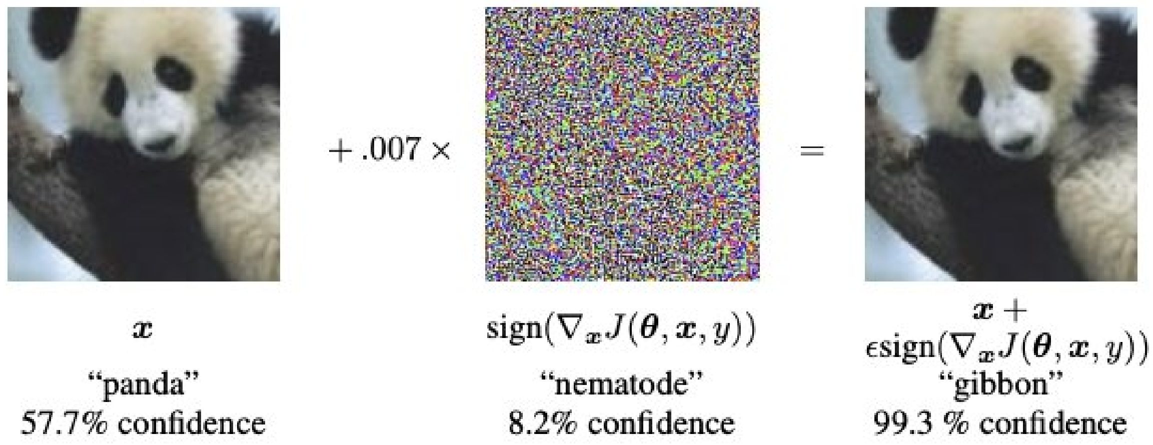
Adversarial attacks

☰ Week	Project

Definition

Adversarial machine learning is a machine learning method that aims to trick machine learning models by providing deceptive input. Hence, it includes both the **generation** and **detection** of adversarial examples, which are inputs
specially created to deceive classifiers.


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

> 👉 Thus, the adversarial attack is an optical illusion for the ML model that misperceives the objects while not visible to the naked eye.

Adversarial example :

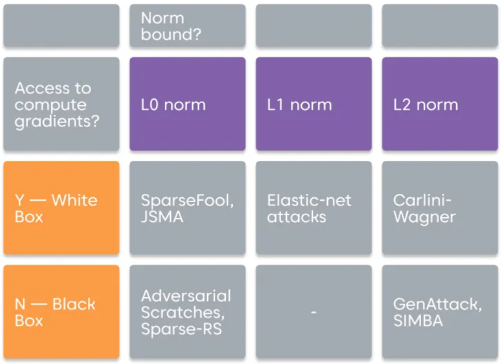
Given an input x, where the model predicts output y, we add a perturbation d that changes the prediction $||d|| < L$:

$$f(x + d)! = y$$

Types of adversarial attacks

- **White box :**
  - Access to the model parameters + data + architecture ⇒ we can compute the gradients
- **Grey box :**
  - Access to data
- **Black box :**
  - No access

- The adversarial attacks can also be classified depending on the norm of the upper bound of the perturbation

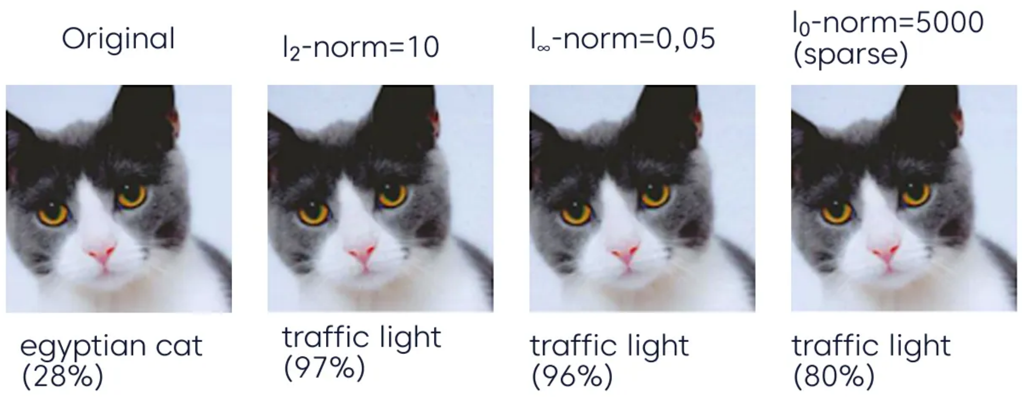


	Norm bound?		
Access to compute gradients?	L0 norm	L1 norm	L2 norm
Y — White Box	SparseFool, JSMA	Elastic-net attacks	Carlini-Wagner
N — Black Box	Adversarial Scratches, Sparse-RS	-	GenAttack, SIMBA

The types of adversarial attacks, considering the access to compute guidelines and perturbation bound. Credits to Malhar

sciforce

Adversarial examples with different norm constraints formed via the projected gradient method (Madry et al., 2017) on a Resnet50, along with the distance between the base image and the adversarial example, and the top class label.



