blocks (from `layer.0` to `layer.3`). As such, `b3.layer.1.add` corresponds to the output of the second residual block in the last group. Figure 2a shows the results for that model, while Figure 2b presents the results for DENSER.

The first layers of both models seem to keep the representations of clean and perturbed samples close to one another, as shown by metric values close to 1. Moreover, the robustness degrades earlier on in the network with L_∞ perturbations than with L_2.

For WRN-28-10, the most drastic drops in the robustness of the hidden representations seem to occur in the last group of residual blocks. In the L_2 scenario, the metric drops to almost zero after the second residual block in that group, while in the L_∞ case, it is almost zero for all layers following the first residual block. Metric values
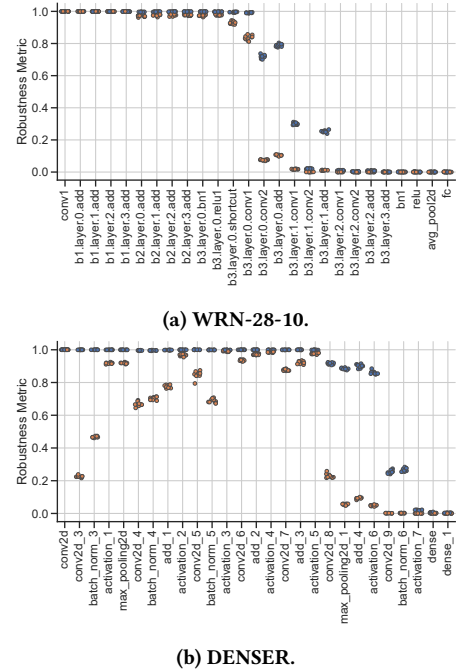
[3]https://github.com/fra31/auto-attack

(a) WRN-28-10.

(b) DENSER.

Figure 2: Layerwise robustness of the models under APGD-CE with L_2 (blue) and L_∞ (orange) perturbations.

close to zero mean that there is practically no overlap between clean and perturbed images on the low-dimensional t-SNE map.

Regarding DENSER, the layers between `conv2d` and `conv2d_3` are omitted since the metric does not drop from 1. In the L_2 scenario, the layerwise robustness starts decreasing after `activation_5`, but the biggest drop occurs after `activation_6`. It only gets close to zero with the last ReLU activation (i.e., `activation_7`). For L_∞ perturbations, the results deviate from what we have observed so far. There is a significant drop in `conv2d_3`, but the model seems to recover, with the values for `activation_5` being close to 1. From this point onward, the metric drops significantly again, reaching zero with `conv2d_9` and all the following layers.

The DENSER model has considerably less convolutional layers than WRN-28-10 (10 vs. 28), but more of those layers from the latter model seem to learn representations that are less robust. Focusing on L_2-robustness, the metric at the 23rd convolution of WRN-28-10 (`b3.layer.1.conv1`) drops below 0.4 and does not increase in any layers that follow, while that only happens at the last convolution of DENSER. That represents more than 20% of the convolutional layers of WRN-28-10, but only 10% of DENSER.

Due to space restrictions, we do not show the obtained results with the remaining two attacks, but similar trends can be observed. We just note that, for DENSER, the metric values remain higher until later in the model than with APGD-CE.

5.2 Visual Inspection

Figure 3a shows the 2D map for the `activation_6` layer of DENSER, only considering the points (both clean and perturbed) that, according to the metric, do not overlap. Differences between clean and perturbed representations are relatively scarce. These representations

start to diverge in the layer that immediately follows `activation_6`, i.e., `conv2d_9`. Figure 3b shows the non-overlapping points for that layer, which are noticeably more than on Figure 3a. Additionally, the perturbed points seem to encircle the clean ones.
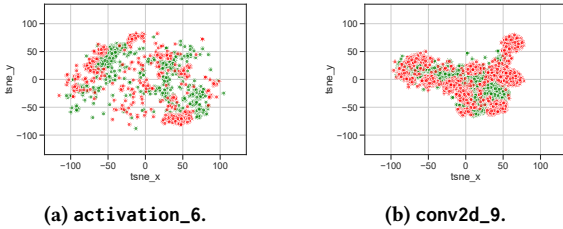

(a) `activation_6`. (b) `conv2d_9`.

