#29 Improving robustness of neural networks against adversarial examples

Martin Gaňo
Faculty of Information Technology, Brno University of Technology
xganom00@fit.vutbr.cz

Excel@FIT 2020

Abstract

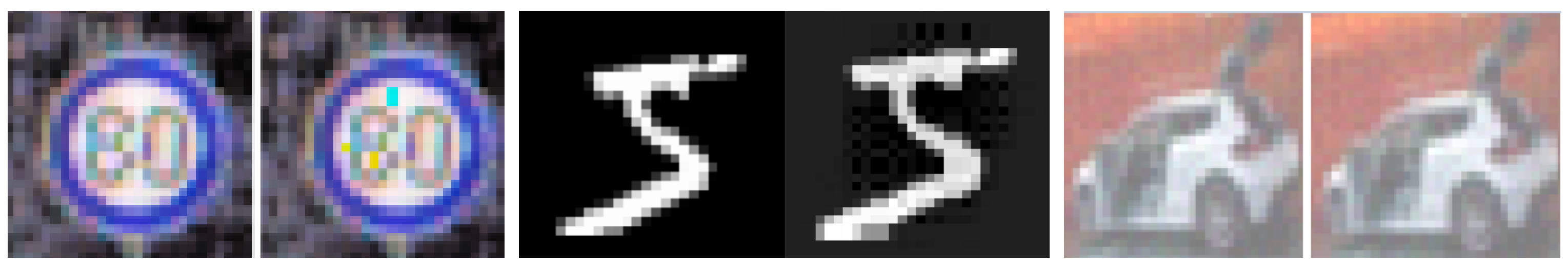
Main goal of this work is to design and implement framework that yields robust neural network model against any adversarial attack, while result models accuracy is not significantly lower comparing to naturally trained model. Our approach is to minimize maximization the loss function of the target model. Related work and our own experiments leads us to use Projected gradient descent method as a target attack therefore we train against data generated by PGD. As a result using the framework we can achieve accuracy more then 90% against sophisticated adversarial attacks on MNIST dataset. Greatest contribution of this work is an implementation of adversarial attacks and defences against them, because there's no public implementation.

1 Problem definition

Recent studies [3] shows an intriguing phenomenon of neural networks vulnerability to so called adversarial examples. Even state-of-the-art neural network architectures with impressive performance are easily fooled by simple adversarial method if they aren't specially trained. Main contribution of this work is providing full implementation of adversarial training introduced by Madry et. al [2]. Our implementation can be used for any model - we can simply load it in h5 format and train.

1.1 Adversarial examples

Sometimes also called *malignant examples*, are a little perturbed and almost indistinguishable from original sample, while they are misclassified by the target model. This phenomenon could seriously endanger using neural networks in safety critical systems. Important terms in this context are ϵ-distance and norm. In context of this work norm is a function that results distance between two points. For the problem of adversarial examples are commonly used two norms - L^2 and L^∞ because they properly simulates similarity for humans. Therefore ϵ is a distance between two points or samples, and if ϵ is below some value it indicates that the two samples reach some level of similarity.



Three couples of natural and adversarial samples. For the first couple the little perturbation changed target model prediction from 80mph to 30mph and for second couple from predicted class "5" to "4". First couple is adopted from [1] and may cause fatal consequences when wrong classify by autonomous car. Third couple is image from CIFAR10 dataset and its perturbation generated by our framework. ϵ for the attack is only 0.03 and accuracy was decreased from 70% to 16.5% just by such small modification. Second image in the third couple was classified as a horse.

Keeping current progress in such speed without studying attack and defence strategies would make all applications managed by neural network an easy target for an enemy.

1.2 Results of adversarial attacks

This section contains basic statistics about attack strategies. All attacks were evaluated on test dataset with $10\,000$ samples with $\epsilon = 0.3$ and all models were trained only against natural examples. For this experiment we used Projected Gradient Descent (PGD), Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM) and it demonstrates that attacks on not prepared models could be really harmfull.

