Robustness

Last updated by | Mateusz Macias | Jun 15, 2020 at 12:22 PM GMT+1

Robustness

Definition

Robustness refer to the correctness of an AI system in the presence of invalid inputs or stressful environmental conditions.

> Let S be a machine learning system. Let E(S) be the correctness of S. Let δ(S) be the machine learning system with perturbations on any machine learning components such as the data, the learning program, or the framework. The robustness of a machine learning system is a measurement of the difference between E(S) and E(δ(S)): i.e. $r = E(S) - E(\delta(S))$. Robustness thus measures the resilience of an ML system's correctness in the presence of perturbations.

Robustness considerations

A key consideration of AI Robustness is the model robustness against adversarial attacks. There are two main forms of attacks:

1. White-box attack: This is where the attackers know the structures of the model as well as it's weights and parameters. Adversarial samples are crafted to be visually similar but intended to "poison" the parameters of the model thus deteriorate its correctness. The main defence mechanism is to train the model with adversarial samples.

2. Black-box attack: This is where the attacker does not know the model structure or parameters. The attacker is intending to copycat or steal the model, as most commercial AI models can be expensive to train. The attackers craft the attack to learn the probabilities, thus gradually learn the model. The main defence mechanisms include; secure API and limit API call frequencies; applying methods to blur the outputs such as adding inverse noise to the probabilities so the attacker cannot learn the model from output.

These attacks can be deliberate but in some cases may be accidental. Consequently it is a good idea to test for robustness and protect from such attacks. To read further details about how adversary attacks happen and can be mitigated, read through this article.

Metrics to measure model robustness

There are a few widely used metrics to measure model robustness:

1. Model accuracy/precision/recall on test data. The higher the better.

2. After re-training with adversarial samples, model accuracy/precision/recall on test data and on adversarial data. The higher the better.

3. Reduction in confidence is computed using this formula:

   (y_ori_classconf - y_adv_classconf) / y_ori_classconf

   The smaller the better.

Tools to detect, generate and mitigate Adversary attacks

All the tools listed below can be used by machine learning developers to construct robust models by using adversarial training, which requires the construction of adversarial examples during the training procedure. It is encouraged to report the accuracy of their models in the adversarial setting.

IBM Adversarial Robustness Toolbox (ART)

The aim of this tool is to protect the machine learning pipeline to ensure its safety at training, test and inference time. This tool provides nice functional API's allowing for the integration of classification models with this library. It supports most of the machine learning libraries such as tensorflow, mxnet, pytorch, scikit-learn, keras, XGBoost, catBoost and gPy. To understand about the different types of adversarial attacks, it is strongly recommended to read through section 2 of this paper.

Following is a list of algorithms available from ART - this list is copied from https://github.com/IBM/adversarial-robustness-toolbox:

Evasion Attacks:

- Threshold Attack (Vargas et al., 2019)
- Pixel Attack (Vargas et al., 2019, Su et al., 2019)
- HopSkipJump attack (Chen et al., 2019)
- High Confidence Low Uncertainty adversarial samples (Grosse et al., 2018)
- Projected gradient descent (Madry et al., 2017)
- NewtonFool (Jang et al., 2017)
- Elastic net attack (Chen et al., 2017)
- Spatial transformation attack (Engstrom et al., 2017)
- Query-efficient black-box attack (Ilyas et al., 2017)
- Zeroth-order optimization attack (Chen et al., 2017)
- Decision-based attack / Boundary attack (Brendel et al., 2018)
- Adversarial patch (Brown et al., 2017)
- Decision tree attack (Papernot et al., 2016)
- Carlini & Wagner (C&W) L_2 and L_inf attacks (Carlini and Wagner, 2016)
- Basic iterative method (Kurakin et al., 2016)
- Jacobian saliency map (Papernot et al., 2016)
- Universal perturbation (Moosavi-Dezfooli et al., 2016)
- DeepFool (Moosavi-Dezfooli et al., 2015)
- Virtual adversarial method (Miyato et al., 2015)
- Fast gradient method (Goodfellow et al., 2014)

Extraction Attacks:

- Functionally Equivalent Extraction (Jagielski et al., 2019)
- Copycat CNN (Correia-Silva et al., 2018)
- KnockoffNets (Orekondy et al., 2018)

Poisoning Attacks:

- Poisoning Attack on SVM (Biggio et al., 2013)
- Backdoor Attack (Gu, et. al., 2017)

Defences - Preprocessor:

- Thermometer encoding (Buckman et al., 2018)
- Total variance minimization (Guo et al., 2018)
- PixelDefend (Song et al., 2017)
- Gaussian data augmentation (Zantedeschi et al., 2017)
- Feature squeezing (Xu et al., 2017)
- Spatial smoothing (Xu et al., 2017)
- JPEG compression (Dziugaite et al., 2016)
- Label smoothing (Warde-Farley and Goodfellow, 2016)
- Virtual adversarial training (Miyato et al., 2015)

