Dauphine | PSL
UNIVERSITÉ PARIS

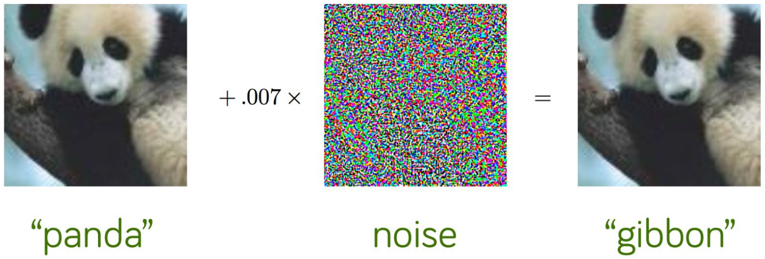
Adversarial attacks

Eureka_Team

Project Data Science IASD
Decembre 2022

Table of Contents

Adversarial machine learning



White box Attacks

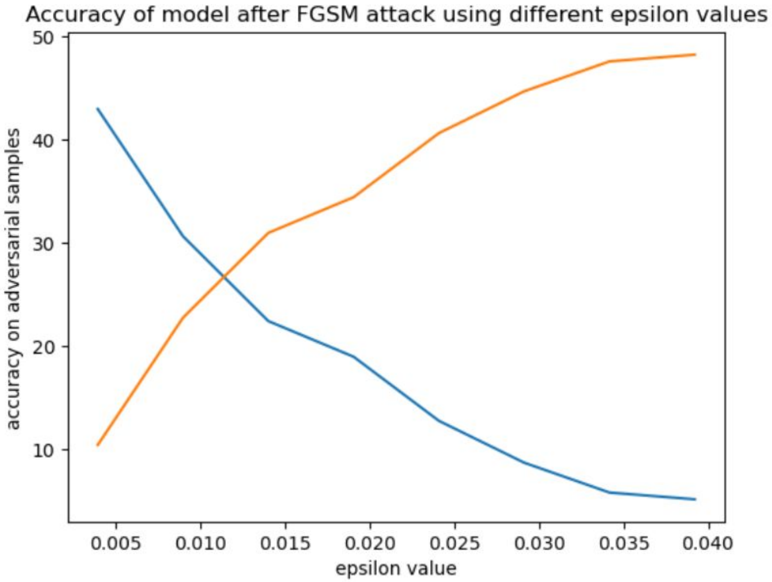
- Fast Gradient Sign Method
- Projected Gradient Descent (Linf, L2)
- Adversarial Model

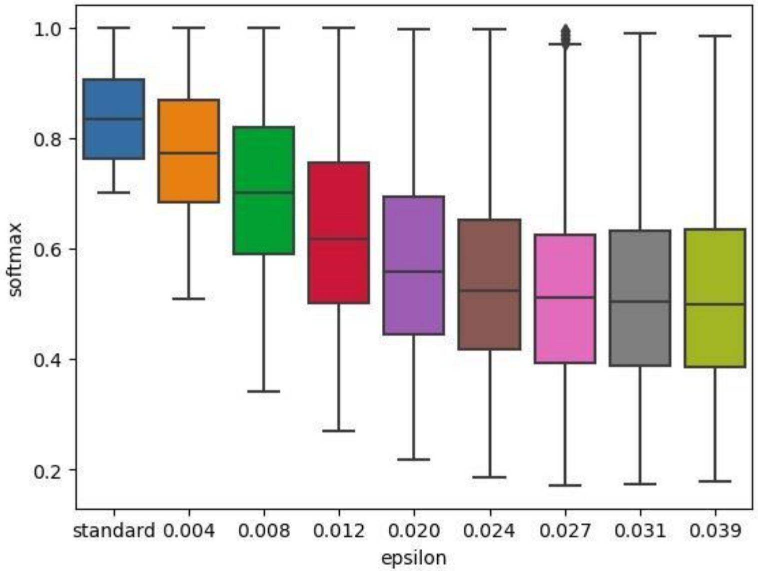
Fast Gradient Sign Method

(Goodfellow, 2015)

One-step gradient update is performed along the direction of gradient.

$$x^{adv} = x + \epsilon \, sign(\nabla_x l(x, y))$$

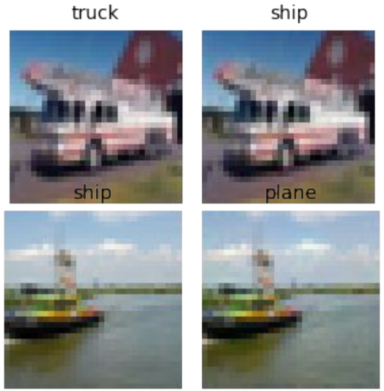


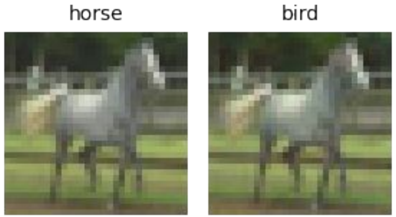


Fast Gradient Sign Method

(Goodfellow, 2015)

One-step gradient update is performed along the direction of gradient.