Source: Are adversarial examples inevitable? (Ali Shafahi et al.)

sciforce

- Adversarial attacks can be mainly classified into the following categories:
- **Poisoning Attacks** :
  - Influences the training **data** or its **labels**
- **Evasion Attacks**
  - The attacker manipulates the data during deployment to **deceive** previously trained classifiers (most common) (exp : spoofing attacks)
- **Model Extraction Attacks**
  - The attacker probes a **black box** machine learning system in order to either reconstruct the model or extract the data it was trained on.

Popular Adversarial Attack Methods

Limited-memory BFGS (L-BFGS)

The Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method is a **non-linear gradient-based** numerical optimization algorithm to minimize the number of perturbations added to images.

- Advantages: Effective at generating adversarial examples.
- Disadvantages: Very computationally intensive, as it is an optimized method with box constraints. The method is time-consuming and impractical.

FastGradient Sign method (FGSM)

A simple and fast gradient-based method is used to generate adversarial examples to minimize the maximum amount of perturbation added to any pixel of the image to cause misclassification.

Advantages: Comparably efficient computing times.

Disadvantages: Perturbations are added to every feature.

Jacobian-based Saliency Map Attack (JSMA)

Unlike FGSM, the method uses feature selection to minimize the number of features modified while causing misclassification. Flat perturbations are added to features iteratively according to saliency value by decreasing order.

- Advantages: Very few features are perturbed.
- Disadvantages: More computationally intensive than FGSM.

Deepfool Attack

This untargeted adversarial sample generation technique aims at minimizing the euclidean distance between perturbed samples and original samples. Decision boundaries between classes are estimated, and
perturbations are added iteratively.

- Advantages: Effective at producing adversarial examples, with fewer perturbations and higher misclassification rates.

- Disadvantages: More computationally intensive than FGSM and JSMA. Also, adversarial examples are likely not optimal.

Carlini & Wagner Attack (C&W)

The technique is based on the L-BFGS attack (optimization problem) but without box constraints and different objective functions. This makes the method more efficient at generating adversarial examples; it was shown to be able to defeat state-of-the-art defenses, such as defensive
distillation and adversarial training.

- Advantages: Very effective at producing adversarial examples. Also, it can defeat some adversarial defenses.
- Disadvantages: More computationally intensive than FGSM, JSMA, and Deepfool.

Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) have been used to generate adversarial attacks, where two neural networks compete with each other. Thereby one is acting as a **generator**, and the other behaves as the **discriminator**. The two networks play a zero-sum game, where the
generator tries to produce samples that the discriminator will misclassify. Meanwhile, the discriminator tries to distinguish real samples from ones created by the generator.

- Advantages: Generation of samples different from the ones used in training.
- Disadvantages: Training a Generate Adversarial Network is very computationally intensive and can be highly unstable.

Zeroth-order optimization attack (ZOO)

The ZOO technique allows the estimation of the gradient of the classifiers without access to the classifier, making it ideal for black-box attacks. The method estimates gradient and hessian by querying the target model with modified individual features and uses Adam or Newton’s method to optimize perturbations.

- Advantages: Similar performance to the C&W attack. No training of substitute models or information on the classifier is required.

- Disadvantages: Requires a large number of queries to the target classifier.

Tested in the original paper

- **Gradient based attacks**
    - white box setting
    - Against the classifier (not aware of the detector)
    - 2 classes : l_∞, l_2
        - l_∞ :
            - fast gradient sign method (FGSM, Goodfellow et al., 2015)
            - projected gradient descent (PGD, Madry et al., 2018)
            - momentum iterative method (MIM, Dong et al., 2018)
        - l_2 :
            - Carlini & Wagner (CW) attack (Carlini & Wagner, 2017a)

> 👉🏻 **Sanity checks** : If a successful defence against white-box gradient-based attacks is due to gradient masking, then this defence is likely to be less effective against attacks that do not differentiate through the victim classifier and the defence

- **Distillation based attacks**
    - Grey box setting : Access to both training data and output probability vectors of the classifiers on the
      training set
    - Black box setting : the attacker only
      has access to queried labels on a given input.
- **SPSA (evolutionary strategies)**

    - Black box setting
    - l_∞

👉 How to evaluate robustness on a model to an attack

To do :

□ Implement a gradient based attack