DeepRobust: a Platform for Adversarial Attacks and Defenses

Yaxin Li*, Wei Jin*, Han Xu, and Jiliang Tang
Computer Science and Engineer,
Michigan State University
{liyaxin1, jinwei2, xuhan1, tangjili}@msu.edu,

Abstract

DeepRobust is a PyTorch platform for generating adversarial examples and building robust machine learning models for different data domains. Users can easily evaluate the attack performance against different defense methods with *DeepRobust*. In this paper, we introduce the functions of *DeepRobust* with detailed instructions. We will demonstrate that *DeepRobust* is a useful tool to measure deep learning model robustness and to identify the suitable countermeasures against adversarial attacks. The platform is kept updated and can be found at https://github.com/DSE-MSU/DeepRobust. More details of instructions can be found in the documentation at https://deeprobust.readthedocs.io/en/latest/.

Introduction

Deep learning has been increasingly adopted by real-world safety-critical applications such as autonomous driving, face recognition, healthcare and education (Xu et al. 2019), it is crucial to examine its vulnerability and safety issues. It was reported in (Szegedy et al. 2013) that Deep Neural Networks (DNNs) are vulnerable to small designed adversarial perturbations. Figure 1 is an example of adversarial example that contains such perturbation. Since then, tremendous efforts have been made on developing attack methods to fool DNNs and designing their countermeasures. As a result, there is a growing need to build a comprehensive platform for adversarial attacks and defenses. Such platform enables us to systematically launch experiments on existing algorithms and efficiently test new algorithms, which could deepen our understandings and immensely foster this research field.

Currently there are some existing platforms, such as *Cleverhans* (Papernot et al. 2018), *advertorch* (Ding et al. 2019). They mainly focused on attack methods in the image domain. However, little attention has been paid on defense methods. Furthermore, the majority of them are dedicated to the image domain while largely ignoring other domains such as graph data. The platform *DeepRobust* aims to fulfill this need. It is an easy-to-use evaluation tool to test adversarial examples against different defense algorithms. It not only provides interfaces for representative algorithms to generate adversarial examples and robust models in the image domain, but also covers attacks and defenses for graph data(Jin et al. 2020b,a). With the evaluation interface, *DeepRobust* can present the defense performance against different attacks. Thus, it can bring insights on finding the strongest attack/defense and give a most reliable evaluation of robustness. To best demonstrate the advantages of *DeepRobust*, we will detail the platform architecture and provide concrete instructions on how to use the platform.

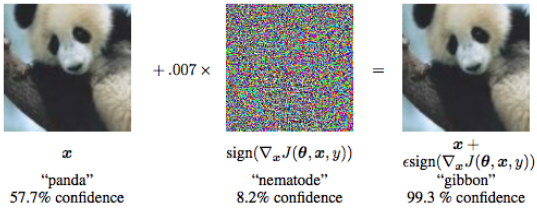

Figure 1: An Illustration of Adversarial Examples.

The Architecture

DeepRobust offers attack and defense generation interfaces for two data domains, i.e., image domain and graph domain. Users can evaluate model robustness against different attacks through the *DeepRobust* evaluation pipeline.

Evaluation Pipeline: Figure 2 is an illustration of the pipeline. Basically, users can generate adversarial attacks through the *attacking API*s (Figure 2, left); robust models can be obtained with *defense API*s (Figure 2, right); and once adversarial attacks and robust models are generated, they can be fed into *evaluation API* to compete against each other, and the model performance under the corresponding attack is reported as both numerical form and figures.

The Image Component: The image component is divided into several sub-packages according to different functions. *Attack* sub-package includes attack base class and 11 attack algorithms. *Defense* sub-package contains defense base class and 8 defense algorithms. *Netmodels* includes different popular classification model classes.

The Graph Component: The graph component contains several sub-packages based on the functions. *Targeted-attack* sub-package includes the targeted attack base class

*Both authors contributed equally in developing the platform.

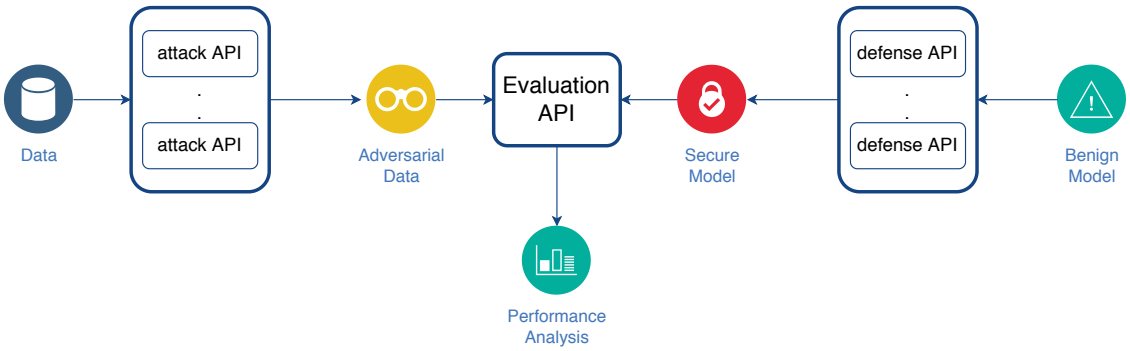

Figure 2: The Architecture of DeepRobust.