Figure 3: Non-overlapping clean (green) and perturbed (red) points on the t-SNE map of different intermediate layers of the DENSER model, considering an APGD-CE attack in L_2.

Lastly, Figures 4a and 4b show the representation of clean and perturbed images at the final FC layer of DENSER, before softmax. Clean images cluster by true labels, while perturbed images cluster with instances from different classes. Almost no clean-perturbed points overlap once they reach this layer.
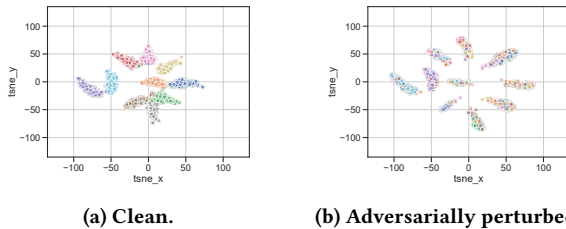

(a) Clean. (b) Adversarially perturbed.

Figure 4: t-SNE map for the last layer of DENSER and an APGD-CE attack in L_2. Colors represent true labels.

The t-SNE maps of WRN-28-10 layers further validate our metric. For instance, the map of the `b3.layer.0.add` layer resembles the one obtained for the `activation_6` layer of DENSER. In the last layer, clean points once again cluster by class and do not overlap with perturbed ones, which are more scattered in the space but still form clusters of mixed labels.

6 CONCLUSION AND FUTURE WORK

Adversarial examples compromise ANN robustness. Since traditional evaluations fall short in multi-layer analysis, we propose a method that quantifies and visually examines the discrepancies between the latent representations of clean and adversarial samples.

Our results show that discrepancies between clean and perturbed data appear still during feature extraction, even before the final convolutional layer. Our layerwise robustness metric aids defense development, with potential uses in improving NE fitness functions or selecting layers for detection-based defenses.

For each architecture, we used a single pre-trained model, which may raise questions on generalizability. Attempts to retrain the models and reproduce the results with the pre-trained WRN were unsuccessful, highlighting the potential influence of the learning strategy on the adversarial robustness of a model.

In the future, the proposed approach needs to be evaluated on more datasets, and with models that have been explicitly designed to be adversarially robust.

REFERENCES
[1] Filipe Assunção, Nuno Lourenço, Penousal Machado, and Bernardete Ribeiro. 2018. DENSER: Deep Evolutionary Network Structured Representation. arXiv:1801.01563 [cs.NE]
[2] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On Evaluating Adversarial Robustness. arXiv:1902.06705 [cs.LG]
[3] Nicholas Carlini and David A. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 39–57. https://doi.org/10.1109/SP.2017.49
[4] Christian Cianfarani, Arjun Nitin Bhagoji, Vikash Sehwag, Ben Zhao, Heather Zheng, and Prateek Mittal. 2022. Understanding Robust Learning through the Lens of Representation Similarities. In *Advances in Neural Information Processing Systems*, Vol. 35. 34912–34925.
[5] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. 2020. RobustBench: a standardized adversarial robustness benchmark. arXiv:2010.09670 [cs.LG]
[6] Francesco Croce and Matthias Hein. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119. PMLR, 2206–2216.
[7] Chaitanya Devaguptapu, Devansh Agarwal, Gaurav Mittal, Pulkit Gopalani, and Vineeth N Balasubramanian. 2021. On Adversarial Robustness: A Neural Architecture Search Perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 152–161.
[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.
[9] Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24 (1933), 417–441.
[10] Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. 2021. Exploring Architectural Ingredients of Adversarially Robust Deep Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 34. 5545–5559.
[11] Alex Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report. University of Toronto.
[12] Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. 2019. NSGA-Net: Neural Architecture Search Using Multi-Objective Genetic Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 419–427. https://doi.org/10.1145/3321707.3321729
[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.
[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.
[15] Inês Valentim, Nuno Lourenço, and Nuno Antunes. 2022. Adversarial Robustness Assessment of NeuroEvolution Approaches. In *2022 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC55065.2022.9870202
[16] Laurens van der Maaten. 2014. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15, 93 (2014), 3221–3245.
[17] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.
[18] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 87.1–87.12.