Dataset	Accuracy natural	Accuracy PGD	Accuracy FGSM	Accuracy BIM
MNIST	99.1%	6.9%	12.8%	7.4%
CIFAR10	70.6%	5.5%	8.1%	12.1%
CIFAR100	29.8%	1.2%	1.3%	2.2%

References

[1] **Safety Verification of Deep Neural Networks** Xiaowei Huang, Marta Kwiatkowska, Sen Wang, Min Wu, Department of Computer Science, University of Oxford, 2016

[2] **Towards Deep Learning Models Resistant to Adversarial Attacks** Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu, MIT, 2019

[3] **EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES** Ian J. Goodfellow, Jonathon Shlens, Christian Szeged, https://arxiv.org/pdf/1412.6572.pdf, 2015

[4] **Is pgd-adversarial training necessary? Alternative training via a soft-quantization network with noisy-natural samples only** Tianhang Zheng, Changyou Chen, Kui Ren, ICLR, 2019

2 Adversarial training

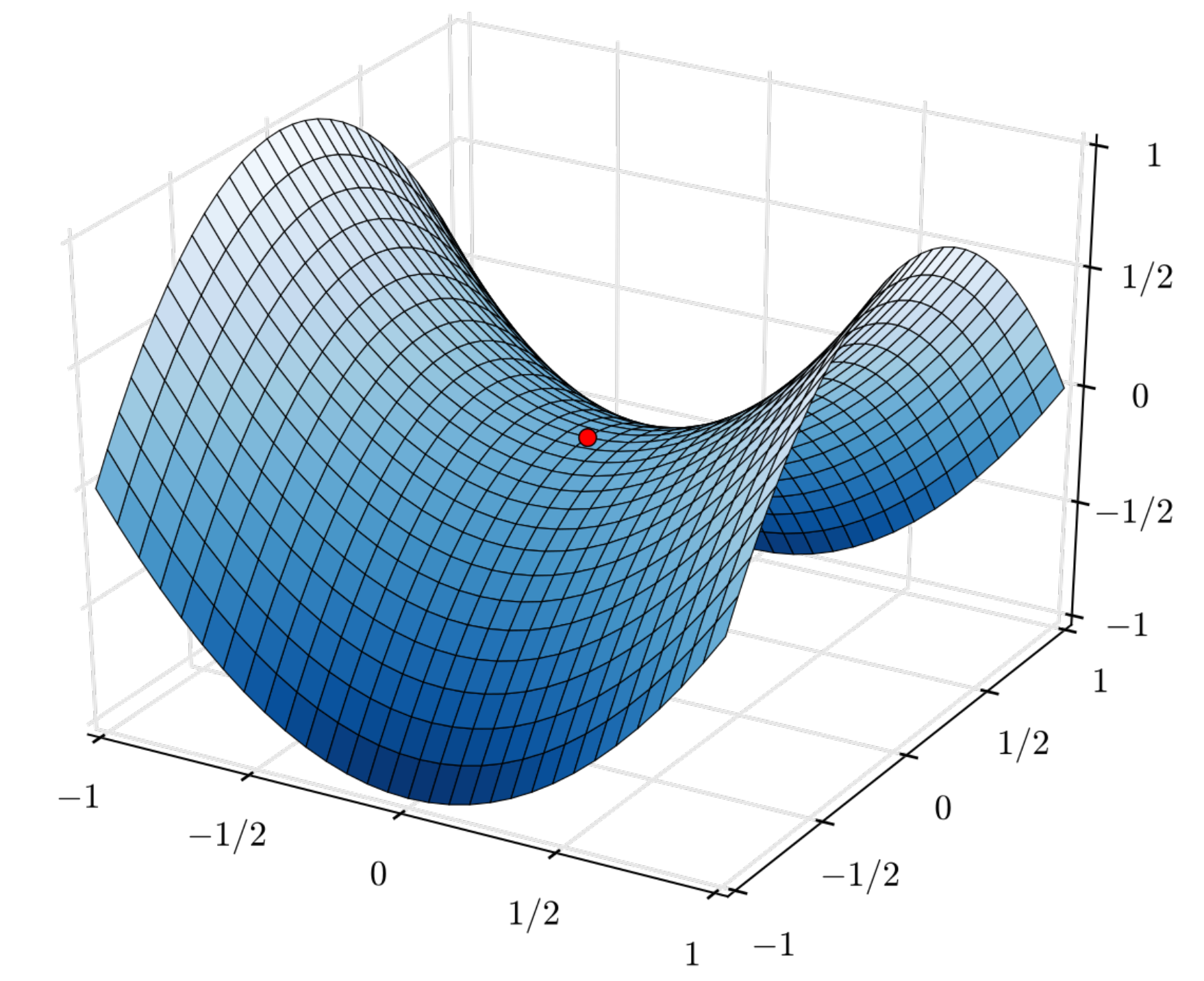
Madry et al. [2] in 2019 proposed defence method against adversarial examples based on minimization maximizing the loss function. Challenging part of this task is to find sufficient method that maximizes an error, while yields samples that don't cause overfitting and this is exactly case of Fast Gradient Sign Method (generally any single-step method). Model trained on FGSM samples would overfit and according to our experiments it would be barely resistant against FGSM but definitely not against other first-order methods. We also experimentally shown sufficiency of samples generated by Projected gradient descent for training robust model and we also provided implementation of such adversarial training in *Python 3*.