and famous targeted attack algorithms. Similarly, global attack base class and global attack algorithms are included in *global-attack* sub-package. *Defense* sub-package provides GCN model and other methods for defending graph adversarial attack. Besides, sub-package *data* provides an easy access to public benchmark datasets including Cora, Cora-ml, Citeseer, Polblogs and Pubmed as well as pre-attacked graph data.

Evaluation

In this section, we go through the robustness measurement process of *DeepRobust* for image and graph domains, separately.

Image Data

Adversarial Example Generation Attackers could use any attack APIs in the *image* package in *DeepRobust* to generate adversarial images. Users can either utilize attacking parameters provided by *DeepRobust* in config file, which are consistent with the common experiment settings, or define customize parameters to fit different evaluation needs. Following code is an example to generate PGD adversarial examples(Madry et al. 2017). Line 1 is to generate an attack object, passing the victim model and device. Call *generate* as in line2 to generate adversarial examples correspondingly with a set of preset parameters.

```
adversary = PGD(model, device)
Adv_img = adversary.generate(x, y,
**attack_params["PGD_CIFAR10"])
```

Robust Model Generation *DeepRobust* also support different defense strategies to train robust model. Users can use the defense APIs in the *defense* package.

```
defense = PGDtraining(model, device)
defense.generate(trainloader,
                 testloader,
**defense_params["PGDtraining_MNIST"])
```

Evaluation and Visualization *DeepRobust* also allows users to evaluate the adversarial images generated by an attack method against a defense model through the evaluation API. With the help of this function, users can evaluate an attack method towards different models, thus bring in an clear illustration of the effectiveness of the attack method. Meanwhile, one robust model can be tested under different attacks, thus can show a comprehensive test result of the robustness of one model. Moreover, through changing the parameters of the competing attack and defense methods, users can get the performance of models under the different attack settings. Basically, users are able to select the attack method and the victim as shown in the following code. Besides, the *DeepRobust* documentation includes more options including: perturbation budget, target labels of attacking, etc.

```
python evaluation_attack.py
--attack_method PGD --attack_model CNN
--dataset MNIST
```

Graph Data

Similar to the *image* package, the *graph* package also provides APIs for users to generate adversarial samples, build robust models and evaluate the robustness of models. The major differences between them are in three aspects: (1) the *graph* package is dealing with discrete graph data, which is more challenging than image data where the pixel values are continuous; (2) graph data is naturally sparse so the APIs in the *graph* package are mostly dealing with sparse matrices to guarantee the efficiency; and (3) since poisoning attack is very common in real-world graph-related applications, the *graph* package provides pre-attacked graphs for users to fast evaluate the model robustness and analyze the properties of graph adversarial attacks.

Conclusion

In this paper, we demonstrate *DeepRobust*, a platform to study adversarial attacks and defenses. We discuss the usage of the platform and provide hand on instructions. In the future, we will keep enriching the platform by including more algorithms and more domains such as texts.

Acknowledgements

This research is supported by the National Science Foundation (NSF) under grant number CNS1815636, IIS1928278, IIS1714741, IIS1845081, IIS1907704, and IIS1955285.

References

Ding, W.; et al. 2019. AdverTorch v0.1: An Adversarial Robustness Toolbox based on PyTorch. *arXiv:1902.07623* .

Jin, W.; Li, Y.; Xu, H.; Wang, Y.; and Tang, J. 2020a. Adversarial Attacks and Defenses on Graphs: A Review and Empirical Study. *arXiv preprint arXiv:2003.00653* .

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020b. Graph Structure Learning for Robust Graph Neural Networks. *arXiv preprint arXiv:2005.10203* .

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

Papernot, N.; et al. 2018. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv:1610.00768* .

Szegedy, C.; et al. 2013. Intriguing properties of neural networks. *arXiv:1312.6199* .

Xu, H.; Ma, Y.; Liu, H.; Deb, D.; Liu, H.; Tang, J.; and Jain, A. 2019. Adversarial attacks and defenses in images, graphs and text: A review. *arXiv preprint arXiv:1909.08072* .