Projected Gradient Descent

Iterative version of attack

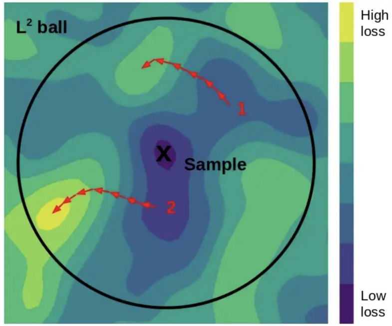
Most complete

Constrained optimization

- L^2 norm
- L^{∞} norm

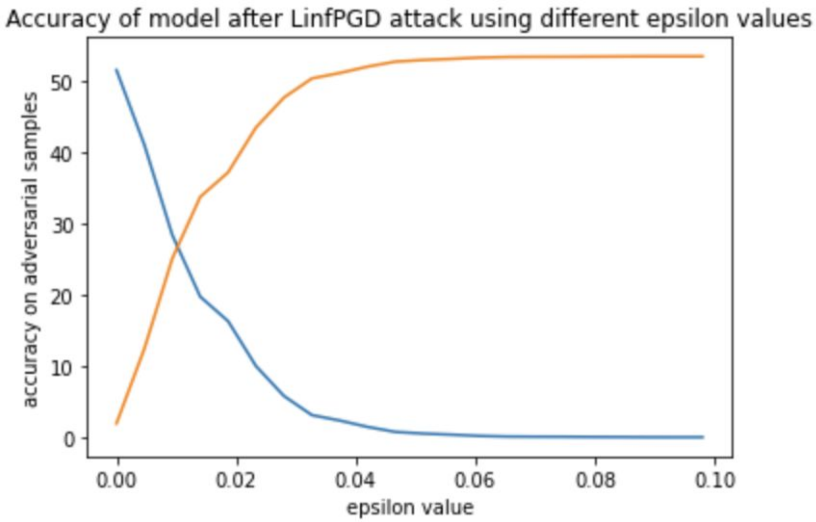
Main Steps

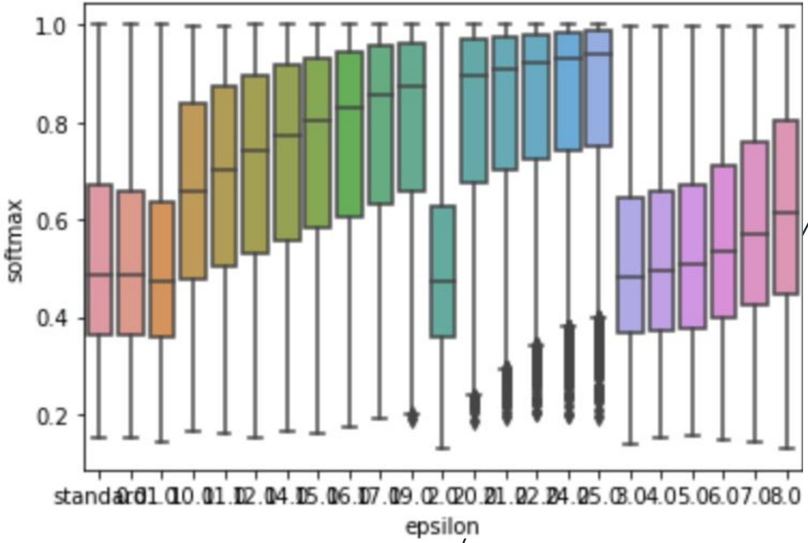
1. Start from a random perturbation in the ball around a sample.
2. Take a gradient step in the direction of greatest loss
3. Project perturbation back into the ball if necessary
4. Repeat until convergence



Projected Gradient Descent Linf

One-step gradient update is performed along the direction of gradient.





Projected Gradient Descent Linf

One-step gradient update is performed along the direction of gradient.