Saddle point problem

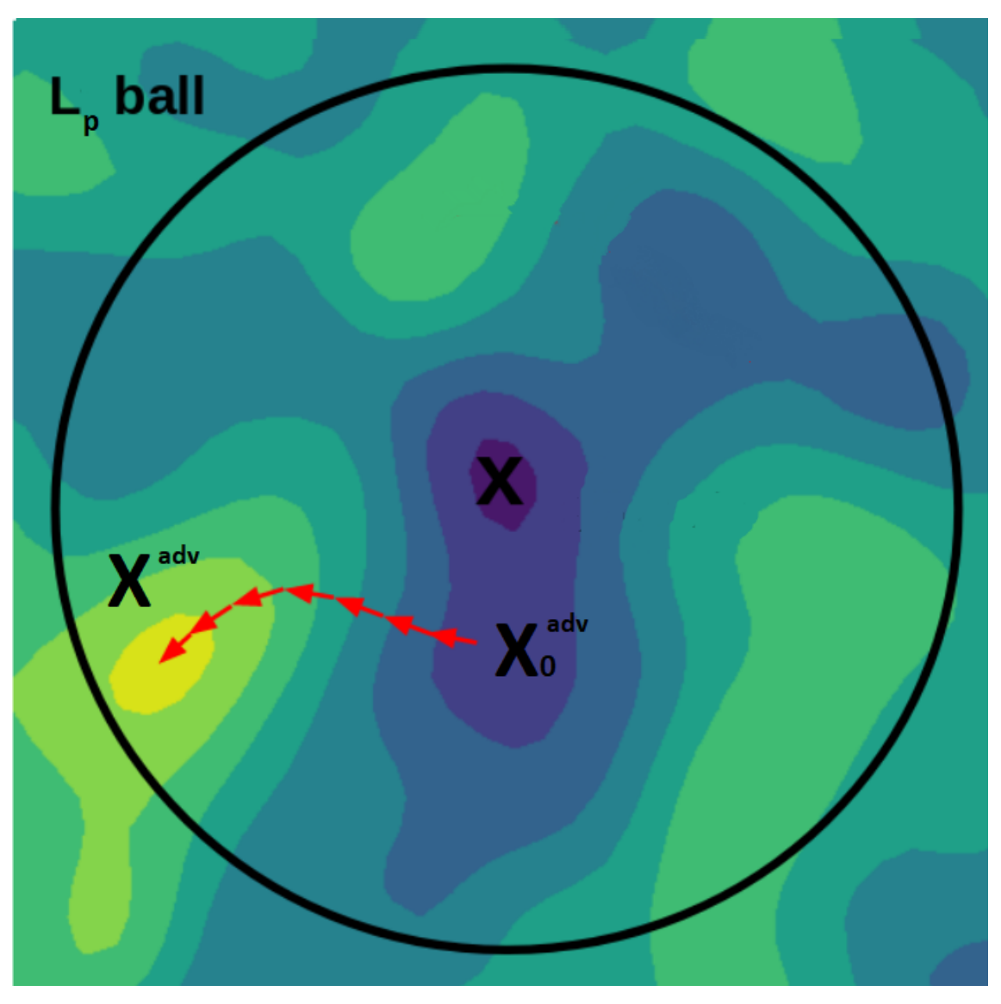
Defence strategy implemented in this work is based on optimization problem called *saddle point* or *minimax* problem. Saddle point is a point on the graph of a function where the derivatives in orthogonal directions are all zero, however it is not a local extreme. We are finding such point to minimize maximization an error of our model using adversarial methods. Formal expression of task for our framework follows:

$$\underset{\theta}{argmin}\ E_{(x,y)\sim D}[\underset{\delta}{max}\ L(x+\delta, y, \theta)]$$

where θ are model parameters, D is data distribution, L is loss function and δ is perturbation such that $\delta \in S$, where S is set of allowed perturbations.



Projected gradient descent



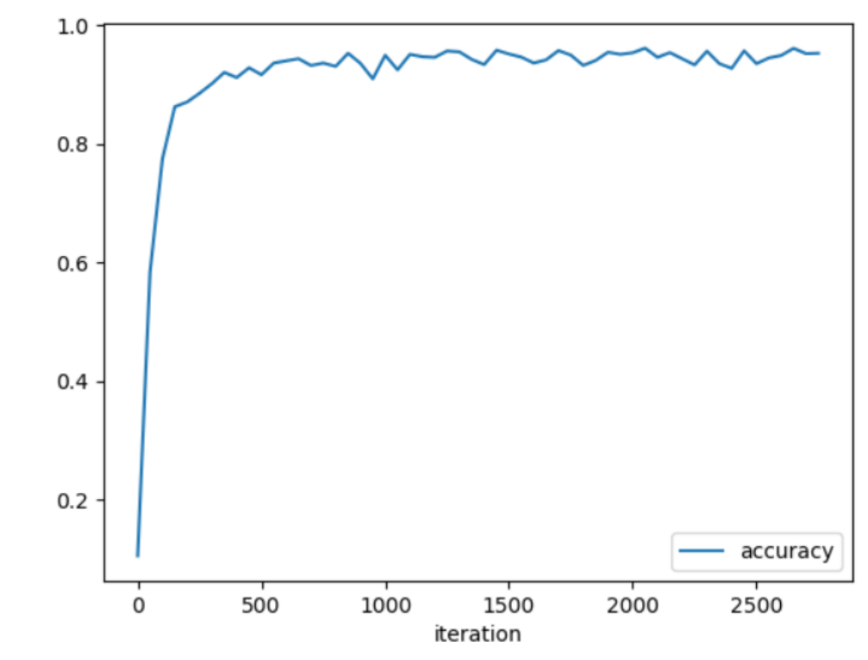
It is essential to understand this adversarial method, since the defence strategy described and implemented in this work is based on PGD. The algorithm is an extension of both FGSM and Basic iterative method. We first set X_0^{adv} to random point within the L^p ball with radius of ϵ. After that we keep applying equation 1 until convergence, i.e. until value of loss function slows down increasing from step to step [4].

$$X_n^{adv} = clip_{X,\epsilon}\{X_{n-1}^{adv} + \alpha * sign(\nabla_x L(X_{n-1}^{adv}, Y)\} \tag{1}$$