Defences - Postprocessor:

- Reverse Sigmoid (Lee et al., 2018)
- Random Noise (Chandrasekaranet al., 2018)
- Class Labels (Tramer et al., 2016, Chandrasekaranet al., 2018)
- High Confidence (Tramer et al., 2016)
- Rounding (Tramer et al., 2016)

Defences - Trainer:

- Adversarial training (Szegedy et al., 2013)
- Adversarial training Madry PGD (Madry et al., 2017)

Defences - Transformer:

- Defensive Distillation (Papernot et al., 2015)

Robustness Metrics, Certifications and Verifications:

- Clique Method Robustness Verification (Hongge et al., 2019)
- Randomized Smoothing (Cohen et al., 2019)
- CLEVER (Weng et al., 2018)
- Loss sensitivity (Arpit et al., 2017)
- Empirical robustness (Moosavi-Dezfooli et al., 2015)

Detection of Adversarial Examples:

- Basic detector based on inputs
- Detector trained on the activations of a specific layer
- Detector based on Fast Generalized Subset Scan (Speakman et al., 2018)

Detection of Poisoning Attacks:

- Detection based on activations analysis (Chen et al., 2018)
- Detection based on data provenance (Baracaldo et al., 2018)

Tool | Implementation guide | Metrics list
Documentation | Examples

FoolBox

Foolbox provides a collection of state-of-the-art gradient-based and decision-based adversarial attacks. It supports limited ML libraries PyTorch, Tensorflow, JAX and numpy. They use distance metrics to measure/quantify the size of adversarial perturbations. According to their tool, Gradient-based attacks can be classified under white-box attack while score-based attacks and decision-based attacks can be classified under black-box attacks. Furthermore this toolkit supports only deep learning models.

Tool | Implementation guide | Metrics list
Documentation | Examples | Tutorial

CleverHans

Cleverhans is a software library from Google that provides state-of-the-art implementations of adversarial example construction techniques and adversarial training. However this supports only deep learning with TensorFlow, pytorch and keras.

For all forms of attacks, the adversarial crafting algorithms take the model and an input and returns adversarial example. To defend from attacks (ie., defend from adversarial examples is to make the model smoother by limiting its sensitivity to small perturbations of its inputs), they have crafted adversarial training algorithm that injects adversarial examples during training to improve the generalisation of the machine learning model.

Tool | Implementation guide
Examples | Tutorial

AdvBox

AdvBox is a tool from Baidu to generate adversarial examples that fool neural networks in PaddlePaddle, PyTorch, Caffe2, MxNet, Keras, TensorFlow and benchmark the robustness of machine learning models. They also provide defense methods to protect from adversarial attacks. They do not seem to have detection mechanisms however this is not completely verified as a lot of their documentation is provided in Mandarin.

The following is a list of algorithms they have provided for use. This list is copied from https://github.com/advboxes/AdvBox/blob/master/advbox.md

White-box attack methods:

- Limited Memory-Broyden-Fletcher-Goldfarb-Shannon
- Fast Gradient Sign Method (FGSM)
- Basic Iterative Method (BIM)
- Iterative Least-Likely Class Method (ILCM)
- MI-Fast Gradient Sign Method
- Jacobian-based Saliency Map (JSMA)
- DeepFool
- Carlini and Wagner (C/W)

Black-box attack methods:

- Single Pixel Attack
- Local Search Attack
- Defense methods:
- Feature Squeezing

- Spatial Smoothing
- Label Smoothing
- Gaussian Augmentation
- Adversarial Training
- Thermometer Encoding

Tool | Implementation guide
Examples | Tutorial - In Mandarin

Artificial Adversary - The only toolkit for NLP which was developed by Airbnb.

When classifying user-generated text, there are many ways that users can modify their content to avoid detection. These methods are typically cosmetic modifications to the texts that change the raw characters or words used, but leave the original meaning visible enough for human readers to understand. Such methods include replacing characters with similar looking ones, removing or adding punctuation and spacing, and swapping letters in words. For example "please wire me 10,000 US DOLLARS to bank of scamland" is probably an obvious scam message, but "pl3@se.wire me 10000 US DoLars to,BANK of Scamland" would fool many classifiers. These kind of samples can be generated using this tool. This is the only tool identified to generate adversarial examples for text data.

Tool

Alibi Detect

This is an open source tool to detect adversarial examples. They are focused on outliers as well as adversarial and concept drift detection. The library aims to cover both online and offline detectors for tabular data, images and time series.

Tool | Implementation algorithms
Documentation | Examples

References

(1) Robustness and Explainability of Artificial Intelligence -- There is a more detailed explanation on the Robustness towards AI from this link. This report talks about the different policy initiatives formed by different expert committees.
(2) Machine Learning Testing: Survey, Landscapes and Horizons -- This paper provides a comprehensive survey to test ML models. It can be accessed from this link.