Epsilon: 0.01

Epsilon: 0.0314

Epsilon: 0.06

Epsilon: 0.09

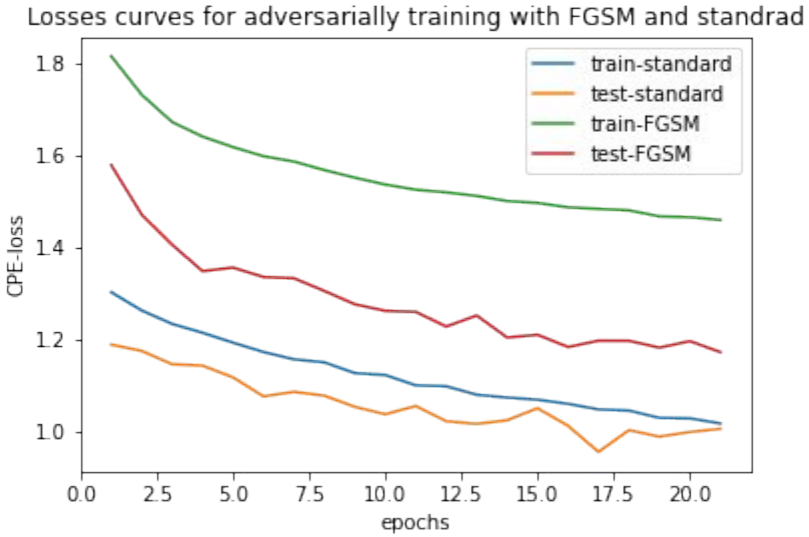
Adversarial Training

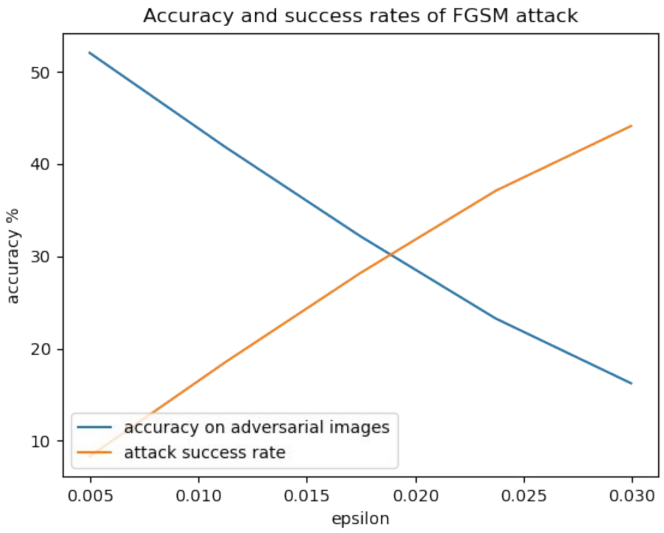
Adversarial training corresponds to the task of training a robust model to adversarial attacks; We want our model to perform well whatever the input received from an adversary agent. It can be modeled as a min-max optimization problem formulated as follows:

$$\min_{\theta} \frac{1}{|S_{train}|} \sum_{(x,y)\in S} \max_{\delta\leq\epsilon} l(h_{\theta}(x+\delta), y)$$

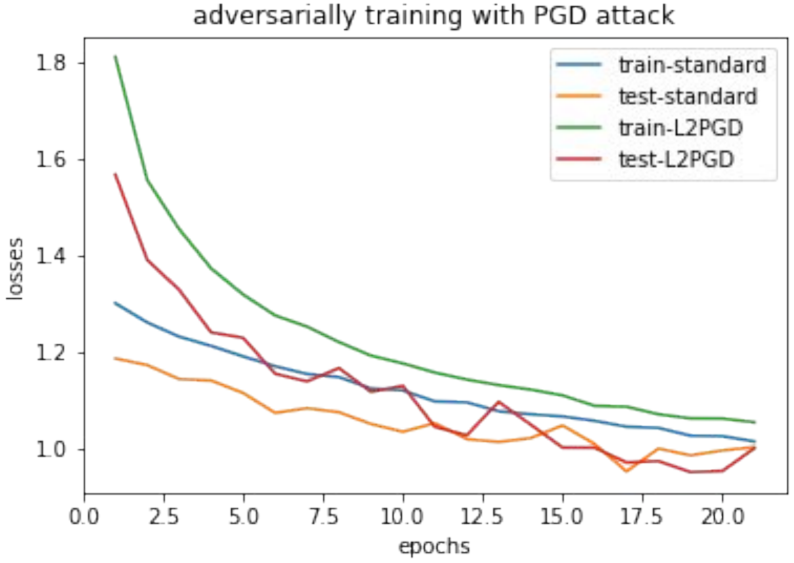
- A simple and intuitive strategy to solve this problem is to incorporate the process of generation of adversarial samples inside the training loop of the model.
- We'll try to use both attacks (FGSM and PGD) to train the classifier and compare losses curves and accuracies on test dataset