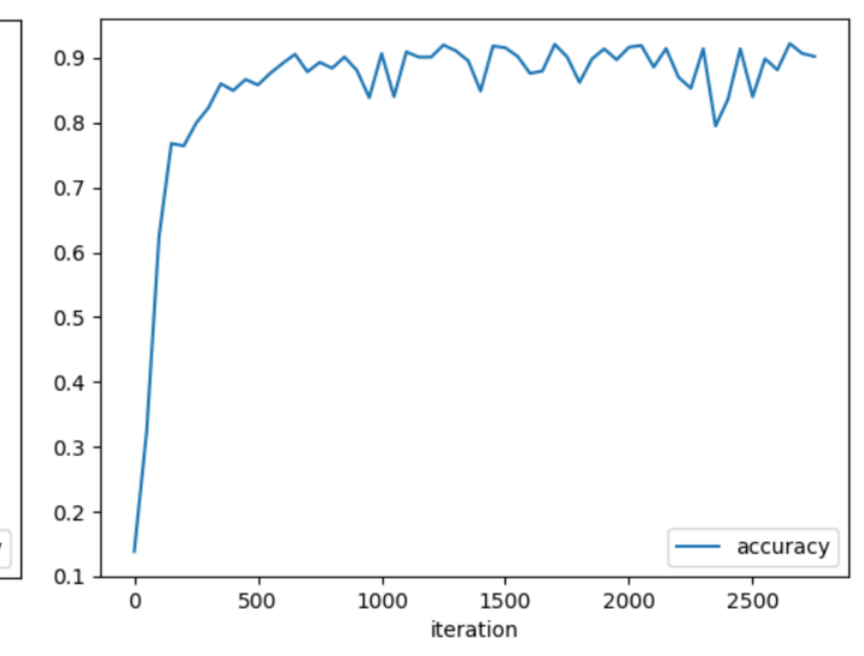
where $sign(x)$ is function that returns -1 if $x < 0$; 0 if $x = 0$ and 1 if $x > 0$ and $clip_{X,\epsilon}(x^{pert})$ is a function that returns x^{pert} if $D(X, x^{pert}) < \epsilon$, otherwise it returns nearest point to x^{pert} that is within ϵ-ball around X.

3 Achieved results

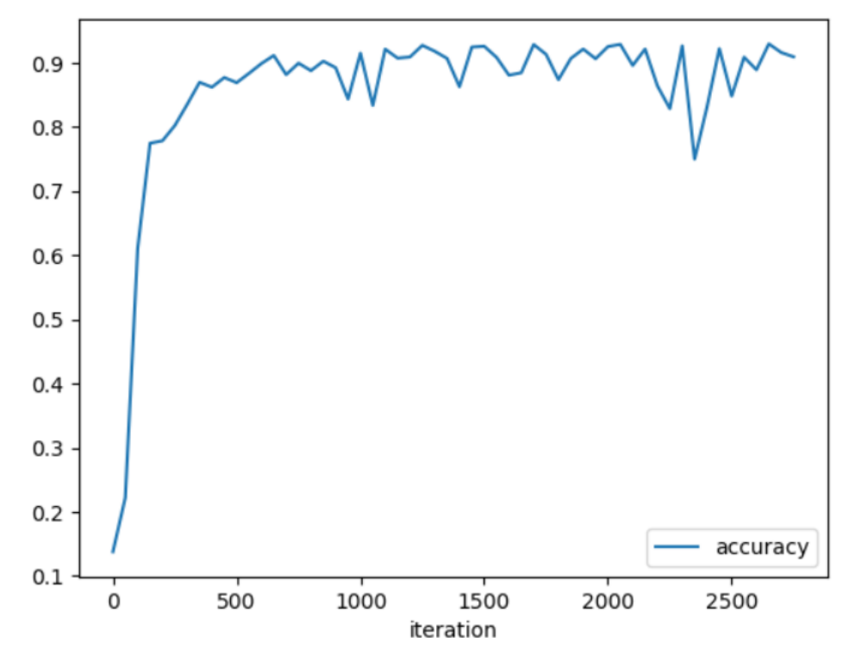
MNIST model trained by our framework achieved impressive results against number of adversarial attacks without significant decreasing its accuracy evaluated on natural data. After performing 500-1000 iterations model achieves sufficient results evaluated on testing dataset containing data generated by adversarial attacks summarized in the Figure below. Performing 1000 iterations takes few hours using an average laptop. An iteration mean generating perturbed samples with user defined size and performing one epoch of training using this data.



FGSM



Basic Iterative Method



Projected Gradient Descent

Madry et al. [2] proposed hypothesis that model trained against PGD should be robust against any first-order adversarial attack and we checked this statement for number of other first-order attacks in our work. That is related to reasoning why we are starting from random point in PGD and don't perform any restarts of method for finding higher local maximums. All maximums found by PGD are almost the same and task to find much higher local maximum would be a hard-to-solve problem. Therefore our adversarial training satisfies current requirements, however it is almost certain that in the future adversarial methods will improve.