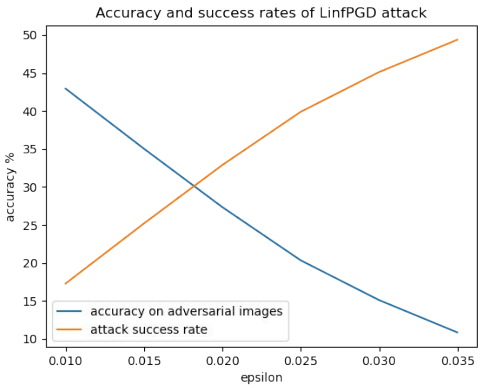
Adversarial Training using FGSM

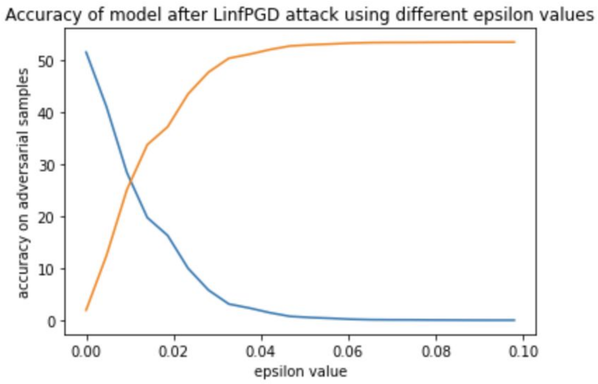




Adversarial Training using PGD





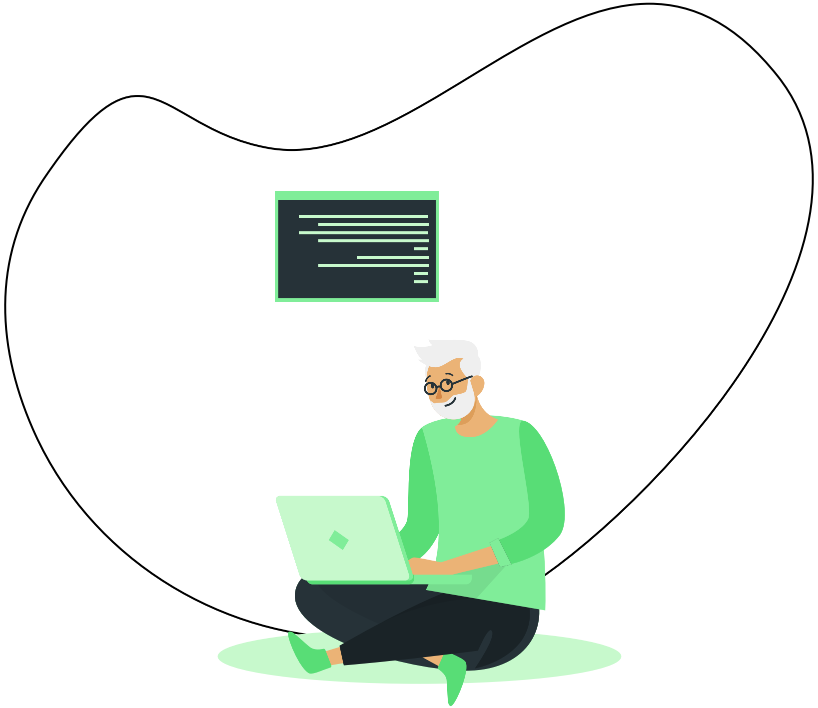


What is next ?

In the 2nd stage of this project, we are looking to implement and try different defense mechanisms :

- Explore the utility of Bayesian neural networks to estimate uncertainty of models and to resist to white box adversary attacks.

Thanks!



References

- Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.
- Madry, Aleksander, et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *arXiv*, 2017, https://doi.org/10.48550/arXiv.1706.06083. Accessed 9 Dec. 2022.
- Gallicchio, Claudio, and Scardapane, Simone. "Deep Randomized Neural Networks." *arXiv*, 2020, https://doi.org/10.48550/arXiv.2002.12287. Accessed 9 Dec. 2022.
- *Adversarial training*. adversarial-ml-tutorial.org. https://adversarial-ml-tutorial.org/adversarial_training/